Output Factsheet

Output title: Modular cross-spanning energy management tool

Summary of the output (max. 2500 characters)

The output represents a tool for integrated energy management of buildings and energy distribution grids. It is constituted of software modules that should be used on the side of buildings and on the side of electricity distribution grids. Interaction of particular software modules on the building and on the grid side is established which enables integrated energy management both in near real time and on longer time scales for the purpose of demand response service contracting, planning and benefits assessment.

As modularity and easy adaptation of the tool is aimed as its key feature, on the building side the modules are divided into three vertical levels covering three key parts of the buildings – level of comfort control in individual building zones, level of central heating, ventilation and air conditioning systems for preparation of heating/cooling media for zones, and level of major energy flows in the building, shortly named building microgrid level. On each of these levels the modules are divided horizontally into the part for variables prediction and estimation (PE) based on incoming measurements from the building, the part for optimal control of the level by using model predictive control (MPC) and the part for interfacing (I) optimal control commands towards the building automaton equipment in the field.

On the grid side the modules encompass longer-term planning of grid operation and the grid operation itself on time windows of day-ahead and intra-day market operations. Longer-term planning includes determination of technical and economical conditions for engaging demand response services from active consumers via multi-annual software module as well as contracting demand response via annual software module. Short-term operation is enabled by day-ahead software module which is in charge for activating contracted flexibility services from different customers in an optimal way for the grid and by intra-day module which is used to reschedule flexibility activation on hourly and sub-hourly time-scales.

Main features of the tool are:

- it is meant as an add-on to the existing automation systems in buildings and grids;
- it operates building and grid elements to minimize costs, including exploitation of demand response opportunities;
- it respects comfort and equipment constraints in buildings and grids;
- it is operable in different configurations which can be selected based on projected costs of needed interventions and expected benefits in operation.

Contribution to the project and Programme objectives (max. 1500 characters)

The main objective of 3Smart is to provide a technological and legislative setup for cross-spanning energy management of buildings, energy grids and major city infrastructures in the Danube region. The output clearly addresses the technological set-up for integrated energy management of buildings and grids, crucial for the coming time of energy system transition which necessitates that the energy system balancing comes at least in part on the side of consumers. It is also very important for shaping the energy regulations in the coming time, as elaborated next.

The DTP major objective is to harmonize policies across different countries in the Danube region in crucial priority fields, one of them being also energy. The tool gives a new unique possibility for technically informed decisions regarding different options in energy regulations for enabling energy transition – from smart meters rollout to decisions on different tariff options that enable decarbonization of the energy sector.

Transnational impact (max. 1500 characters)
The transnational approach in development and piloting of the grid-building cross-spanning energy management tool was important to make it relevant for the entire Danube region. The software modules were developed by teams in several different countries. The tool developed was elaborated and discussed via a sequence of transnational trainings held within first two years of the project execution. The seeds of such region-wide applicable energy management platform are „planted“ in pilots of 5 different countries. Intensive transnational interactions are also enabled via pilot study visits where the tool application on the pilot sites was reviewed and discussed. The engaged local target groups will enable faster convergence of national energy regulatory setups with steeper learning curves building on a broader experience.
Contribution to EUSDR actions and/or targets (max. 1500 characters)
The output contributes to Priority Area 2 "To encourage more sustainable energy" of the EUSDR within which the following actions are required: „To explore the possibility to have an increased energy production originating from local renewable energy sources to increase the energy autonomy”, „To promote energy efficiency and use of renewable energy in buildings and heating systems“, „To facilitate networking and cooperation between national authorities in order to promote awareness and increase the use of renewable energies“. The developed cross-spanning energy management tool not just increases energy efficiency and combines it optimally with renewable energy measures, but also unlocks demand response capacities of buildings as largest consumers of energy. It is very important for enabling higher renewable energy integration since the energy system regulation needs to be brought at least in part on the side of consumers within the process of energy system decarbonization.
Performed testing, if applicable (max. 1000 characters)
The tool is tested within 5 pilots in 5 different countries of the Danube region: Croatia, Slovenia, Austria, Bosnia and Herzegovina, Hungary. It shows very promising results on pilots, and yet considerable testing will be performed in the remaining part of the project.
Integration and use of the output by the target group (max. 2000 characters)
The selected target groups for this output are national regulatory agencies who will be able to yield technically informed regulatory decisions by utilizing the tool, SMEs who will be in a position to develop businesses on application of tools to buildings and grids, higher education and research who will be able to provide expert services in tool further development and adaptations to various pilot sites together with SMEs.
Geographical coverage and transferability (max. 1500 characters)

The tool is already now started to be piloted in 5 different 3Smart pilots, within 5 different countries, there are no specific geographical constraints for its usage. It is transferrable to other buildings and grids by using the performed pilots as seeds, but requires expert knowledge for adaptation to a particular pilot site, which actually represents an opportunity for research-based SMEs, enterprises and higher education and research institutions.
Durability (max. 1500 characters)
There is no specific constraint in time duration of output validity. Moreover, it seems that with the coming decarbonization of the energy system this output will only gain on its importance and significance.
Synergies with other projects/ initiatives and / or alignment with current EU policies/ directives/ regulations, if applicable (max. 1500 characters)
The 3Smart tool has already become a starting point for new developments in several projects like the project PC-ATE Buildings (Development of system for predictive control and autonomous trading of energy in buildings) led by company Klimaoprema and partnered by UNIZGFER, funded by ERDF funds in Croatia or like Store4HUC (Integration and smart management of energy storages at historical urban sites) funded by Interreg Central Europe in which UNIZGFER participates as partner. The tool is very relevant for transposition of EU directive Clean Energy for all Europeans into national policies as it enables to perform technically informed regulatory decisions within the directive adoption process.
Output integration in the current political/ economic/ social/ technological/ environmental/ legal/ regulatory framework (max. 2000 characters)
The tool should become a backbone for the future technology streamline for demand response services provision, as a key instrument for energy system decarbonization.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Estimation and prediction submodules for zones management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Mostar Faculty of Mechanical Engineering, Computing and Electrical Engineering
Authors (Partners)	Anita Martinčević, Hrvoje Novak, Vinko Lešić, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić, Gordan Lješić, Boris Crnokić (SVEMOFSR)
Contact person	Anita Martinčević (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of estimation and prediction submodules on the level of building zones for hierarchical management of building subsystems.
Keyword List	Mathematical Model of Building; Thermodynamics; Heating and Cooling Elements, Fan Coil Unit; Radiator, Floor Heating and Cooling, Disturbance Prediction, Neural Network, Estimation, Prediction

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.4.1	Mario Vašak (UNIZGFER)
v0.5	15 January 2019	Updated version	Mario Vašak (UNIZGFER)
v0.9	7 June 2019	Updated version with contributions of partners	Mario Vašak, Anita Martinčević, Hrvoje Novak (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
v1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing – and the algorithms are implemented via a sequence of submodules.

The submodules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on zone level prediction and estimation submodules.

Each submodule is presented via an interfacing table that explains what data are used by the submodules as inputs and what are the final output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following estimation and prediction submodules are designed, commissioned and tested on the zone level:

Z.PE.1 – submodule for estimation of a thermodynamic and hydraulic model of a fan coil in off-line operation mode and for estimation of heat input from the fan coil to the zone air in on-line operation mode (tested in UNIZGFER, HEP, STREM school, EPHZHB and EON pilot buildings within 3Smart);

Z.PE.2 -- submodule for estimation of a thermodynamic and hydraulic model of a radiator in off-line operation mode and for estimation of heat input from the radiator to the zone air in on-line operation mode (tested in HEP, IDRIJA school and sports centre and STREM school pilot buildings within 3Smart);

Z.PE.3 -- submodule for estimation of a thermodynamic and hydraulic model of a zone floor heating/cooling unit in off-line operation mode and for estimation of heat input from the floor heating/cooling unit to the zone air in on-line operation mode (tested in STREM retirement and care centre pilot building within 3Smart);

Z.PE.4 – submodule for estimation of a simplified room thermodynamic model, used only in off-line operation (tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

Z.PE.5 – submodule for estimation of heat disturbances and simplified zone model states, used only in on-line operation (tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

Z.PE.6 – submodule for estimation of a prediction model for heat disturbance in a zone in off-line operation and for prediction of heat disturbance evolution in on-line operation (tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

Z.PE.7 – submodule for prediction of comfort requirements in a zone, used only in on-line operation (tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

Z.PE.8 – submodule for estimation of a prediction model for zone heating/cooling energy consumption in off-line operation and for prediction of zone heating/cooling energy consumption in on-line operation (tested in EON pilot building within 3Smart);

Z.PE.9 – submodule for estimation of a prediction model for zone temperature in off-line operation and for prediction of zone temperature in on-line operation (tested in EON pilot building within 3Smart).

In the following chapters the mentioned submodules are presented with their interface tables showing which data they use as inputs and which data they provide as outputs to be at the disposal to other submodules. Detailed explanations of algorithms behind each of the submodules are provided in the previously delivered 3Smart document D4.4.1 (related to prediction and estimation). For completeness, D4.4.1 prediction and estimation part is annexed (Annex 1).

Source and sink for the data used by submodules is a properly structured 3Smart database. Its structure in the part concerned by the zone level prediction and estimation submodules is provided in Annex 2.

2 Z.PE.1 submodule

Z.PE.1 submodule is used for estimation of a thermodynamic and hydraulic model of a fan coil in off-line operation mode and for estimation of heat input from the fan coil to the zone air in on-line operation mode. Within 3Smart it is tested in UNIZGFER, HEP, STREM school, EPHZHB and EON pilot buildings.

The submodule interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Required inputs for the Z.PE.1 submodule.

Variable name	Notation	Description
Historical temperature profile from zone (minute-scale of the sampling time)	T	Data taken from the database
Historical profile of fan actuation in the zone (minute-scale of the sampling time)	FS	Data taken from the database
Historical temperature profile of the supply medium from a calorimeter (minute-scale of the sampling time)	$T_{\mathrm{supply,cal}}$	Data taken from the database
Historical temperature profile of the return medium from a calorimeter (minute-scale of the sampling time)	$T_{\mathrm{return,cal}}$	Data taken from the database
Historical profile of the flow from a calorimeter (minute-scale of the sampling time)	Q_{cal}	Data taken from the database
Historical profile of energy (power) recorded on the calorimeter (minute-scale of the sampling time)	E_{cal} (P_{cal})	Data taken from the database
Historical temperature profile from the return medium temperature sensor on a fan coil (minute-scale of the sampling time)	$T_{\mathrm{return,fc}}$	Data taken from the database
Historical temperature profile from the supply medium temperature sensor on a fan coil (minute-scale of the sampling time)	$T_{\mathrm{supply\_fc}}$	(optional) If not existing, measurement of the temperature on the calorimeter should be used, and additionally a characteristic of the temperature drop along the pipeline from the heat loss model should be used

Table 2.2: Outputs of the Z.PE.1 submodule.

Variable name	Notation	Description
Flow shares between fan coils on the same supply line measured by calorimeters (off-line)	η	Required to be able to calculate heating/cooling medium flow through the individual fan coil unit
Parameters of the fan coil model that relates fan coil actuation, room temperature and medium conditions registered on a calorimeter to fan coil energy transmitted to room air in a defined time period; also parameters of a simple relation between heating energy and electrical energy for fans for different supply medium flows and temperatures	$A_{\mathrm{fc}}(Q_{\mathrm{w}})$, $B_{\mathrm{fc}}(Q_{\mathrm{w}})$, $C_{\mathrm{fc}}(Q_{\mathrm{w}})$, $D_{\mathrm{fc}}(Q_{\mathrm{w}})$	Parameters needed for calculation of maximum energy for the MPC module (Z.MPC.1), for the interface submodule functioning (Z.I.1), and for calculation of energy inputs for identification of a simplified building dynamic model and for on-line estimation of its states and disturbances; electricity consumption model needed on the first higher level MPC modules
Estimated heating or cooling energy provided to the zone air (on-line, minute-level of the sampling time)	P_a^e	

3 Z.PE.2 submodule

Z.PE.2 submodule is used for estimation of a thermodynamic and hydraulic model of a radiator in off-line operation mode and for estimation of heat input from the radiator to the zone air in on-line operation mode. Within 3Smart it is tested in UNIZGFER, IDRIJA school and sports centre and STREM school pilot buildings.

The submodule interface is defined in Table 3.1 and Table 3.2.

Table 3.1. Required inputs for radiator identification submodule.

Variable name	Notation	Description
Historical temperature profile from zone (minute-scale of the sampling time)	T_z	Data taken from the database
Historical profile of valve actuation in the zone (minute-scale of the sampling time)	V_x	Data taken from the database
Historical temperature profile of the supply medium from a calorimeter (minute-scale of the sampling time)	T_w^{cal}	Data taken from the database
Historical temperature profile of the return medium from a calorimeter (minute-scale of the sampling time)	$T_{\mathrm{return,cal}}$	Data taken from the database (might not be needed, but will be available)
Historical profile of the flow from a calorimeter (minute-scale of the sampling	q_{cal}	Data taken from the database

time)		
Historical profile of energy (power) recorded on the calorimeter (minute-scale of the sampling time)	$E_{\mathrm{cal}}\ (P_{\mathrm{cal}})$	Data taken from the database
Historical temperature profile from the return medium temperature sensor on a fan coil (minute-scale of the sampling time)	T_{w}^{out}	Data taken from the database
Historical temperature profile from the supply medium temperature sensor on a fan coil (minute-scale of the sampling time)	T_{w}^{in}	(optional) If not existing, measurement of the temperature on the calorimeter should be used, and additionally a characteristic of the temperature drop along the pipeline from the heat loss model should be used

Table 3.2. Outputs of the radiators identification submodule.

Variable name	Notation	Description
Return medium sensors calibration parameters		
Parameters of the radiator model (for off-line operation)	a, b, C, U_0, n	Parameters needed for calculation of maximum energy for the MPC module, for the interface submodule functioning, and for calculation of energy inputs for identification of a simplified building dynamic model and for on-line estimation of its states and disturbances
Heating input from the radiator to the zone air in on-line operation (for on-line operation)	E_{rad}	

4 Z.PE.3 submodule

Z.PE.3 submodule is used for estimation of a thermodynamic and hydraulic model of a zone floor heating/cooling unit in off-line operation mode and for estimation of heat input from the floor heating/cooling unit to the zone air in on-line operation mode. Within 3Smart it is tested in STREM retirement and care centre pilot building.

The submodule interface is defined in **Table 4.1** and **Table 4.2**.

Table 4.1. Required inputs for floor heating identification submodule.

Variable name	Notation	Description
Historical temperature profile from zone (minute-scale of the sampling time)	T_z	Data taken from the database
Historical profile of valve actuation in the zone (minute-scale of the sampling time)	V_x	Data taken from the database
Historical temperature profile of the supply medium from a calorimeter (minute-scale of the sampling time)	T_{w}^{cal}	Data taken from the database
Historical temperature profile of the return medium from a calorimeter (minute-scale of the sampling time)	$T_{\mathrm{return,cal}}$	Data taken from the database (might not be needed, but will be available)
Historical profile of the flow from a calorimeter (minute-scale of the sampling time)	Q_{cal}	Data taken from the database
Historical profile of energy (power) recorded on the calorimeter (minute-scale of the sampling time)	$E_{\mathrm{cal}}\ (P_{\mathrm{cal}})$	Data taken from the database
Historical temperature profile from the return medium temperature sensor on a floor heating system (minute-scale of the sampling time)	T_{w}^{out}	Data taken from the database
Historical temperature profile from the supply medium temperature sensor on a floor heating system (minute-scale of the sampling time)	T_{w}^{in}	(optional) If not existing, measurement of the temperature on the calorimeter should be used, and additionally a characteristic of the temperature drop along the pipeline from the heat loss model should be used

Table 4.2. Outputs of the floor heating identification submodule.

Variable name	Notation	Description
Return medium sensors calibration parameters		
Parameters of the floor heating/cooling element model	$A_{\mathrm{fh}}, B_{\mathrm{fh}},$ $C_{\mathrm{fh}}, D_{\mathrm{fh}}$	Parameters needed for calculation of maximum energy for the MPC module, for the interface submodule functioning, and for calculation of energy inputs for identification of a simplified building dynamic model
Heating/cooling energy input from the element to the zone air		

5 Z.PE.5 submodule

Z.PE.5 is a submodule used for estimation of heat disturbances and simplified zone model states, and it is used only in on-line operation. Within 3Smart it is tested in all 8 pilot buildings, i.e. in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot building.

The submodule interface is defined in the following tables (Table 5.1, Table 5.2).

Table 5.1: Required inputs for the submodule for identification of simplified building thermodynamic model

Variable name	Variable annotation	Variable description
Parameters of the simplified building thermal dynamics model	$A_{room}, B_{room}, C_{room}, D_{room}$	Model identified with the procedure from above
Temperature measurement in rooms/zones of the building (minute-scale of the sampling time)	T	Current temperature measurement in room/zone
Outdoor temperature (minute-scale of the sampling time)	T_{o}	Current outdoor temperature measurement
Solar irradiance estimation on all relevant building surfaces (minute-scale of the sampling time)	I_{solar}	Current amount of the solar radiation on different surfaces of the building (estimated from local measurements)
Energy inputs from heating/cooling elements in zones (minute-scale of the sampling time)	E^{T} (noted herein as E_{t})	Current amount of the energy input from the heating/cooling elements in zones calculated based on the heating/cooling element model

Table 5.2: Outputs of the submodule for identification of simplified building thermodynamic model

Variable name	Variable annotation	Variable description
Estimated states of the simplified building thermal dynamics model	x_0 (noted herein as $\begin{bmatrix} T_1(0) \\ T_2(0) \end{bmatrix}$)	States needed for the MPC module on the zone level
Estimated heat disturbance in zone	E_0^{D} (noted herein as $E_{\mathrm{d}}(0)$)	Current disturbances needed for the MPC module, for the interface submodule, and also for disturbance prediction

6 Z.PE.6 submodule

Z.PE.6 is a submodule for estimation of a prediction model for heat disturbance in a zone in off-line operation and for prediction of heat disturbance evolution in on-line operation. It is tested in all 8 pilot buildings within 3Smart.

The module input and output data are provided within the following tables.

Table 6.1: Required inputs for heat disturbance prediction submodule.

Variable name	Variable annotation	Variable description
Estimated heat disturbance in zone	E_{d}	Profile of the estimated heat disturbance in the past needed for off-line model tuning; recent values needed for on-line execution
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I_{diff}^{h}, I_{dir}^{\mathrm{n}}$ Remaining pilot sites: $T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

Table 6.2: Outputs of the heat disturbance prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_d	Needed for on-line operation of the submodule.
Predicted heat disturbance evolution per zone (for on-line operation of the submodule)	$(E_{\mathrm{d}})_{\mathrm{N}}$	Needed for the MPC module on the zones level.

7 Z.PE.7 submodule

Z.PE.7 is a submodule for prediction of comfort requirements in a zone, and it is used only in on-line operation. Within 3Smart it is tested in all 8 pilot buildings.

The interface for this module is provided within the following tables.

Table 7.1: Required inputs for comfort setpoint prediction submodule.

Variable name	Variable annotation	Variable description
Comfort setpoint in the zone	SP	Profile of the comfort setpoints selected in the past needed for off-line model tuning; recent values needed for on-line operation
Zone control mode	CM	Integer showing which operation mode of the heating/cooling system is selected in the zone (off, auto, fixed fan speed/valve openness).
Building HVAC system operation schedule	SC	Data showing when is the HVAC system for heating/cooling turned on/off.
Possible extension: Connection with the company business data.		Connection point between the EMS and the business information system of a company (travel orders, vacations, sick leaves, different known occupancy schedules for meetings/lectures...)

Table 7.2: Outputs of the comfort setpoint prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{SP}	Needed for on-line operation of the submodule.
Predicted comfort setpoint evolution per zone (for on-line operation of the submodule)	$(SP)_N$	Needed for the MPC module on the zones level.

8 Z.PE.8 submodule

Z.PE.8 is a submodule for estimation of a prediction model for zone heating/cooling energy consumption in off-line operation and for prediction of zone heating/cooling energy consumption in on-line operation. Within 3Smart it is tested in EON pilot building.

The module interface is provided in the following tables.

Table 8.1: Required inputs for thermal energy consumption prediction submodule.

Variable name	Variable annotation	Variable description
Thermal energy consumption in zone	E_t	Profile of the thermal energy consumption in zone needed for off-line model tuning; recent estimates of the thermal energy consumption needed for on-line operation
Weather measurements	$T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, global horizontal and tilted global irradiance.
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

Table 8.2: Outputs of the thermal energy consumption prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{et}	Needed for on-line operation of the submodule.
Predicted thermal energy consumption evolution per zone (for on-line operation of the submodule)	$(E_t)_N$	Needed for the MPC module on the zones level.

9 Z.PE.9 submodule

Z.PE.9 is a submodule for estimation of a prediction model for zone temperature in off-line operation and for prediction of zone temperature in on-line operation. Within 3Smart it is tested in EON pilot building.

The module interface is provided in the following tables.

Table 9.1: Required inputs for zone temperature prediction submodule.

Variable name	Variable annotation	Variable description
Temperature in zone	T_z	Profile of the temperature in zone needed for off-line model tuning; recent measurements of zone temperature needed for on-line operation
Weather measurements	$T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, global horizontal and tilted global irradiance.
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

Table 9.2: Outputs of the zone temperature prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{tz}	Needed for on-line operation of the submodule.
Predicted temperature evolution per zone (for on-line operation of the submodule)	$(T_z)_N$	Needed for the MPC module on the zones level.

Bibliography

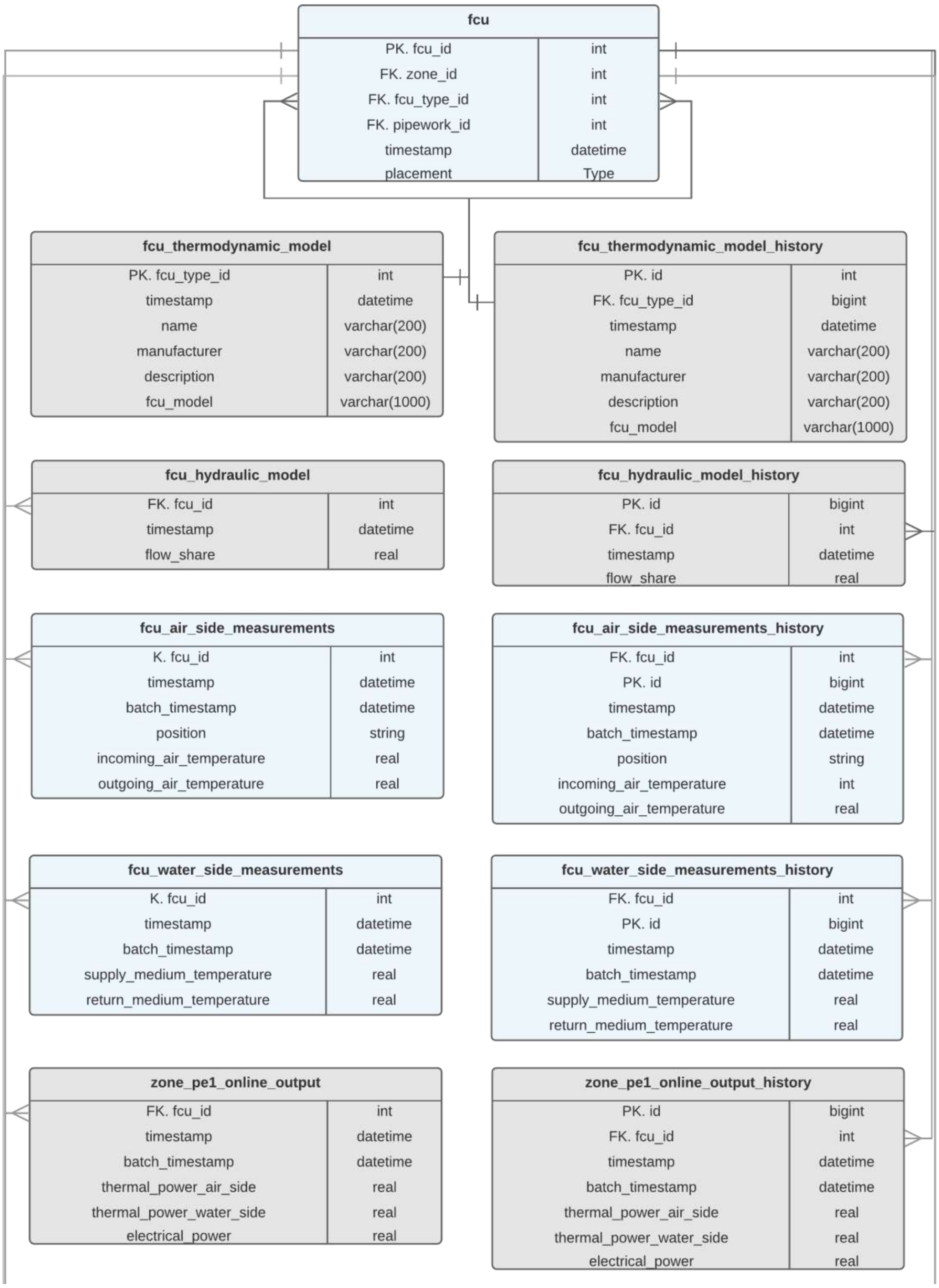
[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 – Open software module for zone consumption management – Estimation and prediction submodules

Annex 1 is provided as a separate document.

Annex 2 – 3Smart database organization for open software module for zone consumption management – Estimation and prediction submodules

Z.PE.1



Z.PE.6.

Input data database structure:

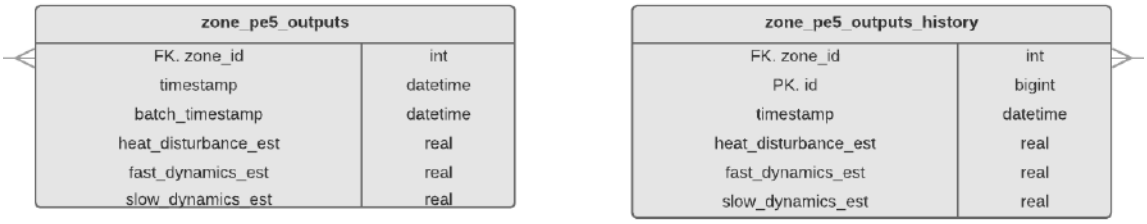


Figure 1. Current and historical estimated heat disturbance zone data database structure.

Figure 2. Weather measurements data database structure.

weather_predictor	
PK. weather_predictor_id	int
weather_predictor_timestamp	datetime
weather_predictor_name	varchar(100)
weather_predictor_description	varchar(100)
weather_predictor_sample_time	int

weather_prediction	
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

weather_prediction_history	
PK. id	bigint
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

Figure 3. Weather forecast data database structure.

Output data database structure:

zone_pe6_outputs	
FK. zone_id	int
timestamp	datetime
heat_disturbance_pred	varchar(2000)

zone_pe6_outputs_history	
FK. zone_id	int
PK. id	bigint
timestamp	datetime
heat_disturbance_pred	varchar(2000)

Figure 4. Current and historical zone heat disturbance prediction data database tables.

Z.PE.7.

Input data database structure:

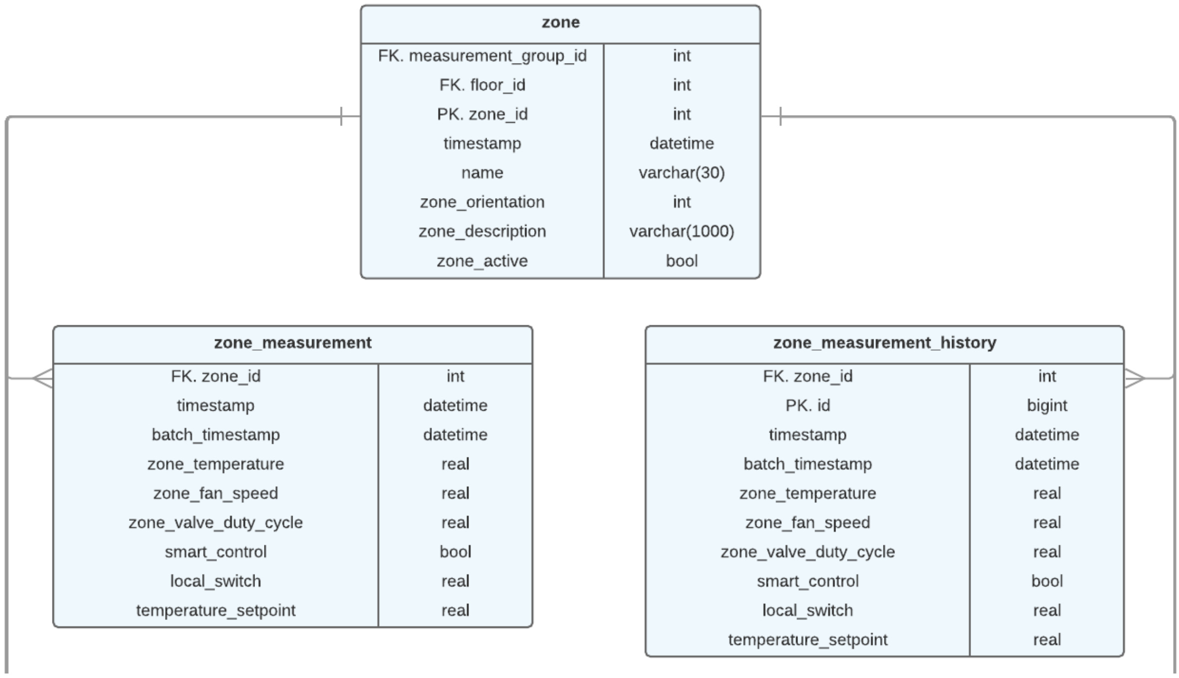


Figure 5. Current and historical zone setpoint and current opreation mode (local_switch variable) data database structure.

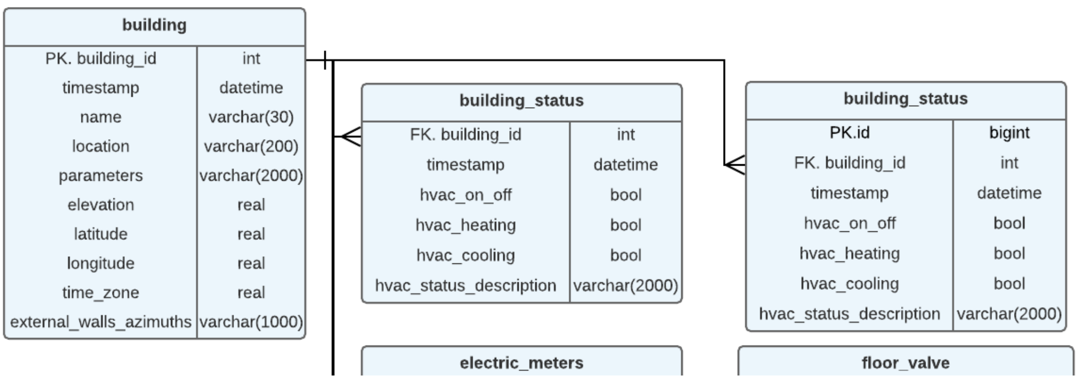


Figure 6. HVAC system operation schedule data database structure.

Output data database structure:

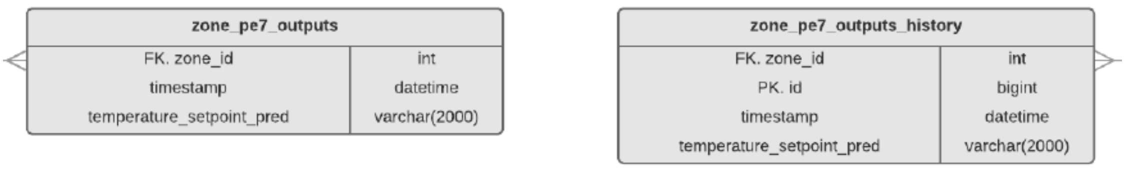


Figure 7. Current and historical zone setpoint prediction data database tables.

Z.PE.8.

Input data database structure:

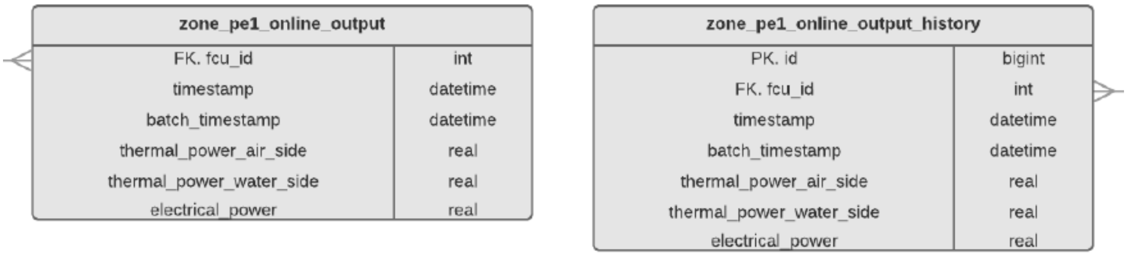


Figure 8. Current and historical zone thermal energy consumption data database structure.

Figure 9. Weather measurements data database structure.

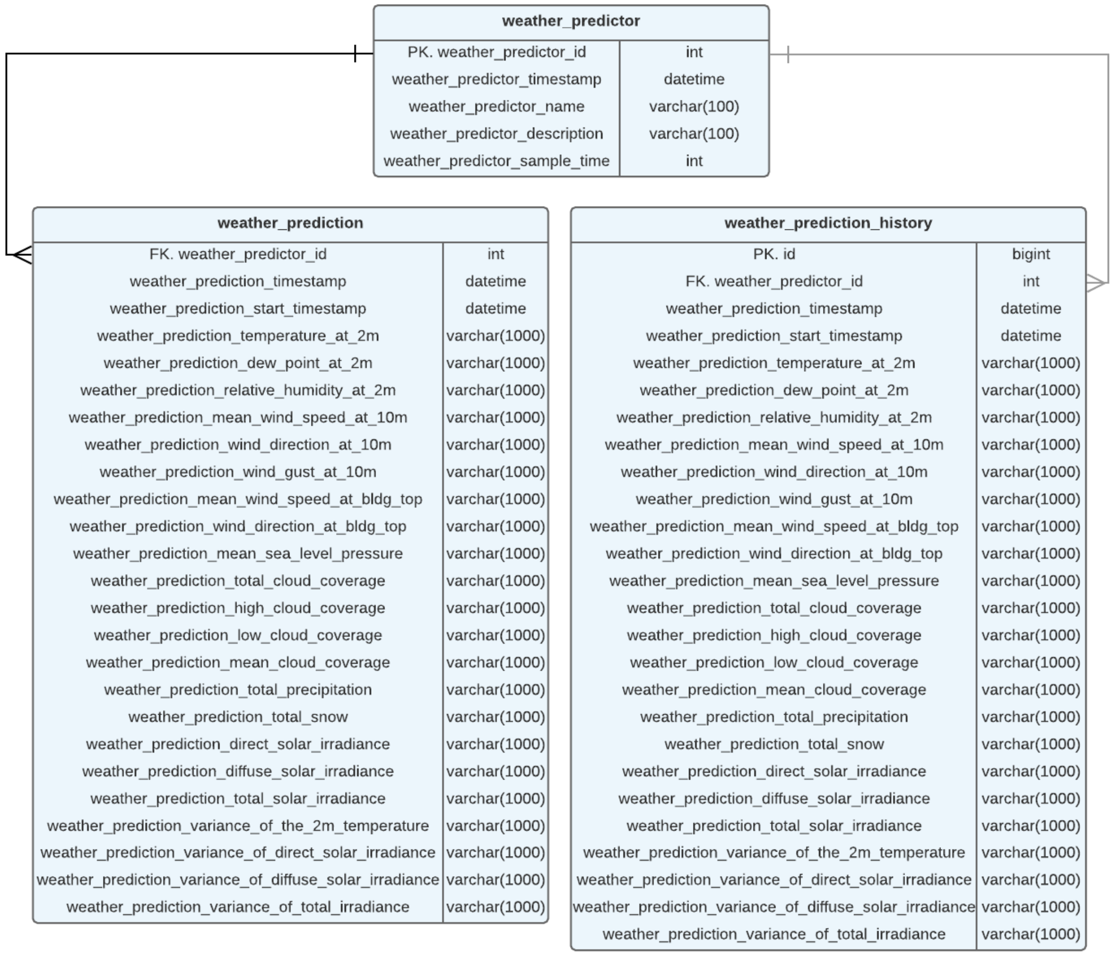


Figure 10. Weather forecast data database structure.

Output data database structure:

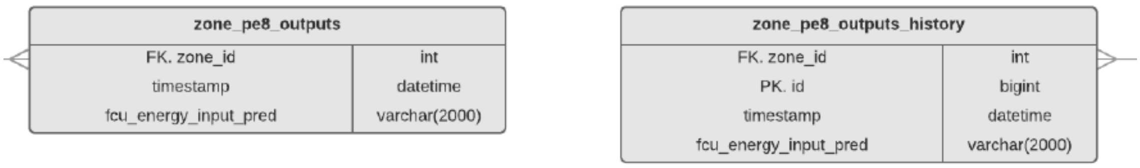


Figure 11. Current and historical zone thermal energy consumption prediction data database tables.

Z.PE.9.

Input data database structure:

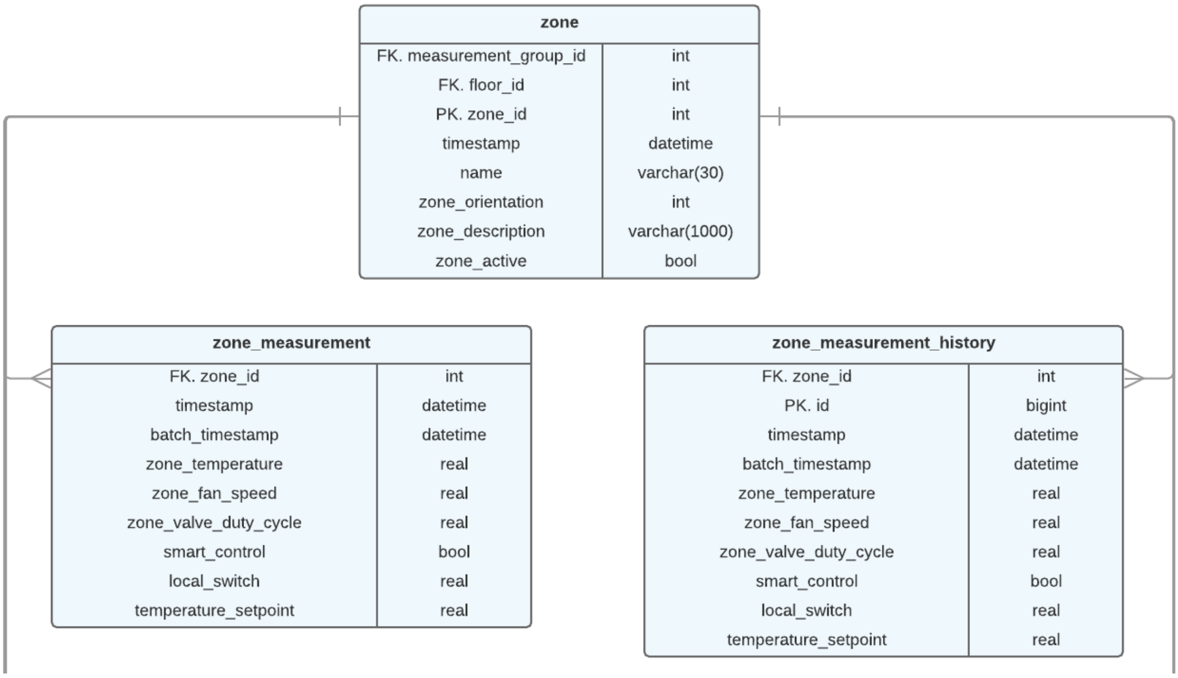


Figure 12. Current and historical zone temperature data database structure.

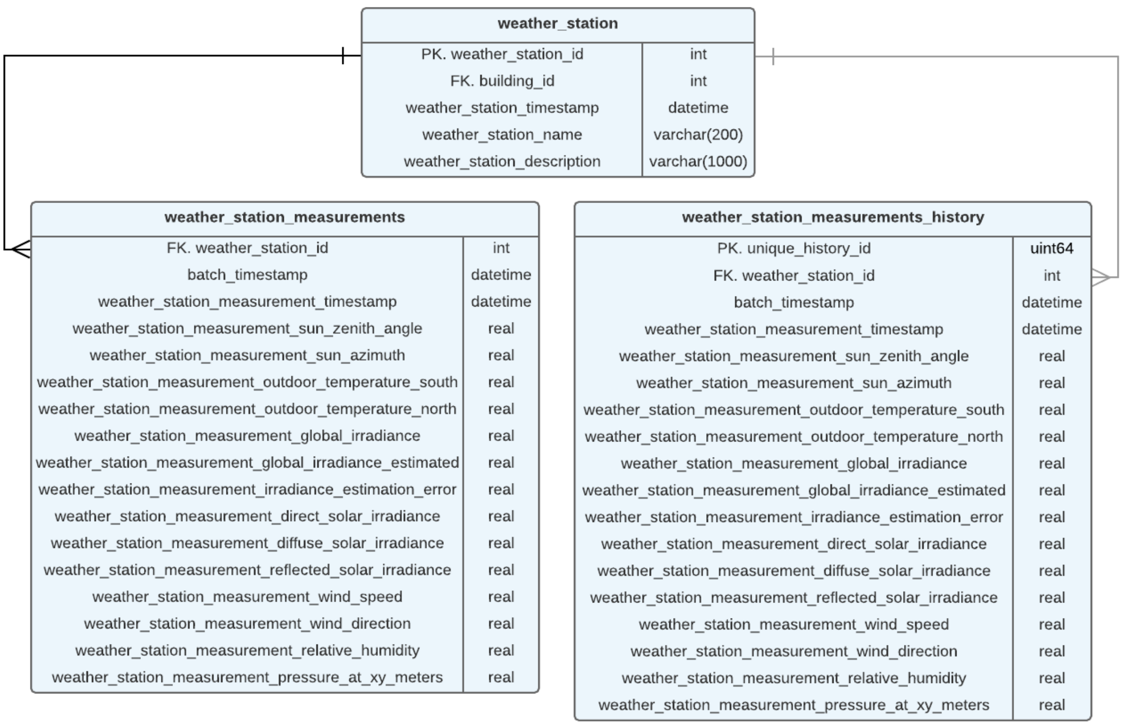


Figure 13. Weather measurements data database structure.

weather_predictor	
PK. weather_predictor_id	int
weather_predictor_timestamp	datetime
weather_predictor_name	varchar(100)
weather_predictor_description	varchar(100)
weather_predictor_sample_time	int

weather_prediction	
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

weather_prediction_history	
PK. id	bigint
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

Figure 14. Weather forecast data database structure.

Output data database structure:

zone_pe9_outputs	
FK. zone_id	int
timestamp	datetime
zone_temperature_pred	varchar(2000)

zone_pe9_outputs_history	
FK. zone_id	int
PK. id	bigint
timestamp	datetime
zone_temperature_pred	varchar(2000)

Figure 15. Current and historical zone temperature prediction data database tables.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 ZONE PREDICTION AND ESTIMATION

Open software module for zone consumption management – Estimation and prediction submodules

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER)
Authors (Partners)	Anita Martinčević, Hrvoje Novak, Vinko Lešić, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić, Gordan Lješić, Boris Crnokić (SVEMOFSR)
Contact person	Anita Martinčević (UNIZGFER) (Chapters 2-5), Hrvoje Novak (UNIZGFER), (Chapters 8-9), Ivo Bevanda (SVEMOFSR) (Chapter 6), Petar Marić (SVEMOFSR) (Chapter 7)
Abstract (for dissemination)	The deliverable gives a comprehensive report on the estimation and prediction techniques used when developing estimation and prediction submodules for hierarchical management of building subsystems.
Keyword List	Mathematical Model of Building; Thermodynamic; Heating and Cooling Elements, Fan Coil Units; Radiator, Floor Heating and Cooling, Disturbance Prediction, Neural Network

Revision history

Revision	Date	Description	Author (Organization)
v0.1	28 September 2017	First draft of Chapter 4 with results only for one zone	Anita Martinčević (UNIZGFER)
v0.2	9 October 2017	Added results for one complete supply duct	Anita Martinčević (UNIZGFER)
v0.3	19 October 2017	Complete draft version of Chapter 4	Anita Martinčević (UNIZGFER)
v0.4.	14. November 2017	Chapter 4 updated with results of medium heat capacity estimation.	Anita Martinčević (UNIZGFER)
v0.5.	14. November 2017	First draft of Chapter 5	Hrvoje Novak (UNIZGFER)
v0.6.	14. November 2017	First draft of Chapter 2 and Chapter 3	Anita Martinčević (UNIZGFER)
v0.7.	15. November 2017	First draft of the Chapters 1-5	Anita Martinčević, Hrvoje Novak, Mario Vašak (UNIZGFER)
v0.8.	16. November 2017	First draft of the radiator and floor heating identification chapters	Ivan Bevanda, Petar Marić (SVEMOFSR)
v0.9	15. December 2017	Final version of radiator and floor heating identification chapters	Ivan Bevanda, Petar Marić (SVEMOFSR)
v1.0	17. December 2017	First draft of the whole deliverable	Anita Martinčević, Hrvoje Novak, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
v1.1	19. December 2017	Final draft version	Anita Martinčević, Hrvoje Novak, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
v1.2	22 December 2017	Final version	Anita Martinčević, Hrvoje Novak, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
v1.3	1 June 2019	Update according to the feedback from operation on pilots	Anita Martinčević, Hrvoje Novak, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
v2.0	30 June 2019	Integration as Annex 1 to D4.5.3 and quality checked	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

This document represents an annex to D4.5.3 zone prediction and estimation modules which describes the logic behind each of the modules. It is organized as follows.

In Chapter 1 the interface of the submodule for identification of simplified building thermodynamic model is presented.

In Chapter 2 general methodology for on-line estimation of heat disturbance flux is described.

Chapter 3 gives a comprehensive report on the identification procedure for the development of thermohydraulic FCU model. The developed models are tested on a real data collected from the Faculty's living lab. The proposed identification procedures stand out due to their simplicity, cost-effectiveness (minimal sensor configuration), noninvasiveness and amount of time it takes to identify the models. The validated models can subsequently be used to predict the real FCU behaviour, as a crucial part of a synthesis of an MPC based controller for zone temperature control, to predict the electrical or thermal loads or as a part of an advanced diagnosis and fault detection algorithms.

In Chapter 4 detailed analysis of radiator thermodynamic behaviour is given with developed procedure for identification of an on-site radiator tested on simulations.

Chapter 5 describes the thermodynamic behaviour of floor heating and cooling systems and gives a procedure for identification of the real system tested on data from verified building simulation program IDA – Indoor climate and energy.

In Chapter 6 submodule for prediction of heat disturbance profile in a building zone is presented through implementation of artificial neural networks.

Chapter 7 outlines the submodule for comfort setpoint prediction in zones.

Chapters 8 and 9 outline submodules that use neural networks for predicting zone energy demands and temperatures.

1 Submodule for identification of a simplified building thermodynamic model (Z.PE.4)

Mathematical model of a building is a basis for the implementation of Model Predictive Control algorithm for zone temperature control. Numerous software packages specialized for modelling of the building thermal behaviour exist on the market [13]. Although those programs can provide very accurate models, they are usually also highly nonlinear and of a high order. The most popular building modelling framework consists of using the resistance-capacitance (RC) network to model thermodynamic processes in buildings. The RC network models are established as simple, computationally efficient and accurate. The problem of RC representation is a fast increase in the model complexity with an increase of building zones. To be applicable for the control system design generally, the model of the process should be simple and yet accurate enough. Besides modelling based on first principles, building model can be found by using the advanced estimation techniques.

Depending on the assumed model structure and numbers of unknowns, models to be estimated can be classified as *i*) white-box models based on first principles *ii*) black-box models where both structure and parameters are unknown or *iii*) grey-box models where only model structure is known based on first principles or some other priory knowledge. Models based on first principles, so called white-box models, have the strongest physical basis at the expense of high order and often nonlinearity. Additionally, information for development of such models are often unavailable or hardly measurable. Black-box models typically require long period of an operational data for learning which are often unavailable or require a long period of data collecting before the implementation of the algorithm. Grey-box models (or often referred to as semi-physical models) encompass the advantages of both white and black-box models. Benefits of grey-box models as opposed to black-box models are in the fact that priory information and physical knowledge can be incorporated directly. This typically results with fewer parameters, which are, due to their physicality, valid over wider ranges of operation. As opposed to white-box models grey-box models tend to give more reproducible results and less bias [14].

Unscented Kalman Filter (UKF) is an algorithm that uses a series of noisy and inaccurate measurements observed over time to estimate unknown system states, parameters or even both [1][5]. The main premise behind the UKF is that it is easier to approximate a Gaussian distribution than an arbitrary nonlinear function. Use of UKF for estimation of building models was already reported in [3][4]. In [3] UKF is used to simultaneously estimate states and parameters of the parameter-adaptive building model of a single zone placed in Michigan. While the model was built on first principles some parameters of the model such as thermal resistance adopted negative values which is not physically possible. Motivated by this, the constraints are introduced into building estimation problem to improve the performance of the estimation, reduce the hyperspace of possible parameters, assure physicality and improve the filter convergence [7][8][9][10].

The submodule interface is defined in the following tables (Table 1.1, Table 1.2). In the following subsections an overview of the methodology used to develop the submodule is given. The UKF estimation principle with detailed descriptions of the Interval Constrained Unscented Transformation and general problem of the simultaneous states and parameters estimation is described. Afterwards, the model form selection is justified and test results of the application of the developed algorithms are given.

Table 1.1: Required inputs for the submodule for identification of simplified building thermodynamic model

Variable name	Variable annotation	Variable description
Historical temperature profile measurement in rooms/zones of the building (minute-scale of the sampling time)	T (noted herein as T_1)	Data from the database
Outdoor temperature profile measurement (minute-scale of the sampling time)	T_o	Data from the database
Solar irradiance estimation on all relevant building surfaces (minute-scale of the sampling time)	I_{solar} (noted herein as I_{dif}, I_{dir})	Data from the data base
Historical profile of energy inputs from heating/cooling elements in zones (minute-scale of the sampling time)	E^{T} (noted herein as E_{t})	Stored in the database or calculated based on the heating/cooling element model

Table 1.2: Outputs of the submodule for identification of simplified building thermodynamic model

Variable name	Variable annotation	Variable description
Parameters of the simplified building thermodynamic model	$A_{room}, B_{room}, C_{room}, D_{room}$	Model to be used for zone-level MPC, states estimation of the simplified model, as well as the estimation of heating/cooling disturbances

1.1 Unscented Kalman Filter

Kalman filter is an algorithm that uses a series of noisy and inaccurate measurements observed over time to estimate unknown system states, parameters or both. Early development of the Kalman filter dates to the early sixties. One of the primary developers of the Kalman filter theory was Rudolf E. Kalman by whom the filter is named [11]. Since then Kalman filter has found its place in many applications: guidance, robotics, control, signal processing, etc. With years many extensions and generalizations of the Kalman filter have been developed to adjust the filter to the specific systems. The most known extensions of the Kalman filter which can be applied to nonlinear systems are Extended Kalman Filter (EKF) and UKF. Although both mentioned filters are suitable for nonlinear systems, it has been shown that UKF outperforms EKF in most applications. The basic idea of the EKF is first order linearization of the nonlinear function and the Kalman filtering of the linearized system. To implement the EKF algorithm first constraint is differentiability of the system functions. Also, calculation of the Jacobian matrices may be difficult for higher order systems. Furthermore, linear approximation may not be appropriate for some systems which can result with corrupted mean and covariance of the states and at the end divergence of the algorithm. The UKF, which is derivative-free, successfully overcomes the disadvantages of the EKF by using a deterministic sampling approach.

The UKF belongs to a bigger class of filters called Sigma-Point Kalman Filters. Instead of analytical linearization of the nonlinear functions UKF uses a deterministic sampling approach to capture mean and covariance of the estimates with a minimal set of sample points. Since the spread of the variables is also considered, UKF can be accurate up to the second order in estimating mean and covariance. Consider the nonlinear system represented with the following standard discrete-time equations:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}, \tag{1-1}$$

$$y_k = h(x_k) + v_k, \tag{1-2}$$

where $x_k \in \mathbb{R}^{\mathrm{n_x}}$ is the system state, $w \in \mathbb{R}^{\mathrm{n_w}}$ the process noise, $v \in \mathbb{R}^{\mathrm{n}_v}$ observation noise, $u \in \mathbb{R}^{\mathrm{n_u}}$ the system input vector and y the noisy observation of the system. The nonlinear functions $f(.)$ and $h(.)$ are not necessarily differentiable. For the identification of a continuous time system $f(.)$ represents integration of the continuous time function over a unit-sample time interval. For the case of simultaneous estimation of parameters and states, unknown parameters are treated as additional states. Even for the linear model case, simultaneous states and parameters estimation makes the resulting identification nonlinear, i.e. performed on a nonlinear state-update model.

For system and measurement noises considered as additive, the UKF algorithm basically consists of the following steps:

- **Initialization at $k = 0$**
  The initial state x_0 is a random vector with known mean $\hat{x}_0 = E[x_0]$ and covariance:

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]. \tag{1-3}$$

- **For $k = 1, 2, 3, \ldots$**

(1) **Prediction and time update**

Let χ_{k-1}^{a} be a set of $2N + 1$ sigma points (where $N = n_x$) defined as:

$$\chi_{i,k-1}^{a} = \begin{cases} \hat{x}_{k-1} & i = 0 \\ \hat{x}_{k-1} + \gamma\sqrt{P_{x_{k-1}}} & i = 1,\ldots,N \\ \hat{x}_{k-1} - \gamma\sqrt{P_{x_{k-1}}} & i = N+1,\ldots,2N \end{cases}, \tag{1-4}$$

Parameter γ is a scaling parameter defined as:

$$\gamma = \sqrt{N+\lambda}, \quad \lambda = \alpha^2(N+\kappa) - N, \tag{1-5}$$

where α and κ are tuning parameters. To guarantee the semi-definiteness of the covariance matrix $\kappa \geq 0$, a good default choice is $\kappa = 0$. The parameter α defines the spread of the sigma points around $\hat{x}$ and is usually set to a small positive value, $1^{-4} \leq \alpha \leq 1$. The i^{th} sigma point is the i^{th} column of the sigma point matrix $\chi_{i,k-1}^{a}$.

Time-update equations

Transform the sigma points through the state-update nonlinear function:

$$\chi_{i,k|k-1}^{x} = f\left(\chi_{i,k-1}^{a}, u_{k-1}\right), \quad i = 0,1,\ldots,2N. \tag{1-6}$$

Calculate the a-priori state estimate and a-priori covariance:

$$\hat{x}_k^{-} = \sum_{i=0}^{2N} W_i^{(m)} \chi_{i,k|k-1}^{x}, \tag{1-7}$$

$$P_{\hat{x}_k}^{-} = \sum_{i=0}^{2N} W_i^{(c)} \left(\chi_{i,k|k-1}^{x} - \hat{x}_k^{-}\right)\left(\chi_{i,k|k-1}^{x} - \hat{x}_k^{-}\right)^T + Q_k, \tag{1-8}$$

where Q_k is the process error covariance matrix. The weights $W^{(m)}$ and $W^{(c)}$ are defined as:

$$W_i^{(m)} \triangleq \begin{cases} \frac{\lambda}{N+\lambda} & i = 0, \\ \frac{\lambda}{2(N+\lambda)} & i = 1,\ldots,2N' \end{cases} \tag{1-9}$$

$$W_i^{(c)} \triangleq \begin{cases} \frac{\lambda}{N+\lambda} + (1-\alpha^2+\beta) & i = 0, \\ \frac{\lambda}{2(N+\lambda)} & i = 1,\ldots,2N' \end{cases} \tag{1-10}$$

where β is the parameter used to incorporate the prior knowledge of the distribution of x. For a Gaussian prior optimal choice is $\beta = 2$.

(2) **Measurement update**

Transform the sigma points through the measurement-update function:

$$Y_{i,k|k-1} = h\left(\chi_{i,k|k-1}^{x}\right) \quad i = 0,1,\ldots,2N, \tag{1-11}$$

and calculate the mean and covariance of the measurement:

$$\hat{y}_k^{-} = \sum_{i=0}^{2N} W_i^{(m)} Y_{i,k|k-1}, \tag{1-12}$$

$$P_{\hat{y}_k} = \sum_{i=0}^{2N} W_i^{(c)} \left(Y_{i,k|k-1} - \hat{y}_k^-\right)\left(Y_{i,k|k-1} - \hat{y}_k^-\right)^T + R_k \,, \tag{1-13}$$

where R_k is the measurement noise covariance matrix. The cross covariance is defined as:

$$P_{x_k y_k} = \sum_{i=0}^{2N} W_i^{(c)} \left(\chi_{i,k|k-1}^x - \hat{x}_k^-\right)\left(Y_{i,k|k-1} - \hat{y}_k^-\right)^T \,, \tag{1-14}$$

Kalman gain is defined as:

$$K_k = P_{x_k y_k} P_{\hat{y}_k}^{-1}, \tag{1-15}$$

and the UKF estimate and its covariance are computed from the standard Kalman update equations:

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - \hat{y}_k^-), \tag{1-16}$$

$$P_{x_k} = P_{\hat{x}_k}^- - K_k P_{\hat{y}_k} K_k^T. \tag{1-17}$$

1.1.1 Interval Constrained Unscented Transformation

The candidates to be constrained are a-priori and a-posteriori state estimates, sigma points, sigma points propagated through the non-linear function, etc. [13]. The basic idea of the constrained UKF is to project the unconstrained set $\hat{x}_k$ onto some constrained set $\tilde{x}_k$ by solving the following convex problem:

$$\begin{aligned} \min_{\tilde{x}_k} \quad & (\tilde{x}_k - \hat{x}_k)^T Z_k (\tilde{x}_k - \hat{x}_k) \\ s.t \quad & A_k \tilde{x}_k \le b_k, \end{aligned} \tag{1-18}$$

It can be shown that for $Z_k = I$ and $A_k = I$, where I is the identity matrix, the solution of (1-18), in the case when $\hat{x}_k$ violates the constraints, is $\tilde{x}_k = b_k$. More generally, for constraints introduced as box constraints solving (1-18) gives the same solution as clipping, i.e. setting all points outside the allowed set to the boundaries. Figure 1.1 shows the clipping approach applied on two-dimensional state variable. The sigma points outside a boundary (the dotted lines) are projected back on the constraint boundary. As it can be seen from the Figure 1.1 the constrained sigma-points may not be symmetric, thus the resulting distribution may not be Gaussian. In this case, the weights of these non-symmetric sigma-points need to be adjusted to preserve the Gaussian distribution. The modified Unscented Transformation (UT) which includes the projection of the sigma points outside a constrained set to the boundaries and adjustment of the weights according to the performed modifications is called Interval Constrained Unscented Transformation (ICUT).

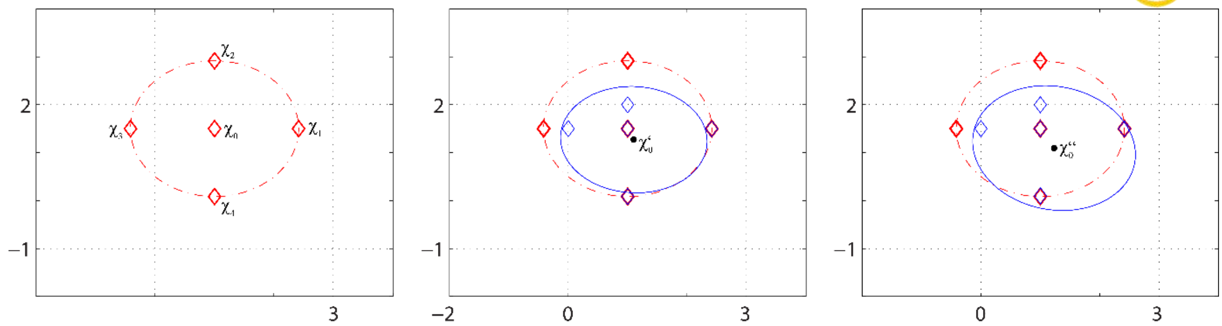


Figure 1.1: Unconstrained and constrained sigma points, constraints indicated by dashed black lines. a) Illustrates the covariance ellipse (dashed line) with unconstrained sigma points (red ◊) and mean $\boldsymbol{x_0}$. Here two of the four unconstrained sigma points $\boldsymbol{\chi_2}$ and $\boldsymbol{\chi_3}$ lie outside of the constraints. In b) only clipping is applied, i.e. all sigma points outside constraints are set to the boundary (blue ◊), result is modified covariance ellipse depicted with solid line and modified mean $\boldsymbol{\chi_0'}$. In c) the interval constraint unscented transformation is applied, result is modified covariance ellipse which include all constrained sigma points (solid line) and modified mean $\boldsymbol{\chi_0''}$.

Standard UT (1-4) is based on a set of deterministically chosen vectors $\chi_{i,k-1}^{a} \in \mathbb{R}^N$ known as sigma points. Sigma points satisfy:

$$\sum_{i=0}^{2N} W_i^{(m)} \chi_{i,k}^{a} = \hat{x}_k \quad \text{and} \quad \sum_{i=0}^{2N} W_i^{(c)} \left(\chi_{i,k}^{a} - \hat{x}_k\right)\left(\chi_{i,k}^{a} - \hat{x}_k\right)^T = P_{x_k} \tag{1-19}$$

with weights $W_i^{(m)}$ and $W_i^{(c)}$ defined by (1-9) and (1-10). In ICUT, $W^{(m)}$for the mean and $W^{(m)}$ for the covariance, are calculated such that when sigma-points do not violate the constraints, the regular weights are selected. If the sigma-points are propagated onto the boundary, the weights vary linearly with the step size [9]. The weights can be positive or negative but, to provide an unbiased estimate, they must obey [12]:

$$\sum_{i=0}^{2N} W_i \quad = 1 \tag{1-20}$$

The ICUT algorithm for calculation of the constrained sigma points with state constraints defined as:

$$L_k \leq x_k \leq U_k, \tag{1-21}$$

where $L_k \in \mathbb{R}^N$ is vector of lower bounds and $U_k \in \mathbb{R}^N$ is vector of upper bounds (the boundaries can vary with each iteration) is listed hereinafter.

Interval Constrained Unscented Transformation

Step size vector ξ_j is defined as:

$$\xi_j \triangleq \min\left(\mathrm{col}_j(\epsilon)\right) \tag{1-22}$$

where $\mathrm{col}_j(\Theta)$ denotes jth column of matrix Θ where matrix Θ is defined as:

$$\epsilon_{i,j} \triangleq \begin{cases} \sqrt{N+\lambda} & if \quad \Phi_{i,j} = 0 \\ \min\left(\sqrt{N+\lambda}, \frac{U_{i,k}-\hat{x}_{k-1}}{\Phi_{i,j}}\right) & if \quad \Phi_{i,j} > 0 \\ \min\left(\sqrt{N+\lambda}, \frac{L_{i,k}-\hat{x}_{k-1}}{\Phi_{i,j}}\right) & if \quad \Phi_{i,j} < 0 \end{cases}, \tag{1-23}$$

$$\Phi = \left[\sqrt{P_{x_k}} \quad -\sqrt{P_{x_k}}\right] \tag{1-24}$$

Constrained sigma-points $\chi_{i,k-1}^{a}$ are:

$$\chi_{i,k-1}^{a} \triangleq \begin{cases} \hat{x}_{k-1} & j = 0 \\ \hat{x}_{k-1} + \xi_j \mathrm{col}_j(\Sigma) & 1 \le j \le 2N \end{cases}. \tag{1-25}$$

Adjusted weights are:

$$W_i^{(m)} \triangleq \begin{cases} \frac{\lambda}{N+\lambda} & i = 0, \\ \frac{\lambda}{2(N+\lambda)\sum_{i=1}^{2N}\xi_i} \cdot \xi_i & i = 1,\ldots,2N \end{cases}, \tag{1-26}$$

$$W_i^{(c)} \triangleq \begin{cases} \frac{\lambda}{N+\lambda} + (1-\alpha^2+\beta) & i = 0, \\ \frac{\lambda}{2(N+\lambda)\sum_{i=1}^{2N}\xi_i} \cdot \xi_i & i = 1,\ldots,2N \end{cases}, \tag{1-27}$$

1.1.2 Kalman Filter and parameter identification

A possible way to encounter with parameter Θ estimation is to treat the unknown parameter vector Θ as a dynamical variable itself.

$$\dot{\Theta} = 0. \tag{1-28}$$

Although the parameters are constant within the state dynamics, they are modified in each recursion step by measurement update equation, as long as their current values deviates from the true one. It is important to note that even if the system model (1-1) is linear the joint model for estimation of states in parameters is bilinear in both states and parameters. The joint identification problem for simultaneous estimation of parameters and states, is as follows:

$$\begin{bmatrix} x_k \\ \Theta_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, u_{k-1}, \Theta_{k-1}) \\ \Theta_{k-1} \end{bmatrix} + w_{k-1}, \tag{1-29}$$

$$y_k = h(x_{k-1}, u_{k-1}, \Theta_{k-1}) + v_k. \tag{1-30}$$

When parameters are the only unknowns (all states are measurable) separate estimation of parameters only can be done with modified form of UKF algorithm [6] where discrete transition function for parameters has the following form:

$$\Theta_k = \Theta_{k-1} + w_{k-1} \tag{1-31}$$

$$y_k = h(x_{k-1}, u_{k-1}, \Theta_{k-1}) + v_k, \tag{1-32}$$

An $\mathcal{O}(ML^2)$ algorithm, as opposed to $\mathcal{O}(L^3)$ is possible by taking advantage of the linear state transition function (1-28).

1.2 Building model form selection

By observing time constants of building thermodynamics processes it is evident that the dominant time constants are related to air temperature. Other, noticeably larger time constants are related to the walls or additional internal masses like furniture due to their higher thermal capacity. The zone thermodynamic can be approximated with two thermal masses: fast dynamic with lower thermal capacity related to the air temperature inside a zone and slow dynamics with a higher thermal capacity related to the solid zone parts (walls and furniture) [2]:

$$p_1^i \dot{T}_1^i = \sum_{j \in \mathcal{V}_i} p_3^{ij} (T_1^j - T_1^i) + p_4^i (T_o - T_1^i) + p_5^i (T_2^i - T_1^i) + p_6^i I_{dir} + p_7^i I_{dif} + P_{t,i}, \tag{1-33}$$

$$p_2^i \dot{T}_1^i = p_5^i (T_1^i - T_2^i), \tag{1-34}$$

where T_1^i is the temperature of the fast-dynamic mass which is assumed to be equal to the temperature of the air inside the ith zone, T_2^i is the temperature of the ith zone slow dynamic mass, T_o is outer air temperature, $\mathcal{V}_i$ is a set of zones adjacent to the ith zone, $P_{t,i}$ is measurable thermal load affecting the ith zone, I_{dif} and I_{dir} are diffuse and direct solar irradiances on the external zone wall(s) respectively. The unknown parameters of the ith zone are denoted with $p_{1:7}^i$. Parameters p_1^i and p_2^i correspond to thermal capacities of the ith zone fast-dynamic mass and ith zone slow-dynamic mass, while parameters $p_{3:7}^i$ correspond to the associated thermal resistances.

In steady state, temperature distribution through a solid wall is linear, i.e. the heat transfer between two adjacent zones is proportional to the temperature difference between those zones, where factor of proportionality is overall heat resistance of a wall that separates them. Modern construction materials assure good thermal insulation between the zones. Additionally, temperature distribution in most commercial and public buildings is almost uniform resulting in small heat flux between the zones which makes the estimation of the overall heat resistance of the wall hard. To avoid the problem, simplified thermodynamic model of every zone is estimated separately by assuming $\mathcal{V}_i = \{\ \}$. To identify the model (1-33), (1-34) numerical integration within the filter sampling time T_s^f or discretization with a discretization time T_s selected as a multiple of the filter sampling time should be applied. The building model discretized by Euler method is as follows:

$$\begin{bmatrix} T_1(k+1) \\ T_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 - (p_{d,1}^i + p_{d,3}^i)T_s & p_{d,1}^{i'} T_s \\ p_{d,2}^i T_s & 1 - p_{d,2}^{i'} T_s \end{bmatrix} \begin{bmatrix} T_1(k) \\ T_2(k) \end{bmatrix} + \begin{bmatrix} p_{d,3}^i & p_{d,4}^i & p_{d,5}^i & p_{d,6}^i \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_o \\ I_{dif} \\ I_{dir} \\ E_{t,i} \end{bmatrix}. \tag{1-35}$$

Continuous system parameters $p_{1:7}^i$ can easily be recalculated from the discrete system parameters $p_{d,1:6}^i$ through algebraic functions.

1.3 Test results

Estimation algorithm is tested on an artificial data set with known parameters. The data set is created by taking real building input measurements for a typical south oriented zone (part of UNIZG-FER skyscraper) during 8 days in 2016, and simulating the model with parameters obtained from batch optimization. The simulated slow and fast dynamics temperatures are then used to validate the algorithm. Estimation results are shown in Figure 1.2. with estimated parameters normalized to interval [0,10]. Since the algorithm is tested on normal operation data set and not on the data set containing highly excited temperatures, parameter convergence time is up to one week. Filter successfully follows the states dynamics all the time.

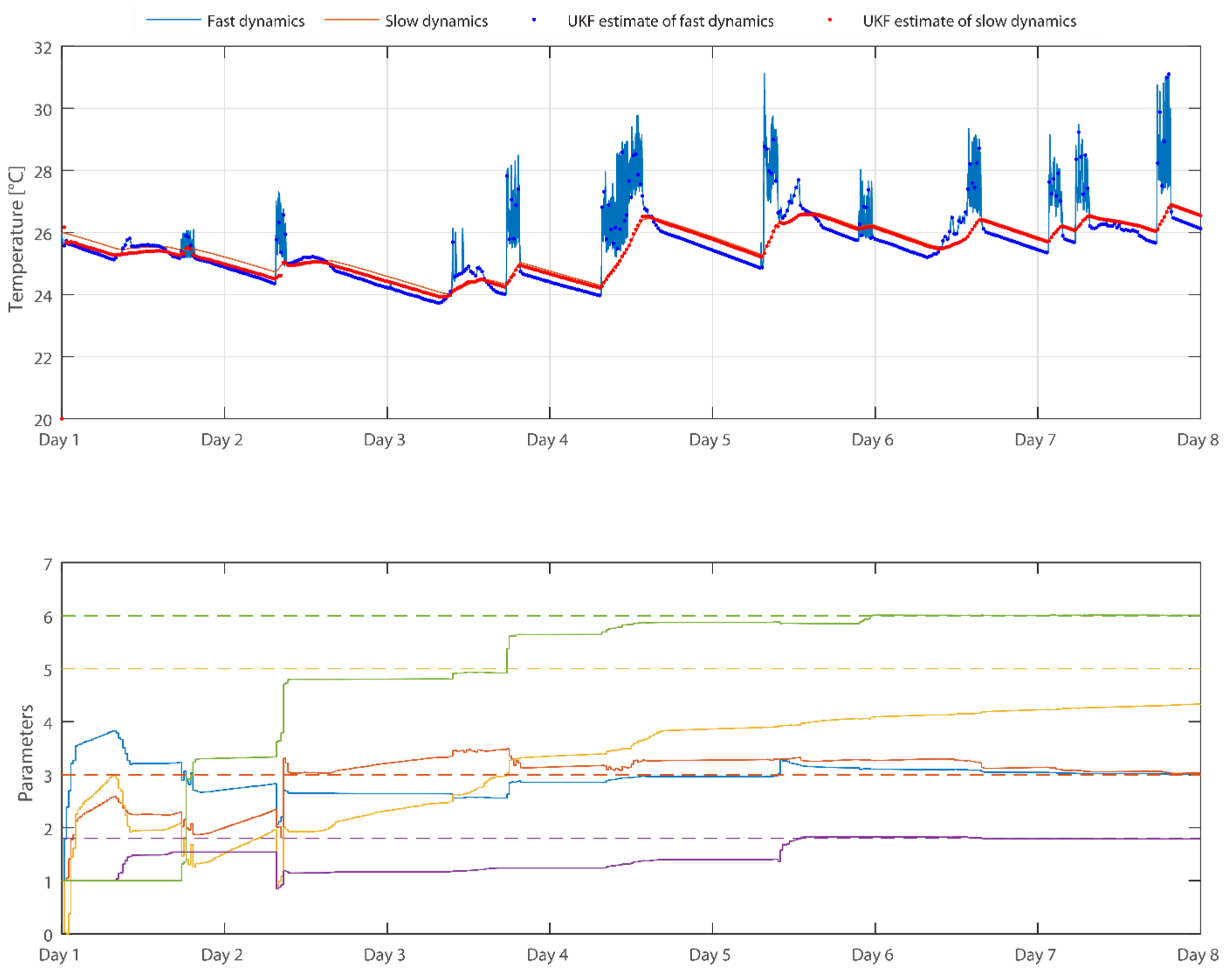


Figure 1.2. Estimation results for a typical zone.

2 Submodule for estimation of the states of the simplified building thermal dynamics model including also the estimation of heat disturbance in zone (Z.PE.5)

Heat disturbance in zone implies sum of additional heat flows in the zone which are not included into the estimated building model or occur due to the changed conditions in the zone compared to the one used for estimation of the zone model. Typical examples of heat disturbances are: window opening, shading position changes, electronic equipment, occupancy, lighting.

Slow dynamics temperature, which is a part of the estimated building model presents substitute variable for all higher thermal capacity element temperatures (e.g. walls and furniture). As such, it is hardly measurable and have to be estimated online.

The submodule interface is defined in the following tables (Table 2.1, Table 2.2).

Table 2.1: Required inputs for the submodule for identification of simplified building thermodynamic model

Variable name	Variable annotation	Variable description
Parameters of the simplified building thermal dynamics model	$A_{room}, B_{room}, C_{room}, D_{room}$	Model identified with the procedure from above
Temperature measurement in rooms/zones of the building (minute-scale of the sampling time)	T (noted herein as T_1)	Current temperature measurement in room/zone
Outdoor temperature (minute-scale of the sampling time)	T_{o}	Current outdoor temperature measurement
Solar irradiance estimation on all relevant building surfaces (minute-scale of the sampling time)	I_{solar} (noted herein as I_{dif}, I_{dir})	Current amount of the solar radiation on different surfaces of the building (estimated from local measurements)
Energy inputs from heating/cooling elements in zones (minute-scale of the sampling time)	E^{T} (noted herein as E_{t})	Current amount of the energy input from the heating/cooling elements in zones calculated based on the heating/cooling element model

Table 2.2: Outputs of the submodule for identification of simplified building thermodynamic model

Variable name	Variable annotation	Variable description
Estimated states of the simplified building thermal dynamics model	x_0 (noted herein as $\begin{bmatrix} T_1(0) \\ T_2(0) \end{bmatrix}$)	States needed for the MPC module on the zone level
Estimated heat disturbance in zone	E_0^{D} (noted herein as $E_{\mathrm{d}}(0)$)	Current disturbances needed for the MPC module, for the interface submodule, and also for disturbance prediction

2.1 Methodology

Heat disturbance E_d is modelled as additive heat flux by introducing the additional heat flux into the system model:

$$\begin{bmatrix} T_1(k+1) \\ T_2(k+1) \end{bmatrix} = A \cdot \begin{bmatrix} T_1(k) \\ T_2(k) \end{bmatrix} + \begin{bmatrix} \mathrm{B} & p_{d,6}^{i} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} T_o(k) \\ I_{dir}(k) \\ I_{diff}(k) \\ E_t(k) \\ E_d(k) \end{bmatrix}, \tag{2-1}$$

where T_1 is zone temperature, T_2 is equivalent temperature of all thermal masses affecting the zone, T_o is outdoor air temperature, direct and diffuse solar irradiance are denoted with I_{dir} and I_{diff}, measured thermal energies from heating/cooling devices are denoted with E_t. System matrices $A \in \mathbb{R}^{2\times2}$ and $B \in \mathbb{R}^{2\times5}$ and parameter $p_{d,6}^{i}$ are obtained through identification procedure for determining the zone temperature dynamics model (Chapter 1). Heat disturbance is denoted with E_d. Due to the similar nature of thermal fluxes E_t and E_d , their impact on the zone temperature is modelled with the same parameter $p_{d,6}^{i}$.

For known outdoor temperature, solar irradiances I_{dir} and I_{diff}, thermal energy E_t and zone temperature T_1, the remaining unknowns are the slow dynamics temperature T_2 and heat disturbance energy input E_d. Stochastic dynamics of the heat disturbance is defined as:

$$E_d(k+1) = E_d(k) + w_d(k), \tag{2-2}$$

where $w_d(k) \sim N(0, Q_d)$ is high covariance system noise with covariance Q_d. By introducing the dynamics of heat disturbance into equation (2-1) the following form, suitable for implementation of linear Kalman Filter [11] is obtained:

$$\begin{bmatrix} T_1(k+1) \\ T_2(k+1) \\ E_d(k+1) \end{bmatrix} = \begin{bmatrix} A & p_{d,6}^{i} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T_1(k) \\ T_2(k) \\ E_d(k) \end{bmatrix} + \mathrm{B} \cdot \begin{bmatrix} T_o(k) \\ I_{dir}(k) \\ I_{diff}(k) \\ E_t(k) \end{bmatrix} + \begin{bmatrix} w_{t1}(k) \\ w_{t2}(k) \\ w_d(k) \end{bmatrix}, \tag{2-3}$$

where process noise regarding temperatures is denoted with w_{t1} and w_{t2}. To successfully capture transient behaviours which can occur on a minute scale the filter sampling time is set to one minute. Results of disturbance heat flux estimation on artificial data set from IDA-ICE [13] are shown in Figure 2.1.

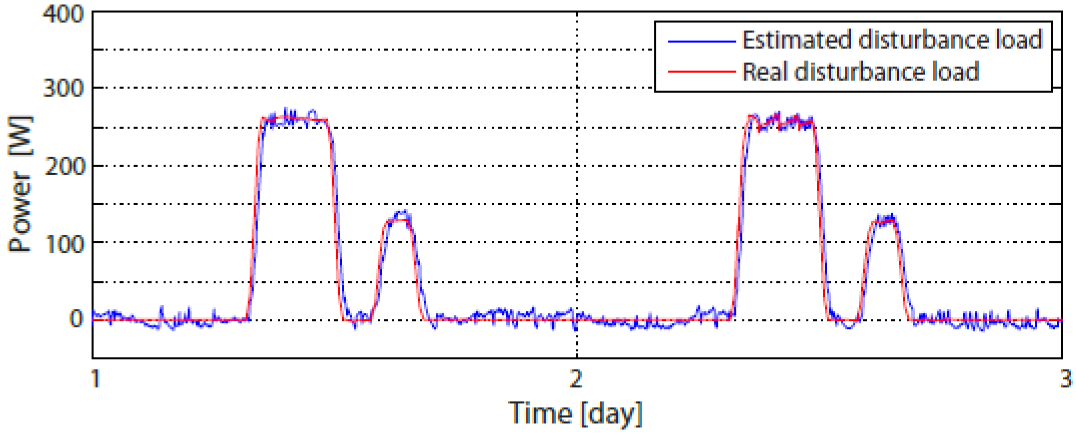


Figure 2.1: Estimation of unmeasured disturbance heat flux in zone.

As it can be seen, on-line estimation of disturbances with fixed model parameters successfully captures the disturbance behaviour. Accurate estimation of disturbance loads is of great importance for monitoring the space usage and thus the more efficient zone temperature control through: better prediction of future disturbance behaviour based on historical data, detection of window openness and provision of offset-free control when zone temperature is regulated by using predictive control algorithms.

3 Fan coils identification submodule (Z.PE.1)

A Fan Coil Unit (FCU) consists of a fan and at least one air-water heat exchanger. Due to their improved performance over classic radiators, FCUs are widely used for localized heating and cooling. However, advanced control, diagnosis and fault detection algorithms, which are well-established in other industries, are still hardly used for FCUs due to a lack of appropriate mathematical models that are easy to parametrize. While experimental analysis of general heat exchanger thermodynamic behaviour has been subject of numerous articles [23][24][26], experimental analysis of a FCU as a whole has been investigated in just a few papers concerning mainly one FCU [25][27]. In general, the developed models can be divided into three groups: *i)* physical models build on first principles [19][21], *ii)* non-physical models totally dependent on the experimental data [22][26] and *iii)* semi-physical models offering the compromise between first two approaches [23][24]. Models built on first principles typically require detailed physical properties of FCUs, such as fin thickness, tube dimensions, etc., which are often omitted from manufacturers data sheets and hardly measurable on the final on-site product [21]. On the other hand, the accuracy of the non-physical models, which are usually in a form of simple linear approximations around an operating point [27] or neural networks, could be an issue when applying them outside the training range. Also, the final model forms may be too complex for control purposes and real-time implementations. Semi-physical modelling presents a compromise between the first two modelling approaches by taking physical considerations into account when identifying the model based on the experimental data. Besides the thermodynamic performance, a hydrodynamic performance of a FCU and the sensor free solution for determining the medium mass flow through the unit is often neglected. In [24] authors suggested use of the pressure drop sensors to determine mass flow through the heat exchanger which tends to be cost-intensive when applied to individual FCUs.

In this chapter, a replicable, robust, simple and fast methodology for identification of a thermohydraulic model of an on-site FCU is derived by consolidating the advantages of first principles modelling, identification methods and data sheet information. Consolidation of all the available tools downsized the number of required sensors to a single sensor per FCU, zone temperature sensor and calorimeter installed on major supply ducts. Use of energy meters is optionally suggested as a cost-effective solution for on-line monitoring of FCU performance. Although one single FCU consumes relatively small amount of electricity, due to many units installed in a whole building and long operation hours, the inclusion of the electrical consumption model into control algorithm can lead to significant electrical energy savings. Identification and validation of the identified models are performed within the living-lab pilot on predictive building zones control.

The submodule interface is defined in the following tables (Table 3.1, Table 3.2). Test site configuration is described in Section 3.1. In Section 3.2, an analogue electro-hydraulic model for a test site is developed and optimized to find a flow distribution along the test site. A thermodynamic model of a FCU unit is introduced in Section 3.3. In Section 3.4 an experimental analysis of both thermodynamic and hydraulic performance is conducted to find the unknown model parameters. After the parameters have been found both models are tested and validated experimentaly as shown in Section 3.4.

Table 3.1: Required inputs for the fan coils identification submodule.

Variable name	Notation	Description
Historical temperature profile from zone (minute-scale of the sampling time)	T (noted herein as T_1)	Data taken from the database
Historical profile of fan actuation in the zone (minute-scale of the sampling time)	FS (noted herein as x)	Data taken from the database
Historical temperature profile of the supply medium from a calorimeter (minute-scale of the sampling time)	$T_{\text{supply,cal}}$	Data taken from the database
Historical temperature profile of the return medium from a calorimeter (minute-scale of the sampling time)	$T_{\text{return,cal}}$	Data taken from the database (might not be needed, but will be available)
Historical profile of the flow from a calorimeter (minute-scale of the sampling time)	Q_{cal}	Data taken from the database
Historical profile of energy (power) recorded on the calorimeter (minute-scale of the sampling time)	E_{cal} (P_{cal})	Data taken from the database
Historical temperature profile from the return medium temperature sensor on a fan coil (minute-scale of the sampling time)	$T_{\text{return,fc}}$ (noted herein as $T_{\text{w}}^{\text{out}}$)	Data taken from the database
Historical temperature profile from the supply medium temperature sensor on a fan coil (minute-scale of the sampling time)	$T_{\text{supply\_fc}}$ (noted herein as T_{w}^{in})	(optional) If not existing, measurement of the temperature on the calorimeter should be used, and additionally a characteristic of the temperature drop along the pipeline from the heat loss model should be used

Table 3.2: Outputs of the fan coils identification submodule.

Variable name	Notation	Description
Parameters of the fan coil model that relates fan coil actuation, room temperature and medium conditions registered on a calorimeter to fan coil energy transmitted to room air in a defined time period; also parameters of a simple relation between heating energy and electrical energy for fans for different supply medium flows and temperatures	$A_{\text{fc}}(Q_{\text{w}})$, $B_{\text{fc}}(Q_{\text{w}})$, $C_{\text{fc}}(Q_{\text{w}})$, $D_{\text{fc}}(Q_{\text{w}})$	Parameters needed for calculation of maximum energy for the MPC module, for the interface submodule functioning, and for calculation of energy inputs for identification of a simplified building dynamic model and for on-line estimation of its states and disturbances; electricity consumption model needed on the first higher level MPC modules

3.1 Test site configuration

The considered living lab of UNIZG-FER spans over two floors of university skyscraper (9th and 10th floor). Zones on the north side and on the south side are supplied via separate supply lines such that each floor has separate north-side and south-side piping. The central part of the living lab is a supervisory control and data acquisition (SCADA) system operating on a minute time-scale. The heating/cooling is a hydronic two-pipe system that provides seasonal cooling or heating. The FCUs, produced by manufacturer Trane (models FCC06 and FCC04) \cite{TRANE}, are equipped with a centrifugal fan with four different fan speeds (Zero, Low, Medium and High, denoted respectively by 0, L, M and H) and three-way valve (on-off type). Both fan speed and valve position are controlled by Siemens RXC21.1/RXC21.5 zone temperature controllers operating on LonWorks network. Each controllable zone includes a separate user interface for temperature reference selection (QAX34.1 device). The existing communication network is enhanced such that the RXC controllers are reconfigured to be able to pass the information to a central database (current zone temperature, fan speed and valve actuation) or can receive the commands from the database (fan speed, valve actuation). All FCUs in the same zone are actuated simultaneously. The system is further upgraded by installing low-cost 1-wire digital temperature sensors DS18B20 on the FCUs return pipes and Siemens calorimeters UH50-A50-00 operating on M-Bus protocol on every floor supply duct. Calorimeters measure supply and return medium temperature, temperature difference, medium flow, thermal power and consumed thermal energy with one-minute time resolution. Overall electrical consumption of all FCUs is measured by one common electrical energy meter, Schneider Electric PM3200 operating on Modbus protocol. All systems are integrated together with a network controller unit employed to enable two-way communication between devices operating on different protocols. Logical organization of the described system is shown in Figure 3.1.

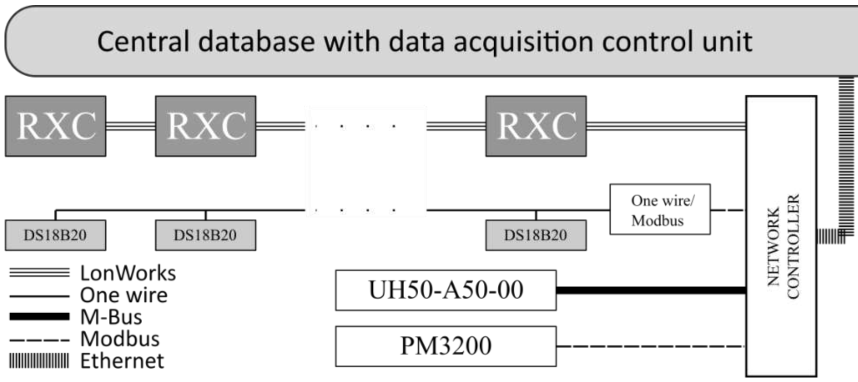


Figure 3.1. Logical organization of Living-lab sensor-actuator network.

We focus on the south-side piping on the 9th floor consisted of 13 zones with 17 vertical FCUs mounted on the floor, 12 units of type FCC06 and 5 units of type FCC04 (Figure 3.2). The arrangement of units with included geometry of supply pipes (length and diameter) is given in Table 3.3. The equivalent length of vertical supply and return pipes (including fittings) is identical for all units and amounts $l + \sum l_{\mathrm{eq}} = 6.26$ m.

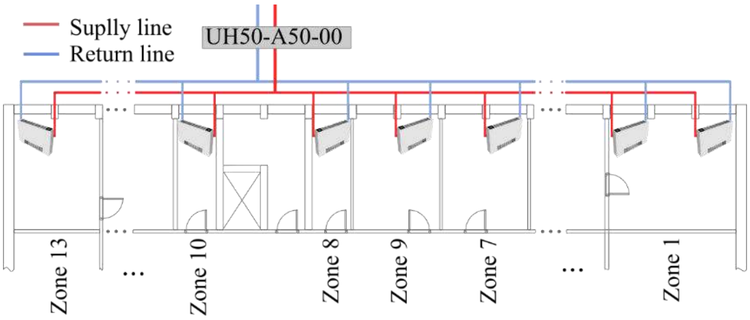


Figure 3.2: Layout of the southern supply duct on the 9th floor.

Table 3.3: Duct sizing information.

Zone ID	*Supply pipe diameter [mm]*	*Supply pipe length [m]*	*FCU unit type*
Zone 1	18	1.7	FCC06
	22	3.5	FCC06
Zone 2	28	1.7	FCC06
	35	3.5	FCC06
Zone 3	35	3.5	FCC06
Zone 4	35	3.5	FCC06
Zone 5	42	3.5	FCC06
Zone 6	42	1.7.	FCC04
	42	3.5	FCC04
Zone 7	42	3.5	FCC06
Zone 8	42	1.7	FCC06
Zone 9	42	2.1	FCC04
Zone 10	28	5	FCC06
Zone 11	28	1.7	FCC06
Zone 12	28	3.5	FCC04
	22	1.7	FCC04
Zone 13	18	3.5	FCC06

3.2 Hydraulic model of heating/cooling installations

The medium flow through a FCU depends on the pressure drop across the various elements that form up the entire installation. A practical way of modelling complex hydraulic systems is the transition to an analogous electric model where medium mass flow q_{w}, pressure drop Δp and hydraulic resistance R_h behave equivalently to electrical current, voltage and electrical resistance, respectively. In a series electrical circuit, the current through all the elements is the same and voltage drops along the elements are additive. The voltage drop across all elements connected in parallel is the same while the total current is equal to the sum of the currents through each of the branches. The same logic is applied to a hydraulic installations network consisted of hydraulic elements such as pipes, heat exchangers, tees, elbows, etc. The equation relating pressure drop and mass flow through a hydraulic network element is equal to:

$$\Delta p = R_h \cdot q_{\mathrm{w}}^{\alpha}, \tag{3-1}$$

where R_h is a constant hydraulic resistance and α is an exponent. The values of α depend on the methodology used for calculation of R_h and the element type.

Pressure loss in pipes consists of three components: (i) hydrostatic pressure loss Δp_h, (ii) frictional pressure loss Δp_f and (iii) kinetic pressure loss. For most applications, kinetic losses are minimal and can be ignored. Thus, the equation that describes the overall pressure loss in pipes is expressed as a sum of two major terms:

$$\Delta p_p = \Delta p_f + \Delta p_h, \tag{3-2}$$

The hydrostatic pressure drop occurs only when there are differences in elevation from the inlet to the outlet of a pipe segment:

$$\Delta p_h = \rho \cdot g \cdot \Delta h, \tag{3-3}$$

where g is acceleration of gravity and Δh is change in pipe elevation. The frictional pressure drop in a circular pipe with constant inner diameter d and length l is defined by Darcy-Weisbach equation:

$$\Delta\, p_f \;= f_D \frac{8 \cdot\; l}{\rho \cdot \pi^2 \cdot\; d^5} \cdot\; q_{\mathrm{w}}^2, \tag{3-4}$$

where ρ is the density of heating/cooling medium and f_D is the friction factor. For hydraulically smooth pipes f_D is defined by Blasius equation:

$$f_D = 0.3164 \cdot\; Re^{-0.25}, \tag{3-5}$$

where Re is Reynolds number defined as:

$$Re = \frac{4}{\mu \cdot\; d \cdot \pi} \cdot q_{\mathrm{w}}, \tag{3-6}$$

and μ is dynamic viscosity of the medium. In addition to the losses due to the friction or elevation difference, there are also losses associated with flow through valves and fittings. These, so called minor pressure losses, are accounted by using the equivalent length method [59]. The method uses empirical tables to convert each fitting into an equivalent length of the straight pipe l_{eq} which is then added to the pipe length l. The l_{eq}/d ratio for most common types of fitting is given in Table 3.4.

Table 3.4. Equivalent length of fittings [59],[60].

Type of fitting	$\boldsymbol{l_{eq}/d}$
Tee - along the straight	20
Tee - to the branch	60
Elbow 90 (smooth radius)	30
Three-way valve (fully opened - through flow)	30
Sudden pipe diameter expansion	4*
Sudden pipe diameter contraction	20*
** used with inlet velocity*	

By inserting (3-6) and (3-5) into (3-4) and including the minor losses, the final form of frictional pressure drop across the circular pipe section is defined as:

$$\Delta\, p_{\mathrm{fc}} = R_{\mathrm{fc}} \cdot\; q_{\mathrm{w}}^{\alpha_{\mathrm{fc}}}, \tag{3-7}$$

where R_{fc} and α_{fc} are parameters to be found based on the experiments or pressure drop data from the manufacturers' catalogue.

Based on the electric-hydraulic analogy and known heating/cooling network topology and geometry of the pipes, the equivalent electrical scheme of the supply/return piping around FCUs is derived Figure 3.3.

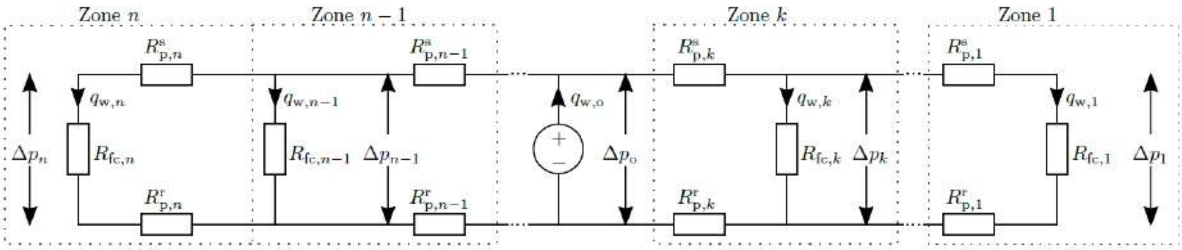


Figure 3.3: Equivalent electrical scheme of the 9th floor south supply duct heating/cooling installations.

Supply pipe, return pipe and FCU hydraulic resistances are denoted as $R_{\mathrm{p}}^{\mathrm{s}}$, $R_{\mathrm{p}}^{\mathrm{r}}$ and R_{fc}, respectively. For clarity, hydraulic resistances of pipes in parallel branches are not shown. For every closed loop of the circuit, once the hydraulic resistances and mass flows are known, the pressure drop is defined with Kirchhoff's circuit laws $\forall\, j = 1, \dots, n-1$:

$$\Delta\, p_j = \begin{cases} \Delta p_{j+1} - \sum_{i=1}^{j} q_{\mathrm{w},i}^{1.75}\,(R_{\mathrm{p,j}}^{\mathrm{s}} + R_{\mathrm{p,j}}^{\mathrm{r}}) & \text{for} \quad j \le k \\ \Delta p_{j+1} + \sum_{i=j}^{n} q_{\mathrm{w},i}^{1.75}\,(R_{\mathrm{p,j}}^{\mathrm{s}} + R_{\mathrm{p,j}}^{\mathrm{r}}) & \text{for} \quad j > k, \end{cases} \tag{3-8}$$

where $\Delta\, p_j$ is the overall pressure drop in a parallel branch including pressure drop through FCU and pressure drop in associated vertical supply and return pipes, $\Delta p_{k+1} = \Delta p_{\mathrm{o}}$ is the overall network pressure drop of the entire duct, n is the total number of the FCUs connected to the duct and $q_{\mathrm{w},i}$ is medium mass flow though i^{th} network branch. For known overall medium mass flow denoted with $q_{\mathrm{w,o}}^{\mathrm{m}}$ the individual FCU mass flows are found by solving the following optimization problem:

$$\begin{aligned} \min_{\Delta p_{\mathrm{o}}} \quad & \left| q_{\mathrm{w,o}}^{\mathrm{m}} - q_{\mathrm{w,o}} \right| \\ s.t \quad & q_{\mathrm{w,o}} = \sum_{i=1}^{n} q_{\mathrm{w},i}, \\ & \text{(3-2),(3-3),(3-7),(3-8),(3-9)} \end{aligned} \tag{3-9}$$

The optimization problem (3-8) belongs to a class of nonlinear programs which can be efficiently solved with e.g. genetic algorithms [58]. For installations with operable valves, where flow distribution is time-variable and based on the valve positions, the procedure is extended by introducing variable valves hydraulic resistances in the network.

3.3 Thermodynamic fan coil unit model

Heat transfer within a FCU consists of three parts: convection of the heating/cooling medium, heat conduction through the heat exchanger and convection of air to be heated or cooled. For modelling, the following assumptions are made:

- air mass flow q_a inside the FCU variates with the fan speed,
- mean water temperature inside the FCU is approximately the arithmetic average of water inlet temperature T_w^in and water outlet temperature $T_\mathrm{w}^\mathrm{out}$, i.e. $\overline{T}_\mathrm{w} = 0.5(T_\mathrm{w}^\mathrm{in} + T_\mathrm{w}^\mathrm{out})$,
- heat transfer from water to air is driven by the temperature difference $(\overline{T}_\mathrm{w} - T_\mathrm{a}^\mathrm{in})$,
- air intake temperature T_a^in a is assumed to be equal to zone temperature,
- properties of air and water are assumed to be constant.

Furthermore, due to the parallel connection of the FCUs and constant pressure drop within the system maintained by a central circulation pump, the FCUs are considered independent. With set assumptions, the following dynamics equations are derived for each FCU:

$$m_\mathrm{a} c_\mathrm{a}\, \dot{T}_\mathrm{a}^\mathrm{out} = q_\mathrm{a} c_\mathrm{a}(T_\mathrm{a}^\mathrm{in} - T_\mathrm{a}^\mathrm{out}) + U_\mathrm{o}\big(\overline{T}_\mathrm{w} - T_\mathrm{a}^\mathrm{in}\big), \tag{3-10}$$

$$m_\mathrm{w} c_\mathrm{w}\, \dot{T}_\mathrm{w}^\mathrm{out} = q_\mathrm{w} c_\mathrm{w}(T_\mathrm{w}^\mathrm{in} - T_\mathrm{w}^\mathrm{out}) - U_o\big(\overline{T}_\mathrm{w} - T_\mathrm{a}^\mathrm{in}\big), \tag{3-11}$$

where T_a^out is the outgoing air temperature, c_a , c_w and $U_\mathrm{o} = f(q_\mathrm{a}, q_\mathrm{w})$ are the specific heat capacity of air, specific heat capacity of water and the heat transfer coefficient, respectively. Parameter m_a is the mass of air inside the fan coil unit, m_w is the mass of water inside the coil and it is easily obtained from manufacturers data sheets. According to the FCU dimensions, a mass of the air inside the FCU is less than 0.1 kg, thus time constant of the air $m_\mathrm{a}/q_\mathrm{a}$ is less than 1 s, which makes it negligible when compared to the significantly larger time constant of the water. Due to a very small time constant, thermal process from the air side is observed as a stationary process (e.g. $\dot{T}_\mathrm{w}^\mathrm{out} = 0$).

$$0 = \underbrace{Q_\mathrm{a} c_\mathrm{a}(T_\mathrm{a}^\mathrm{in} - T_\mathrm{a}^\mathrm{out})}_{P_\mathrm{a}} + \underbrace{U_\mathrm{o}\big(\overline{T}_\mathrm{w} - T_\mathrm{a}^\mathrm{in}\big)}_{P_\mathrm{t}}, \tag{3-12}$$

This further means that the thermal power affecting the zone P_a is equal to the overall transmitted thermal power P_t . The important feature of this approach is that the hardly measurable and unreliable T_a^out measurement is omitted. During the cooling season relation (3-10), due to the possible phase-change of the water vapour contained in the air, goes to:

$$m_\mathrm{a} c_\mathrm{a}\, \dot{T}_\mathrm{a}^\mathrm{out} = q_\mathrm{a} c_\mathrm{a}(T_\mathrm{a}^\mathrm{in} - T_\mathrm{a}^\mathrm{out}) + U_\mathrm{o}\big(\overline{T}_\mathrm{w} - T_\mathrm{a}^\mathrm{in}\big) + P_l, \tag{3-13}$$

where P_l is latent power defined as:

$$P_l = q_\mathrm{a} \lambda(\omega_\mathrm{a}^\mathrm{out} - \omega_\mathrm{a}^\mathrm{in}), \tag{3-14}$$

λ is the latent heat of vaporization of water and $\omega_\mathrm{a}^\mathrm{in}$ and $\omega_\mathrm{a}^\mathrm{out}$ are absolute humidities of the air at the FCU air intake and exhaust.

3.4 Experimental analysis of hydraulic and thermodynamic FCU behaviour

To identify the thermodynamic model and experimentally validate the algorithm developed within Section 3.2, an identification procedure was ran on all FCUs. The identification is performed by shutting down all the units connected to the same duct (or assure their constant operation) and running a test sequence on the particular unit. Valves remained fully opened for all units. In such a set-up the calorimeter on the duct's inlet measures the heat consumption of the particular unit with a constant offset equal to the thermal power of the remaining duct part. To assure constant losses, supply medium mass flow and temperature are required to be constant during the test. If supply temperature is not constant, identification procedure can still be done by identifying variable thermal losses during the test.

Return medium temperature sensors and calorimeters are modelled as ideal elements with variable transport delay τ:

$$H(s) = e^{-\tau \cdot s}. \tag{3-15}$$

Transport delay is estimated manually for every data set by comparing the response time of the sensors with the time when fan speed change occurred. Measurements obtained after running the test in Zone 7 during the heating season 2016/17 are shown in Figure 3.4. In the following subsections transmission heat losses are neglected due to the good thermal insulation of the pipeline. This means that the FCU water inlet temperature is considered to be equal to the supply temperature measured by the calorimeter T_{cal} ($T_{\mathrm{w}}^{\mathrm{in}} = T_{\mathrm{cal}}$). If this is not the case, temperature drop along the network should also be modelled or additional temperature sensors have to be mounted at the FCU water inlet.

3.4.1 One wire return medium temperature sensors calibration

Indirect measurement of the return medium temperature with the 1-wire sensor mounted on the FCU return pipe is subject to various effects (e.g. lead wires acting as a thermal sink, sensor insulation, effects of ambient temperature, etc.) that cause deviation from the real temperature. To eliminate the offset, measurements have to be compared with a trusted calibrated sensor in at least two operating points. Ideally, one close to the lower and one close to the upper bound of the operating range. This so-called two-point calibration method essentially re-scales the output and is capable of correcting both slope and offset errors. Specific to the test-site, due to the well-insulated supply pipelines, large thermal conductivity of the copper pipes and 1-wire sensor mounted close to a bypass branch (Figure 3.5), offset characteristics is determined using historical measurements of supply medium temperature T_{cal} from the calorimeter with switched off fan and closed FCU valves (total flow goes through the bypass branch).

To avoid the transient impact of the medium stalled inside the heat exchanger, only stationary values are used. Figure 3.6 shows the calibration curve obtained by calibrating the 1-wire sensor mounted on the FCU return in Zone 7. The functional dependence is intentionally fitted only to raw sensor measurements $T_{\mathrm{w},i}^{\mathrm{out,raw}}$, instead of fitting it to the temperature difference between the sensor and its surroundings, since ambient temperature is always within narrow user comfort range and has negligible effect.

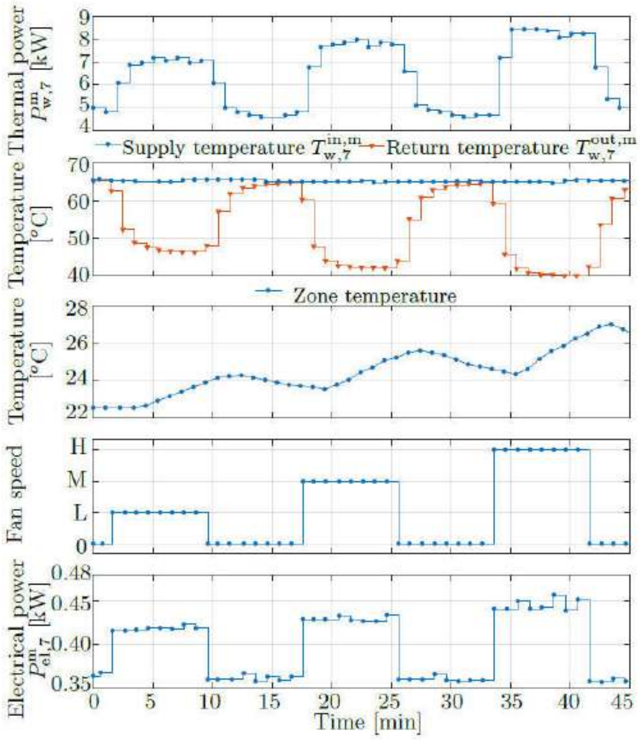


Figure 3.4. Measurements obtained after running the identification procedure in the exemplary Zone 7.

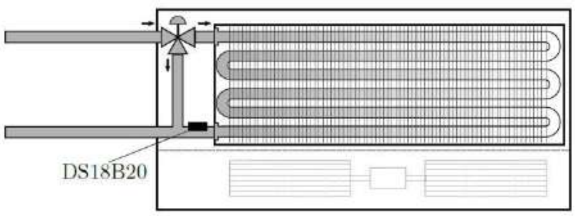


Figure 3.5. One wire return medium temperature sensor placement.

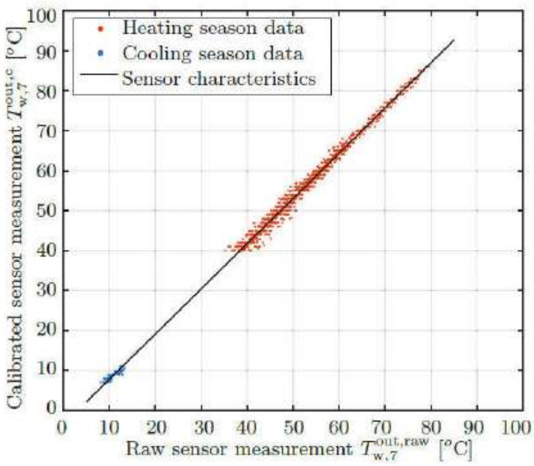


Figure 3.6: One wire return medium temperature sensor characteristics.

When FCUs are operated (fan speed different than zero and valves fully opened) high thermal conductivity of the cooper pipes combined with large temperature difference between the supply and return pipe additionally skew the measurements. Since supply and return pipes are thermally coupled through the bypass, large thermal gradient between them influence the sensor measurements proportionally to temperature difference between the pipes. True temperature measurement $T^{\text{out,m}}_{\text{w},i}$ is thus defined as:

$$T^{\text{out,m}}_{\text{w},i} = T^{\text{out,c}}_{\text{w},i} - \phi(T^{\text{in}}_{\text{w},i} - T^{\text{out,m}}_{\text{w},i}) \quad (3\text{-}16)$$

where $T_{\mathrm{w},i}^{\mathrm{out,c}}$ is the calibrated sensor measurement of the i^{th} FCU, $T_{\mathrm{w},i}^{\mathrm{in}} = T_{\mathrm{cal}}$ and ϕ is the unknown heat transfer coefficient. For ideal mixing of the return medium from different FCUs and only the i^{th} FCU operating at the time, the following holds:

$$q_{\mathrm{w},i}\left(T_{\mathrm{w},i}^{\mathrm{in}} - T_{\mathrm{w},i}^{\mathrm{out,m}}\right) = q_{\mathrm{w,o}}\Delta T_{\mathrm{cal},i}\ , \tag{3-17}$$

where $\Delta T_{\mathrm{cal},i}$ is temperature difference measured on the calorimeter reduced for the constant temperature drop assessed in the steady operation of the remaining pipework and FCUs. Since $\sum_{i=1}^{n} q_{\mathrm{w},i} = q_{\mathrm{w,o}}$, by combining (3-16) and (3-17) heat transfer coefficient ϕ is defined as:

$$\phi = 1 - \frac{1}{\sum_{j=1}^{N} \frac{\Delta T_{\mathrm{cal,j}}}{\sum_{i\in\nu_j}\left(T_{\mathrm{w},i}^{\mathrm{in}} - T_{\mathrm{w},i}^{\mathrm{out,c}}\right)}} \tag{3-18}$$

where N is the total number of zones and ν_j is the set of FCUs placed in the j^{th} zone. To get the unique estimate of ϕ, independent tests were performed in all zones with FCUs active in only one zone at the time and valves fully opened. Estimated ϕ value for the test-site is ϕ =0.1534.

3.4.2 Hydraulic model identification

The correlation between pressure drop and mass flow for both FCU types is found by identifying the unknown coefficients R_{fc} and α_{fc} based on the data from the manufacturers' catalogue (see e.g. Figure 3.7 for FCC06 FCU type).

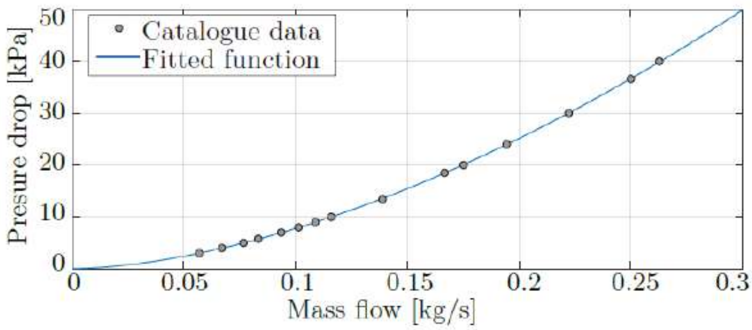


Figure 3.7. Identified pressure drop function for Trane model FCC06.

With known coefficients and topology and geometry of the pipes, an analogous electrical model of the test-site hydraulic installations is developed. To set up the optimization problem (3-9) single measurement of the overall medium mass flow $q_{\mathrm{w,o}}^{\mathrm{m}} = q_{\mathrm{cal}}^{\mathrm{m}}$ from the calorimeter is used. The flow distribution through the entire network is found by solving the optimization problem (3-9) in MATLAB using genetic algorithms [58]. The resulting flow distribution, defined as $\eta_i = q_{\mathrm{w},i}/q_{\mathrm{w,o}}$ rounded to two decimals is listed in column 1 in Table Table 3.6.

The validity of the developed approach is verified by comparing the results with the solutions of the following optimization problem solved for all identification data sets separately:

$$\left(\eta_{e,i}^{*}, P_d^{*}\right) = \underset{\eta_{e,i}, P_d}{\operatorname{argmin}} \left\| \eta_{e,i} \cdot P_{\mathrm{w}}^{\mathrm{a}} - \left(P_{\mathrm{cal}}^{\mathrm{m}} - P_d\right) \right\|^2 \tag{3-11}$$

where $P_{\mathrm{w}}^{\mathrm{a}} = q_{\mathrm{w,o}}^{\mathrm{m}} c_{\mathrm{w,cal}} (T_{\mathrm{w}}^{\mathrm{in,m}} - T_{\mathrm{w}}^{\mathrm{out,m}})$ is a-priori FCU water side thermal power calculated with nominal water heat capacity used by calorimeter $c_{\mathrm{w,cal}}$ (usually set to 4180 J/(kg°C)) and $P_{\mathrm{cal}}^{\mathrm{m}}$ is is thermal power measurement from the calorimeter and P_d is constant thermal power consumed by the rest of the duct. The identified individual flow shares according to (3-11), for 8 tests performed in the exemplary Zone 7 during winter 2015 and 2016 are shown in Table 3.5. The mean flow share is $\bar{\eta}_{e,7} = 7.40$ %, which deviated from the calculated value based on the electric-hydraulic analogy only by 2.84 % (see Table 3.5).

Table 3.5: Estimated flow share for Trane FCC06 installed in zone 9.

	Test 1	*Test 2*	*Test 3*	*Test 4*	*Test 5*	*Test 6*	*Test 7*	*Test 8*
$\bar{q}_{w,7}\left[\frac{kg}{h}\right]$	*0.027*	*0.028*	*0.027*	*0.028*	*0.029*	*0.023*	*0.022*	*0.030*
$100 \cdot \eta_{e,7}^{*}$ [%]	7.26	7.35	7.35	7.57	7.34	7.26	7.78	7.31

The identified flow distribution of the test-site is listed in column 6 in Table 3.6. Since all FCUs in a single zone are actuated simultaneously, for zones with more than one FCU, mean flow share of all units is calculated instead of individual shares. Average relative error, mainly due to sensor accuracy, is 1.16 %, which proves the adequate accuracy of electric-hydraulic analogy based calculation of flow distribution through the supply line. In zones marked with 'x' measurements were unavailable.

Table 3.6: Estimated flow share for the south supply line on the 9th floor for $q_{w,o} \approx 0.36$ kg/s.

Zone ID	***Calculated flow share*** η_i [%]	***Estimated flow share*** $\eta_{e,i}$ [%]	***Relative error*** $\frac{\lvert\eta_i - \eta_{e,i}\rvert}{\eta_i}$
Zone 1	4.62 4.85	4.73	0.04
Zone 2	5.40 5.60	5.49	0.13
Zone 3	5.81	5.88	1.22
Zone 4	6.09	5.97	1.87
Zone 5	6.49	6.50	0.24
Zone 6	5.31 5.51	X	X
Zone 7	7.20	7.40	2.84
Zone 8	7.58	7.66	1.06
Zone 9	6.01	6.15	2.27
Zone 10	7.08	X	X
Zone 11	6.77	6.71	0.79
Zone 12	5.00 4.77	4.94	1.13
Zone 13	5.62	x	x

3.4.3 Air flow estimation

A typical calculation of FCU airflow includes the use of temperature sensors at air intake and exhaust and/or anemometers. Both approaches %, used for straightforward measurement of the airflow, proved to be impractical due to highly variable accuracy dependent on the sensors installation position [25]. The calculation is additionally corrupted by dominant effects of temperature stratification or natural airflows. Here we propose the methodology for indirect measurement of the airflow by using an electrical energy meter and fan performance data from the catalogue.

$$q_{\mathrm{a}} = f_{el}^{x}(P_{el}), \qquad x \in \{\mathrm{L}, \mathrm{M}, \mathrm{H}\} \tag{3-10}$$

where P_{el} is electric power consumed by the FCU fan. Since FCUs' air intake filters performances vary slowly over time, one electrical energy meter can be used to monitor electrical consumption of all FCUs connected to the same line or all FCUs placed on the same or couple of floors, offering thus more cost-effective solution to monitor the FCU performance over time. Figure 3.8 shows the functional dependence between the electrical energy consumption and FCU airflow for Trane FCC06. In Figure 3.9 the results of on-line monitoring of the FCU performance in Zone 7 are shown. Tests are performed by turning the high speed of the fan for two minutes during unoccupied hours. To see the impact of the air path blockage on, e.g. the chosen date of 11 October, the FCU air exhaust was blocked by placing the obstacle on approximately 70 % of the exhaust. The result is degradation of the airflow by approximately 11 %. Similar procedure is derived on-line based only on FCU operational data and information on its fan state by running disaggregation algorithms [30]. By performing the experiment, it is proved that the air flow does not deviate significantly from the nominal air floor rate listed in manufacturers' catalogue if there is no external impact blocking the air path. Thus, in the following subsections air flow is assumed to be constant and equal to the nominal value for each speed.

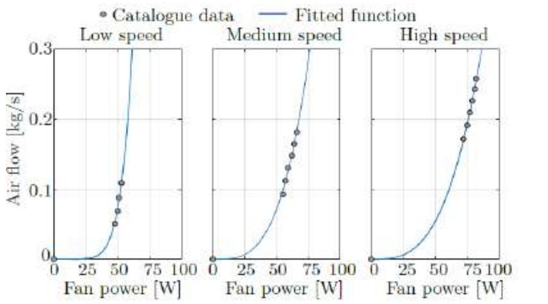


Figure 3.8: Identified FCU electrical consumption model for Trane FCC06..

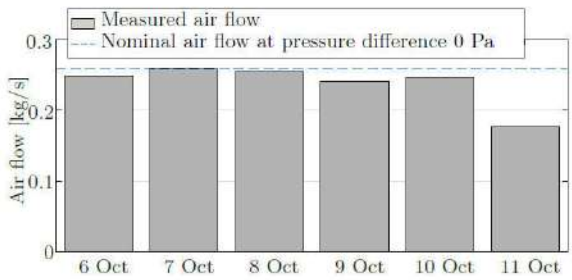


Figure 3.9: Air mass flow variation for 6 days period in 2017.

3.4.4 Thermodynamic model estimation

For the hydraulic model calculated as described in Section 3.2, the following identification of the thermodynamic FCU model can be performed on the normal operation FCU data. In most heating and cooling systems medium mass flow is controlled to a constant value while supply temperature is altered to meet the building thermal demand. To estimate $U_{\mathrm{o}} = f(q_{\mathrm{a}}, q_{\mathrm{w}})$ the existing non-uniform distribution of the flow through the units is exploited. Water heat capacity c_{w} and overall heat transfer coefficient U_{o} for fixed air and medium mass flows are calculated by minimizing the squared error between the measured return medium temperature $T_{\mathrm{w}}^{\mathrm{out,m}}$ and the simulated one on a minute-time scale (3-11):

$$\begin{aligned} (U_{\mathrm{o}}^{*}, c_{\mathrm{w}}^{*}) = \underset{U_{\mathrm{o}}, c_{\mathrm{w}}}{\arg\min} \quad & \left\| T_{\mathrm{w},i}^{\mathrm{out,m}} - T_{\mathrm{w},i}^{\mathrm{out}} \right\|^{2} \\ s.t. \quad & q_{\mathrm{w},i} = \eta_i \cdot q_{\mathrm{cal}}, \\ & (3-11). \end{aligned} \tag{3-13}$$

where $U_{\mathrm{o}} \coloneqq \{U_{\mathrm{o}}^{0}, U_{\mathrm{o}}^{\mathrm{L}}, U_{\mathrm{o}}^{\mathrm{M}}, U_{\mathrm{o}}^{\mathrm{H}}\}$ is the set of fixed overall heat transfer coefficients for zero, low, medium and high speed defined as:

$$U_{\mathrm{o}} \triangleq \begin{cases} U_{\mathrm{o}}^{0} & \text{for} \quad x = 0, \\ U_{\mathrm{o}}^{L} & \text{for} \quad x = \mathrm{L}, \\ U_{\mathrm{o}}^{\mathrm{M}} & \text{for} \quad x = \mathrm{M}, \\ U_{\mathrm{o}}^{\mathrm{H}} & \text{for} \quad x = \mathrm{H}. \end{cases} \tag{3-13}$$

Identified heat capacity of water c_{w}^{*} for a set of data containing 32 test runs in different zones during heating season 2015/2016, 2016/2017 and cooling season 2017 is shown in Figure 3.10.

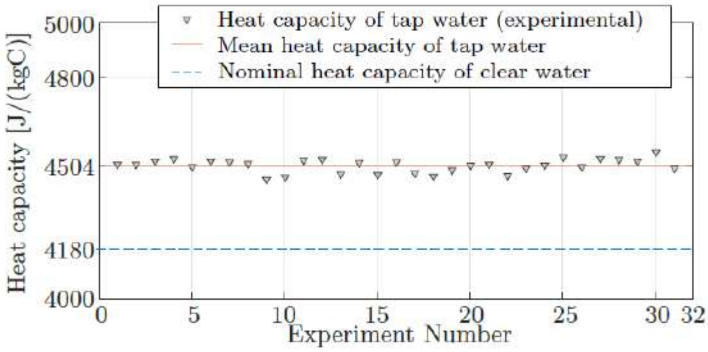


Figure 3.10: Estimated heat transfer coefficient.

The effect of the latent heat during the cooling season is investigated by installing additional ZigBee humidity and pressure sensors in several zones. By monitoring the FCU performance during the cooling season of 2015, it is found that latent heat amounts for no more than 10 % of sensible heat. To avoid the need to install additional sensors in every zone, only sensible heat P_{a} is considered while P_{l} is treated as a dynamic disturbance. Time responses of the identified model during a heating season, tested on the verification data with the same medium mass flow as the identification data set, are shown in Figure 3.11. Estimated heat capacity of the medium is considered by scaling the calorimeter power measurements with $P_{\mathrm{cal}}^{*} = P_{\mathrm{cal}}^{\mathrm{m}} \cdot \frac{c_{\mathrm{w}}^{*}}{c_{\mathrm{w,cal}}} - P_{\mathrm{d}}^{*}$. As it can be seen from the figure, the model successfully captures the FCU dynamics.

Overall heat transfer coefficient function estimation

During the cooling season thermodynamic performance of the floor mounted FCUs is downgraded. This is because cooled air tends to settle at the floor and may ``short-circuit" FCU air intake without mixing with the zone air. The estimated U_{o}^{*} values during the cooling season are thus much lower than expected. To deal with the problem, empirical correction coefficients $\varepsilon := \{\varepsilon^{\mathrm{L}}, \varepsilon^{\mathrm{M}}, \varepsilon^{\mathrm{H}}\}$ are introduced in the model to anticipate the effects of large vertical temperature stratification. During the heating season parameter ε is equal to one for all fan speeds. Physically meaningful U_{o} relation dependent on the airflow q_{a} and mass flow q_{w} is [23],[24]:

$$U_{\mathrm{o}}(q_{\mathrm{a}}, q_{w}) = \varepsilon \cdot \frac{a \cdot q_{\mathrm{a}}^{c}}{1 + b\left(\frac{q_{\mathrm{a}}}{q_{w}}\right)^{c}}, \tag{3-16}$$

where a, b, c and ε are parameters to be determined.

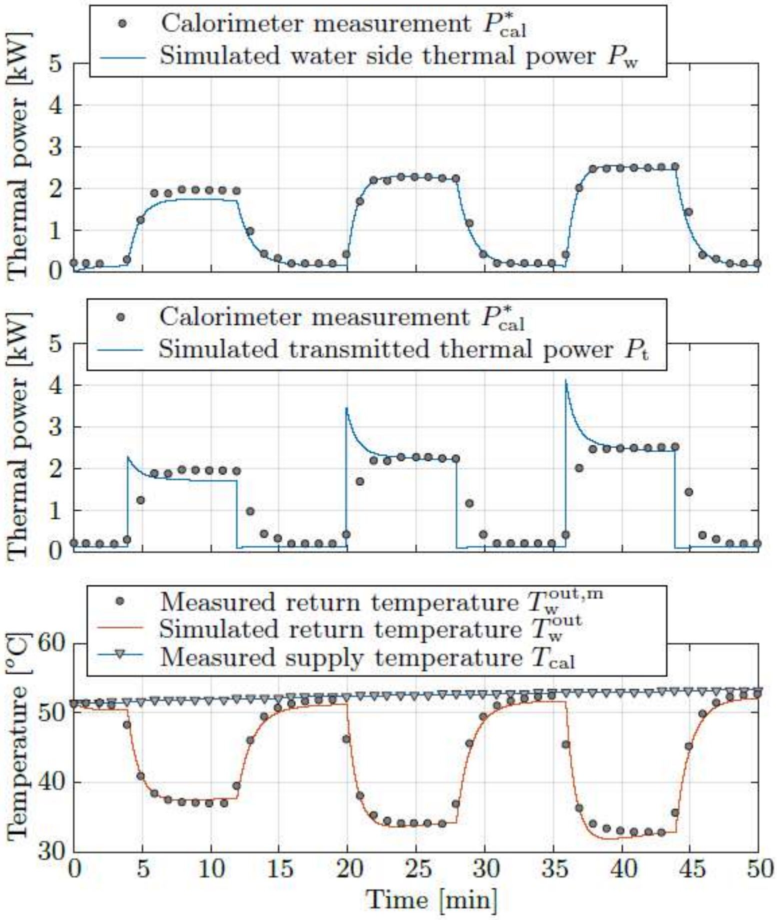


Figure 3.11. Identified FCU model response over the verification data set.

Manufacturers' catalogue performance data usually cover the operating range with constant U_{o} and higher medium mass flows while operating range of the on-site FCUs usually covers lower medium mass flows. To develop a model applicable over the entire operating range, the catalogue data are combined with real on-site measurements. Only sensible heat data from the catalogue is used. The catalogue data, extracted from heating and cooling capacity tables of Trane FCC06 FCU consist of the stationary values of the sensible power data $P_{\mathrm{a}}^{\mathrm{cd}}$, supply $T_{\mathrm{w}}^{\mathrm{in,cd}}$and return $T_{\mathrm{w}}^{\mathrm{out,cd}}$ water temperature data and entering air temperature data $T_{\mathrm{a}}^{\mathrm{in,cd}}$for low, medium and high speed. The overall heat transfer coefficient for every speed is thus calculated from the stationary condition (3-12) as:

$$U_{\mathrm{o}}^{x} = \frac{P_{\mathrm{a}}^{\mathrm{cd}}}{\left(0.5 \cdot \left(T_{\mathrm{w}}^{\mathrm{in,cd}} + T_{\mathrm{w}}^{\mathrm{out,cd}}\right) - T_{\mathrm{a}}^{\mathrm{in,cd}}\right)}, \qquad x = \{\mathrm{L, M, H}\} \tag{3-16}$$

The estimated functional dependence between the mass flows and the overall heat transfer coefficient for three non-zero fan speeds of Trane FCC06 unit is shown in Figure 3.12 with estimated correction coefficients ε listed in Table 3.7 and estimated unknown model parameters listed in Table 3.8.

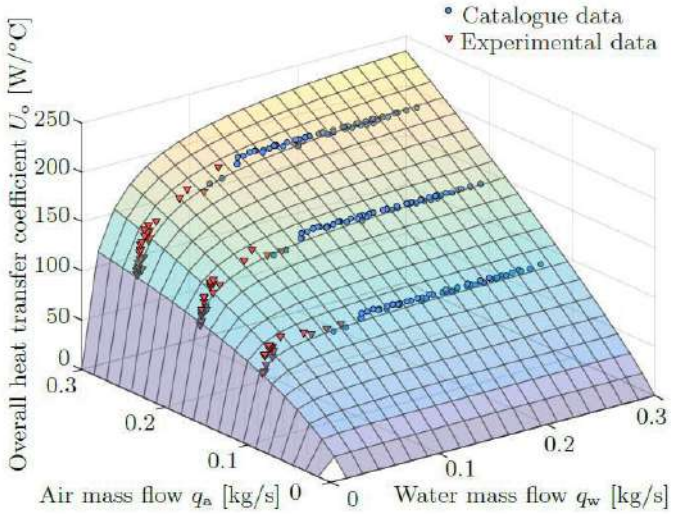


Figure 3.12: Overall heat transfer coefficient function.

Table 3.7: Cooling season correction coefficients $\boldsymbol{\varepsilon}$.

Fan speed $\boldsymbol{x}$	***L***	***M***	***H***
Correction coefficients ε^x	0.35	0.47	0.52

Table 3.8: Estimated $\boldsymbol{U}_{\mathrm{o}}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{q}_{\boldsymbol{w}})$ function parameters for Trane FCC06.

Model parameters	***a***	***b***	***l***
Estimated values	751.496	0.082	0.900

Standard preventive maintenance of FCUs within the living lab implies a change of air intake filters every two years. By running the identification procedure several times between 2015 and 2017, it is found that the filter dusting does not affect the airflow significantly. Airflow in all the tests was around the nominal airflow from the catalogue at pressure difference 0 Pa. Since the airflow does not deviate over time significantly (if there are no external impacts blocking the air path), it is reasonable to estimate the separate functional dependencies for all three fan speeds avoiding thus the need for knowing the exact information on the airflow, which is very convenient for the case $x = 0$ when air flow information is unavailable. For switched-off fan ($x = 0$) a FCU behaves as a normal radiator unit with a constant heat transfer coefficient:

$$U_{\mathrm{o}}(q_{\mathrm{w}}, x) = D, \qquad \mathrm{x} = \{0\}, \qquad q_{\mathrm{w}} \neq 0. \tag{3-18}$$

By fixing the airflow information (3-16) for non-zero fan speeds obtains the form:

$$U_{\mathrm{o}}(q_{\mathrm{w}}, x) = \varepsilon \cdot \frac{A}{1 + B \cdot q_{\mathrm{w}}^{-C}}, \quad x = \{\mathrm{L,M,H}\}. \tag{3-18}$$

Unknown parameter sets $A := \{a^{\mathrm{L}}, a^{\mathrm{M}}, a^{\mathrm{H}}\}$, $B := \{b^{\mathrm{L}}, b^{\mathrm{M}}, b^{\mathrm{H}}\}$ and $C := \{c^{\mathrm{L}}, c^{\mathrm{M}}, c^{\mathrm{H}}\}$ and parameter D are identified by running a simple non-linear least squares curve fitting in MATLAB [29]. Identified parameters for Trane FCC06 are shown in Table 3.9 with U_{o}^{0} estimated to be equal to 5.30. Parameter set C does not depend on the airflow so one common parameter for all three speeds is identified. The functional dependence $U_{\mathrm{o}}(q_{\mathrm{w}}, x)$ for three non-zero fan speeds of Trane FCC06 is shown in Figure 3.13.

Table 3.9: Identified parameters of overall heat transfer coefficient function for Trane FCC06.

Model parameters/ Fan speed x	A	B	C
L	96.45	$1.73 \cdot 10^{-3}$	
M	152.90	$3.58 \cdot 10^{-3}$	1.86
H	201.80	$5.40 \cdot 10^{-3}$	

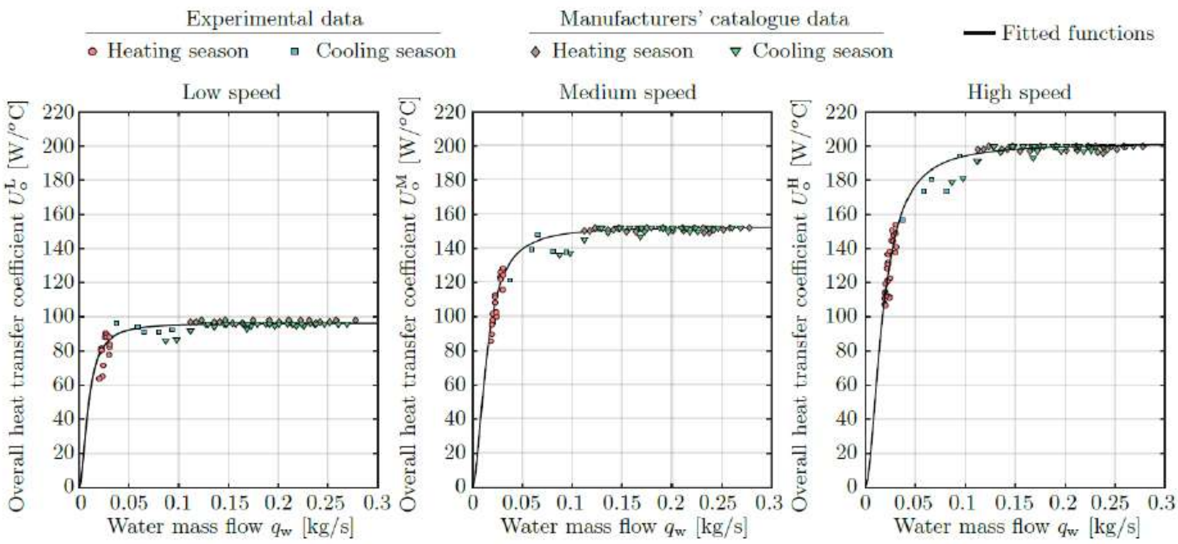


Figure 3.13. Heat transfer coefficients as a function of medium mass flow for fan coil type Trane FCC06.

The final thermodynamic model of FCU is :

$$\dot{T}_{\mathrm{w}}^{\mathrm{out}} = \left[-\frac{q_{\mathrm{w}}}{m_{\mathrm{w}}} - \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2m_{\mathrm{w}}c_{\mathrm{w}}}\right] T_{\mathrm{w}}^{\mathrm{out}} + \left[\frac{q_{\mathrm{w}}}{m_{\mathrm{w}}} - \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2m_{w}c_{w}} \quad \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2m_{w}c_{w}}\right] \begin{bmatrix} T_{\mathrm{w}}^{\mathrm{in}} \\ T_{\mathrm{a}}^{\mathrm{in}} \end{bmatrix}, \tag{3-19}$$

$$\mathrm{P_a} = \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2} T_{\mathrm{w}}^{\mathrm{out}} + \left[\frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2} \quad -U_{\mathrm{o}}(q_{\mathrm{w}},x)\right] \begin{bmatrix} T_{\mathrm{w}}^{\mathrm{in}} \\ T_{\mathrm{a}}^{\mathrm{in}} \end{bmatrix} \tag{3-20}$$

$$U_{\mathrm{o}}(q_{\mathrm{w}},x) = \begin{cases} D & \text{for} \quad x = 0 \\ \varepsilon \cdot \dfrac{A}{1 + B \cdot q_{\mathrm{w}}^{-C}} & \text{for} \quad x = \{\mathrm{L, M, H}\}, \end{cases} \tag{3-21}$$

$$q_{\mathrm{w}} = \eta \cdot q_{\mathrm{cal}}. \tag{3-22}$$

3.5 Maximal thermal power with fixed mass flow and supply temperature

Maximal thermal power of a FCU presents a physical limit of the available thermal energy when optimizing thermal energy inputs per zone. Operating sampling time of submodule for optimization of thermal energies per zones makes time constant of water inside FCU negligible, transforming (3-11) to:

$$0 = q_{\mathrm{w}} c_{\mathrm{w}} (T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}}) - U_{\mathrm{o}}(q_{\mathrm{w}},x)(\overline{T}_{\mathrm{w}} - T_{\mathrm{a}}^{\mathrm{in}}), \tag{3-23}$$

from where analytical expression for calculation of return medium temperature $T_{\mathrm{w}}^{\mathrm{out}}$ is expressed:

$$T_{\mathrm{w}}^{\mathrm{out}} = \frac{\left(q_{\mathrm{w}}c_{\mathrm{w}} - \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2}\right)}{\left(q_{\mathrm{w}}c_{\mathrm{w}} + \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2}\right)} T_{\mathrm{w}}^{\mathrm{in}} + \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{\left(q_{\mathrm{w}}c_{\mathrm{w}} + \frac{U_{\mathrm{o}}(q_{\mathrm{w}},x)}{2}\right)} T_{\mathrm{a}}^{\mathrm{in}}. \tag{3-24}$$

From (3-24) maximal power is generated when high speed is on. By inserting (3-24) into (3-20) and defining $x = H$ it follows:

$$P_{\mathrm{a,max}}(q_{\mathrm{w}}) = \frac{2q_{\mathrm{w}}c_{\mathrm{w}}U_{\mathrm{o}}(q_{\mathrm{w}},H)}{2q_{\mathrm{w}}c_{\mathrm{w}} + U_{\mathrm{o}}(q_{\mathrm{w}},H)} (T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{a}}^{\mathrm{in}}) \tag{3-25}$$

4 Radiator identification submodule (Z.PE.2)

A radiator is a type of heat exchanger designed to transfer heat from one medium to another for the purpose of heating or cooling. Regardless of its purpose (heating the environment or cooling the fluid supplied to it), it is always a source of heat to its environment. Although the term “radiator” suggests that radiation is the dominating process in heat transfer between the radiator and the environment, that is not correct. Regardless of their material of production and irrespective of their designs, the majority of radiators will heat the environment mostly through the mechanism of convection (approximately 80 % of the total heat transfer between the radiator and the environment), leaving the remaining 20 % to be emitted through radiation.

Figure 4.1. Steel panel radiator

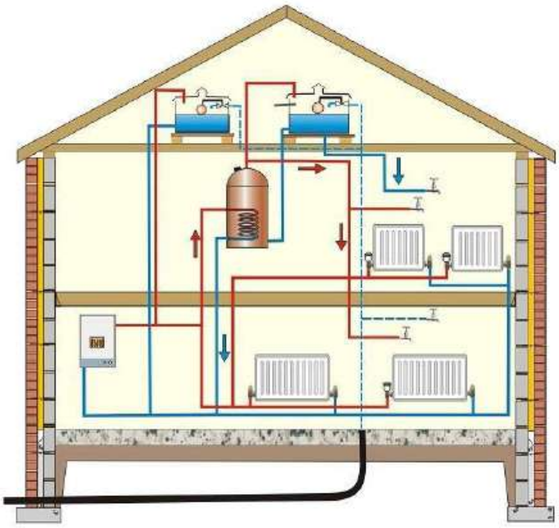


Figure 4.2. Central heating system

Design, material, size and colours of radiators can be different. Comparisons between different radiators are based on volume and projection area, weight, thermal inertia, ease of installation, water volume, life, corrosion, aesthetic, security, the required amount of heating surface, pressure resistance and price. It is possible to classify radiators according to construction material as cast iron radiators, steel radiators and aluminum radiators. Panel radiators, panel radiators with extended surfaces, convectors, low surface temperature radiators and towel radiators are examples of steel radiators. Even though steel is the more common material in radiators, aluminum is also used in the production of radiators. Better heat conductivity properties of aluminum 4 improves heat dissipation rate. Short heat-up period and immediate response to desired temperature make In addition, better corrosion resistance is achieved by using aluminum item since formation of surface layer of aluminum oxide when exposed to air. Reduced load of aluminum radiators ease to place them [31].

This article presents the mathematical model of a panel radiator that is used to heat buildings. Differential equations are used to describe the heat transfer processes inside the radiator and between the radiator and the environment. Also, experimental setup for identification of a radiator model is described.

The submodule interface is defined in the following tables (Table 4.1, Table 4.2).

4.1 Submodule inputs

Table 4.1. Required inputs for radiator identification submodule.

Variable name	Notation	Description
Historical temperature profile from zone (minute-scale of the sampling time)	T_z	Data taken from the database
Historical profile of valve actuation in the zone (minute-scale of the sampling time)	V_x	Data taken from the database
Historical temperature profile of the supply medium from a calorimeter (minute-scale of the sampling time)	T_w^{cal}	Data taken from the database
Historical temperature profile of the return medium from a calorimeter (minute-scale of the sampling time)	$T_{\text{return,cal}}$	Data taken from the database (might not be needed, but will be available)
Historical profile of the flow from a calorimeter (minute-scale of the sampling time)	q_{cal}	Data taken from the database
Historical profile of energy (power) recorded on the calorimeter (minute-scale of the sampling time)	$E_{\text{cal}}\ (P_{\text{cal}})$	Data taken from the database
Historical temperature profile from the return medium temperature sensor on a fan coil (minute-scale of the sampling time)	T_{w}^{out}	Data taken from the database
Historical temperature profile from the supply medium temperature sensor on a fan coil (minute-scale of the sampling time)	T_{w}^{in}	(optional) If not existing, measurement of the temperature on the calorimeter should be used, and additionally a characteristic of the temperature drop along the pipeline from the heat loss model should be used

4.2 Submodule outputs

Table 4.2. Outputs of the radiators identification submodule.

Variable name	Notation	Description
Parameters of the radiator model	a, b, C, U_0, n	Parameters needed for calculation of maximum energy for the MPC module, for the interface submodule functioning, and for calculation of energy inputs for identification of a simplified building dynamic model and for on-line estimation of its states and disturbances

4.3 Mathematical model of a panel radiator

A model is a representation of reality that retains its salient features. The first task is to identify the system under study. Modelling usually implies approximating the real geometry to an ideal geometry (assuming perfect planar, cylindrical or spherical surfaces, or a set of points, a given interpolation function, and its domain), approximating material properties (constant values, isotropic values, reference material values, extrapolated values), and approximating the heat transfer equations (neglecting some contributions, linearizing some terms, assuming a continuum media, assuming a discretization, etc.). Modelling material properties introduces uncertainties because density, thermal conductivity, thermal capacity, emissivity, etc., depend on the base materials, their impurity contents, bulk and surface treatments applied, actual temperatures, the effects of aging, etc. In most cases materials properties are modelled as uniform in space and constant in time for each material, but the worthiness of this model and the right selection of the constant-property values, requires insight [32].

For radiators, there are three mechanisms of heat transfer that need to be described with mathematical equations:

1. Forced heat convection from fluid (water in the tube) to the inner wall of the tube
2. Heat conduction through the tube wall
3. Coupled radiation and natural convection from the outer tube wall to the outside fluid (air)

The following assumptions are made:

- Since the water inlet and outlet temperature differ significantly, the arithmetic average of those two temperatures was taken as the water temperature in the radiator:

$$T_{\mathrm{w}}^{\mathrm{av}} = \frac{T_{\mathrm{w}}^{\mathrm{in}} + T_{\mathrm{w}}^{\mathrm{out}}}{2}, \tag{4-1}$$

- No heat losses from the room to the external environment were taken into account
- System is hydraulically balanced

The heat transfer processes are described by the following differential equation:

$$m_w c_w \frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = q_{\mathrm{w}} c_w (T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}}) - U_0 (T_{\mathrm{w}}^{\mathrm{av}} - T_{\mathrm{z}})^n, \tag{4-2}$$

or:

$$\frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = \frac{q_{\mathrm{w}}}{m_w} (T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}}) - \frac{U_0}{m_w c_w} (T_{\mathrm{w}}^{\mathrm{av}} - T_{\mathrm{z}})^n, \tag{4-3}$$

where m_w is the mass of water inside the radiator, c_w is the specific heat capacity of the water, q_w is the medium mass flow of the water through the radiator and T_z is the room temperature. $P_w = q_{\mathrm{w}} c_w (T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}})$ represents the thermal power on the water side. $P_{\mathrm{t,r}} = U_0 (T_{\mathrm{w}}^{\mathrm{av}} - T_{\mathrm{z}})^n$ represents the thermal power of a radiator affecting the zone where U_0 is the overall heat transfer coefficient and n is the radiator exponent describing the type of the radiator. For standard panel radiators the value of n is around 1.33. This is a dynamic model of a water radiator when $q_w \neq 0$. In the identification process the unknown parameters will be $a = \frac{1}{m_w}$, $b = \frac{U_0}{m_w c_w}$ and n, so the equation can be written in the following form:

$$\frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = aq_w\left(T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}}\right) - b(T_{\mathrm{w}}^{\mathrm{av}} - T_{\mathrm{z}})^n, \tag{4-4}$$

Through the process of identification the values of a, b and n will be estimated, but there is still unknown parameter U_0 that needs to be identified because the transmitted thermal power between the water side and the air in the room needs to be known for the energy input control. It will be identified using the steady state values of transmitted thermal power $P_{\mathrm{t,r}}$. The whole identification process is described in the section 4.5.

It is also important to take into account situation where there is no medium mass flow ($q_w = 0$). Values of medium mass flow and inlet water temperature are inserted in the equation ($q_w = 0$ and $T_{\mathrm{w}}^{\mathrm{in}} = 0$) and dynamic model for the situation with no medium mass flow is:

$$\frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = -\frac{U_0}{m_w c_w}(T_{\mathrm{w}}^{\mathrm{out}} - T_{\mathrm{z}})^n, \tag{4-5}$$

or

$$\frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = -b(T_{\mathrm{w}}^{\mathrm{out}} - T_{\mathrm{z}})^n, \tag{4-6}$$

4.4 Radiator model simulation in Simulink

Detailed water radiator model has been made and simulated in Simulink. Simulink is a block diagram environment for multidomain simulation and model-based design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

The heat transfer processes in the radiator and between the radiator and the environment have been modelled by using the following equations:

$$m_w c_w \frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = q_{\mathrm{w}} c_w (T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}}) - U_{\mathrm{w}}(T_{\mathrm{w}}^{\mathrm{av}} - T_{\mathrm{r}}), \tag{4-7}$$

$$m_r c_r \frac{dT_r}{dt} = U_{\mathrm{w}}(T_{\mathrm{w}}^{\mathrm{av}} - T_{\mathrm{r}}) - U_a (T_r - T_z)^n \tag{4-8}$$

where m_w is the mass of water inside the radiator, c_w is the specific heat capacity of the water, m_r is the mass of the radiator skin, c_r is the specific heat capacity of the radiator, q_w is the medium mass flow of the water through the radiator, m_a is the mass of the air in the room, c_a is the specific heat capacity of the water, T_r is the radiator skin surface temperature and T_z is the room temperature. This detailed mathematical model of a radiator has been taken as a representation of a real radiator and was simulated in Simulink. System inputs are inlet water temperature $T_{\mathrm{w}}^{\mathrm{in}}$, medium mass flow q_w and room temperature T_z. System outputs are outgoing radiator temperature $T_{\mathrm{w}}^{\mathrm{out}}$ and radiator skin surface temperature T_r.

For the purpose of identification, the system inputs values have been changed over time during the simulation to obtain the reliable data of system behaviour with different values so the system parameters could be found more accurately. The medium mass flow and inlet water temperature never changed their values at the same time so that their effect on outgoing water temperature could have been seen clearly. The change in either one of these two variables occurred every 25 minutes during the simulation. The inlet water temperature has changed from 62 °C to 75 °C and medium mass flow has changed from 0.025 kg/s to 0.038 kg/s. Continued state-space model of one room in FER building was used to get the room temperature T_z. Inputs in that steady-state model were outside temperature and power transmitted between the radiator and the air in the room, while output was temperature in that room which was then used as an input in radiator model. The graphs of those values are shown on the bottom pictures.

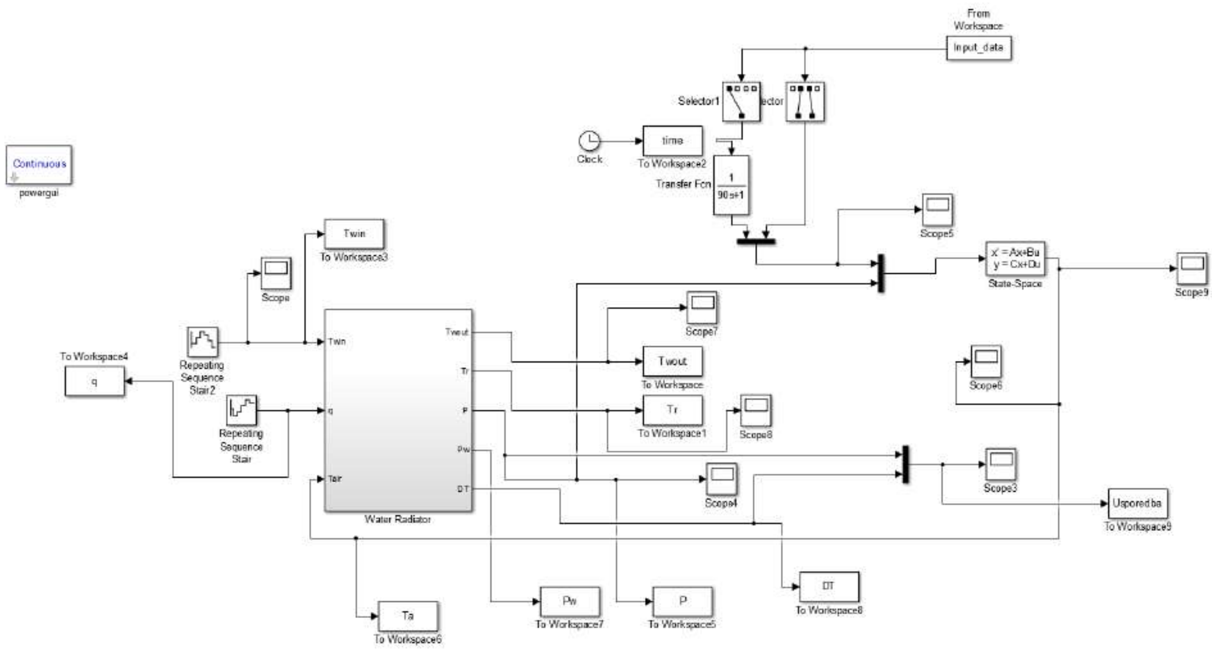


Figure 4 .4.3. Dynamic radiator model in Simulink

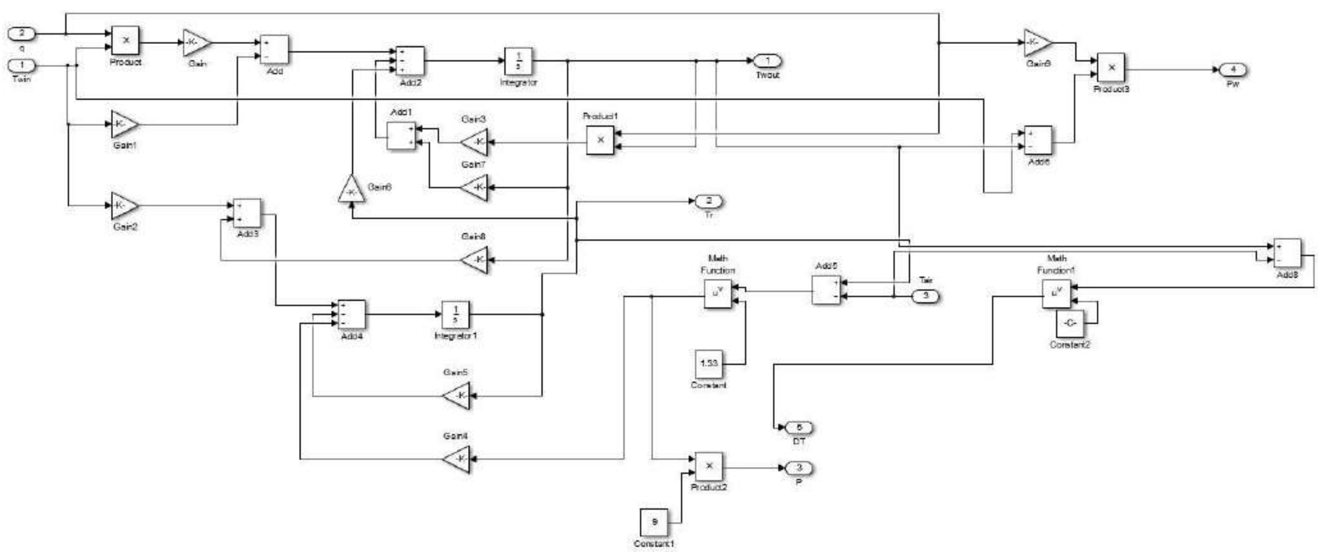


Figure 4.4. Dynamic model in Simulink

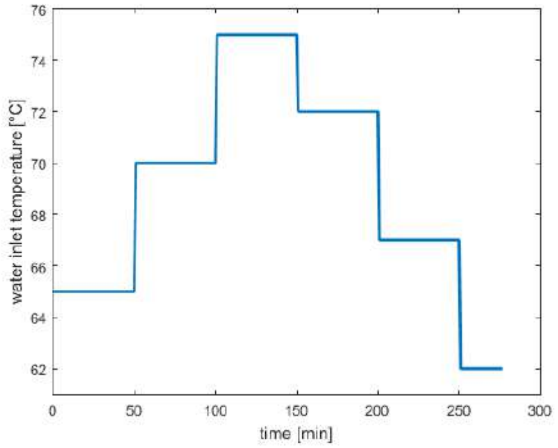


Figure 4.5. Temperature of the water inlet during the simulation.

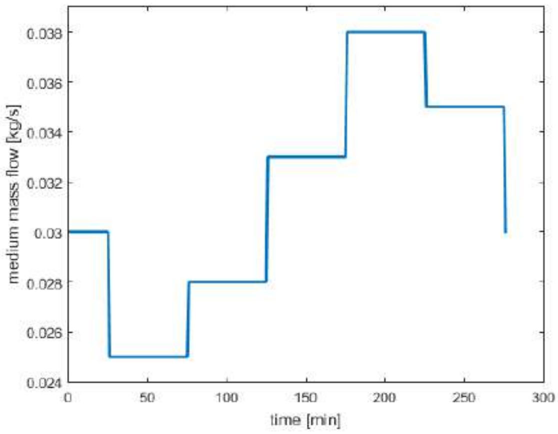


Figure 4.6. Medium mass flow during the simulation.

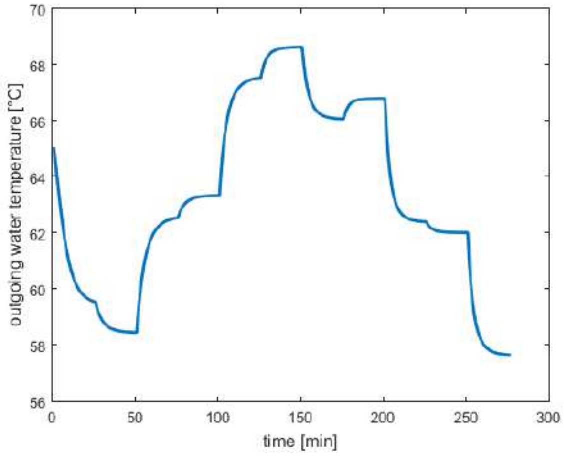


Figure 4.7. Temperature of the radiator outgoing water.

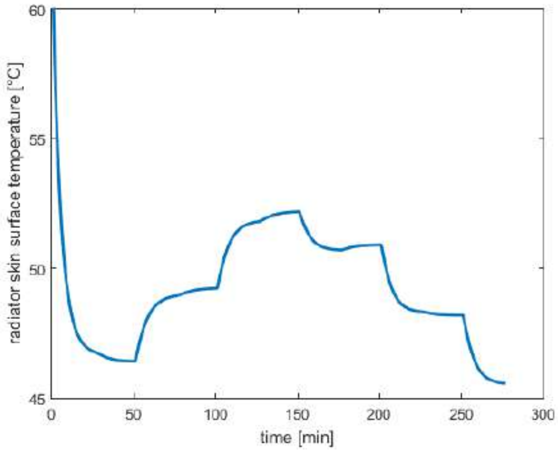


Figure 4.8. Temperature of the radiator skin surface during the simulation.

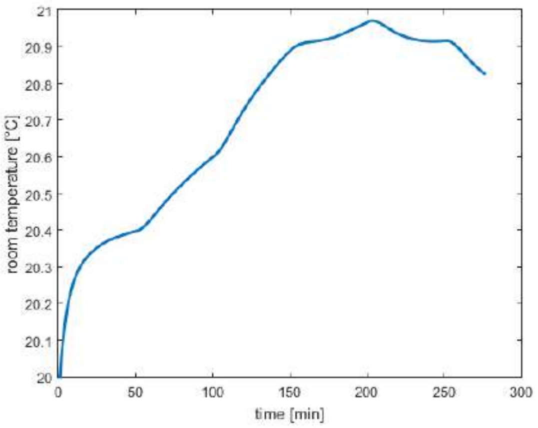


Figure 4.9. Room temperature during the simulation.

4.5 System identification

The nonlinear optimization procedure was carried out in MATLAB in order to estimate the unknown parameters *a*, *b* and *n*. After that the parameter U_0 is identified by using the steady-state approach. In chapter 4.3. an assumption has been made that the water temperature inside the radiator is the arithmetic average of water inlet temperature and water outlet temperature and that value has also been used in detailed mathematical model of a radiator in Simulink, but with real measurements on radiators in pilot buildings that temperature will maybe significantly differ from arithmetic average of $T_{\mathrm{w}}^{\mathrm{in}}$and $T_{\mathrm{w}}^{\mathrm{out}}$. To avoid errors, another parameter *C* is added to identification procedure and it is used to find out how much does each one of these two water temperatures affect the water temperature inside the radiator. The equation (4-4) is now written in the following form:

$$\frac{dT_{\mathrm{w}}^{\mathrm{out}}}{dt} = aq_w\left(T_{\mathrm{w}}^{\mathrm{in}} - T_{\mathrm{w}}^{\mathrm{out}}\right) - b(\,CT_{\mathrm{w}}^{\mathrm{in}} + (1-C)T_{\mathrm{w}}^{\mathrm{out}} - T_z)^n, \tag{4-9}$$

Since we used the arithmetic average of water inlet temperature and water outlet temperature in Simulink model, there is no need for parameter *C* to be used now because we know that its value is 0.5, but it will be used on real radiator measurement to eventually improve the identification.

The equation (4-10) is used to calculate the minimized squared error between the measured temperature of outgoing water from the Simulink simulation and simulated temperature of the outgoing water in MATLAB environment.

$$\min_{a,b,n,C} \ \| T_{measured} - T_{sim} \|^2 \ . \tag{4-10}$$

By the identification process the values of unknown parameters have been obtained and they are shown in the Table 4.5.

Table 4.5: Estimated radiator parameters**.**

Model parameters	*a*	*b*	*n*	*C*
Estimated values	*0.105*	*2.259·10^{-4}*	*1.163*	*0.500*

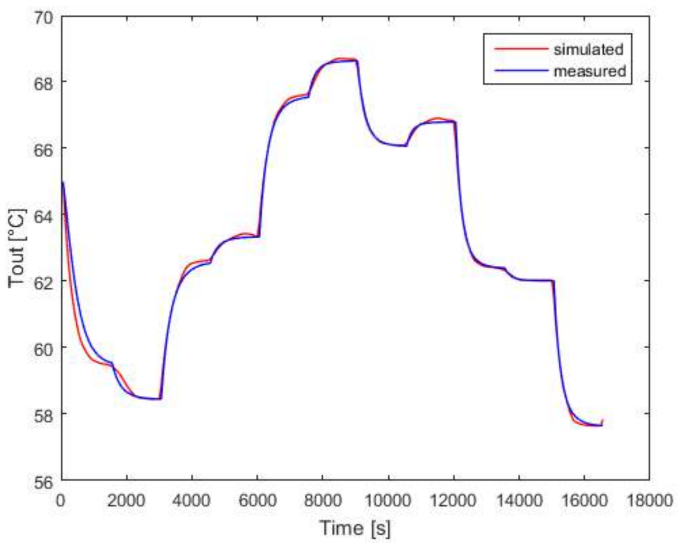


Figure 4.10. Measured and identified outgoing water temperature .

After the parameters *a*,*b*,*n* and *C* have been found, there is still unknown value of parameter U_0 that needs to be estimated, since it has been hidden in the parameter *b* ($b = \frac{U_0}{m_w c_w}$) and thus it can not be estimated directly. The steady-state approach is used to find out U_0. In steady state there is no change in $T_{\mathrm{w}}^{\mathrm{out}}$ and the equation (4-2) is simplified:

$$0 = q_w c_w\left(T_{\mathrm{w}}^{in} - T_{\mathrm{w}}^{\mathrm{out}}\right) - U_0\left(T_{\mathrm{w}}^{\mathrm{av}} - T_z\right)^n, \tag{4-11}$$

In this case, power on the water side P_w is equal to the power on the air side $P_{\mathrm{t,r}}$:

$$q_w c_w\left(T_{\mathrm{w}}^{in} - T_{\mathrm{w}}^{out}\right) = U_0\left(T_{\mathrm{w}}^{\mathrm{av}} - T_z\right)^n, \tag{4-12}$$

The heat transfer coefficient U_0 can now be written as:

$$U_0 = \frac{q_w c_w\left(T_{\mathrm{w}}^{in} - T_{\mathrm{w}}^{out}\right)}{\left(T_{\mathrm{w}}^{\mathrm{av}} - T_z\right)^n} \tag{4-13}$$

Since all the other values in equation are known and *n* is already estimated, U_0 can now be identified. 10 steady state values of water-side power P_w have been taken and divided by $\left(T_{\mathrm{w}}^{\mathrm{av}} - T_z\right)^n$ to get 10 values of U_0 and then the average value is taken as U_0. Air-side power $P_{\mathrm{t,r}}$ is now known and it is compared to water side power measured in Simulink. The comparison is shown on picture 4.5. The biggest difference between these two variables is 14 Watts.

Table 4.6: Estimated radiator parameter**.**

Model parameter	***U_0***
Estimated value	*9.17*

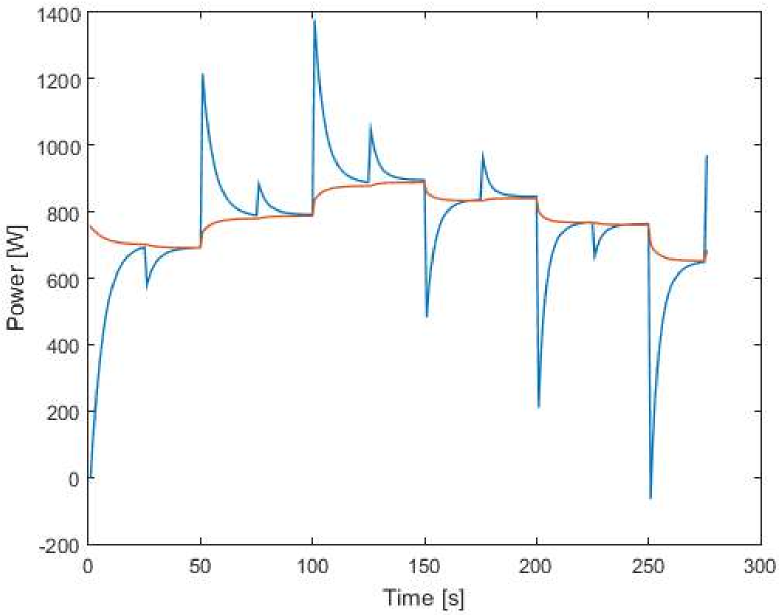


Figure 4.11. Measured water-side power and identified air-side power.

When unknown parameters are identified, the thermal energy aquired by zone actuators can be calculated using the right-hand side part of equation (4-6):

$$\int_{0}^{Td} U_0 (T_{\mathrm{w}}^{\mathrm{av}} - T_z)^n dt. \tag{4-14}$$

This equation becomes simpler for the situation with no medium mass flow ($q_w = 0$):

$$\int_{0}^{Td} U_0 \left(T_{\mathrm{w}}^{\mathrm{out}} - T_z\right)^n dt\,. \tag{4-15}$$

4.6 An experimental setup for identification of a radiator model

The measurements that need to be performed in order to make identification of the radiator models installed in pilot buildings (HEP building in Zagreb, Primary school and Sports centre in Idrija, school in Strem) are described in this section. There are five parameters that are crucial for the identification: T_z (temperature in the the room), T_{w}^{cal} (temperature of the water at the duct inlet), $T_{\mathrm{w}}^{\mathrm{out}}$ (temperature of the radiator outgoing water), $q_{\mathrm{w}}^{\mathrm{cal}}$ (overall mass flow of the water through the duct) and $E_{\mathrm{t,r}}$ (overall energy consumption).

Supply and return water temperature, overall flow of the water and overall energy consumption can be measured by calorimeter. These devices can be put on each radiator in the building to measure the required parameters, but since they are rather expensive the optimal would be to reduce their number as much as possible by modifying the way of measurement. For example, some of the buildings have two major heating loops per floor (north-south side or east-west side) and it will be enough to put two calorimeters on each floor, one for each loop. The assumption is that radiators are parallel connected and that pressure drop is constant within the whole system. If all radiators are of the same type, then the water mass flow of each radiator q_w is equal to water mass flow of the loop $q_{\mathrm{w}}^{\mathrm{cal}}$, that is measured by calorimeter, divided by the number of radiators connected to the loop.

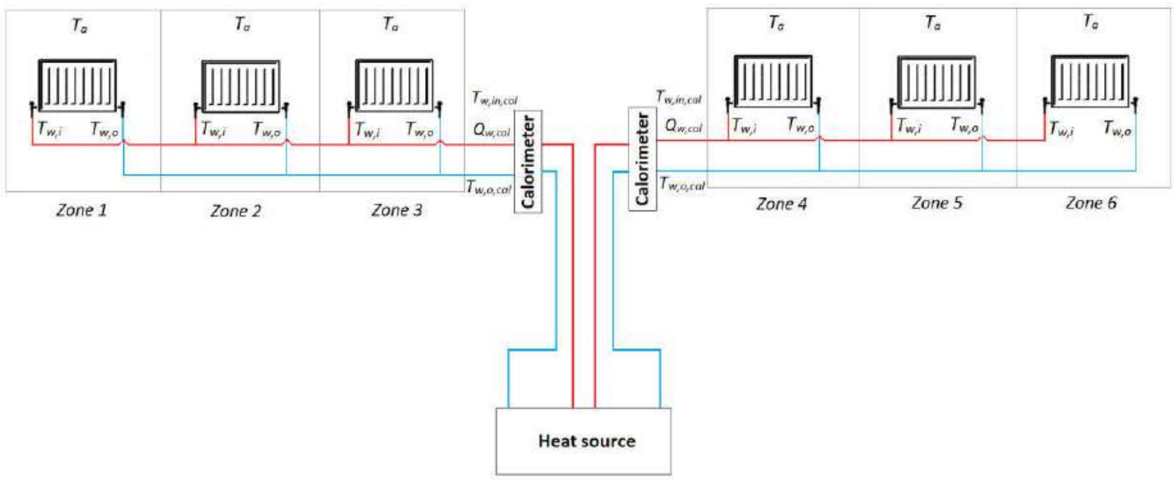


Figure 4.14. Required measurements for one floor of the building.

Also, all radiators should be equipped with sensors measuring the temperature of the radiator outgoing water $T_{\mathrm{w}}^{\mathrm{out}}$ and temperature of the water at the radiator inlet $T_{\mathrm{w}}^{\mathrm{in}}$. If there is no sensor for measurement of $T_{\mathrm{w}}^{\mathrm{in}}$, measurement of the temperature on the calorimeter should be used, and additionally a characteristics of the temperature drop along the pipeline should be assessed.

In every zone there should be a temperature sensor measuring the air layer in the room T_z and that sensor has to be placed on height that is at least the 50 % of the height between the floor surface and the ceiling.

Algorithm 1: Identification test algorithm

INITIALIZATION PHASE
- Sensor calibration if necessary

TEST PHASE
- Shut down all units connected to the same duct or assure their constant operation
- Run the identification procedure on a particular unit
- Data acquisition

DATA PROCESSING PHASE
- radiator model identification (finding *a*,*b, n* and *C* - Eq.4-9)
- overall heat transfer coefficient U_0 estimation (Eq. 4-13)
- calculate air-side power $P_{t,r}$

Heat losses

Since the temperature at the radiator inlet will not be measured directly, the temperature measured by calorimeter at the duct inlet will be used. Temperature of the water at the radiator inlet is equal to the temperature of the water at the duct inlet that is measured by calorimeter reduced by total temperature heat losses occurred due to transmission of the medium through the pipeline [4]. The following measuring procedure should be done to determine those losses: On the first day of measurement the inlet water temperature should be set on, for example, 60 °C and the water temperature on the radiator inlet should be observed. The next day that temperature could be set on 80 °C and the same measuring procedure should be done. Two measurements will be enough to make linear approximation of heat losses in form $T_{loss}=aT_{w,i,cal}+b$. In ideal case, if the pipes were perfectly isolated, both temperatures would be the same. The pipes in each of the pilot buildings are unfortunately not isolated, so T_{w}^{in} will differ from $T_{\mathrm{w}}^{\mathrm{cal}}$. Main goal will be to determine if those two temperatures differ significantly, or that difference can be neglected and the $T_{\mathrm{w}}^{\mathrm{cal}}$ can be taken as the radiator inlet temperature.

5 Floor heating identification submodule (Z.PE.3)

A floor heating/cooling is one another controlled type of heating or cooling application. It achieves thermal comfort through conduction, convection and radiation. Floor heating/cooling has been widely used in buildings with the advantage of uniform room temperature distribution. Other important advantage of the floor heating/cooling is that it can use low energy resources such as solar hot water, heat pumps and condensing boilers.

In the design and the setup of the floor heating/cooling system, surface temperature is one of the most important parameters to be considered. In order to avoid condensation, surface temperature should be always higher than the temperature of the surrounding air at the dew point. Based on this conclusion, it is recommended that the surface temperature should be kept between 19°C and 29°C [35].

There are two kinds of floor heating/cooling constructions. One is called homogenous and the other multilayer floor system [36]. With the homogenous floor just one type of material (concrete, gypsum cement or mortar) is placed above the water pipes [37]. The material of surface layer in multilayer floor can be wood, tile or some other material.

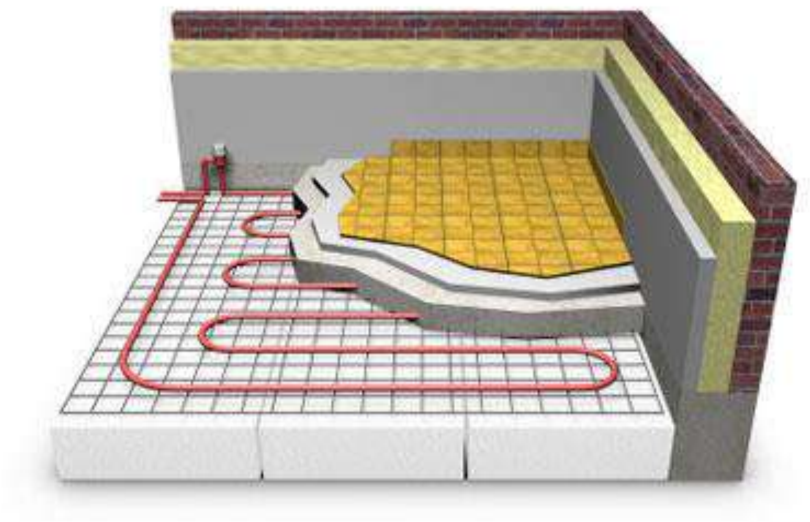


Figure 5.1. Multilayer floor heating/cooling system.

5.1 Submodule inputs

The submodule interface is defined in the following tables (**Table 5.1**, **Table 5.2**).

Table 5.1. Required inputs for floor heating identification submodule.

Variable name	Notation	Description
Historical temperature profile from zone (minute-scale of the sampling time)	T_z	Data taken from the database
Historical profile of valve actuation in the zone (minute-scale of the sampling time)	V_x	Data taken from the database
Historical temperature profile of the supply medium from a calorimeter (minute-scale of the sampling time)	T_{w}^{cal}	Data taken from the database
Historical temperature profile of the return medium from a calorimeter (minute-scale of the sampling time)	$T_{\mathrm{return,cal}}$	Data taken from the database (might not be needed, but will be available)
Historical profile of the flow from a calorimeter (minute-scale of the sampling time)	Q_{cal}	Data taken from the database
Historical profile of energy (power) recorded on the calorimeter (minute-scale of the sampling time)	$E_{\mathrm{cal}}\ (P_{\mathrm{cal}})$	Data taken from the database
Historical temperature profile from the return medium temperature sensor on a floor heating system (minute-scale of the sampling time)	T_{w}^{out}	Data taken from the database
Historical temperature profile from the supply medium temperature sensor on a floor heating system (minute-scale of the sampling time)	T_{w}^{in}	(optional) If not existing, measurement of the temperature on the calorimeter should be used, and additionally a characteristic of the temperature drop along the pipeline from the heat loss model should be used

5.2 Submodule outputs

Table 5.2. Outputs of the floor heating identification submodule.

Variable name	Notation	Description
Parameters of the floor heating/cooling element model	$A_{\mathrm{fh}}, B_{\mathrm{fh}}, C_{\mathrm{fh}}, D_{\mathrm{fh}}$	Parameters needed for calculation of maximum energy for the MPC module, for the interface submodule functioning, and for calculation of energy inputs for identification of a simplified building dynamic model

5.3 Thermodynamic floor heating/cooling unit model

For the purpose of subsequent developments it is assumed that the water pipes are placed above the insulation layer. Concrete layer is then placed above the water pipes and finally a tile surface layer is applied on the top. Mathematical model of a system will be written and reviewed from the side of the outgoing water temperature circulating through the pipes.

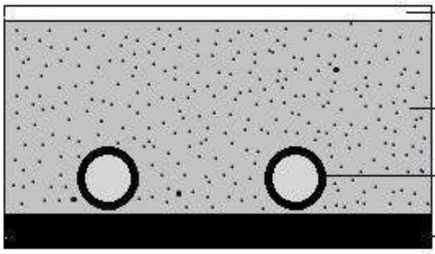


Figure 5.2. Vertical cross-section of a multilayer floor heating/cooling.

Heat transfer within the system occurs through convection, conduction and the radiation of the heat [39]. First the heat is convected from the water circulating in the pipe to the inner wall of the pipe. After that the heat is conducted through the pipe material and through the concrete layer above the pipes. The final stage of the heat transfer into the zone is done through the convection and radiation of the heat from the surface layer to the air layer in the room.

The differential equation for the temperature of the water in the pipe $T_{w,out}$ can be written in the following form :

$$m_w c_w \frac{dT_w^{out}}{dt} = \begin{cases} Q_w c_w\left(T_w^{in} - T_w^{out}\right) - U(T_c - T_z)^n & Q_w \neq 0, \\ U(T_c - T_z)^n & Q_w = 0, \end{cases} \tag{5-1}$$

where m_w and Q_w are respectively mass of the water inside the pipe and medium mass flow of the water through the pipe. Parameter c_w represent specific heat capacity of the water. T_w^{out} stands for the temperature of the outgoing water from the pipes in the room and T_w^{in} stands for the temperature of the inlet water going through the pipes in to the room. T_z stands for the temperature of the air in the room(zone) and T_c stands for the temperature of the concrete layer above pipes of the floor heating/cooling system.

When we look the equation (5-1) from the energy perspective we can see that the first part of the right hand side of the equation represents the amount of energy inserted in to the system(concrete layer) whereas the second part represents the amount of energy that is transmitted from the concrete layer to the air in the room(zone). In that perspective the parameter U stands for the heat transfer coefficient between the concrete and air in the zone.

Looking from the thermodynamical perspective of the floor heating/cooling system one can surely notice that due to the physical properties of the pipes the temperature of the outgoing water $T_{w,out}$ is never going to be greater than the temperature of the concrete layer T_c above the pipes. More or less, these two will be very similar. Based on this assumption we can now rewrite the equation (5-1):

$$m_w c_w \frac{dT_w^{out}}{dt} = \begin{cases} Q_w c_w\left(T_w^{in} - T_w^{out}\right) - U(T_w^{out} - T_z)^n & Q_w \neq 0, \\ U(T_w^{out} - T_z)^n & Q_w = 0, \end{cases} \tag{5-2}$$

From mathematical and identification point of view the equation (5-2) is much less complicated for the identification procedure in comparison with the equation (5-1). The temperature of concrete layer T_c is not going to be available for the identification procedure in pilot building in Strem, Austria.

Now we can write the final differential equation of our model with parameters *a,b* and *n* which can be seen in the following equation.

$$\frac{dT_w^{out}}{dt} = aQ_w\left(T_w^{in} - T_w^{out}\right) - b(T_w^{out} - T_z)^n \tag{5-3}$$

Where *a* stands for the relation $1/m_w$, *b* stands for the relation $U/m_w c_w$. Three unknown parameters *a,b* and *n* will later be optimized and identified with the help of the MATLAB.

5.4 Model simulation in IDA ICE software

The floor heating/cooling system model has been developed and simulated for the purposes of parameter identification in IDA ICE software. One can surely notice that the heat losses in the mathematical model equations are neglected so the simulated zone (room) has been constructed with minimal losses to the environment. The dimensions of the test room that was constructed in IDA ICE software were 4 meters in length, 4 meters in width and 3 meters in height.

The floor heating system in IDA ICE software had a maximum heating power of 100 W/m^2 with the pipes immersed into the concrete. The thickness of the concrete layer in the floor was set up to 10 cm and the pipes were immersed in it on the depth of 5 cm. Under the concrete layer the insulation layer was placed with the thickness of 10 cm. Above the concrete layer the floor coating layer approximately 1 cm thick was placed.

The outputs of the floor heating/cooling simulation were temperature of the outgoing water T_w^{out}, inlet water temperature T_w^{in}, medium mass flow of the water Q_w and the temperature of the air in the zone T_z. The inputs to the simulation were water temperature T_w^{in} and medium mass flow of the water Q_w. These inputs were adjusted according to **Figure 5.3** and **Figure 5.4**. The simulation has been carried out for 119521 samples with the sampling time of 1 minute.

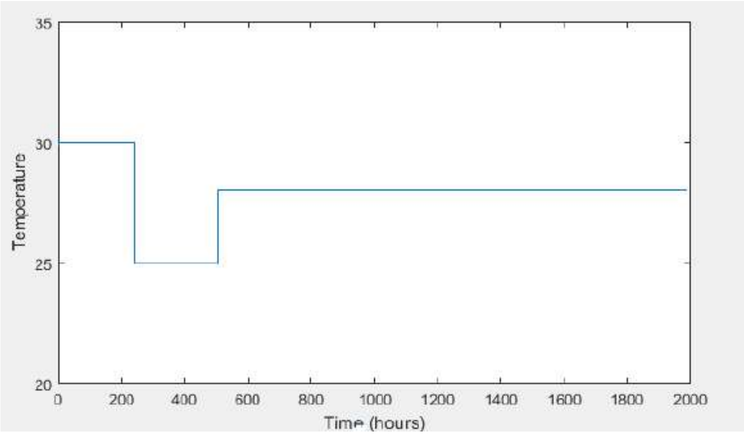


Figure 5.3. Inlet water temperature T_w^{in} during the simulation.

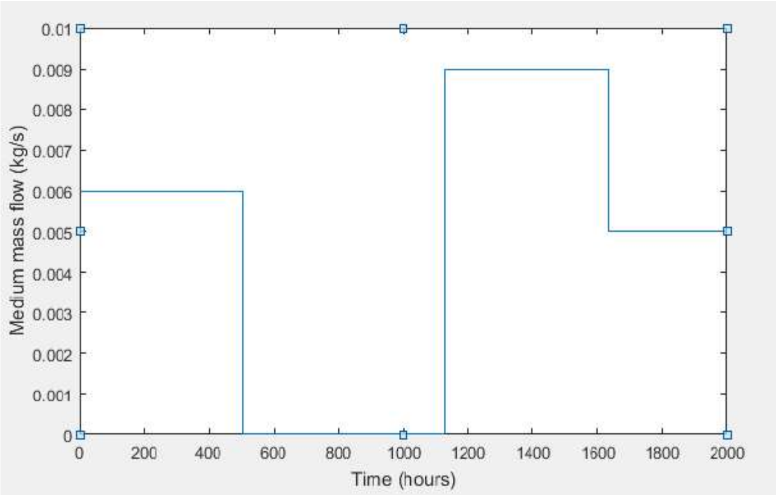


Figure 5.4. Medium mass flow Q_w during the simulation

The outputs of the floor heating/cooling simulation were temperature of the outgoing water T_w^{out} and the temperature of the air in the zone T_z. These outputs are shown in **Figure 5.5** and **Figure 5.6**.

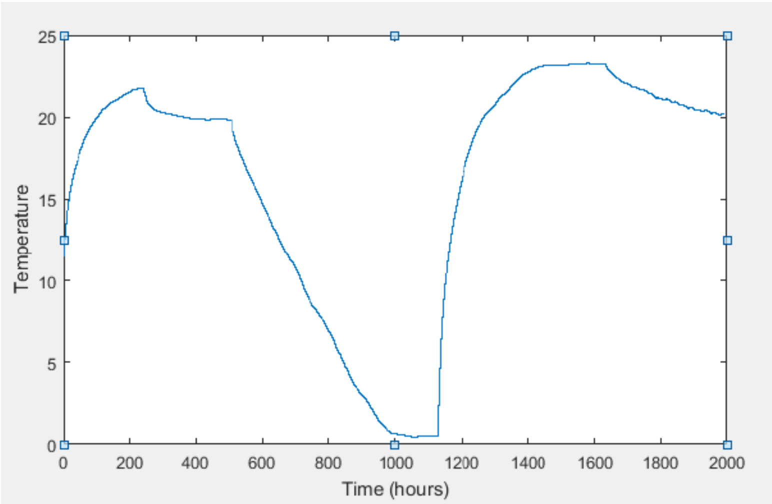


Figure 5.5. Mean air temperature in the zone (room).

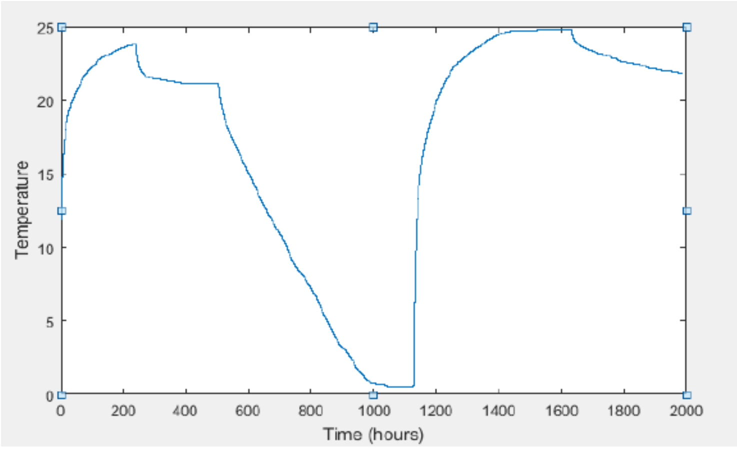


Figure 5.6. Temperature of the outgoing water during the simulation.

5.5 Thermodynamical model parameter estimation

In order to estimate the unknown parameters a, b and n from the equation and (5-3) the nonlinear optimization procedure in the MATLAB environment has to be applied based on the simulated data from the IDA ICE software simulation.

$$\min_{a,n,b} \quad \|T_{measured} - T_{sim}\|^2 . \qquad (5\text{-}4)$$

The equation (5-4) calculates the minimized squared error between the measured temperature of outgoing water from the IDA ICE simulation and simulated temperature of outgoing water in MATLAB environment. Time response of the simulated model outgoing water temperature with respect to the measured values of the outgoing water temperature can be seen in **Figure 5.7**.

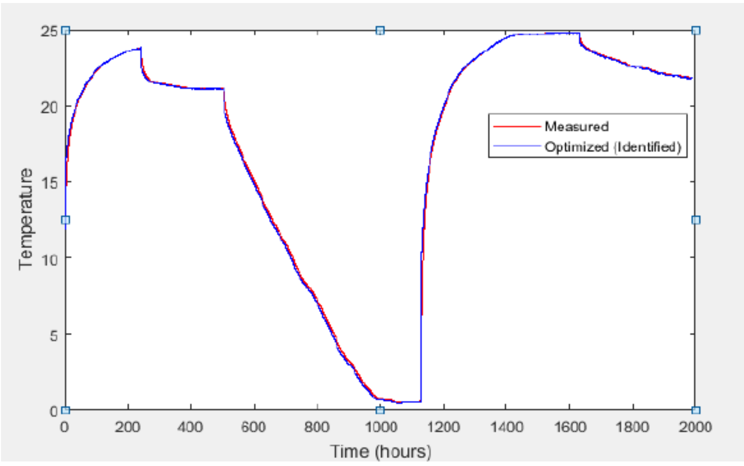


Figure 5.7. Measured outgoing water temperature compared to the one simulated with the identified parameters.

We can see that the simulated time response (blue line) of the outgoing water for the optimized parameters truly well represents the measured (red line) data from the simulation. This nonlinear optimization problem has been solved in MATLAB environment giving the outputs described as parameter a, n and b. This problem has the possibility to be solved in MATLAB using *fminsearch* or *fmincon* command. These parameters are given by the following values:

$$b = 0.0250,$$

$$n = 1.055,$$

$$a = 1.25$$

As it can be seen from the equation (5-3) the parameter U can only be optimized together with m_w and c_w where the U parameter is going to be in the numerator and the $m_w c_w$ is going to be in the denominator. So using the optimization methods in MATLAB it is possible that the real value of U will not be properly found due to the dependency on the parameters m_w and c_w.

In order to fix this problem by finding the right value of U so that the energy transmitted in to the zone could be calculated we need to apply the following procedure.

First thing that needs to be done is to find the points of the energy transmitted from the water in the steady state. These points can be found when the energy transmitted from the water is equal to the energy transmitted from the concrete on the air in the zone (room). Since we have 5 transition states there will be 5 points in steady state where the water side energy is equal to the air side energy. The steady state where the medium mass flow is set to 0 will not be taken in to account. These points are easily read from the energy graphs derived from the simulation in IDA ICE software. These energies can be seen on the figure 5.8. and 5.9.

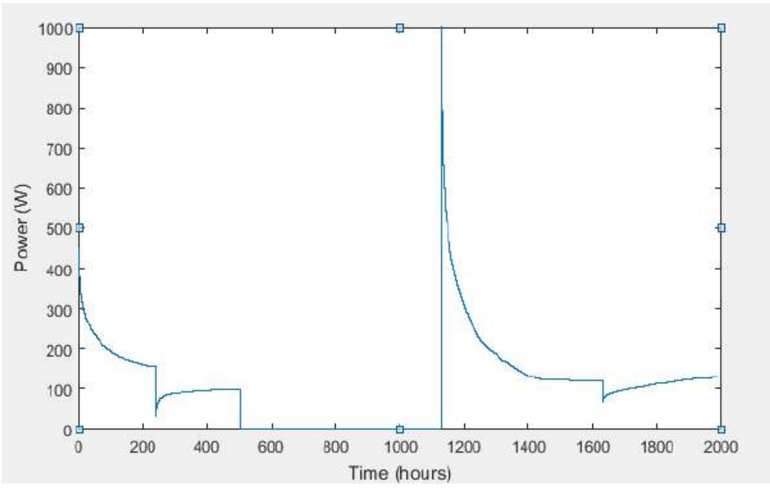


Figure 5.8. Measured energy transmitted from the water during the simulation.

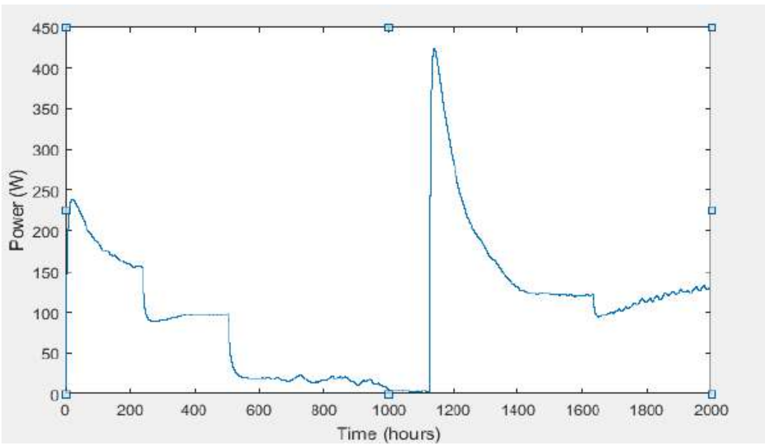


Figure 5.9. Measured energy transmitted on to the air in the zone during the simulation.

As we have now these 4 points of steady state energies we now have to calculate the heat transfer coefficient for each of these steady state by relying on the following equation:

$$U = \frac{P_{ss}}{(T_w^{out} - T_{air})^n} \tag{5-5}$$

where P_{ss} represents the energy points in 4 steady states. With the help of the equation (5-5) we now can calculate the average heat transfer coefficient *U* from the 4 steady states and use it to calculate the thermal energy given to the air in the zone. The average value of *U* calculated from the equation (5-5) is 75.5 *W/K.*

We now have needed parameter values and we are able to calculate the thermal energy acquired by zone actuators using the right hand side part of equation (5-2):

$$\int_0^{Td} U(T_w^{out} - T_{air})^n dt. \tag{5-6}$$

For easier computation can the outgoing temperature T_w^{out} at the outlet of the actuator be lineary interpolated.

With the optimized parameter *n* from the equation (5-4) and with the parameter *U* from the equation (5-5),they can be inserted in to equation (5-6) so that the energy given to the air in the zone (room) can be calculated and compared with the energy given to the air from the IDA ICE software what can be seen in Figure 5.10.

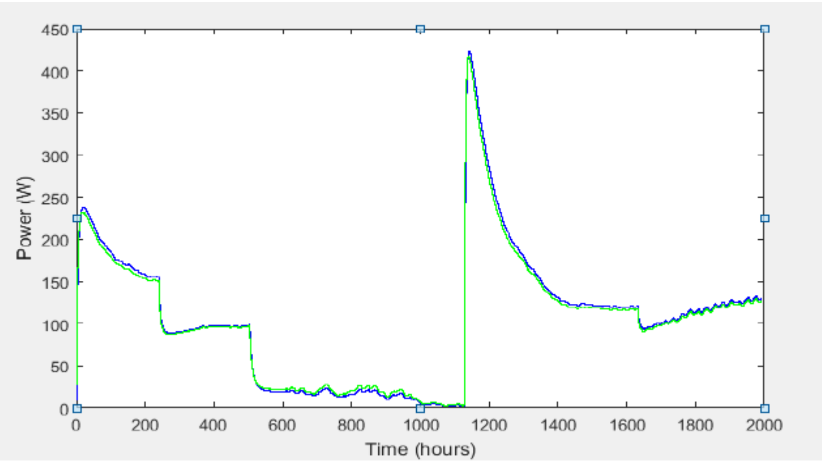


Figure 5.10.Calculated energy transmitted to the air compared to the one measured from the IDA ICE software.

As it can be seen from the Figure 5.10. the blue line that represents the energy derived from IDA ICE software truthfully follows the green line that represents the energy calculated with the optimized parameters *U* and *n.*

5.6 Outgoing water temperature prediction

One of our main variables that is going to be measured on site is the outgoing water temperature T_w^{out}. In the pilot building for elderly care in Strem (Austria) there are some constraints that need to be assessed regarding the on site measurements.

The floor heating/cooling system in building for elderly care in installed in a way that outgoing water from two or more zones in a building is mixed before it can be measured with a sensor. This in a way complicates the procedure for the identification because mathematical model for a particular zone relies completely on the T_w^{out} measurement from the zone that is being considered.

For this reason a new building model in IDA ICE software has been created with two separate zones (rooms) with the installed floor heating/cooling systems. The dimensions of the zones were the same (*4m x 4m x 3m)* and the floor heating/cooling systems with the power of 100 *W* were installed into the each room. In order to represent more truthfully our real system in Strem, a fluid mixing element is added to our IDA ICE simulation software project. This fluid mixing element is created outside the zones (rooms) so that the two outgoing pipes from each zone should be the input pipes to this element. Since we can measure now the outgoing water temperatures from the each zone and the water temperature going from the fluid mixing element, it is possible to find the mathematical relation that could estimate outgoing water temperature from the each zone as a function of zone mass flows and temperature going from the fluid mixing element.

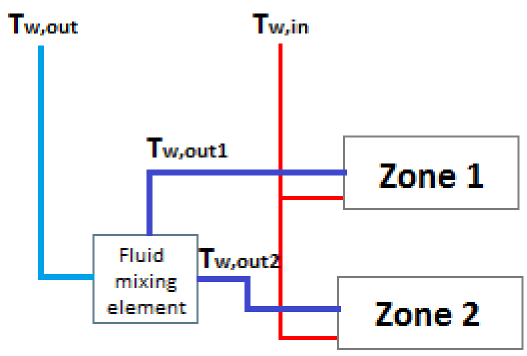


Figure 5.11. Schematic of fluid mixing principle in IDA ICE software

As we can see on the Figure 5.8., the outgoing water temperatures ($T_w^{out1}$and T_w^{out2}) from the Zone 1 and Zone 2 are not going to be measured but will have to be estimated. The temperature of the outgoing water from the fluid mixing element T_w^{out} is going to be measured.

First thing that we are going to do is to set the medium mass flow in each room to maximum an then we are going to measure the water temperature from a fluid mixing element. In a real time system we will not be able to measure T_w^{out1} and T_w^{out2} but in IDA ICE simulation these values will be accessible.

The equation (5-7) is going to be used to evaluate outgoing water temperatures in each zone.

$$T_w^{out} = \frac{Q_{w1}T_w^{out1} + Q_{w2}T_w^{out2}}{Q_{w1} + Q_{w2}} \quad (5\text{-}6)$$

where subscripts 1 and 2 represents zones (rooms) 1 and 2 created and simulated in IDA ICE. After the measuring the T_w^{out} with the maximum medium mass flows Q_{w1} and Q_{w2} we are now going to set the mass flow in one zone to minimum value or 0. This way we are going to be able to calculate the outgoing water temperature for zone (room) where the medium mass flow is not 0. The same can be done for the other room.

6 Heat disturbance prediction submodule (Z.PE.6)

Submodule for prediction of the heat disturbance evolution per zone.

6.1 Submodule inputs

Table 6.1: Required inputs for heat disturbance prediction submodule.

Variable name	Variable annotation	Variable description
Estimated heat disturbance in zone	E_d	Profile of the estimated heat disturbance in the past needed for off-line model tuning; recent values needed for on-line execution
Weather measurements	UNIZG-FER pilot site: $T_\mathrm{env}, I_{diff}^h, I_{dir}^\mathrm{n}$ Remaining pilot sites: $T_\mathrm{env}, I_\mathrm{glo}^h, I_\mathrm{glo}^\mathrm{t}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_\mathrm{env})_\mathrm{N}, (I_\mathrm{dir}^\mathrm{n})_\mathrm{N}, (I_\mathrm{diff}^\mathrm{h})_\mathrm{N}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).

Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

6.1.1 Solar irradiance data

Depending on the availability of solar irradiance measurements on different pilot sites throughout the project, two separate sets of weather measurements inputs are used.

On the UNIZG-FER pilot site, where direct normal and diffuse horizontal irradiance measurements are available, they are used as submodule inputs and paired with the same forecasted variables during submodule operation.

Due to high costs of direct and diffuse irradiance sensors other pilot sites provide measurements of global horizontal and tilted global irradiations which are then used as submodule inputs. Since measured and forecasted irradiances are now different, during submodule operation, forecasted direct and diffuse irradiance, solar angles (obtained through the use of Pysolar python library), geographical pilot site data and current datetime, are used for calculation of global horizontal and tilted global irradiances thus matching the measured and forecasted irradiance variables.

6.2 Submodule outputs

Table 6.2: Outputs of the heat disturbance prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_d	Needed for on-line operation of the submodule.
Predicted heat disturbance evolution per zone (for on-line operation of the submodule)	$(E_\mathrm{d})_\mathrm{N}$	Needed for the MPC module on the zones level.

6.3 Methodology

Based on a detailed description of artificial neural networks (ANN) given in [36], in the following sections a condensed description of ANNs structures and learning algorithms is given, together with a description of prediction module structure and operation schemes.

6.3.1 Artificial neural networks

Understanding of the human brain functioning and its learning and adaptation abilities made researchers try imitating its structure in order to imitate its capabilities in the computer systems. The basic element of the brain is a neural cell or neuron. Human brain contains 10^{11} neurons interconnected in the network with more than 10^{15} links. Although the neuron structure is rather simple, because of the immense number of links among them, a brain can perform the most complex operations. Schematic representation of a biological neuron is shown in Figure 6.1.

Neuron is composed of the cell body (soma), axon and a number of dendrites. Front end of an axon is connected to the cell body and its back end is split in a large number of branches. These branches are terminated by telodendria with their terminal buttons that touch dendrites of the other neurons. The terminal buttons contain numerous small bags with transmitters. A small distance between a telodendron of one neuron and a dendrite of another is called a synapse. Axon of one neuron forms synaptic interactions with many other neurons. Impulses generated in the cell body travel through an axon to a synapse. Depending on the efficiency of each synaptic transfer, action potentials of different intensity come over dendrites to the cell body where they are then collected and processed. If their cumulative value is greater than the neuron sensitivity threshold, a cell body generates an action potential which is spread over the axon to the other neurons, and if it is lower, the neuron remains inactive and does not generate an action potential. From the signal processing perspective, neuron operation can be divided in *synaptic operation* which gives a certain relevance (weight) to each input signal and *somatic operation* which collects all the "weighted" input signals, and due to their cumulative values, generates or does not generate a signal which is transferred towards other neurons.

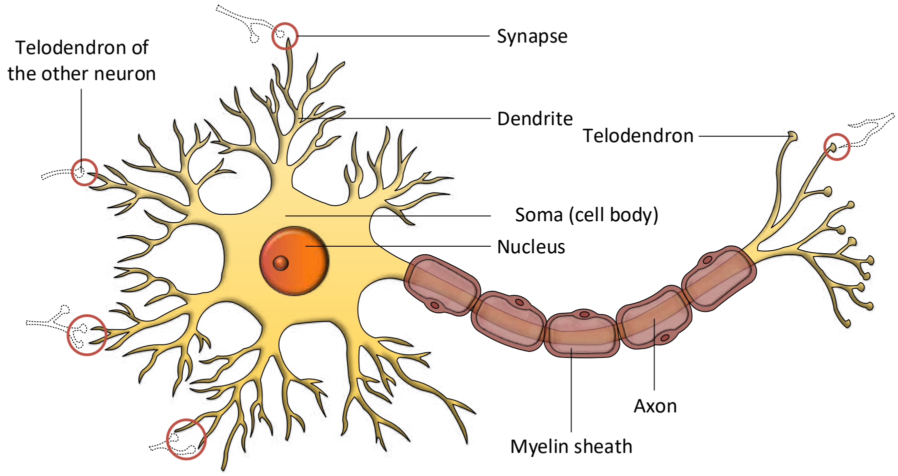


Figure 6.1: Schematic representation of a biological neuron.

6.3.1.1 Artificial neuron model

Early research in the field of artificial neurons was published by McCulloh and Pitts in 1943 and 1947 [37], [38]. Their model was based on a simple implementation of synaptic and somatic operations and was called a perceptron. Schematic representation of a perceptron is shown in Figure 6.2.

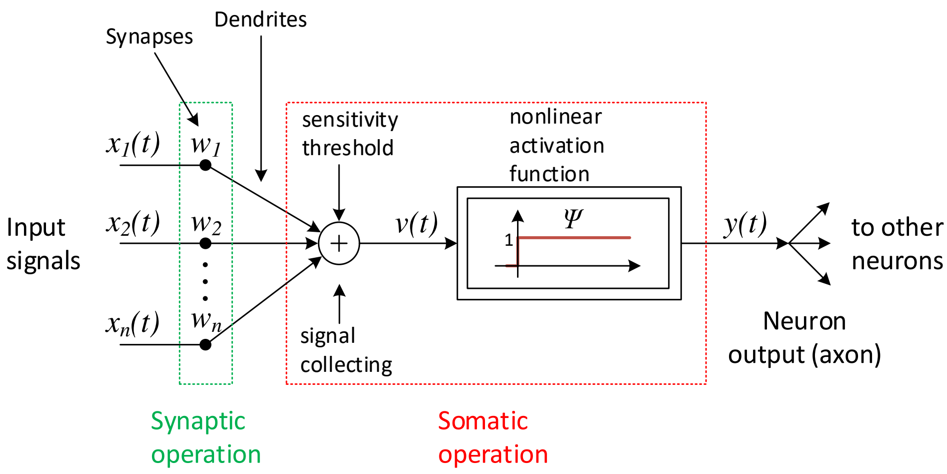


Figure 6.2: Schematic representation of a perceptron.

Synaptic operation is performed by multiplying input signals x_i with their weight coefficients w_i. Sum of all weighted signals is compared to a neuron sensitivity threshold w_{n+1}. If this sum is greater than a sensitivity threshold, nonlinear activation function ψ generates an output signal y equal to 1, and if it is less, neuron output is zero.

Mathematically, a perceptron can be described using these relations:

$$v(t) = \sum_{t=1}^{n} w_i(t)\, x_i(t) - w_{n+1}, \qquad (6\text{-}1)$$

$$y(t) = \psi(v), \qquad (6\text{-}2)$$

where:

$\boldsymbol{x_u} = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is a vector of neuron input signals;

$\boldsymbol{w_s} = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is a vector of neuron input signals;

$\boldsymbol{w_{n+1}}$ is a neuron sensitivity threshold;

$\boldsymbol{v(t)}$ is a similarity measure between input signals and synaptic weight coefficients (result of the confluence operation);

$\boldsymbol{\psi(t)}$ is a nonlinear activation function;

$\boldsymbol{y(t)}$ is a neuron output.

However, because of the too simple model of a neuron, especially because of the discontinuity in nonlinear activation function, perceptron is not able to solve some simple operations. These constraints of the perceptron can be overcome by applying a continuous differentiable activation function. Sigmoid functions are commonly used as activation functions because it was proved that the ANNs composed of at least three layers of neurons with sigmoid functions can represent any continuous function. One of the most commonly used activation functions is *tansig* defined by the following expression:

$$\psi(v) = \frac{2}{1 + e^{-2g_0 v}} - 1, \tag{6-3}$$

where g_o is an activation gain and it is usually set to 1. Because of an extension of the initial model, in literature neurons with sigmoid activation functions are also referred to as perceptrons.

Neuron models can be divided in two groups: static and dynamic models. Static neuron models, as opposed to dynamic ones, do not contain dynamic elements and their output depends exclusively on current values of input signals and weight coefficients. In this deliverable only ANNs with static neuron models are analysed.

6.3.1.2 Multilayer perceptron

Static neural networks are most commonly used ANNs, especially in identification and control applications. A basic element of the static ANN is a static neuron. In static ANNs neurons are organised in a feedforward way, i.e.: each neuron can be connected to the network inputs and/or to other neurons, but in the way that no feedback connections are formed. Therefore, static ANNs do not contain any dynamic elements and that makes them statically stable which is their most important advantage in relation to dynamic ANNs. However, in order to model a dynamic system, delayed input and output signals have to be explicitly included in the vector of input signals of the static ANN. The most commonly used static ANNs are multilayer perceptrons (MLP) whose structure is presented in Figure 6.3. MLPs consist of perceptrons organized in serially connected layers. Layers are often labelled with numbers $0, 1, 2, \cdots, L$, while for the number of nodes in the l-th layer we use label $n(l)$. The zeroth layer only transfers the input vector to an input of the first layer, L-th layer is an output layer, while layers between them are called hidden layers. Every neuron in a hidden layer is connected to all the neurons in two neighbouring layers with unidirectional feedforward connections. Connections between neurons of the neighbouring layers are represented by synaptic weight coefficients which act as signal gains on the corresponding connections. Values of the synaptic weight coefficients determine the network behaviour, i.e.: its ability of approximating a nonlinear function.

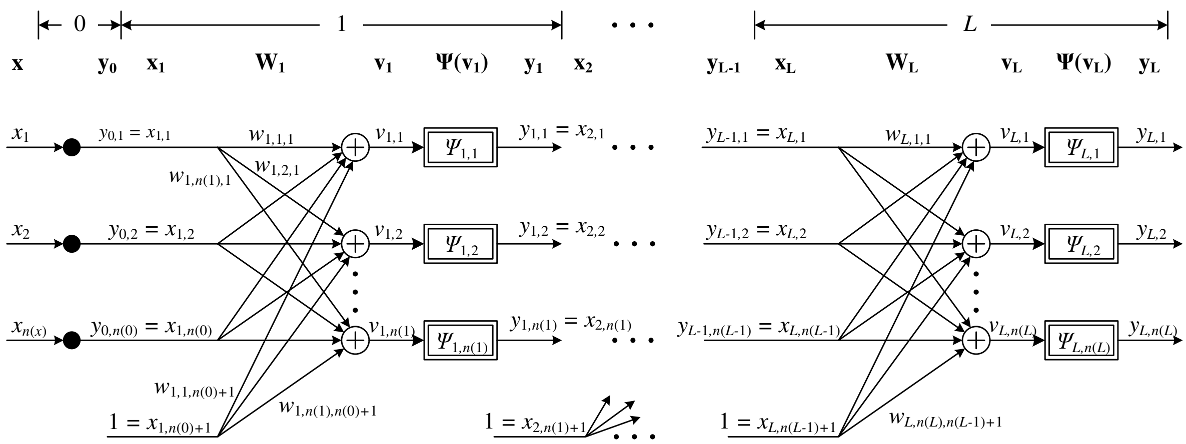


Figure 6.3: Schematic representation of a multilayer perceptron.

Mathematically, MLPs can be described by the following relations:

$$y_0 = x, \tag{6-4}$$

$$x_l = [y_{l-1}^T, 1]^T, \qquad 1 \le l \le L, \tag{6-5}$$
$$v_l = W_l \cdot x_l, \qquad 1 \le l \le L, \tag{6-6}$$
$$y_l = \psi(v_l), \qquad 1 \le l \le L, \tag{1-7}$$

where:

$\boldsymbol{x} = \left[x_1, x_2, \cdots, x_{n(x)}\right]^T$ is a vector of the network input od dimension $n(x)$;

$\boldsymbol{y_0} = \left[y_{0,1}, y_{0,2}, \cdots, y_{0,n(0)}\right]^T$ is an output vector of the 0-th layer of dimension $n(0)$;

$\boldsymbol{x_l} = \left[x_{l,1}, x_{l,2}, \cdots, x_{l,n(l-1)}, x_{l,n(l-1)+1}\right]^T$ is an input vector to the l-th layer (input $x_{l,n(l-1)+1} = 1$ multiplied by corresponding weight coefficient gives a scalar bias to neurons of the l-th layer);

$\boldsymbol{v_l} = \left[v_{l,1}, v_{l,2}, \cdots, v_{l,n(l)}\right]^T$ is an output vector of the confluence operation of the l-th layer;

$\boldsymbol{y_l} = \left[y_{l,1}, y_{l,2}, \cdots, y_{l,n(l)}\right]^T$ is an output vector of the l-th layer;

$\boldsymbol{W_l} = \begin{bmatrix} \boldsymbol{w_{l,1,1}} & \cdots & \boldsymbol{w_{l,1,j}} & \cdots & \boldsymbol{w_{l,1,n(l-1)}} & \boldsymbol{w_{l,1,n(l-1)+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{w_{l,i,1}} & \cdots & \boldsymbol{w_{l,i,j}} & \cdots & \boldsymbol{w_{l,i,n(l-1)}} & \boldsymbol{w_{l,i,n(l-1)+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{w_{l,n(l),1}} & \cdots & \boldsymbol{w_{l,n(l),j}} & \cdots & \boldsymbol{w_{l,n(l),n(l-1)}} & \boldsymbol{w_{l,n(l),n(l-1)+1}} \end{bmatrix}$ is a weight coefficient matrix of the synaptic connections of the l-th layer, dimension of which is $n(l) \times (n(l-1)+1)$;

$\boldsymbol{\Psi_l}(\boldsymbol{v_l}) = \left[\Psi_{l,1}(v_{l,1}), \Psi_{l,2}(v_{l,2}), \cdots, \Psi_{l,n(l)}(v_{l,n(l)})\right]^T$ is an activation function vector of the l-th layer (usually $\Psi_{l,1} = \Psi_{l,2} = \cdots \Psi_{l,n(l)}$).

The most commonly used activation function in the hidden layer is *tansig*, while in the output layer linear activation function is used. The activation gain is usually set to one.

The most important properties of the ANNs are universal approximation, learning and adaptation. ANN property of approximating any continuous function to an arbitrary accuracy is its most important property from the perspective of modelling, identification and control of nonlinear processes. Learning and adaptation properties enable that an adequately calibrated ANN has the generalization ability when the data that was not present in the calibrating data set comes to its input.

6.3.1.3 Neural network learning algorithms

Learning algorithm tunes network parameters in order to achieve its desired behaviour. In identification and control of nonlinear dynamic systems desired behaviour of a neural network is usually known, so error-based algorithms are used for the learning/calibrating procedure. Schematic representation of the error-based algorithm for neural network learning is shown in Figure 6.4.

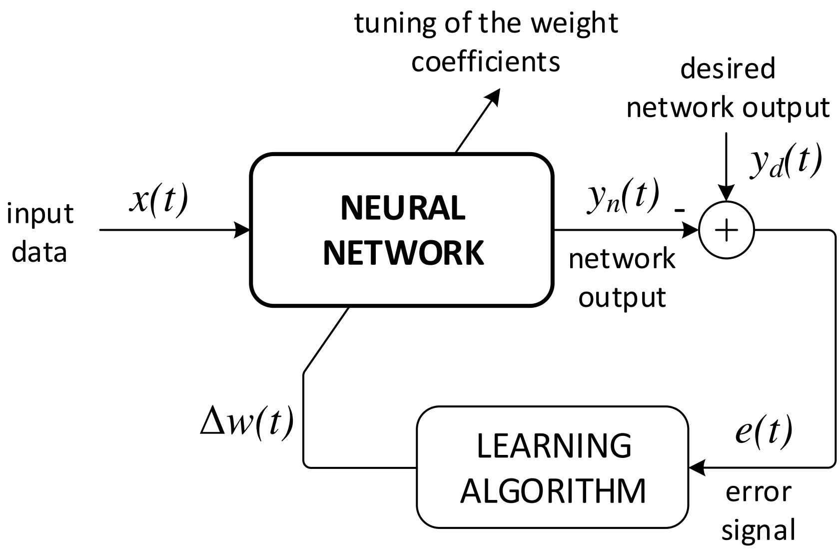


Figure 6.4: Schematic representation of the error-based algorithm for neural network learning.

Resulting neural network response y_n to the input data is compared to the external reference signal y_d, which represents desired network behaviour, generating error signal e based on which the learning algorithm changes synaptic weight coefficients of the network in order to improve its behaviour, i.e.: to decrease the error. As an error measure a criterion function $\Im(\Theta)$ is used and it can be any positive scalar function dependent on ANN parameters $\mathbf{\Theta}$. The most commonly used criterion function is defined as:

$$\Im(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{v=1}^{N} e(v,\boldsymbol{\Theta}) \cdot e^T(v,\boldsymbol{\Theta}) = \frac{1}{2}\sum_{v=1}^{N}\sum_{i=1}^{n(L)} e_i^2(v,\boldsymbol{\Theta}) = \frac{1}{2}e^{*T}(\boldsymbol{\Theta}) \cdot e^*(\boldsymbol{\Theta}), \tag{6-8}$$

where ν is a number of the measured sample, N is an overall number of measured samples, $e^*(\Theta)$ is the error vector of the whole measured data set, which is of dimension $N_e = N \cdot n(L)$.

There are two basic approaches in minimizing the criterion function $\Im(\Theta)$: non-recursive and recursive. According to the non-recursive approach, function $\Im(\Theta)$is minimized such that network parameter changes are determined based on the complete set of N measured samples. According to the recursive approach, function $\Im(\Theta)$ is minimized based on a local criterion function $\Im_\nu(\Theta)$, i.e. network parameters are changed after each measured sample.

Learning algorithm tunes network parameters until the criterion function reaches its minimum. Minimum of the criterion function $\Im(\Theta)$ can be formally defined by its Taylor series expansion in vicinity of the parameter vector Θ^0 for which the minimum is obtained, and by ignoring its third and higher order terms:

$$\Im(\boldsymbol{\Theta}) \cong \Im(\boldsymbol{\Theta^0}) = \nabla\Im^T(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta^0}} \cdot \Delta\boldsymbol{\Theta} + \frac{\mathbf{1}}{\mathbf{2}}\Delta\boldsymbol{\Theta^T} \cdot \boldsymbol{H}(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta^0}} \cdot \Delta\boldsymbol{\Theta}, \tag{6-9}$$

where:

$\Delta\boldsymbol{\Theta} = \boldsymbol{\Theta} - \boldsymbol{\Theta^0}$;

$\nabla\Im(\boldsymbol{\Theta})$ is a gradient vector of the criterion function:

$$\nabla\Im(\boldsymbol{\Theta}) = \left[\frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}}, \frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}}, \cdots, \frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}}\right]; \tag{6-10}$$

$\boldsymbol{H}(\boldsymbol{\Theta}) = \nabla^2\Im(\boldsymbol{\Theta})$ is a Hessian matrix of the criterion function:

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \begin{bmatrix} \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_1^2} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}\partial\theta_{\boldsymbol{n}(\theta)}} \\ \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}\partial\theta_{\mathbf{1}}} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_2^2} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}\partial\theta_{\boldsymbol{n}(\theta)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}\partial\theta_{\mathbf{1}}} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}^2} \end{bmatrix}. \tag{6-11}$$

For the criterion defined by (1-8), gradient vector and Hessian matrix become:

$$\nabla\Im(\boldsymbol{\Theta}) = \boldsymbol{J}^T(\boldsymbol{\Theta}) \cdot e^*(\boldsymbol{\Theta}), \tag{6-12}$$

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \nabla^2\Im(\boldsymbol{\Theta}) = \boldsymbol{J}^T(\boldsymbol{\Theta}) \cdot \boldsymbol{J}(\boldsymbol{\Theta}) + \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N_e}} e_i^*(\boldsymbol{\Theta})\nabla^2 e_i^*(\boldsymbol{\Theta}), \tag{6-13}$$

where $\boldsymbol{J}(\boldsymbol{\Theta})$ is a Jacobian matrix:

$$\boldsymbol{J}(\boldsymbol{\Theta}) = \begin{bmatrix} \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}} & \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}} \\ \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}} & \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}} & \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}} \end{bmatrix}. \tag{6-14}$$

Parameter vector $\Theta = \Theta^*$ will be the minimum argument of the function $\Im(\Theta)$ if the following conditions are fulfilled:

$$\nabla\Im(\Theta^*) = 0, \tag{6-15}$$

$$\Delta\Theta^{\mathrm{T}} \cdot \mathrm{H}(\Theta^*) \cdot \Delta\Theta > 0. \tag{6-16}$$

Therefore, tuning of the ANN parameters Θ is in fact a *nonlinear optimisation* problem where the criterion function $\Im(\Theta)$ is the objective function of the optimisation problem. *Gradient methods* are most commonly used nonlinear optimisation techniques. The main problem in applying gradient methods in ANN learning procedure is calculating a gradient vector of the criterion function over the network parameters. This problem has slowed research and application of ANNs for a while, but was successfully solved using the backpropagation algorithm. More details can be found in [34].

Tuning of the ANN parameter vector Θ is based on an iterative procedure:

$$\Theta(k+1) = \Theta(k) + \Delta\Theta(k) = \Theta(k) + \alpha(k)s_d(k), \tag{6-17}$$

where:

$s_d(k)$ is the minimum searching direction in the k-th iteration of the optimisation procedure (it is based on an information on a function $\mathfrak{I}(\Theta)$);

$\alpha(k)$ is the learning coefficient in the k-th iteration of the optimisation procedure (it determines the step size in the searching direction).

Depending on the procedure of determining the minimum searching direction $s_d(k)$, gradient methods can be divided into four groups:

- *Steepest descent methods*: $s_d(k) := -\nabla\mathfrak{I}(\Theta(k))$;
- *Conjugate gradient methods:* $s_d(k) := -\nabla\mathfrak{I}(\Theta(k)) + \beta(k) \cdot s_d(k-1)$, where $\beta(k)$ is a scalar parameter which ensures conjugacy;
- *Newton methods:* $s_d(k) := -[\nabla^{2\mathfrak{I}}(\Theta(k))]^{-1}\nabla\mathfrak{I}(\Theta(k))$;
- *Quasi-Newton methods* [39], [40]*:* $s_d(k) := -S(k)\nabla\mathfrak{I}(\Theta(k))$ where $S(k) \cong [\nabla^{2\mathfrak{I}}(\Theta(k))]^{-1}$.

ANN learning algorithms are named based on the corresponding nonlinear optimisation methods which are used: steepest descent algorithms, conjugate gradient algorithms etc.

6.3.2 Applying neural networks to system modelling

In the last 20 years neural network applications for predicting variables in ecological and technical systems have become a well-known procedure in a research community [41]. In the early phases of their applications, ANNs were considered as a novel approach in system modelling and the majority of published papers in that period were related to applying ANNs in different systems and exploring their advantages in relation to the well-known statistic approaches [42]. Many review papers in this research area did not only affirm a potential of using the ANNs in prediction systems, but they also noted an importance of developing a standard methodology in the model development procedure using ANNs. Clearly defined methodology is an important procedure for all modelling methods, but especially in ANN modelling because models are developed based on the available data and they are not explicitly based on the physical system that is modelled, therefore, a possibility of developing a model which is not very meaningful is increased.

Main steps in developing the prediction model using ANNs are shown in Figure 6.5. Flow of data and outcomes for each step are also shown. First step in model development process is a choice of appropriate model outputs (variables which are going to be predicted) and potential inputs. A choice of potential inputs is based on *a priori* knowledge on the modelled process and on data availability. Selected data have to be processed (scaled, filtered, lagged) for being in an appropriate form for the next model development steps.

A general ANN prediction model can be expressed in the following form:

$$\mathrm{Y} = \mathrm{f}(\mathrm{X}, \mathrm{W}) + \mathrm{e}, \qquad (6\text{-}18)$$

where Y is a model output vector, X is a model input vector, W is a model parameter vector (weight coefficients), f is a function which defines input-output relationship and e is a model error vector. Therefore, in model development process we need to define model inputs X, a functional relationship f defined by the ANN structure and ANN parameter vector W. Model inputs are determined using the so called Input Variable Selection (IVS) procedures which are described in subsection 6.3.3. Result of this step are model development data which are then divided in calibration and validation data sets. Calibration data are used in ANN learning algorithms for determining the optimal model parameters, while validation data are used for validating the calibrated model on the independent data set. If implicit regularization is used as a stopping criterion of the learning algorithm, calibration data are divided in training and testing data sets.

The main objective of the ANN learning process is to find the global minimum of the criterion function $\mathfrak{J}(\Theta)$. However, in modelling of dynamic systems which inherently contain noise, the global minimum of the criterion function is not the optimal solution because the obtained model does not assure the best generalization properties. In the first phase of the ANN learning process a decrease of the criterion function $\mathfrak{J}(\Theta)$ on the training data leads to a decrease of the criterion function $\mathfrak{J}^{t}(\Theta)$ on the testing data. However, after certain number of iterations, value of the criterion function $\mathfrak{J}^{t}(\Theta)$ starts increasing although $\mathfrak{J}(\Theta)$ is still decreasing and, therefore, further adjusting of the ANN parameters leads to a deterioration of its generalization properties. This problem can be solved by early stopping of the learning process when a criterion function value on the testing data starts increasing. This procedure is called an *implicit regularization*.

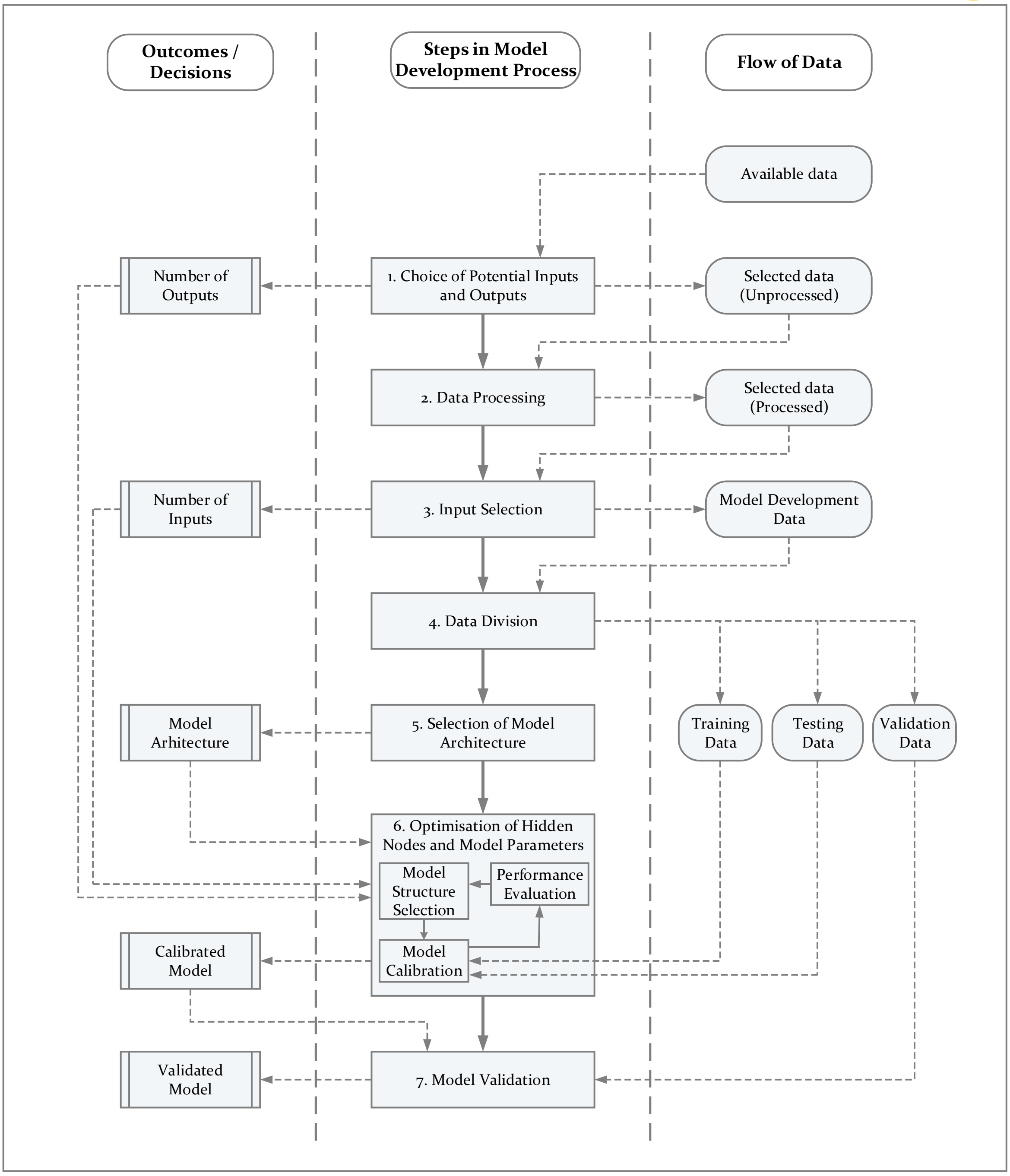


Figure 6.5: Main steps in the model development process using artificial neural networks [42].

Next step implies choosing a number of hidden layers and a number of neurons in each layer. The optimal structure of ANN is usually determined iteratively [42]. For a fixed structure, optimal parameters of the ANN are determined using learning procedure and they depend on the choice of learning algorithm and on initial ANN parameters. In general case criterion function is nonconvex and applying gradient methods can trap model parameter vector in a local minimum of the criterion function which is not the optimal solution. Therefore, a calibration process implies a number of calibration instances for different initial values of model parameters. ANN, defined by its structure and parameters, which has the minimal criterion function value on the calibration data is then validated on the validation data set. To ensure that a model development process results in the best possible model, it is required that training, testing and validation data sets have the same statistical properties [43].

6.3.3 Input variable selection procedure

One of the most important steps in modelling of complex systems is selection of the appropriate input variables. However, this step is usually not concerned to be of an extreme importance and most of the input variables are determined heuristically or based on *a priori* knowledge of the system which can result in including too many or too little input variables [44].

As a consequence of omitting one or more relevant input variables, model will not be able to describe the whole dynamics and phenomena of the system. Possibility of omitting relevant input variables is much greater for time series in which input candidates are not only different variables, but also their lagged values (unless dynamic ANNs are used) which significantly increases the number of potential input variables. Including too many input variables can be caused by poorly assessed relevance of an input variable or by existence of a redundancy among them, where some of the chosen variables contain some useful information, but are interdependent, so they contain a redundant information. This case leads to an increase in a number of local minima in the criterion function [42] and makes it harder to determine the optimal model parameters if a gradient method is used for ANN learning. On the other hand, with an increase of input variables, a number of model parameters is also increased which, as a consequence, leads to decreased speed and quality of the learning procedure. Furthermore, existence of an input variable which does not affect the output variable can lead to a deterioration of ANN generalization properties, i.e. the model will perform poorly on data that were not used during model calibration procedure.

These considerations indicate that the optimal ANN input variable set consist of the minimal set of variables which can describe the system behaviour well enough. A number of IVS algorithms were developed and they can be classified in *wrapper* and *filter* algorithms [45].

6.3.3.1 Wrapper algorithms

IVS using wrapper algorithms is based on developing a number of ANNs with different input vectors and the choice of an appropriate input set is determined based on performance of the corresponding ANN. The main drawback of this approach is that such a procedure can last very long because it is required to develop a large number of ANNs whereas the development of each implies an appropriate choice of the ANN structure and the learning algorithm. Additionally, appropriateness of the input variables chosen for a certain ANN architecture is not guaranteed for another architecture, so the application of the obtained input set is rather limited [44].

For d potential input variables, a number of possible input subsets is $2^d - 1$. Therefore, because of the large computational and time requirements, all possible input variable combinations are almost never tested. The most commonly used wrapper algorithms are forward selection, backward elimination and genetic algorithms [45].

Forward selection is an incremental procedure for forming the optimal input variable set in which a number of variables is incrementally increased. In the beginning, one out of d variables, for which an ANN with the best performance is obtained, is chosen. Then, the input set is enlarged by the next one out of $d-1$ remained variables for which an ANN performance is most improved. A procedure is repeated until adding a new variable to the input set does not lead to a significant improvement of the ANN performance.

Backward elimination is a procedure inverse to a forward selection, i.e. the input variable set is incrementally reduced. The procedure starts with an input set which contains all the potential input variables and the least relevant variables are progressively eliminated from the input set. This procedure is computationally more intensive than the forward selection because a large number of inputs requires learning an ANN with much larger number of parameters.

Genetic algorithms introduce stochastic elements in the procedure of selecting the optimal input variable set, increasing a possibility of finding the optimal set. Genetic algorithms show their advantages in relation to forward selection and backward elimination when the candidate set contains variables which only combined with other variables show their relevance to an output variable, while taken separately, do not have an excessive importance.

6.3.3.2 Filter algorithms

Unlike wrapper, filter algorithms use statistical measure of dependence between an output variable and potential inputs as a criterion for input selection. Uncoupling IVS procedure and model calibration does not only increase the modelling efficiency, but also extends possible applications of the obtained input set. However, efficiency of a filter algorithm is highly dependent on the statistical measure employed [44].

The most commonly used statistical measure of dependence is a linear correlation coefficient whose main drawback is that it only determines the linear dependence between variables which is particularly problematic in the model development using ANNs because they are used as an alternative to linear regression when a dependence between model inputs and output is nonlinear. Therefore, it is more meaningful to use an appropriate nonlinear statistical measure of dependence, like mutual information [42]. Unlike linear correlation coefficient, mutual information is also sensitive to dependences which are reflected in higher input-output correlation moments – mutual information is equal to zero if and only if two variables are strictly independent [46].

Apart from inputs relevance, IVS procedures should also consider redundancy of the input variables. In order to do so, a suitable algorithm based on partial mutual information (PMI) was developed and it is described in the next subsection.

6.3.3.3 Input variable selection algorithm based on partial mutual information

For a given continuous random variable X with a codomain $C(X)$, Shannon entropy is defined as:

$$\mathrm{H(X)} = -\int_{\mathrm{C(X)}} \mathrm{f(x)} \ln \mathrm{f(x)}\ \mathrm{dx}, \tag{6-19}$$

where x is an outcome of random variable X and $f(x)$ is its probability density function (pdf). Entropy is a term well-known in the information theory and it represents an informational description of random events and defines a measure of the information content, i.e. random variable uncertainty. Mutual information of two random variables, X and Y, is defined as:

$$\mathrm{I(X;Y)} = \int_{\mathrm{C(Y)}} \int_{\mathrm{C(X)}} \mathrm{f(x,y)} \ln\left(\frac{\mathrm{f(x,y)}}{\mathrm{f(x)f(y)}}\right) \mathrm{dxdy}, \qquad (1\text{-}20)$$

where $f(x)$ and $f(y)$ are pdfs of the variables X and Y, respectively, and $f(x,y)$ is a joint pdf of the random vector (X,Y). Mutual information can be expressed using entropies as:

$$\mathrm{I(X;Y) = H(X) + H(Y) - H(X,Y)}, \qquad (6\text{-}21)$$

where $H(X)$ and $H(Y)$ are entropies of the random variables X and Y, respectively, and $H(X,Y)$ is a joint entropy of the random vector (X,Y). Mutual information represents a reduction in uncertainty of the random variable Y knowing the random variable X and *vice versa*. Figure 6.6 depicts the dependency among mutual information and entropies of the random variables X and Y.

Here, $H(Y|X)$ is conditional entropy of Y given X, that is, the amount of uncertainty in the random variable Y when the value of X is known, and it is formally defined as:

$$\mathrm{H(Y|X)} = \int_{\mathrm{C(Y)}} \int_{\mathrm{C(X)}} \mathrm{f(x,y)} \ln\left(\frac{\mathrm{f(x)}}{\mathrm{f(x,y)}}\right) \mathrm{dxdy}. \qquad (6\text{-}22)$$

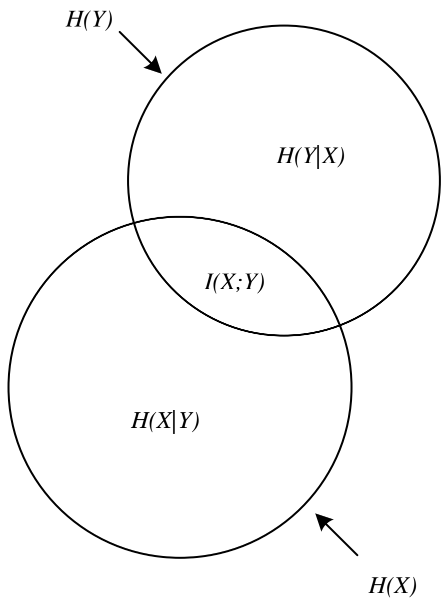


Figure 6.6: Venn diagram showing a relationship among mutual information and entropies of random variables X and Y.

Let us now consider the third random variable, Z. A part of a mutual information $I(Z;Y)$ which is not contained in X, $I(Z;Y|X)$, is called a partial mutual information and it is determined using the following expression:

$$\mathrm{I(Z;Y|X) = H(X,Z) + H(X,Y) - H(X) - H(X,Y,Z)}. \qquad (6\text{-}23)$$

Given X and the already reduced uncertainty $H(Y|X)$ shown in Figure 6.6, the PMI $I(Z;Y|X)$ is defined as the further reduction in uncertainty of the random variable Y that is gained by the additional mutual observation of the random variable Z.

Figure 6.7 depicts the dependence among PMI, individual and joint entropies of the random variables X, Y and Z. PMI is invariant under strictly monotonic transformations which makes it robust against possibly nonlinear distortions among random variables [47] and this is one of its most important advantages in relation to the linear correlation. However, a problem in determining a mutual information is that pdfs of the random variables have to be known. In practice, the real pdfs are not known and it is needed to estimate them. This topic is covered in the next subsection.

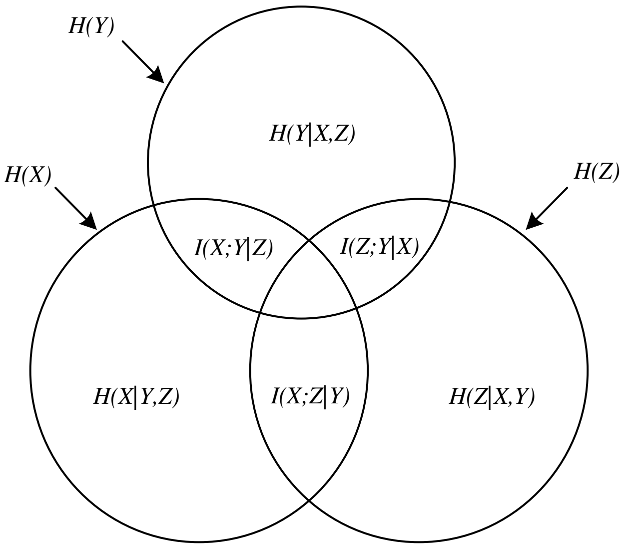


Figure 6.7: Venn diagram showing a relationship among partial mutual information and entropies of the random variables X, Y and Z.

PMI-based IVS algorithm is presented in [48]. Details of the algorithm are presented here:

Algorithm 1: Partial mutual information-based input variable selection

---

Input: output variable Y, potential input variables C
Result: chosen input variables X

Initialise $X \leftarrow \emptyset$
while $C \neq \emptyset$ **do**
  for each $c \in C$
    Estimate $I(c, Y|X)$
  Determine $c_s \in C$ that maximises $I(c, Y|X)$
  if *algorithm termination criterion is satisfied* **then**
    Stop running the algorithm
  Move c_s to X

In [44] a number of algorithm termination criteria are analysed. In this work a predefined number of the most relevant input variables was used as a termination criterion.

6.3.3.4 Estimating partial mutual information

Considering the expression (1-19) it can be seen that for estimating an entropy of the random variable, it is first required to determine its pdf which is estimated from the available historical data, i.e. from the considered random variable outcomes. There are two main approaches in estimating a pdf: *parametric* and *non-parameteric*.

The parametric approach assumes that data are drawn from a known parametric family of distributions, for example the normal distribution with mean μ and variance σ^2. Estimating the pdf then becomes a problem of estimating the parameters μ and σ^2. The non-parametric approach does not assume a form of the pdf, so non-parametric methods are usually much more robust and accurate than the parametric ones. A review of the most commonly used non-parametric estimation methods can be found in [49].

One of the most commonly used non-parametric pdf estimation methods is *kernel density estimation* and this method is proposed in [48] in the original version of Algorithm 1. However, this approach has some drawbacks -- apart from the fact that it is computationally very intensive and that it requires relatively large number of data samples for an accurate estimation, its behaviour is dependent on the kernel function parameters. This problem becomes even harder when a dimension of the random variable is increased [50]. Much more accurate and computationally less intensive pdf estimation method is *k-th nearest neighbour method*. The method in which an entropy of the random variable is directly determined is presented in [47] and it is described here.

Let us consider three continuous time series, $\{x_t\}$, $\{y_t\}$ and $\{z_t\}$, which represent the outcomes of random processes $\{X_t\}$, $\{Y_t\}$ and $\{Z_t\}$, respectively. For each vector $v_t \equiv \{x_t, y_t, z_t\}$, $t = 1,2,\cdots,N$ and a fixed integer k, $1 \le k \ll N$, a distance $\varepsilon_k(t)$ to its k-th neighbour is defined. It means that a set $\{v_{t^*}\}$, where $t^* = 1,2,\cdots,N$, $t^* \ne t$, contains $k-1$ vectors with distances from v_t less than $\varepsilon_k(t)$ and $N-k-1$ vectors with the distance greater than $\varepsilon_k(t)$.

Therefore, for each t distance of v_t to each element of $\{v_{t^*}\}$ is determined:

$$\varepsilon(\mathrm{t}) = \{||\mathrm{v}_{\mathrm{t}^*} - \mathrm{v}_\mathrm{t}||\}. \tag{6-24}$$

This set is then sorted and distance $\varepsilon_k(t)$ is determined by selecting the k-th element of the sorted set. The distance is determined using max norm, i.e. $||\cdot|| = \max\{||\cdot||_x, ||\cdot||_y, ||\cdot||_z\}$, where $||\cdot||_x$, $||\cdot||_y$ and $||\cdot||_z$ can be any norm, but this algorithm suggests using max norm as well. Let us now define a vector $w_t \equiv \{x_t, z_t\}$, $t = 1,2,\cdots,N$.

For each t a number of vectors in $\{w_{t^*}\}$ with distances strictly less than $\varepsilon_k(t)$ is determined:

$$\mathrm{N}_{\mathrm{xz}}(\mathrm{t}) = \#\{\mathrm{t}^* \ne \mathrm{t};\ ||\mathrm{w}_{\mathrm{t}^*} - \mathrm{w}_\mathrm{t}|| < \varepsilon_\mathrm{k}(\mathrm{t})\}. \tag{6-25}$$

where # denotes a number of elements in the set. In a similar way $N_{xy}(t)$ and $N_x(t)$ are defined, for which w_t is defined using vectors $\{x_t, y_t\}$ and $\{x_t\}$, respectively. PMI is estimated using the following expression:

$$\hat{I}(Z;Y|X) = \frac{1}{N}\sum_{t=1}^{N}\left[h_{N_{xz}(t)} + h_{N_{xy}(t)} - h_{N_x(t)}\right] - h_{k-1}, \tag{6-26}$$

where h_n is the n-th negative harmonic number defined as $h_n = -\sum_{i=1}^{n} i^{-1}$ [47].

The k-th nearest neighbour method is computationally much faster than kernel methods are and, regardless of a number of considered variables dimension, it requires defining only one scalar parameter, k.

Here, we analyse the properties of the PMI estimator in case of the normal distribution for which PMI can be determined analytically, as shown in [47]. Multivariate normal distribution of the random vector $X \in R^n$ with mean $a \in R^n$ and covariance matrix $R \in R^{n \times n}$ is defined by its pdf:

$$\mathrm{f}(X) = \frac{1}{(2\pi)^{n/2}\sqrt{R}} exp\left(\frac{1}{2}(x-a)^T R^{-1}(x-a)\right), \tag{6-27}$$

and it is denoted as $X \sim \mathcal{N}_n(a, R)$ where $|R|$ denotes a determinant of the covariance matrix R. For n-dimensional normal distribution $\mathcal{N}_n(a, R)$ entropy is determined using the following expression:

$$H(X) = \frac{n}{2}(1 + ln2\pi) + \frac{1}{2}ln|R|. \tag{6-28}$$

6.3.4 Structure of the prediction model

This section analyses an identification procedure for prediction models with time horizon of 12-36 hours. One of the main issues in developing such a multiple-output system is how to assess its performance, i.e.: how to define a criterion which will tell us if one model is better than the other. The response is trivial if each output of one model outperforms the corresponding output of the other model, but generally it is not the case. The simplest approach is to define a local criterion function for each output and a global criterion function could be e.g.: a sum of the local criterion functions. The first drawback of this approach is that we are usually more concerned about sooner prediction hours than about hours at the end of a prediction horizon, so we do not want to give the same weight to each local criterion function. An alternative is to use weighted sum of the local criterion functions as a global criterion, but a question of how to choose these weights remains open. The second drawback is that such a model has the same input vector which is used for describing input-output relationship for each output, which generally does not have to be the optimal choice. Certainly, developing a separate model for each output can at least perform as well as one model with multiple outputs. The first advantage of this approach is that defining a criterion function is trivial because for single-output models the local criterion corresponds to the global criterion. The second advantage is that such an approach does not necessarily imply a unique input vector for each model. The main drawback of this approach is that the whole developing process, including IVS, defining the optimal model structure and model calibration has to be carried out multiple times which can be computationally very intensive for a large prediction horizon. The concept of this approach is depicted in Figure 6.8.

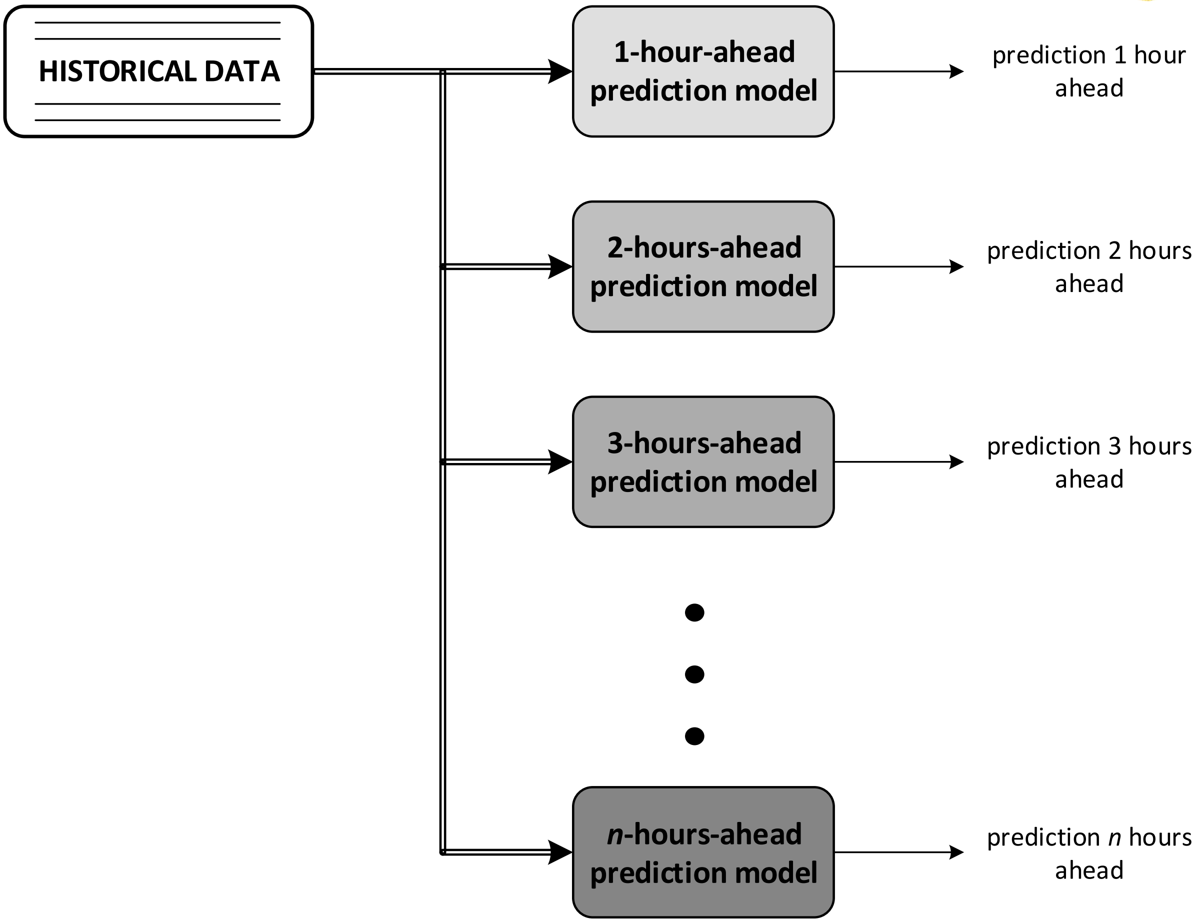


Figure 6.8: A static approach of the prediction system which uses a separate model for each system output.

Unlike the above-mentioned *static* approaches, the third approach uses the fact that the prediction system is considered as dynamic, i.e.: its output depends on past outputs. This *dynamic* approach is depicted in Figure 6.9. The main idea behind this approach is that the model does not have to use all the actual data, but also the provisional data, e.g.: output of the 1-hour-ahead model is a prediction for one hour ahead and this value can be used by the same model for predicting for two hours ahead. Analogously, this procedure can be repeated for obtaining the prediction for k hours ahead. It is expected that this approach will be less accurate than the one shown in Figure 6.8 because in this case a prediction error of the model is accumulated over the whole prediction horizon. However, if the performance of such an approach is not much worse than the one of the static approach, from the computational point of view, applying dynamic approach is much more efficient and contains significantly less parameters. Additionally, in some applications a larger prediction horizon may be required. Extension of the existing prediction model to a larger prediction horizon using dynamic approach is trivial; for the static approach this is not the case. Therefore, a dynamic approach is chosen for the prediction system.

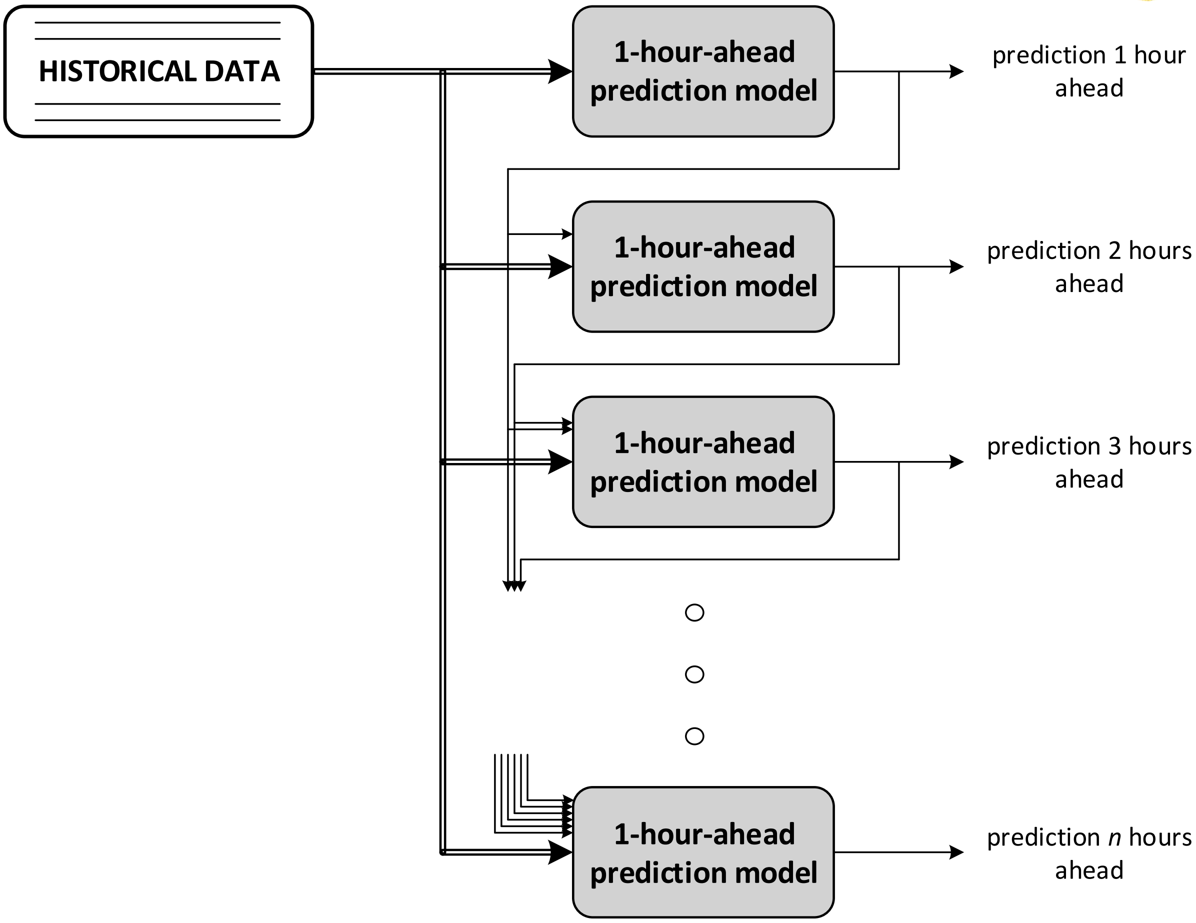


Figure 6.9: A dynamic approach of the prediction system which uses a single model for estimating system outputs for the whole prediction horizon.

6.3.4.1 Adaptive structure of the prediction system

It is often the case that historical data used for calibrating the prediction model do not cover the complete set of possible input-output vectors or that predicted variable values that occurred in past differs from values for the coming period due to factors which were not considered or did not have a significant impact on the variable during model calibration process. Occurrence of these factors can lead to poor predicting abilities of the existing prediction model. Therefore, for robust operation of the prediction system the model should be able to adapt to possible changes in the system.

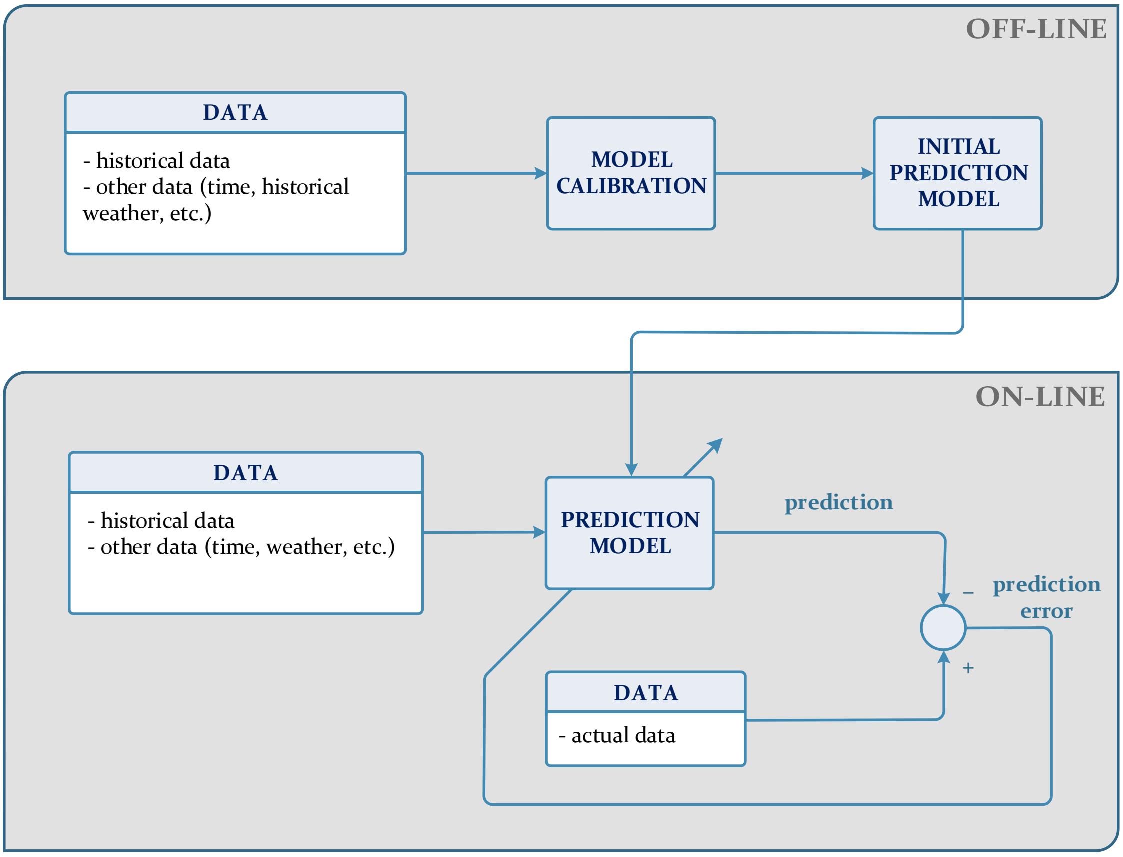


Figure 6.10: Adaptive module structure / a principle overview.

Modified structure of the prediction module is shown in Figure 6.10. The system is composed of two parts: *off-line* and *on-line*. In the off-line part historical data are used for obtaining the initial prediction model and this procedure is described in Subsection 6.3.2. The on-line part of the module uses the initial model developed in the off-line part in order to generate predictions. When the data are available, they are compared to the corresponding predictions which results in the prediction error for the certain time instant. Model parameters are then tuned such that the prediction error is decreased. The presented procedure of using the feedback information on prediction accuracy for model parameters tuning introduces an adaptation ability to module.

6.3.4.2 Possible approaches to the on-line tuning of model parameters

Most real systems are time-variant. In order to track changes in the system, its model parameters should be continuously estimated. The on-line part of the prediction system, mentioned in the previous section, is the tool for continuous tuning of the model parameters such that the model tracks the actual predicted variable evolution as accurately as possible.

Artificial neural network (ANN) is a flexible model structure that can be easily and systematically calibrated and adapted. There is a large number of methods suggested in literature for the so called *recursive* neural network learning. Some of them are based on the recursive approximation of typical gradient methods [34], [51]. On the other hand, some recursive methods are based on the methodology for dynamic system state estimation [52]-[55]. These methods are based on the state-space representation of the ANN model [56]:

$$\mathrm{w_{k+1} = w_k + r_k,} \tag{6-29}$$
$$\mathrm{d_k = G(x_k, w_k) + e_k,} \tag{6-30}$$

where G is a function which defines the input-output mapping and is determined by the ANN structure, x_k is an input vector, w_k is a vector of ANN parameters and e_k is an error vector. In (1-29) a vector of parameters w_k corresponds to a stationary process with identity state matrix, driven by process noise r_k. ANN model written in this form enables using extended Kalman filter (EKF) or unscented Kalman filter (UKF) for the ANN parameter estimation. However, the ANN models with relatively large number of inputs and nodes in the hidden layer result in a large number of parameters, and applying EKF or UKF becomes intractable due to numerical stability issues [57]. On the other hand, recursive gradient methods for ANN learning are quite robust and their application is not limited to ANNs with a small number of parameters. Therefore, this approach in recursive ANN learning is analysed hereinafter.

6.3.4.3 Applying the on-line tuning procedure in normal operation

We use the prediction model developed within subsection 6.3.4 as an initial prediction model for the on-line part of prediction system (see Figure 6.10). Gradient descent method with momentum term is used for the recursive ANN learning. ANN parameters Θ are updated based on the following relation:

$$\Delta\Theta(\mathrm{k}) = -\alpha\nabla\Im_{\mathrm{v}}\big(\Theta(\mathrm{k})\big) + \gamma_{\mathrm{m}}\Delta\Theta(\mathrm{k}-1), \tag{6-31}$$

where $\Delta\Theta(k) = \Theta(k+1) - \Theta(k)$, α is the learning coefficient, $\nabla\Im_{\mathrm{v}}\big(\Theta(k)\big)$ is the gradient of local criterion function on the corresponding data set and γ_m is the nonnegative momentum term which speeds up the learning convergence while attenuating the parasitic oscillations [34]. If the parameter vector Θ is to be updated using more than one data sample, we consider two different learning styles: (i) *incremental learning* in which the model parameters are updated consecutively after each data sample is presented to the model; and (ii) *batch learning* in which the parameters are updated once after all the data samples are presented. The recursive ANN learning is performed using MATLAB® Neural Network Toolbox [58].

The on-line tuning parameters, learning coefficient α and momentum term γ_m can be determined based on the initial set of data that were used for obtaining the initial model. However, those data might not contain an evident variation in predicted variable, thus no significant difference in the performance of off-line and on-line model would be observed. Therefore, on-line tuning parameters can be determined based on the performance of on-line prediction model on the modified testing data –e.g. a linear trend is added to the original data such that predicted variable mean increases by 50% of the initial mean per month.

6.3.4.4 Concept of conditional adaptation (outliers handling)

In addition to the normal operation, another possible scenarios which affect the prediction system can occur. In the normal operation scenario we assumed that data do not contain potentially irregular or corrupted data samples (referred to as *outliers*). However, it is often the case that data on actual data are corrupted -- using these data samples within the on-line tuning procedure could cause an undesirable model behaviour. Instantaneous change in mean may be a result of many different external factors that influence the predicted variable, but it may also be caused by a meter problem – in the latter case data are characterised as corrupted.

The basic idea in avoiding the on-line tuning procedure using corrupted data is by marking those data, i.e. if a data sample is suspected to be an outlier, it is marked and that data sample will not be used in the on-line tuning procedure. In order to recognize an outlier occurrence, min/max values of the model inputs are used as boundaries for filtering the outliers.

7 Comfort setpoint prediction submodule (Z.PE.7)

Submodule for prediction of the comfort setpoint in the zone.

7.1 Submodule inputs

Table 7.1: Required inputs for comfort setpoint prediction submodule.

Variable name	Variable annotation	Variable description
Comfort setpoint in the zone	SP	Profile of the comfort setpoints selected in the past needed for off-line model tuning; recent values needed for on-line operation
Zone control mode	CM	Integer showing which operation mode of the heating/cooling system is selected in the zone (off, auto, fixed fan speed/valve openness).
Building HVAC system operation schedule	SC	Data showing when is the HVAC system for heating/cooling turned on/off.
Possible extension: Connection with the company business data.		Connection point between the EMS and the business information system of a company (travel orders, vacations, sick leaves, different known occupancy schedules for meetings/lectures...)

7.2 Submodule outputs

Table 7.2: Outputs of the comfort setpoint prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{SP}	Needed for on-line operation of the submodule.
Predicted comfort setpoint evolution per zone (for on-line operation of the submodule)	$(SP)_N$	Needed for the MPC module on the zones level.

7.3 Methodology

Methodology of the comfort setpoint prediction submodule is based on logical processing of the current comfort setpoint set in the zone, current control mode of the zone and the main assumption that the current setpoint doesn’t change over time, meaning that the users within zones know exactly what is the desired setpoint and it remains fixed throughout the day.

Depending on the selected current control mode off the heating/cooling system within the zone, different cases are considered:

- control mode set to **off** (no heating/cooling), or a **fixed FCU fan speed**, **radiator valve openness**: since the heating/cooling element in the zone is currently outside the 3Smart control system historical setpoint setup for the same day of the previous week is copied as the predicted setpoint for the entire prediction horizon. Historical setpoint setup may include again off times as well as times when the users provided setpoint.
- control mode set to **auto** (setpoint provided): under the assumption that the users don't change their setpoint, current setpoint is predicted on the entire prediction horizon with the exception of time intervals when the HVAC system in the building is out of operation (e.g. during the night, during the weekend or national holidays etc.). HVAC system operation schedule is obtained from the database or provided manually by the pilot hosts.

Zone setpoint prediction submodule can be further extended with the use of company business data such as travel orders, vacations, sick leaves, different known occupancy schedules for meetings/lectures etc. which are then intersected with the current setpoint prediction meaning that if a zone has no users present (according to the obtained company business data) no setpoint is predicted, thus giving the ability to the zone level MPC module to further manipulate the zone temperature within the building protect temperature limits and offer additional savings possibilities.

8 Zone thermal energy consumption prediction submodule (Z.PE.8)

Submodule for prediction of the heating/cooling energy consumption in the zone.

8.1 Submodule inputs

Table 8.1: Required inputs for zone thermal energy consumption prediction submodule.

Variable name	Variable annotation	Variable description
Thermal energy consumption in zone	E_t	Profile of the thermal energy consumption in zone needed for off-line model tuning; recent estimates of the thermal energy consumption needed for on-line operation
Weather measurements	$T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, global horizontal and tilted global irradiance.
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and

		day of the year. Calculated from current and historical datetimes.

8.1.1 Solar irradiance data

EON pilot sites provides measurements of global horizontal and tilted global irradiations which are then used as submodule inputs. Since measured and forecasted irradiances are different, during submodule operation, forecasted direct and diffuse irradiance, solar angles (obtained through the use of Pysolar python library), geographical pilot site data and current datetime, are used for calculation of global horizontal and tilted global irradiances thus matching the measured and forecasted irradiance variables.

8.2 Submodule outputs

Table 8.2: Outputs of the zone thermal energy consumption prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{et}	Needed for on-line operation of the submodule.
Predicted thermal energy consumption evolution per zone (for on-line operation of the submodule)	$(E_t)_N$	Needed for the MPC module on the zones level.

8.3 Methodology

Methodology of the zone thermal energy consumption prediction submodule is based on Artifical Neural Networks (ANN), and therefor is identical to the methodology of the heat disturbance prediction module which is described in detail in Subsection 6.3.

9 Zone temperature prediction submodule (Z.PE.9)

Submodule for prediction of the temperature in the zone.

9.1 Submodule inputs

Table 9.1: Required inputs for zone temperature prediction submodule.

Variable name	Variable annotation	Variable description
Temperature in zone	T_z	Profile of the temperature in zone needed for off-line model tuning; recent measurements of zone temperature needed for on-line operation
Weather measurements	$T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables:

		temperature, global horizontal and tilted global irradiance.
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I^{\mathrm{n}}_{\mathrm{dir}})_{\mathrm{N}}, (I^{\mathrm{h}}_{\mathrm{diff}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

9.1.1 Solar irradiance data

EON pilot sites provides measurements of global horizontal and tilted global irradiations which are then used as submodule inputs. Since measured and forecasted irradiances are different, during submodule operation, forecasted direct and diffuse irradiance, solar angles (obtained through the use of Pysolar python library), geographical pilot site data and current datetime, are used for calculation of global horizontal and tilted global irradiances thus matching the measured and forecasted irradiance variables.

9.2 Submodule outputs

Table 9.2: Outputs of the temperature prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{tz}	Needed for on-line operation of the submodule.
Predicted temperature evolution per zone (for on-line operation of the submodule)	$(T_z)_N$	Needed for the MPC module on the zones level.

9.3 Methodology

Methodology of the zone temperature prediction submodule is based on Artifical Neural Networks (ANN), and therefor is identical to the methodology of the heat disturbance prediction module which is described in detail in Subsection 6.3.

Bibliography

[1] E.A. Wan and R.V. D. Merwe, “The Unscented Kalman Filter,” in Kalman Filtering and Neural Networks, pp. 221–280, Wiley, 2001.

[2] Y. Ma, Model Predictive Control for Energy Efficient Buildings. Ph.D. dissertation, University of California, Berkeley, 2014.

[3] P. Radecki and B. Hencey, “Online building thermal parameter estimation via unscented kalman filtering,” in American Control Conference (ACC), Montreal, CA, pp. 3056 –3062, june 2012.

[4] M. Maasoumy, B. Moridian, M. Razmara, M. Shahbakhti, and A. Sangiovanni-Vincentelli, “Online simultaneous state estimation and parameter adaptation for building predictive control,” ASME Conference Proceedings, vol. 2013, 2013.

[5] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” Proceedings of the IEEE, vol. 92, no. 3, pp. 401–422, 2004.

[6] R. V. D. Merwe and E. A. Wan, “The square-root unscented kalman filter for state and parameter-estimation,” in in International Conference on Acoustics, Speech, and Signal Processing, pp. 3461–3464, 2001.

[7] R. Kandepu, B. Foss, and L. Imsland, “Applying the unscented kalman filter for nonlinear state estimation,” Journal of Process Control, vol. 18, no. 7-8, pp. 753–768, 2008.

[8] S. Kolås, B. A. Foss, and T. S. Schei, “Constrained nonlinear state estimation based on the ukf approach,” Computers and Chemical Engineering, vol. 33, no. 8, pp. 1386–1401, 2009.

[9] S. Murtuza Baker, C. H. Poskar, F. Schreiber, and B. H. Junker, “An improved constraint filtering technique for inferring hidden states and parameters of a biological model,” Bioinformatics, vol. 29, no. 8, pp. 1052–1059, 2013. 49

[10] B. O. S. Teixeira, L. A. B. Tôrres, L. A. Aguirre, and D. S. Bernstein, “On unscented kalman filtering with state interval constraints,” Journal of Process Control, vol. 20, no. 1, pp. 45–57, 2010.

[11] R. E. Kalman, “An approach to linear filtering and prediction problems,” Journal of Basic Engineering, vol. 81, pp. 35–45, 1960.

[12] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” Proceedings of the IEEE, vol. 92, no. 3, pp. 401–422, 2004.

[13] EQUA Simulation AB, IDA Indoor Climate and Energy. Sweden.

[14] N. Kristensen, H. Madsen, and S. Jrgensen, “Parameter estimation in stochastic grey-box models,” Automatica, vol. 40, no. 2, pp. 225–237, 2004.

[15] International Energy Agency (IEA), World energy outlook 2016, OECD Publishing, 2016 .

[16] Europe’s buildings under the microscope. Buildings Performance Institute Europe, 2011.

[17] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, M. Morari, ”Use of Model Predictive Control and Weather Forecasts for Energy Efficient Building Climate Control”, Energy and Buildings, vol. 45, pp. 15–27, 2012.

[18] D. Sturzenegger, D. Gyalistras, M. Morari and R. S. Smith, ”Model Predictive Climate Control of a Swiss Office Building: Implementation, Results, and Cost-Benefit Analysis”, IEEE Transactions on Control Systems Technology, vol. 24, no. 1, pp. 1–12, 2016.

[19] L. Ratna Raju and T. Nandi, “Effective NTU of a counterflow heat exchanger with unbalanced flow and longitudinal heat conduction through fluid separating and outer walls,” Applied Thermal Engineering, vol. 112, pp. 1172–1177, 2017.

[20] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” Automatica, vol. 36, no. 6, pp. 789–814, 2000.

[21] S. Wongwises and Y. Chokeman, “Effect of fin pitch and number of tube rows on the air side performance of herringbone wavy fin and tube heat exchangers,” Energy Conversion and Management, vol. 46, no. 13-14, pp. 2216–2231, 2005.

[22] K. Yigit and H. Ertunc, “Prediction of the air temperature and humidity at the outlet of a cooling coil using neural networks,” International Communications in Heat and Mass Transfer, vol. 33, no. 7, pp. 898–907, 2006.

[23] G.-Y. Jin, W.-J. Cai, Y.-W. Wang, and Y. Yao, “A simple dynamic model of cooling coil unit,” Energy Conversion and Management, vol. 47, no. 1516, pp. 2659 – 2672, 2006.

[24] Y.-W. Wang, W.-J. Cai, Y.-C. Soh, S.-J. Li, L. Lu, and L. Xie, “A simplified modeling of cooling coils for control and optimization of HVAC systems,” Energy Conversion and Management, vol. 45, no. 1819, pp. 2915 – 2930, 2004.

[25] José Fernández-Seara, Rubén Diz, Francisco J. Uhía, J. Alberto Dopazo, Experimental analysis on pressure drop and heat transfer of a terminal fan-coil unit with ice slurry as cooling medium, In International Journal of Refrigeration, Volume 33, Issue 6, 2010, Pages 1095-1104, ISSN 0140-7007

[26] Yigit, K.S., Ertunc, H.M. Prediction of the air temperature and humidity at the outlet of a cooling coil using neural networks (2006) International Communications in Heat and Mass Transfer, 33 (7), pp. 898-907.

[27] M. Castilla, J. D. Álvarez, J. E. Normey-Rico, and F. Rodríguez, “Thermal comfort control using a non-linear MPC strategy: A real case of study in a bioclimatic building,” J. Process Control, vol. 24, no. 6, pp. 703–713, 2014.

[28] J.Široḱy, F. Oldewurtel, J. Cigler, S. Pŕıvara, ”Experimental analysis of model predictive control for an energy efficient building heating system”, Applied Energy, vol. 88, no. 9, pp. 3079-3087, 2011

[29] MATLAB, version 8.5.0 (R2015a). The MathWorks Inc., 2015.

[30] Jing, B., & Chen, M. (2015). Advanced nonintrusive load monitor for equipment diagnostics on fan coil unit. Paper presented at the IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, 339-344. doi:10.1109/IECON.2015.7392122

[31] A. B. Erdogmus, (2011), Simulation of the heater test room defined by EN 442 standard and virtual testing of different type of heaters, Graduate School of Engineering and Sciences of İzmir Institute of Technology

[32] Martinez, I. (1992.) Termodinámica básica y aplicada, Ed. Dossat

[33] Johansson, P-O. (2010.), Heat Output from space heating radiator with add-on-fan blowers, Lund University, Department of Energy Sciences, Faculty of Engineering, Division of Efficient Energy Systems

[34] Trüschell, A. (1999), Värmesystem med luftvärmare och radiatorer, En analys av funktion och prestanda, Licentiate Thesis, Chalmers

[35] A. Martinčević, F. Rukavina, V. Lešić, and M. Vašak, “Comfort control in buildings with adherence to the required thermal energy input in zones,” in IEEE International Symposium on Industrial Electronics, 2017, pp. 1477–1482.

[36] I. Petrović, M. Baotić, and N. Perić, Inteligentni sustavi upravljanja: Neuronske mreže, evolucijski i genetički algoritmi. University in Zagreb Faculty of Electrical Engineering and Computing, 2011.

[37] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133, dec 1943.

[38] W. Pitts andW. S. McCulloch, “How we know universals: The perception of auditory and visual forms,” Bulletin of Mathematical Biophysics, vol. 9, pp. 127–147, 1947.

[39] K. Levenberg, “A method for the solution of certain problems in least squares,” Quart. Applied Math., vol. 2, pp. 164–168, 1944.

[40] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” SIAM Journal on Applied Mathematics, vol. 11, no. 2, pp. 431–441, 1963.

[41] H. R. Maier and G. C. Dandy, “Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications,” Environmental Modelling & Software, vol. 15, no. 1, pp. 101 – 124, 2000.

[42] H. R. Maier, A. Jain, G. C. Dandy, and K. Sudheer, “Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions,” Environmental Modelling & Software, vol. 25, no. 8, pp. 891 – 909, 2010.

[43] R. May, H. Maier, and G. Dandy, “Data splitting for artificial neural networks using SOM based stratified sampling,” Neural Networks, vol. 23, no. 2, pp. 283 – 294, 2010.

[44] R. J. May, H. R. Maier, G. C. Dandy, and T. G. Fernando, “Non-linear variable selection for artificial neural networks using partial mutual information,” Environmental Modelling & Software, vol. 23, no. 10–11, pp. 1312 – 1326, 2008.

[45] K. Suzuki, Artificial Neural Networks - Methodological Advances and Biomedical Applications. InTech, 2011.

[46] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," Phys. Rev. E, vol. 69, p. 066138, Jun 2004.

[47] S. Frenzel and B. Pompe, "Partial mutual information for coupling analysis of multivariate time series," Phys. Rev. Lett., vol. 99, p. 204101, Nov 2007.

[48] A. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — A strategy for system predictor identification," Journal of Hydrology, vol. 239, no. 1–4, pp. 232 – 239, 2000.

[49] B.W. Silverman and P. J. Green, Density Estimation for Statistics and Data Analysis. London: Chapman and Hall, 1986.

[50] T. Duong and M. L. Hazelton, "Cross-validation bandwidth matrices for multivariate kernel density estimation," Scandinavian Journal of Statistics, vol. 32, no. 3, pp. 485–506, 2005.

[51] L. Ngia and J. Sjoberg, "Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm," IEEE Transactions on Signal Processing, vol. 48, no. 7, pp. 1915–1927, 2000.

[52] I. Rivals and L. Personnaz, "A recursive algorithm based on the extended Kalman filter for the training of feedforward neural models," Neurocomputing, vol. 20, no. 1–3, pp. 279–294, 1998.

[53] X.Wu and Y.Wang, "Extended and Unscented Kalman filtering based feedforward neural networks for time series prediction," Applied Mathematical Modelling, vol. 36, no. 3, pp. 1123–1131, 2012.

[54] R. Zhan and J.Wan, "Neural network-aided adaptive unscented Kalman filter for nonlinear state estimation," IEEE Signal Processing Letters, vol. 13, no. 7, pp. 445–448, 2006.

[55] R. Van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 6, pp. 3461–3464, 2001.

[56] S. Haykin, Kalman Filtering and Neural Networks. Adaptive and Learning Systems for Signal Processing, Communications and Control Series,Wiley, 2004.

[57] L. Padilla and C. Rowley, "An adaptive-covariance-rank algorithm for the unscented Kalman filter," in Proceedings of the IEEE Conference on Decision and Control, pp. 1324–1329, 2010.

[58] MATLAB, version 8.1.0 (R2013a). Natick, Massachusetts: The MathWorks Inc., 2013.

[59] Kraus, J. Welty, and A. Aziz, "Flow in pipes and pipe networks," in Introduction to Thermal and Fluid Engineering. New York: Taylor & Francis, 2011, ch. 17, pp. 523–562.

[60] E. S. Menon and P. S. Menon, "Pressure loss through piping systems," in Working Guide to Pumps and Pumping Stations. Boston: Gulf Professional Publishing, 2010, pp. 69–112.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Model predictive control module for zones management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER)
Authors (Partners)	Anita Martinčević, Vinko Lešić, Mario Vašak (UNIZGFER)
Contact person	Anita Martinčević (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of model predictive control module on the level of building zones for hierarchical management of building subsystems. The details are provided in the annexed document.
Keyword List	Thermal Energy Input in Zones, Model Predictive Control

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.4.1	Mario Vašak (UNIZGFER)
v0.2	15 January 2019	Updated version	Mario Vaašak (UNIZGFER)
v0.3	20 May 2019	Updated version based on feedback from operation on pilots	Anita Martinčević (UNIZGFER)
v0.9	7 June 2019	Updated version	Mario Vašak (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of modules.

The modules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on zone level model predictive control.

The zone level model predictive control module is presented via an interfacing table that explains what data are used by it as inputs and what are the final output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following model predictive control module is designed, commissioned and tested on the zone level:

Z.MPC.1 – module for model predictive control that decides on the thermal energy inputs into each of the building zones (tested in UNIZGFER, HEP, IDRIJA buildings, STREM school, STREM retirement and care centre and EPHZHB buildings within 3Smart).

In the following chapter the module is presented with its interface tables showing which data it uses as inputs and which data it provide as outputs to be at the disposal to other modules and submodules. Detailed explanations of algorithms behind it are provided in the previously delivered 3Smart document D4.4.1 which is refreshed through feedback obtained via pilots operation and is provided as Annex 1 to this document.

Source and sink for the data used by the module is a properly structured 3Smart database. Its structure in the part concerned by the module is provided in Annex 2.

2 Z.MPC.1 module

Z.MPC.1 module is used for model predictive control that decides on the thermal energy inputs into each of the building zones. Within 3Smart it is tested in UNIZGFER, HEP, IDRIJA buildings, STREM school, STREM retirement and care centre and EPHZHB buildings.

The module is executed in on-line and off-line mode – in on-line mode it computes the planned optimal operation along the prediction horizon in coordination with the other two MPC modules (central HVAC and microgrid levels), in off-line mode it is in charge to compute the optimal daily operation on the zone level in coordination with other two MPC modules – a common outcome from the MPC modules in off-line operation is also the optimal flexibility bid towards the grid.

The module interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Required inputs for the model predictive control module for zones comfort control

Variable name	Notation	Description
Estimates of the current states of the simplified building thermal dynamics model	x_0	Non-measured states of the simplified model are estimated.
Predicted profile of comfort setpoints	$\mathrm{SP}_{\mathrm{pred}}$	Temperature and other comfort setpoints that are predicted based on models tuned on historical data
Predicted profile of actuation in rooms/zones that are outside the 3Smart EMS control	manual actuation commands for the room heating/cooling elements	Users may be in control to set a manual command for the room heating/cooling elements or some rooms/zones may be permanently outside the 3Smart EMS control, and the

		EMS should be able to respect it; currently just extrapolation of the current manual selection will be used for predictions
Predicted profile of cumulated heating/cooling disturbances in zones	$E_{\mathrm{d}}^{\mathrm{pred}}$	Disturbances predicted based on models tuned on historical data
Predicted profile of outdoor temperature	T_{opred}	Predicted profile of outdoor temperature received from the weather forecast service
Predicted profile of solar radiation on building envelope sides	$I_{\mathrm{solarpred}}$ (herein noted as $I_{\mathrm{dif}}, I_{\mathrm{dir}}$)	Predicted profiles of solar radiation received from the weather forecast service
Identified parameters of the simplified building thermal dynamics model	$A_{room}, B_{room}, C_{room}, D_{room}$ (herein noted as $A, [B_u, B_d], C$)	Parameters identified through a procedure provided in D4.4.1 prediction and estimation module
Predicted profile of temperature of the heating/cooling medium	T_{supply_med}	Values computed through optimization on the central HVAC level
Predicted profile of flow of the heating/cooling medium	q_{supply_med}	Values computed through optimization on the central HVAC level
Profile of the energy price for the heating/cooling energy	$J^*(E_t)$	Profile of prices for the heating/cooling demand that is generated by the first higher module (prices and boundaries where they hold)
Identified parameters of a model that relates attainable **heating/cooling energy** on a zone element with respect to the predicted medium profile and flow, as well as to the room temperature	$A_{HE}, B_{HE}, C_{HE}, D_{HE}$ (common annotation for either $A_{FCU}, B_{FCU}, C_{FCU}, D_{FCU}$ or $A_{RAD}, B_{RAD}, C_{RAD}, D_{RAD}$ or $A_{FH}, B_{FH}, C_{FH}, D_{FH}$)	Parameters identified through procedures described in D4.4.1 prediction and estimation module
Identified parameters of **the heat loss model**	C_{loss}, D_{loss}	Parameters of the model that relates the supply medium flow and temperature at the output of the central HVAC system with the medium temperature at the inlet of the heating/cooling elements
Adjustable parameter of the optimization problem for price-comfort weighing	δ (herein noted as η)	Parameter for the comfort part of the criterion function in predictive control
Adjustable parameter of the optimization problem for allowed comfort setpoint violation	σ (herein noted as Δ)	Parameter for the comfort part of the criterion function in predictive control

Table 2.2: Outputs of the model predictive control module for zones comfort control.

Variable name	Notation	Description
Optimal profile of heating/cooling energy from actuators in zones	E^{T} (herein noted as U)	Heating/cooling consumption profile calculated for each room controlled through the EMS, that is transferred to the interface submodule of the zone level, and also to the HVAC level MPC module. For zones not in 3Smart EMS control prediction of required energies is calculated if they are in automatic control mode, otherwise if in manual operation the energy requirement is considered as 0.
Optimized profile of temperatures in zones	T (herein noted as Y)	Predicted temperature profiles for zones, needed on the central HVAC level. For zones not in 3Smart control the temperature profile is determined through simulation along the prediction horizon.

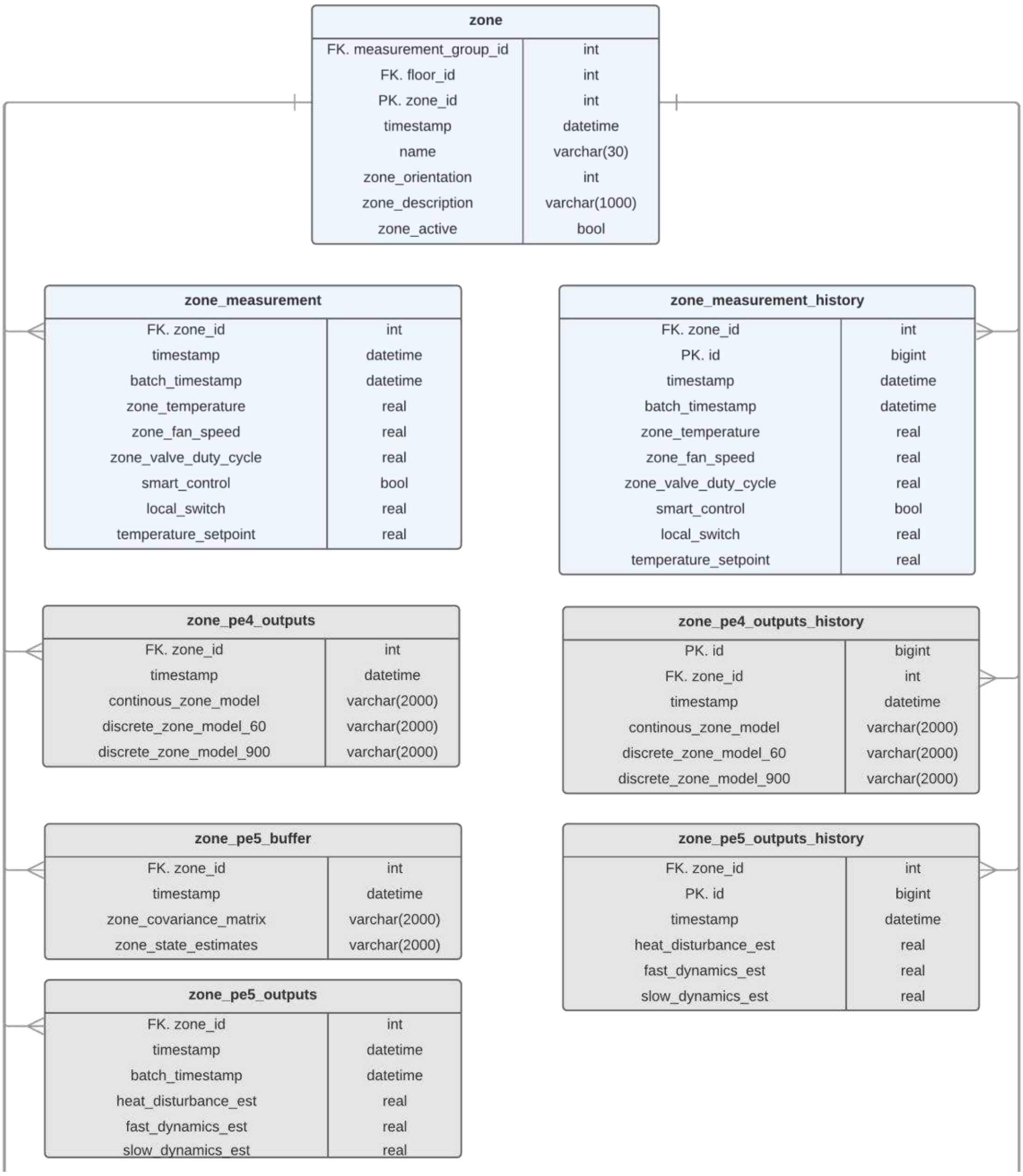
Bibliography

[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 – Open software module for zone consumption management – Model predictive control module

Annex 1 is provided as a separate document.

Annex 2 – 3Smart database organization for open software module for zone consumption management – Model predictive control module



zone_pe6_outputs	
FK. zone_id	int
timestamp	datetime
heat_disturbance_pred	varchar(2000)

zone_pe6_outputs_history	
FK. zone_id	int
PK. id	bigint
timestamp	datetime
heat_disturbance_pred	varchar(2000)

zone_pe7_outputs	
FK. zone_id	int
timestamp	datetime
temperature_setpoint_pred	varchar(2000)

zone_pe7_outputs_history	
FK. zone_id	int
PK. id	bigint
timestamp	datetime
temperature_setpoint_pred	varchar(2000)

zone_pe10_outputs	
FK. zone_id	int
timestamp	datetime
occupancy_pred	varchar(2000)

zone_pe10_outputs_history	
FK. zone_id	int
PK. id	bigint
timestamp	datetime
occupancy_pred	varchar(2000)

zone_mpc	
PK. zone_mpc_id	int
FK. building_id	int
timestamp	datetime
sampling_time	integer
solver	integer
description	varchar(10000)

zone_mpc1_inputs_offline	
FK. zone_mpc_id	int
FK. zone_id	int
timestamp	datetime
offline_calculated_lp_program_matrices	varchar(10000000)

zone_mpc1_outputs	
FK. zone_mpc_id	int
FK. zone_id	int
timestamp	datetime
batch_timestamp	datetime
energy_request_start_time	datetime
zone_energy_request	real
zone_energy_profile	varchar(1000)
zone_temperature_profile	varchar(1000)
instance_id	int
zone_mpc_status	int
mpc_setpoint	real
mpc_disturbance	real

zone_mpc1_outputs_history	
PK. id	bigint
FK. instance_id	int
FK. zone_id	int
timestamp	datetime
batch_timestamp	datetime
energy_request_start_time	datetime
zone_energy_request	real
zone_energy_profile	varchar(1000)
zone_temperature_profile	varchar(1000)
instance_id	int
zone_mpc_status	int
mpc_setpoint	real
mpc_disturbance	real

zone_occupancy	
FK. zone_id	int
timestamp	datetime
zone_occupancy	int
zone_occupancy_schedule	varchar(1000)

zone_windows	
FK. Zone ID	int
Timestamp	datetime
zone_window_state	int

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 MODEL PREDICTIVE CONTROL IN ZONES

Open software module for zone consumption management - Model predictive control module for zones comfort control

Project Acronym	3Smart			
Grant Agreement No.	DTP1-502-3.2-3Smart			
Funding Scheme	Interreg Danube Transnational Programme			
Project Start Date	1 January 2017			
Project Duration	30 months			
Work Package	4			
Task	4.5			
Date of delivery	**Contractual:** 30 June 2019	**Actual:** 30 June 2019		
Code name	**Version:** 2.0	Final ☐	Final draft ☐	Draft ☐
Type of deliverable	Report			
Security	Public			
Deliverable participants	UNIZGFER,			
Authors (Partners)	Anita Martinčević, Vinko Lešić, Mario Vašak (UNIZGFER)			
Contact person	Anita Martinčević (UNIZGFER)			
Abstract (for dissemination)	This deliverable gives formulation of the model predictive temperature control problem in buildings and its fair comparison with conventional controllers with the same level of flexibility.			
Keyword List	Model predictive control, Zone heating/cooling control			

Revision history

Revision	Date	Description	Author (Organization)
v0.1	13 November 2017	First draft of D4.4.1	Anita Martinčević, Vinko Lešić, Mario Vašak (UNIZGFER)
v0.2	15 December 2017	Chapter concerning actuator types and their physical limitations added	Anita Martinčević, Mario Vašak (UNIZGFER)
v1.0	20 December 2017	Final version of D4.4.1	Anita Martinčević, Mario Vašak (UNIZGFER)
v1.1	15 January 2019	Taken over the respective D4.4.1 document and updated based on feedback from pilots operation	Anita Martinčević, Mario Vašak (UNIZGFER)
v1.2	7 June 2019	Updated based on feedback from pilots	Mario Vašak (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Model predictive control has been recognized as one of the essential solutions to achieve considerable energy savings in buildings. However, its performance on a building zone level can be inferior to a well-tuned conventional controller, especially in situations with constant energy prices and conservative comfort constraints. Optimization problem in the background has to be chosen to guarantee recursive feasibility and considerable energy savings without compromising the users comfort at the same time.

This annex to D4.5.3 document gives a formulation of the model predictive temperature control problem in buildings and its fair comparison with conventional controllers with the same level of flexibility allowed in zone temperature control. All controllers are tested for a system with seasonal heating and cooling, which is the most common case in real applications. It is shown that the introduced formulation leads to the model predictive controller that significantly outperforms conventional controllers both in energy consumption and users comfort.

1 Introduction

In the last few decades, the increased awareness of the limitations of fossil fuels combined with increasing energy demand worldwide and noticeable effects of irrational energy consumption has resulted in energy-efficiency policies for advocating and encouraging rational energy consumption. The improvement of the buildings sector energy efficiency becomes critical to attain a balance in many sectors. This is most notably the case with the power sector, as almost half of all the energy consumed today is used in buildings [1]. Given the large share of energy consumed in buildings, improvement of buildings energy efficiency is crucial to ensure long-term energy security. Model Predictive Control (MPC) framework, due to its distinct advantages, significantly outstands among other conventional methods applicable for the building control design. Conventional control algorithms mostly rely on the calibration of algorithms designed for a typical building according to the approximate rule of thumb or trial and error method. Two most common conventional controllers used within Building Energy Management Systems (BEMSs) on a zone level are standard proportional-integral (PI) controller and hysteresis-type (on/off) controller. The MPC is an optimization-based control approach where control actions are calculated by solving finite horizon optimal control problem and applied in a receding horizon fashion [2]. To achieve energy savings and outperform conventional controllers, this optimization problem needs to be chosen very carefully. Potential energy savings are up to 40% [3]–[7], but they must be evaluated in a fair set-up. In most of the reported studies, energy savings are gained by setting users comfort zone very wide, mostly within the interval 20-25 °C or even wider [8]–[11]. The most commonly recommended temperature that ensures comfort is 24 °C and most users are not so flexible to allow temperature deviations of $\pm$2.5 °C or even larger [12], [13]. An additional problem of MPC formulations reported in [8] is a lack of recursive feasibility, e.g. if the initial state violates the control problem constraints, small enough sampling time under control input constraints will make the control problem infeasible. One way to deal with feasibility issues of this type is to replace hard constraints with so-called soft constraints. Another solution is to replace classic hard constraints by chance constraints [9], [10], but for south oriented building zones with a large glazing area and only cooling or heating available at the certain time, this formulation often results in infeasibility as well. Furthermore, for MPC formulations with wide comfort zones, it is very hard to determine obtained gains since conventional PI controller has a task to follow the reference and on-off type hysteresis controller usually has much narrower hysteresis bounds. In addition to extensive literature with claimed gains of MPC over conventional controllers, there are several reports showing that for standard applications performance of the MPC controller on a zone level is approximately the same, or even worse than the performance of a well-tuned conventional controller [14], [15]. Standard application implies constant energy prices and disabled advanced options such as peak shaving, uncertainty handling, etc. This deliverable is focused on a deterministic MPC application in building zones temperature control, and its comparison with standard hysteresis and PI controllers, where MPC constraints are matched with hysteresis bounds. The MPC and hysteresis approach are both tested for three standard cases with allowed temperature deviation from set-point set to $\pm$[0.2 0.5 0.7] °C, which corresponds to limits of cyclic temperature variations of A, B and C classes of the thermal environment defined by ISO 7730 standard [12]. The MPC problem formulation presented in this deliverable enables considerable energy savings without compromising the users comfort. Simulations are performed for the test-site comprised of the 9th floor of University of Zagreb, Faculty of Electrical Engineering and Computing (FER) skyscraper building with 23 zones equipped with fan-coils. The gain is obtained by optimally manipulating the temperature in the given comfort interval, with respect to current and predicted outdoor conditions.

This deliverable is organized as follows. Submodule interfaces are in detail described in Chapter 2. Chapter 3 gives the formulation of MPC with a description of the building model and detailed description of the optimization criterion. In Chapter 4, conventional controllers used for performance comparison are presented as well as the simulation set-up with used comfort metrics and results. In Chapter 5 variable physical limitations of the actuators are introduced. Chapter 6 concludes the deliverable.

2 Module interface

Table 2.1: Required inputs for the model predictive control module for zones comfort control

Variable name	Notation	Description
Estimates of the current states of the simplified building thermal dynamics model	x_0	Non-measured states of the simplified model are estimated.
Predicted profile of comfort setpoints	SP	Temperature and other comfort setpoints that are predicted based on models tuned on historical data
Predicted profile of actuation in rooms/zones that are outside the EMS control	manual actuation commands for the room heating/cooling elements	Users may be in control to set a manual command for the room heating/cooling elements or some rooms/zones may be permanently outside the EMS control, and the EMS should be able to respect it; currently just extrapolation of the current manual selection will be used for predictions
Predicted profile of cumulated heating/cooling disturbances in zones	$E_{\mathrm{d}}^{\mathrm{pred}}$	Disturbances predicted based on models tuned on historical data
Predicted profile of outdoor temperature	T_{o}	Predicted profile of outdoor temperature received from the weather forecast service
Predicted profile of solar radiation on building envelope sides	I_{solar} (herein noted as $I_{\mathrm{dif}}, I_{\mathrm{dir}}$)	Predicted profiles of solar radiation received from the weather forecast service
Identified parameters of the simplified building thermal dynamics model	$A_{room}, B_{room}, C_{room}, D_{room}$ (herein noted as $A, [B_u, B_d], C$)	Parameters identified through a procedure provided in D4.4.1 prediction and estimation module
Predicted profile of temperature of the heating/cooling medium	T_{supply_med}	Values computed through optimization on the central HVAC level
Predicted profile of flow of the heating/cooling medium	q_{supply_med}	Values computed through optimization on the central HVAC level
Profile of the energy price for the heating/cooling energy	$J^*(E_t)$	Profile of prices for the heating/cooling demand that is generated by the first higher module
Identified parameters of a model that relates attainable **heating/cooling energy** on a zone element with respect to the predicted medium profile and flow, as well as to the room temperature	A_{HE} B_{HE}, C_{HE}, D_{HE} (common annotation for either $A_{FCU}, B_{FCU}, C_{FCU}, D_{FCU}$ or $A_{RAD}, B_{RAD}, C_{RAD}, D_{RAD}$ or $A_{FH}, B_{FH}, C_{FH}, D_{FH}$)	Parameters identified through procedures described in prediction and estimation module

Identified parameters of **the heat loss model**	C_{loss}, D_{loss}	Parameters of the model that relates the supply medium flow and temperature at the output of the central HVAC system with the medium temperature at the inlet of the heating/cooling elements
Adjustable parameter of the optimization problem for price-comfort weighing	δ (herein noted as η)	Parameter for the comfort part of the criterion function in predictive control
Adjustable parameter of the optimization problem for allowed comfort setpoint violation	σ (herein noted as Δ)	Parameter for the comfort part of the criterion function in predictive control

Table 2.2: Outputs of the model predictive control module for zones comfort control.

Variable name	Notation	Description
Optimal profile of heating/cooling energy from actuators in zones	E^{T} (herein noted as U)	Heating/cooling consumption profile calculated for each room controlled through the EMS, that is transferred to the interface submodule of the zone level, and also to the HVAC level MPC module
Optimized profile of temperatures in zones	T (herein noted as Y)	Predicted temperature profiles for zones that will be needed on the central HVAC level

3 Model predictive control module for zones comfort control

Possible energy savings by using MPC have been widely reported in the literature. The aim of this deliverable is to identify the best formulation of MPC problem such that it outperforms conventional controllers even under the same conditions. Distinct advantages of MPCs lie in (*i*) using the relevant future information in making control decisions; (*ii*) inherent handling of multi-input multi-output (MIMO) systems; (*iii*) routine respecting of system constraints (e.g. finite amount of heating/cooling power, comfort intervals) and (*iv*) explicit orientation of the control actions towards the specified goal such as economic, environmental or their combination. All the mentioned advantages make the MPC a favourable choice for the BEMS design.

3.1 Building model

Mathematical model of a building is a basis for MPC implementation. The most popular building modelling framework consists of using resistor-capacitor (RC) network to model thermodynamic processes in buildings [8],[18],[19]. The basic idea of this methodology is to represent building elements (or complete zones) with as few thermal circuit elements as possible. The resulting linear dynamic models are established as simple, computationally efficient and accurate enough models. Due to the hourly resolution of the weather forecast given for the building location by Croatian Meteorological and Hydrological Service, a model of the 9th floor of the FER building built on RC principles is discretized with an hourly sampling time T_s. Forecast of solar irradiance given at a time instant k is cumulative solar irradiance received on a unit surface within one sampling interval. Forecast of the outdoor air temperature T_{out} is given as predicted value at k. To utilize this information, influence of outdoor air temperature is discretized by employing first-order hold while the rest of the system is discretized by zero-order hold. The resulting discrete system is as follows:

$$x(k+1) = Ax(k) + B_u u(k) + B_d^1 T_{out}(k) + B_d^2 T_{out}(k+1) + B_d^* d(k), \tag{3-1}$$

$$y(k+1) = Cx(k+1) \tag{3-2}$$

where $x(k) \in \mathbb{R}^n$ is the system state vector, $u(k) \in \mathbb{R}^q$ is the thermal energy input to each of q controllable zones from k to $k+1$, $T_{out}(k) \in \mathbb{R}^1$ is outdoor air temperature, $d(k) \in \mathbb{R}^p$ is disturbance input (solar irradiance, internal gains, etc.) and $y(k) \in \mathbb{R}^m$ is the vector of room temperatures at time step k. Matrices A, B_u, B_d^1, B_d^2 and $B_d^*$are matrices of appropriate dimensions.

3.2 MPC problem formulation

The resulting performance of the control system with employed MPC depends completely on the MPC problem formulation. A special care needs also to be paid to the fact that the computed controls are implemented in a receding horizon fashion. The MPC uses the dynamic model of the building and information on the future disturbance profiles to predict future building behaviour and, based on these predictions, computes the optimal control input trajectory. Predicted states and outputs along the prediction horizon $N \in \mathbb{Z}$ are conveniently written as:

$$Y = \alpha x_{t|t} + \beta U + \gamma_1 D_1 + \gamma_2 D_2, \tag{3-3}$$

where Y is a stack of future outputs:

$$Y = \begin{bmatrix} y_{t+1|t}^T & y_{t+2|t}^T & \dots. & y_{t+N|t}^T \end{bmatrix}^T, \tag{3-4}$$

where U is a stack of future outputs:

$$U = \begin{bmatrix} u_{t|t}^T & u_{t+1|t}^T & \dots & u_{t+N-1|t}^T \end{bmatrix}^T, \tag{3-5}$$

where D_1 and D_2 are stacks of future disturbances:

$$D_1 = [T_{o,t|t} \quad T_{o,t+1|t} \quad \dots \quad T_{o,t+N|t}]^T, \tag{3-6}$$

$$D_2 = \begin{bmatrix} d_{t|t}^T & d_{t+1|t}^T & \dots & d_{t+N-1|t}^T \end{bmatrix}^T, \tag{3-7}$$

and α, β, γ_1 and γ_2 are matrices based on the discrete building model matrices (3-1). Notation $y_{t+k|t}^T$ denotes predicted rooms temperature at time $t+k$, obtained by applying the input sequence U to the system starting from current state $x_{t|t}$.

The most frequent MPC problem formulation for temperature control in buildings consists of a simplistic minimization of energy consumption with respect to the temperature constraints set by the end-users and physical limitations of actuators [8]:

$$\begin{aligned} \min_U \quad & J(U, x_{t|t}, D_1, D_2) \\ s.t. \quad & SP - \Delta \leq y \leq SP + \Delta \\ & P_{\min} \leq U \leq P_{\max} \end{aligned} \tag{3-8}$$

where:

$$J(U, x_{t|t}, D_1, D_2) = \sum_{k=0}^{N-1} \left| R_{t+k|t} u_{t+k|t} \right| , \tag{3-9}$$

SP is a stack of future set-points profiles per zone:

$$SP = \begin{bmatrix} SP_{t+1|t}^T & SP_{t+2|t}^T & \dots & SP_{t+N|t}^T \end{bmatrix}^T, \tag{3-10}$$

and $P_{\max}$ and $P_{\min}$ are the physical limitation of actuators present in the zone, calculated based on the known equations for calculation of a maximum/minimum power and prediction of medium mass flow and supply temperature [22]. Operator $|.|_1$ denotes L_1 norm and $P_{\min}$ and $P_{\max}$ are limitations on power inputs U. Set-points $SP_{t+k|t} \in \mathbb{R}^m$ and allowed deviations from them along the prediction horizon $\Delta \in \mathbb{R}^{N \cdot m}$, are defined by the end-users for each of m zones. $R_{t+k|t} \in \mathbb{R}^{q \times q}$ is a weighting matrix typically set to identity matrix of appropriate size. Formulations like these handle users temperature constraints as hard constraints, which often results in infeasibility, especially when sign change of U is not possible (only heating or only cooling available).

To solve this problem, temperature constraints are ”softened” by introducing them into the cost function through slack variables $\sigma_{t+k|t} \in \mathbb{R}^m$ with large weights $G_{t+k|t} \in \mathbb{R}^{m \times m}$ (e.g. $G_{t+k|t} = 10^6 I_{m \times m}, \forall k$). The resulting optimization problem is as follows:

$$\begin{aligned} \min_{U,\Sigma} \quad & J(U, x_{t|t}, D_1, D_2, \Sigma) \\ s.t. \quad & SP - \Delta - \Sigma \leq y \leq SP + \Delta + \Sigma, \\ & P_{\min} \leq U \leq P_{\max}, \\ & \Sigma \geq 0 \end{aligned} \tag{3-11}$$

where:

$$J(U, x_{t|t}, D_1, D_2, \Sigma) = \sum_{k=0}^{N-1} \left| R_{t+k|t} u_{t+k|t} \right|_1 + \sum_{k=1}^{N} \left| G_{t+k|t} \sigma_{t+k|t} \right|_1 \tag{3-12}$$

with Σ defined as:

$$\Sigma = \begin{bmatrix} \sigma_{t+1|t}^T & \sigma_{t+1|t}^T & \dots & \sigma_{t+N|t}^T \end{bmatrix}^T, \tag{3-13}$$

In most of the situations, occupants want the exact temperature to the one they have chosen on the zone thermostat. This can be obtained by defining the MPC problem as a classic reference tracking problem:

$$\begin{aligned} \min_{U} \quad & J(U, x_{t|t}, D_1, D_2) \\ s.t. \quad & P_{\min} \leq U \leq P_{\max}, \end{aligned} \tag{3-14}$$

where:

$$J(U, x_{t|t}, D_1, D_2) = \sum_{k=0}^{N-1} \left| R_{t+k|t} u_{t+k|t} \right|_1 + \sum_{k=1}^{N} \left| Q_{t+k|t} (y_{t+k|t} - SP_{t+k|t}) \right|_1 \tag{3-15}$$

and $Q_{t+k|t} \in \mathbb{R}^{m \times m}$ is a weighting matrix. This MPC formulation, combined with a receding horizon strategy, often results in either minimum energy performance at the cost of completely disregarded temperature comfort or in permanent set-point following with disregarded energy consumption. A compromise between the two options is made through the mentioned weighting matrices.

To tackle the opposing criteria of reference following and energy saving, weighting matrices $R_{t+k|t}$ and $Q_{t+k|t}$ have to be chosen in a way which enables smart switching between these two requirements based on predicted disturbance profiles. Optimization cost J presented in this deliverable is comprised of two terms, J_1 and J_2. Term J_1 is related to minimization of energy consumption:

$$J_1(U, x_{t|t}, D_1, D_2) = \sum_{k=0}^{N-1} \left| R_{t+k|t} u_{t+k|t} \right|_1, \tag{3-16}$$

where $R_{t+k|t} \in R^1$ is the energy cost important for subsequent coordination with the microgrid or HVAC level. Temperature demands of the end-users are forced by the term J_2:

$$\begin{aligned} J_2(U, x_{t|t}, D_1, D_2, \Sigma) &= \sum_{k=1}^{N} \left|G_{1,t+k|t}\sigma_{1,t+k|t}\right|_1 + \sum_{k=1}^{N} \left|G_{2,t+k|t}\sigma_{2,t+k|t}\right|_1 \\ &+ \eta \sum_{k=1}^{N} \left|R_{t+k|t}\, Q_{t+k|t}(y_{t+k|t} - SP_{t+k|t})\right|_1 \end{aligned} \tag{3-17}$$

where $\eta \geq 0 \in \mathbb{R}^1$ is arbitrary weighting coefficient. Asymmetric slack variables σ_1 and σ_2 defined as in Eq. (3-13) guarantee minimal temperature requirements, i.e. different weighting factors $G_{1,t+k|t} \neq G_{2,t+k|t}$ can be used for penalizing upper and lower limit temperature constraints violation. J_2 can be interpreted as four-segmented convex PieceWise Affine (PWA) penalty function (Figure 3.1).

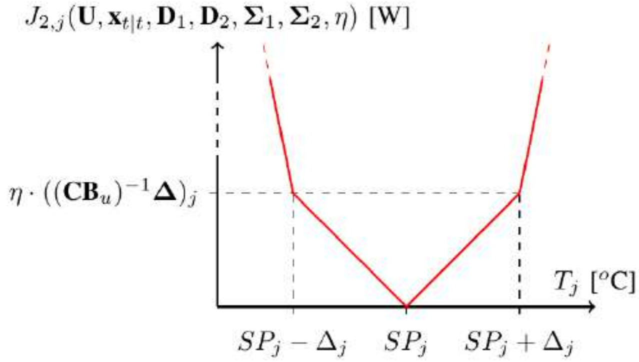


Figure 3.1: Convex PWA penalty function for the $\boldsymbol{j}$th zone and $\boldsymbol{N = 1}$.

To be comparable, both parts of optimization cost have to be expressed in the same units. Since J_1 is defined in watts, weighting matrix $Q_{t+k|t}$ is utilized to convert temperature related cost part J_2 from degrees Celsius to watts. Building model (3-1) is linear so the amount of energy which can be saved by allowing the zone temperature to slide below the set-point during the heating season or above the setpoint during the cooling season, under the same weather conditions, is linear function of system dynamics. Sensitivity of the energy consumption to the zones temperature is defined as:

$$\frac{\partial U}{\partial Y} = \frac{\partial(\beta^{-1}(Y - \alpha x_{t|t} - \gamma_1 D_1 - \gamma_2 D_2))}{\partial Y} = \beta^{-1}. \tag{3-18}$$

Matrix $\beta^{-1}$is lower bidiagonal matrix with all elements on the main diagonal equal to $(CB_u)^{-1}$ and to the $-(B_u^{-1}AC^{-1})$ on the secondary diagonal. By setting weighting matrices along the horizon to:

$$Q_{t+k|t} = \begin{bmatrix} (CB_u)^{-1} \\ -(B_u^{-1}AC^{-1}) \end{bmatrix}, \qquad k = 1, \dots, N-1, \tag{3-19}$$

and $Q_{t+N|t} = (CB_u)^{-1}$, J_2 is converted from degrees to watts. Final optimization cost is thus:

$$J(U, x_{t|t}, D_1, D_2, \Sigma_1, \Sigma_2, \eta) = J_1(U, x_{t|t}, D_1, D_2) + J_2(U, x_{t|t}, D_1, D_2, \Sigma_1, \Sigma_2, \eta) \tag{3-20}$$

Weighting factor η determines the importance of reference tracking with respect to the minimization of energy consumption (Figure 3.2).

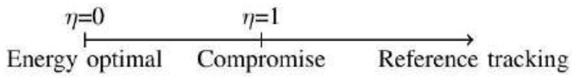


Figure 3.2: Dependence between weighting factor $\boldsymbol{\eta}$ and system performance.

For $\eta = 1$ both energy consumption and reference tracking have the same weights, so the controller will decide what is best from the energy viewpoint. For $\eta > 1$ bigger weight is set on reference tracking so the system will slide from the set-point when it is unavoidable or when this will results with considerable energy savings. Final optimization problem written in compact form is as follows:

$$\begin{aligned} \min_{U,\Sigma_1,\Sigma_2} \quad & J\big(U, x_{t|t}, D_1, D_2, \Sigma_1\,, \Sigma_2, \eta\big) \\ s.t. \quad & SP - \Delta - \Sigma_1 \leq y \leq SP + \Delta + \Sigma_2 \\ & P_{\min} \leq U \leq P_{\max}, \\ & \Sigma_1 \geq 0 \\ & \Sigma_2 \geq 0 \end{aligned} , \tag{3-21}$$

With such a criterion, in the heating season, solar irradiance influence that can result in overheating is heavily penalized, which adversely forces the system to minimize overheating, i.e. to use the free energy from outdoors starting from the lower edge of the allowed range. Effectively, the actuators are controlled such that the lower bound of the temperature range is reached prior to the stream of free energy from outdoors. In the cooling season system is forced to quit cooling the zone when free cooling can be utilized.

4 Results for fixed physical limitations of actuators

The test-case studied in this deliverable is 9th floor of the FER building (Fig. 4). Overall studied area of the test-site is about 700 m^2 large.

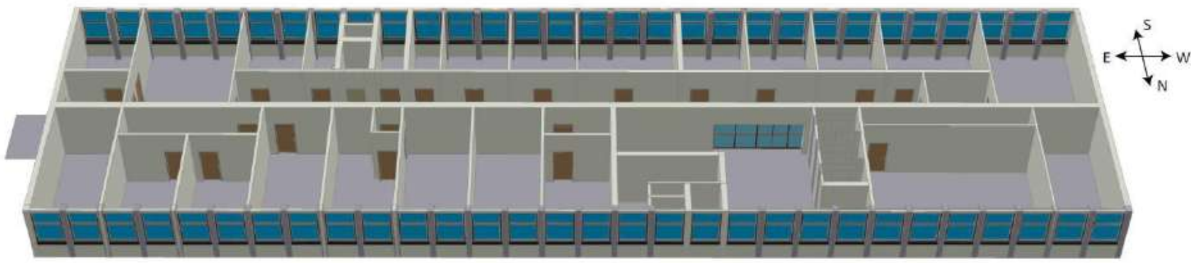


Figure 4.1: 3D drawing of the 9th floor of FER skyscraper building.

All controllers are employed to control directly the thermal powers required to achieve the desired temperature behaviour. Fan-coils have very fast dynamics, i.e. every feasible power can be achieved in negligible time, so in this study lower level controller required to calculate direct control actions to fan-coils, required for real implementation of PI and MPC algorithms, is not considered. Instead, it is assumed that power references can be tracked perfectly. It is assumed that all unit can generate fixed maximal and minimal power $P_{\max}$ and $P_{\min}$.

Data used for external conditions (outdoor air temperature and solar irradiances) are historical measurements from the year 2014 taken on a meteorological station close to the FER building. Weather disturbances are, in such a setup, assumed to be perfectly forecasted, and all other disturbances are neglected. For real building implementations, with a lot of uncertain and unpredictable disturbances, their compensation is performed by introducing an estimator into the control loop. Optimization horizon is 24~h long with hourly sampling time. Hysteresis and PI controller are continuous-time controllers operated in a standard closed-loop fashion.

It is assumed that at a certain moment either only heating or only cooling is available. This corresponds to standard two pipe implementation of heating/cooling system present in many buildings. The cooling season starts at 1st June and lasts until 1st October. The building operates in two working modes: daily mode, from 6 to 18 hours, during which temperature requirements of the end-users are set to 24 °C for both seasons, and night mode, from 18 to 6 hours. During night mode ($t + k$ is within the interval from 18 to 6 hours) reference following part is omitted from the temperature related cost function part J_2. This implies $Q_{t+k|t} = 0_{m\times m}$ for $t + k$ within night mode interval and as in (3-19) otherwise.

Allowed deviations during night mode are set to $\Delta_j^* = 6$ °C, which effectively ensures the minimum temperature of $SP_j - \Delta_j^*$ to prevent the building from cooling down too much during the heating season. In the cooling season, night mode additionally implies unavailable cooling powers, i.e. $P_{\min} = P_{\max} = 0$.

Simulations are performed within MATLAB environment [16]. Optimization problems are solved by using YALMIP [20] and CPLEX [21].

4.1 Conventional controllers

Conventional PI controller represents a typical decentralized control approach which can be found in many building applications. Comparison with PI controller will give the baseline for energy consumption since PI controller ensures tracking of user reference all the time when it is possible. The synthesis of PI controller is performed automatically within the MATLAB environment to ensure the best performance regarding reference tracking [16].

The test-site zones are currently controlled by an industrial RXC controller [17] based on hysteresis control of fan-coils. To assess the possible energy savings on FER building, achievable by improvement of its BEMS with MPC, hysteresis control is also included into comparison. The fan-coils operate at 3 different fan speeds (FS). The amount of power at certain speed depends on the temperature and mass flow of the heating/cooling medium and is considered as constant. To be as close as possible to the real system, power amounts are estimated from calorimeter readings available on each major supply duct of the heating/cooling system on the considered floor of the Faculty building. The RXC controller switches between available power outputs based on the temperature difference between current jth zone temperature T_j and set-point value SP_j set by the end-users (Figure 4.2). Hysteresis width 2Δ is predefined and equal for all zones.

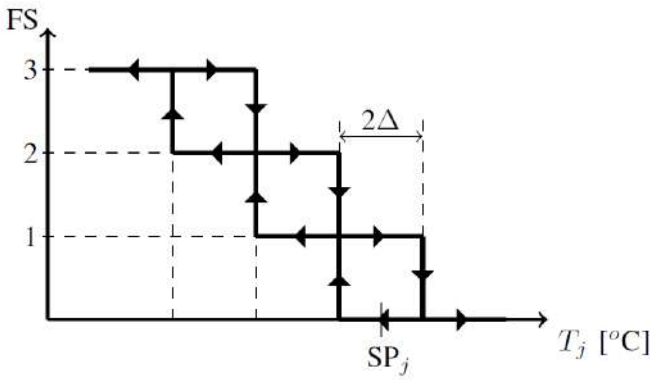


Figure 4.2: Hysteresis control law for the ***j***th zone in the heating season.

4.2 Comfort metrics

Average deviation AD from the set-point is calculated as the ratio of the sum of all the deviation amounts during daily mode and overall number of samples during daily mode, where M is number of measurements per zone:

$$AD = \frac{1}{mM}\sum_{t=0}^{M}\left|y_{t|t} - SP_{t|t}\right|_1. \tag{4-01}$$

4.3 Results

Comparison of overall energy consumption during the whole year of 2014, for different types of control, different flexibilities Δ and different weights η is given in Figure 4.3.

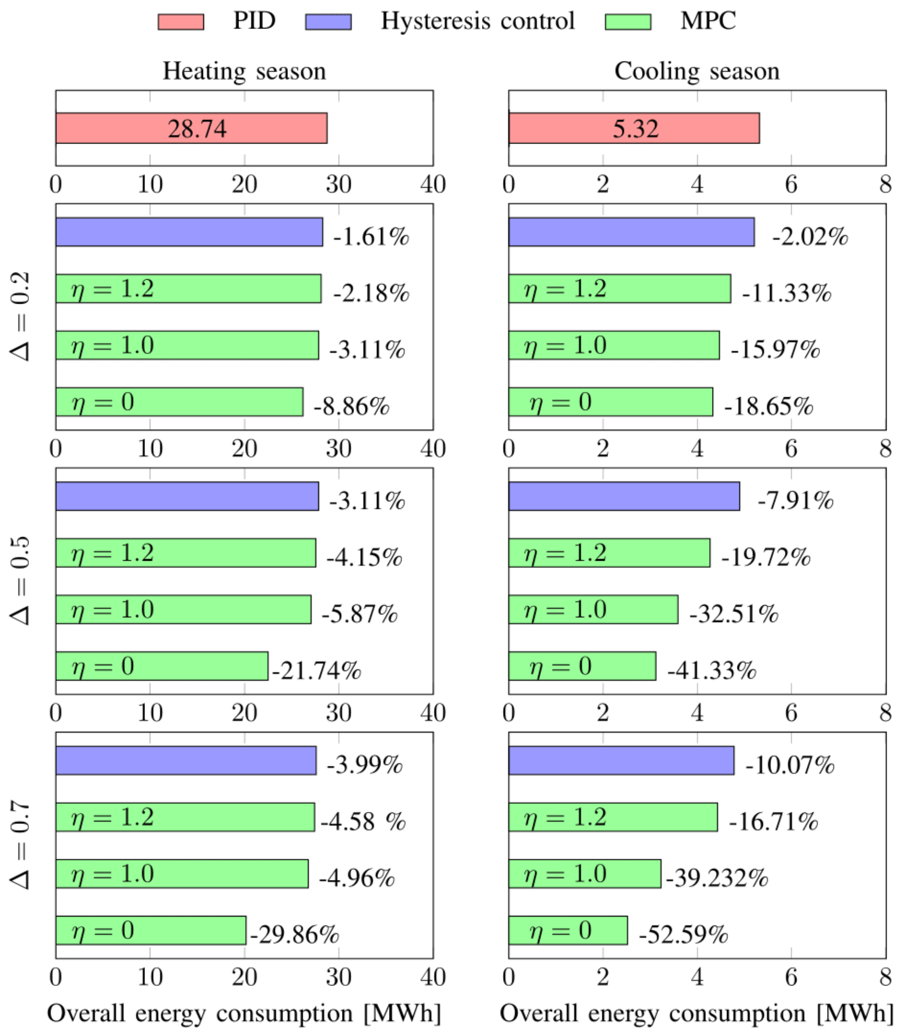


Figure 4.3: Comparison of the overall energy consumption of different controllers in 2014.

All saving percentages in the Figure 4.3 are calculated in relation to the PI controller. Energy savings achievable by replacing the RXC controller with the presented MPC with the same temperature constraints and $\eta = 1$ are up to 30% in the cooling season and up to 3% in the heating season. For the case with $\eta = 0$ savings percentages are even higher, but at the cost of totally disregarded users comfort. By choosing η the user can define does he want to be energy efficient and save the energy or does he strictly want the reference following behaviour. Obtained numbers show large potential of the presented MPC formulation, especially because for $\eta \geq 1$ energy savings are not obtained at the account of totally violated users comfort. Moreover, following from Figure 4.4 it may be observed that users comfort is improved, i.e. average deviations of MPC are smaller than the ones of hysteresis controller.

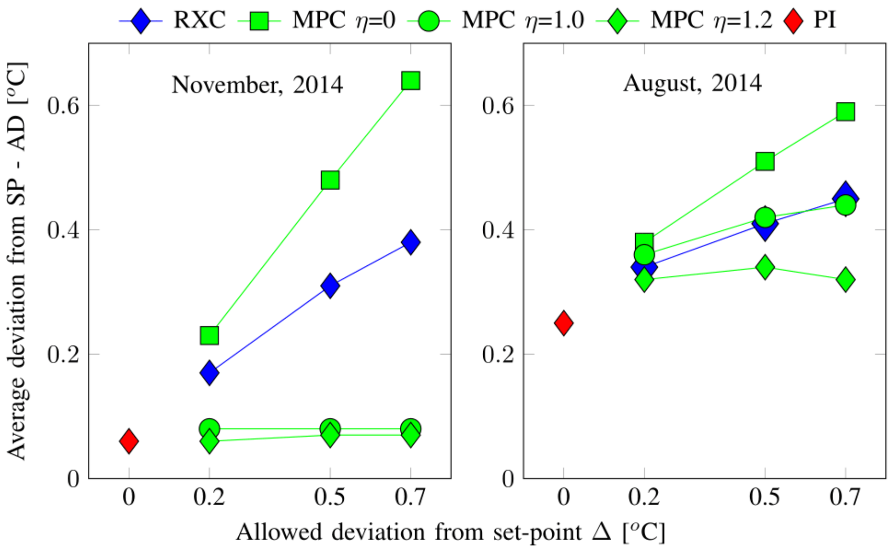


Figure 4.4: Average deviation from SP for different types of controllers.

The results show that for the presented case study energy savings can be increased beyond the 20% without becoming worse in average set-point deviation than hysteresis controller. Figure 4.4 shows a comparison of time responses of two selected building zones controlled by the hysteresis controller and the MPC during one week in November 2014 with $\Delta = 0.2$ °C and $\eta = 1.1$.

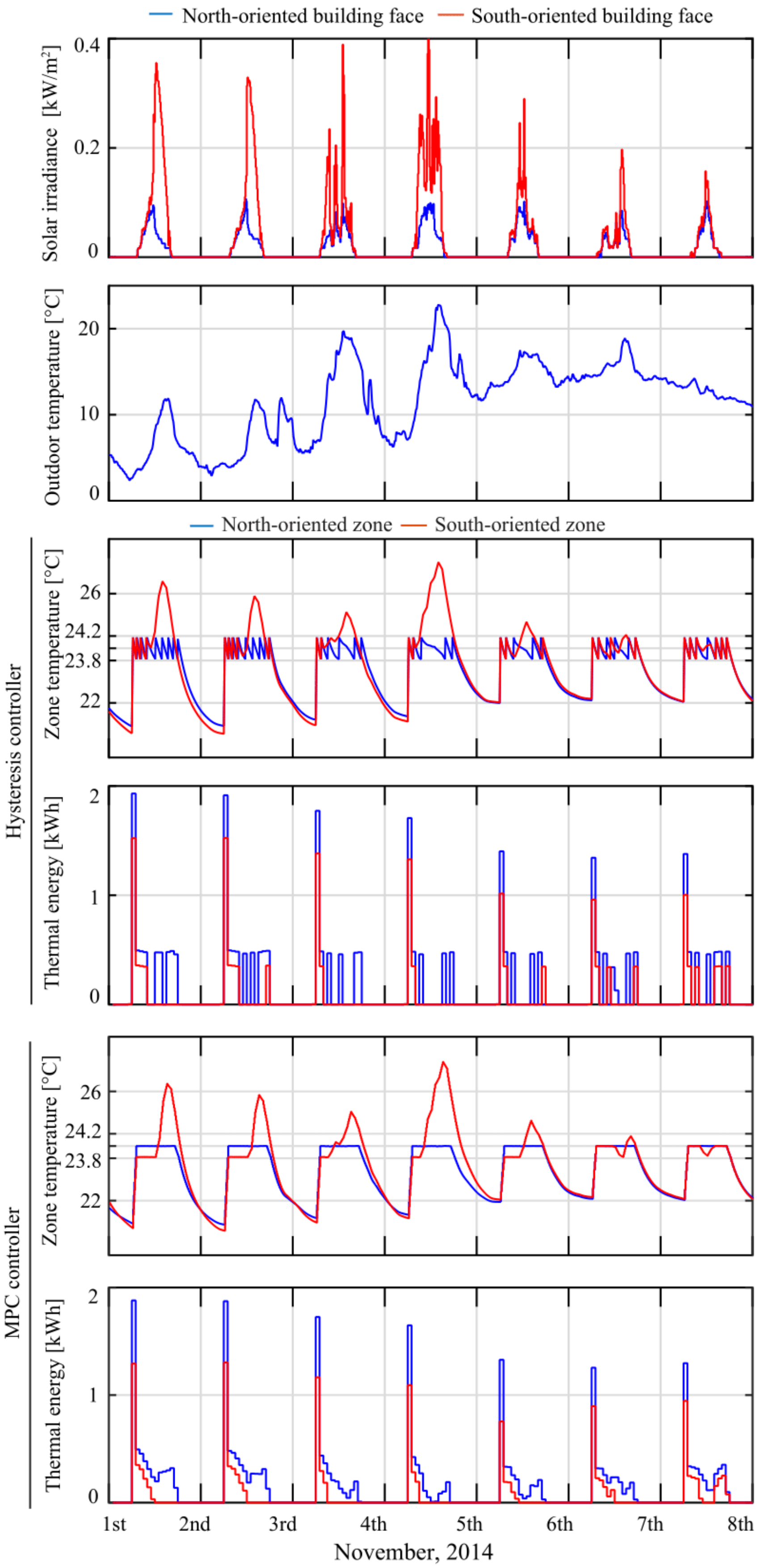


Figure 4.5: Comparison of time responses of two selected building zones controlled by hysteresis controller and MPC during one week of heating season for $\boldsymbol{\Delta} = \mathbf{0.2}$ °C and $\boldsymbol{\eta} = \mathbf{1.1}$.

5 Variable actuator thermal power limitations

Attainable thermal power depends on the actuator type and available heating/cooling medium temperature and mass flow defined by optimizing the energy on the HVAC level or as a constant value specified from the system administrator. In the following subsection three types of problems regarding three different types of thermal actuators will be discussed.

5.1 Fan coil units

Since time constant of the FCU is relatively small compared to the time constants of the building, the attainable thermal power of the Fan Coil Unit (FCU) is defined with the following algebraic equation:

$$P_{\max}(k) = \frac{2Q_{\mathrm{w}}(k)c_{\mathrm{w}}U_{\mathrm{o}}^{H}(k)}{2Q_{\mathrm{w}}(k)c_{\mathrm{w}}+U_{\mathrm{o}}^{H}(k)}\left(T_{\mathrm{w}}^{\mathrm{in}}(k) - T_{\mathrm{a}}^{\mathrm{in}}(k)\right), \tag{5-1}$$

where Q_{w} is medium mass flow through the unit, c_{w} is heat capacity of the heating/cooling medium considered either constant or estimated from the experiments, $T_{\mathrm{w}}^{\mathrm{in}}$ is supply temperature of the unit calculated by using the information on the supply medium temperature on the ducts inlet reduced by temperature losses or considered as equal to the ducts inlet temperature for well insulated pipelines. Air intake temperature $T_{\mathrm{a}}^{\mathrm{in}}$ is considered to be equal to zone temperature (elements of Y vector in (3-4)). Overall heat transfer coefficient for high speed U_{o}^{H} and fixed medium flow Q_{w} is defined as:

$$U_{\mathrm{o}}^{H}(k) = \frac{a^{H}}{1 + b^{H}Q_{w}(k)^{-c^{H}}}, \tag{5-2}$$

Where a^H, b^H and c^H are known parameters found by performing identification procedure as described in Deliverable 4.4.1 related to estimation and prediction. For known FCU model parameters and information on medium mass flow and supply temperature, the variable thermal power limitations can easily be accounted when calculating optimal energy flow into the zone by replacing $P_{\max}$ in (3-8) with (5-1) and setting $P_{\min}$ to zero during a heating season and by replacing $P_{\min}$ in (3-8) with (5-1) and setting $P_{\max}$ to zero during cooling season. The modified optimization problem is still linear while the newly introduced physical constraints take into account the real nature of the FCU.

5.2 Radiators and floor heating

The radiator and floor heating dynamics are much slower than the FCU's. To enable direct control of the radiator and floor heating thermal power and satisfy physical limitations of the system, dynamics limitations have to be introduced into the zone level MPC synthesis. The idea is to utilize the estimated radiator and floor heating models (same form of the model):

$$T_{\mathrm{w}}^{\mathrm{out}}(k+1) = (1 - aq_{w}T_{s})T_{\mathrm{w}}^{\mathrm{out}}(k) + aq_{w}T_{s}T_{\mathrm{w}}^{\mathrm{in}}(k) - T_{s}u(k), \tag{5-3}$$

where q_{w} is medium mass flow through the unit, a is the coefficient related to the heat transfer from heating/cooling medium to the radiator or floor heating system identified by using procedures described in D4.4.1. Estimation and Prediction, T_s is model discretization time which is usually set to the same value as the control sample time, $T_{\mathrm{w}}^{\mathrm{in}}$ is supply temperature of the unit and $T_{\mathrm{w}}^{\mathrm{out}}$ is return medium temperature of the unit. Maximal thermal power is defined as:

$$P_{\max}(k) = U_{\mathrm{o}}(T_{\mathrm{w}}^{\mathrm{out}} - T_{\mathrm{z}})^{n} \ , \tag{5-4}$$

where T_{z} is zone temperature (elements of Y vector in (3-4)), U_o is estimated heat transfer coefficient from radiator/floor surface to the zone air and n is system related parameter which is also identified. Since equation (5-4) is nonlinear, to keep the zone level MPC problem linear, in every time step k it is linearized around return medium temperature and zone temperature predicted in the previous optimization step :

$$P_{\max}(k) = a \cdot \Delta T_{\mathrm{w}}^{\mathrm{out}}(k) + b \cdot \Delta T_{z}(k) \, , \tag{5-5}$$

Where a and b are coefficients obtained after linearization with fixed supply medium temperature and medium mass flow and Δ operator denotes deviation from the linearization points. Linearized expression for $P_{\max}$ is then introduced in (3-8). Since radiators are used for heating only $P_{\min}$ is always zero. Floor heating is used for both cooling and heating, so $P_{\min}$ and $P_{\max}$ change places seasonally as described for FCUs. To be able to perform the optimization initial values of the return medium temperature at every time step have to be available.

6 Conclusion

Possible gains of Model Predictive Control (MPC) in building climate energy savings depend largely on the formulation of the MPC optimization problem. The most important criterion for the problem formulation is to guarantee recursive feasibility to enable real system application. Another important aspect is care about user's comfort. Users are usually most comfortable when the temperature in the zone is exactly on the value they have chosen at a thermostat. The definition of the MPC problem presented in this article enables users to define the desired temperature and their own comfort region. Temperature drifts from the set-point only when this is economically justifiable or when there is no possibility to compensate the effects of external disturbances. MPC can significantly outperform conventional controllers without compromising users comfort. Moreover, it is shown that the users comfort is improved. The expected savings are even much higher with MPC's full potential exploited in zone control, especially in terms of accounted hour-to-hour variable energy prices and in terms of coordination with central heating/cooling medium production and demand response.

Bibliography

[1] Europe's Buildings under the microscope. Buildings Performance Institute Europe (BPIE), 2011.

[2] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," Automatica, vol. 36, no. 6, pp. 789–814, 2000.

[3] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," IEEE Transactions on Control Systems Technology, vol. 20, no. 3, pp. 796–803, 2012.

[4] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, "Use of model predictive control and weather forecasts for energy efficient building climate control," Energy and Buildings, vol. 45, pp. 15–27, 2012.

[5] M. Maasoumy and A. Sangiovanni-Vincentelli, "Total and peak energy consumption minimization of building hvac systems using model predictive control," IEEE Design and Test of Computers, vol. 29, no. 4, pp. 26–35, 2012.

[6] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, "Experimental analysis of model predictive control for an energy efficient building heating system," Applied Energy, vol. 88, no. 9, pp. 3079–3087, 2011.

[7] J. Ma, J. Qin, T. Salsbury, and P. Xu, "Demand reduction in building energy systems based on economic model predictive control," Chemical Engineering Science, vol. 67, no. 1, pp. 92–100, 2012.

[8] M. Vašak, A. Starčić, and A. Martinčević, "Model predictive control of heating and cooling in a family house," in MIPRO, 2011 Proceedings of the 34th International Convention, May 2011, pp. 739–743.

[9] F. Oldewurtel, A. Parisio, C. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using stochastic model predictive control and weather predictions," 2010, pp. 5100–5105.

[10] F. Oldewurtel, C. Jones, A. Parisio, and M. Morari, "Stochastic model predictive control for building climate control," IEEE Transactions on Control Systems Technology, vol. 22, no. 3, pp. 1198–1205, 2014.

[11] M. Vašak and A. Martinčević, "Optimal control of a family house heating system," in Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on, May 2013, pp. 907–912.

[12] European Committee for Standardization, "Ergonomics of the thermal environment - analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria," EN ISO 7730:2005, 2005.

[13] V. Bradshaw, The Building Environment: Active and Passive Control Systems. Wiley, 2006.

[14] M. Stemmann and A. Rantzer, "Temperature control of two interacting rooms with decoupled pi control," vol. 19, 2014, pp. 11 722–11 727.

[15] J. Cigler, J. Siroky, M. Korda, and C. Jones, “On the selection of the most appropriate mpc problem formulation for buildings,” in 11th REHVA World Congress CLIMA 2013, 2013.

[16] MATLAB, version 8.5.0 (R2015a). The MathWorks Inc., 2015.

[17] Siemens Building Technologies, DESIGO RXC Applications Library, Application FNC02.

[18] J. Seem, Modeling of Heat Transfer in Buildings. University of Wisconsin–Madison, 1987.

[19] M. M. Gouda, S. Danaher, and C. P. Underwood, “Low-order model for the simulation of a building and its heating system,” Building Services Engineering Research and Technology, vol. 21, no. 3, pp. 199–208, 2000.

[20] J. Löfberg, “Yalmip: A toolbox for modeling and optimization in MATLAB,” in Proceedings of the CACSD Conference, 2004.

[21] ILOG, Inc, “Ilog cplex: High-performance software for mathematical programming and optimization,” 2013.

[22] 3Smart Deliverable 4.4.1 Open software module for zone consumption management – Estimation and Prediction

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Interfacing submodules for zones management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Mostar Faculty of Mechanical Engineering, Computing and Electrical Engineering
Authors (Partners)	Anita Martinčević, Vinko Lešić, Mario Vašak (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
Contact person	Anita Martinčević (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of modules that interface heating energy commands to the actuating elements in zones to enable energy management of buildings, with their respective interfaces. Details on modules internal logic are provided in the annex.
Keyword List	Mathematical Model of Building; Thermodynamics; Heating and Cooling Elements; Fan Coil Unit; Radiator, Floor Heating and Cooling; Heating Energy Input; Disturbance

Revision history

Revision	Date	Description	Author (Organization)
v1.0	1 September 2018	Initial version based on D4.4.1	Mario Vašak (UNIZGFER)
v1.1	15 January 2019	Updated version	Mario Vašak, Anita Martinčević (UNIZGFER)
v1.2	7 June 2019	Updated version	Mario Vašak (UNIZGFER)
v2.0	30 June 2019	Finalized quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of submodules.

The submodules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on zone level interfacing submodules.

Each submodule is presented via an interfacing table that explains what data are used by the submodules as inputs and what are the final output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following interfacing submodules are designed, commissioned and tested on the zone level:

Z.I.1 – submodule for interfacing commands of heating energy from fan coils to zone air (tested in UNIZGFER, HEP, STREM school and EPHZHB pilot buildings within 3Smart);

Z.I.2 -- submodule for interfacing commands of heating energy from radiators to zone air (tested in HEP, IDRIJA school and sports centre and STREM school pilot buildings within 3Smart);

Z.I.3 -- submodule for interfacing commands of heating energy from floor heating and cooling units to zone air (tested in STREM retirement and care centre pilot building within 3Smart).

In the following chapters the mentioned submodules are presented with their interface tables showing which data they use as inputs and which data they provide as outputs to be at the disposal to other submodules or to be used for building actuators in zones. Detailed explanations of algorithms behind each of the submodules are provided in the previously delivered 3Smart document D4.4.1 (related to interfacing on the zone level) which is further updated based on feedback operation from pilots and provided here as Annex 1.

Source and sink for the data used by the submodules is a properly structured 3Smart database. Its structure in the part concerned by the zone level interfacing submodules is provided in Annex 2.

2 Z.I.1 submodule

Z.I.1 submodule is used for interfacing commands of heating energy from fan coils to zone air. Within 3Smart it is tested in UNIZGFER, HEP, STREM school and EPHZHB pilot buildings.

The submodule interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Required inputs for the fan coils energy input control submodule

Variable name	Notation	Description
Energy input references for fan coils (one or several, depending whether more rooms are handled at once for coordination reasons)	$E_{\mathrm{ref}}^{\mathrm{T}}$ (noted herein as $u_{t\|t}^{*}$)	Energy input command for fan coils that needs to be followed, computed by MPC module on the zone level
Parameters of the **simplified building thermal dynamics model** (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{room}, B_{room}, C_{room}, D_{room}$ (noted herein as $A_b, [B_d, B_u]$ C_b, D_b)	Model obtained through the identification procedure above

Parameters of the **fan coil model** that relates fan coil actuation, room temperature and medium conditions registered on a calorimeter to fan coil energy transmitted to room air in a defined time period (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{\mathrm{fc}}(Q_{\mathrm{w}})$, $B_{\mathrm{fc}}(Q_{\mathrm{w}})$, $C_{\mathrm{fc}}(Q_{\mathrm{w}})$, $D_{\mathrm{fc}}(Q_{\mathrm{w}})$	Model obtained through the identification procedure
Current **setpoint temperature** / comfort setpoint (one or several, depending whether more rooms are handled at once for coordination reasons)	SP_0	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Setpoint used for the particular time and zone/zones in last MPC computation of the required thermal input from heating/cooling elements	$\mathrm{SP}_{\mathrm{MPC}}$	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Currently estimated heat disturbance for the zone (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^{D} (noted herein as $E_{\mathrm{d},0}$)	Needed to correct the required thermal heat input from actuators if the estimated disturbance has changed from the time of MPC computation
Heat disturbance for the zone used by the central zone MPC algorithm	$E_{d,t\|t}^{\mathrm{CMPC}}$	Needed to correct the required thermal heat input from actuators when heat disturbance for the zone used by the central MPC algorithm deviates from the real currently estimated heat disturbance.

Table 2.2: Outputs of the fan coils energy input control submodule

Variable name	Notation	Description
Computed current **commands to fan coils actuators**	FS_0 (can be also fan coil valve command if both can be actuated)	To be applied to the fan coil / fan coils
Computed **future planned actuations of the fan coils actuators**	FS	Possibly needed to better estimate electricity load and heating/cooling profile in the building in near future

3 Z.I.2 submodule

Z.I.2 submodule is used for interfacing commands of heating energy from radiators to zone air. Within 3Smart it is tested in UNIZGFER, IDRIJA school and sports centre and STREM school pilot buildings.

The submodule interface is defined in Table 3.1 and Table 3.2.

Table 3.1: Required inputs for the radiators energy input control submodule

Radiators energy input control submodule		
Frequency of submodule calls: every minute		
Variable name	**Variable annotation**	**Variable description**
Submodule inputs		
Energy input references for radiators (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^T	Energy input command for radiators that needs to be followed, computed by MPC module on the zone level
Parameters of the simplified building thermal dynamics model (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{room}, B_{room}, C_{room}, D_{room}$	Model obtained through the identification procedure above
Parameters of the radiators model that relates radiators actuation, room temperature and medium conditions registered on a calorimeter to radiators energy transmitted to room air in a defined time period (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{rad}\ (V_x)$ $B_{rad}\ (V_x)$ $C_{rad}\ (V_x)$ $D_{rad}\ (V_x)$	Model obtained through the identification procedure
Current setpoint temperature / comfort setpoint (one or several, depending whether more rooms are handled at once for coordination reasons)	SP_0	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Setpoint used for the particular time and zone/zones in last MPC computation of the required thermal input from heating/cooling elements	SP_{MPC}	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Currently estimated heat disturbance for the zone (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^D	Needed to correct the required thermal heat input from actuators if the estimated disturbance has changed from the time of MPC computation

Heat disturbance for the zone used by the central zone MPC algorithm	$E_{d,t\|t}^{\text{CMPC}}$	Needed to correct the required thermal heat input from actuators when heat disturbance for the zone used by the central MPC algorithm deviates from the real currently estimated heat disturbance.

Table 3.2: Outputs of the radiators energy input control submodule

Submodule outputs		
Computed current commands to radiators actuators	V_{x0}	Command to be applied to the valve/valves of radiators (one or several, depending whether more rooms are handled at once for coordination reasons) For 0-1 open-close valves the submodule computes the time within the 15-minute interval when the valve actuation should change from the current actuation. For valves whose position is continuously controllable it is computed the minimal change in actuation needed in order to accomplish the required thermal energy input from the radiator element.

4 Z.I.3 submodule

Z.I.3 submodule is used for interfacing commands of heating energy from floor heating and cooling units to zone air. Within 3Smart it is tested in STREM retirement and care centre pilot building.

The submodule interface is defined in **Table 4.1** and **Table 4.2**.

Table 4.1. Required inputs for the floor heating energy input control submodule

Variable name	Notation	Description
Energy input references for floor heating/cooling (one or several, depending whether	$E_{\text{ref}}^{\text{T}}$ (noted herein as $u_{t\|t}^{*}$)	Energy input command for floor heating/cooling that needs to be followed, computed by MPC

more rooms are handled at once for coordination reasons)		module on the zone level
Parameters of the **simplified building thermal dynamics model** (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{room}, B_{room}, C_{room}, D_{room}$ (noted herein as $A_b, [B_d, B_u]$ C_b, D_b)	Model obtained through the identification procedure of the building.
Parameters of the **floor heating model** that relates valve actuation, room temperature and medium conditions registered on a calorimeter to heating/cooling energy transmitted to room air in a defined time period (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{\mathrm{fhc}}(V_{\mathrm{x}})$, $B_{\mathrm{fhc}}(V_{\mathrm{x}})$, $C_{\mathrm{fhc}}(V_{\mathrm{x}})$, $D_{\mathrm{fhc}}(V_{\mathrm{x}})$	Model obtained through the identification procedure of the floor heating/cooling units.
Current **set point temperature** / comfort set point (one or several, depending whether more rooms are handled at once for coordination reasons)	SP_0	Needed to check whether the user has changed a set point in order to quickly adapt to the new set point (on a sampling time lower than the sampling time of MPC)
Set point used for the particular time and zone/zones in last MPC computation of the required thermal input from heating/cooling elements	$\mathrm{SP}_{\mathrm{MPC}}$	Needed to check whether the user has changed a set point in order to quickly adapt to the new set point (on a sampling time lower than the sampling time of MPC)
Currently estimated heat disturbance for the zone (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^{D} (noted herein as $E_{\mathrm{d},0}$)	Needed to correct the required thermal heat input from actuators if the estimated disturbance has changed from the time of MPC computation
Heat disturbance for the zone used by the central zone MPC algorithm	$E_{d,t\|t}^{\mathrm{CMPC}}$	Needed to correct the required thermal heat input from actuators when heat disturbance for the zone used by the central MPC algorithm deviates from the real currently estimated heat disturbance.

Table 4.2: Outputs of the floor heating/cooling energy input control submodule

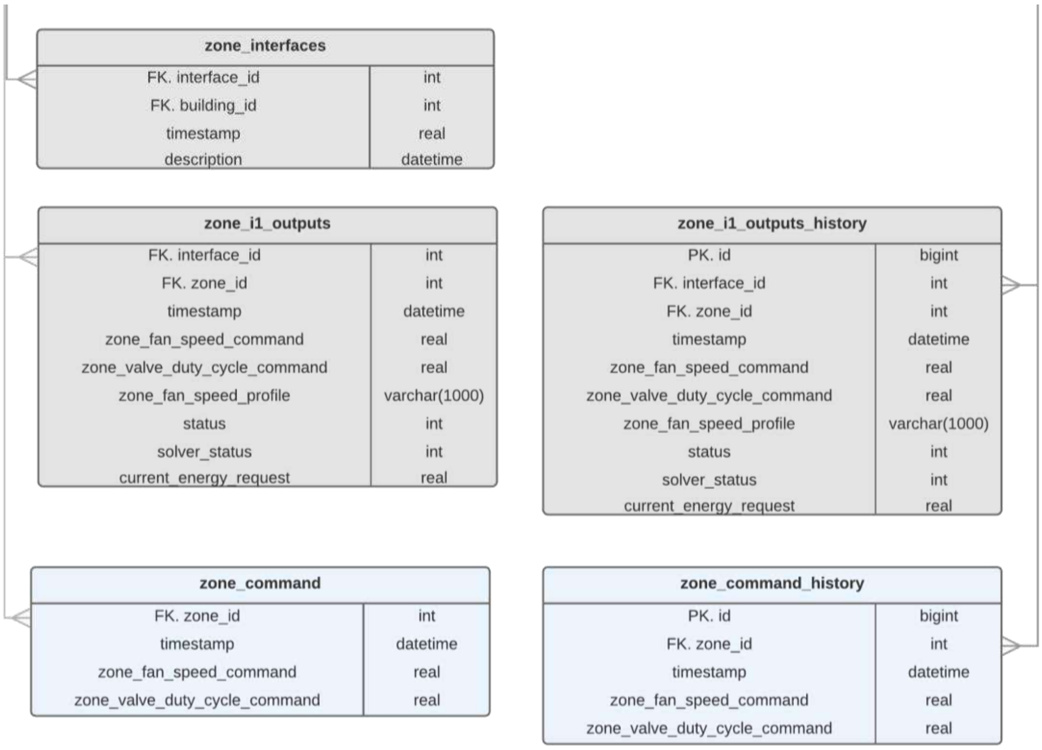
Submodule outputs		
Computed current commands	V_{x0}	Command to be applied to the

to valve actuator		valve of the floor heating/cooling units (one or several, depending whether more rooms are handled at once for coordination reasons) For valves whose position is continuously controllable (only such exist on the project) it is computed the minimal change in actuation needed in order to accomplish the required thermal energy input from the floor heating/cooling element within the 15 minutes period.

Annex 1 – Open software module for zone consumption management – Zone interfacing submodules

Provided as a separate document.

Annex 2 – 3Smart database organization for open software module for zone consumption management – Zone interfacing submodules



Project Deliverable Report
Smart Building – Smart Grid – Smart City
http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 INTERFACES IN ZONES

Open software module for zones consumption management – Zone interfacing submodules

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Mostar Faculty of Mechanical Engineering, Computing and Electrical Engineering (SVEMOFSR)
Authors (Partners)	Anita Martinčević, Mario Vašak, Vinko Lešić (UNIZGFER), Petar Marić, Ivan Bevanda (SVEMOFSR)
Contact person	Anita Martinčević (UNIZGFER)
Abstract (for dissemination)	This annex focusses on implementation of the energy management in building zones with fan coil units, radiators and floor heating/cooling units. The implementation is based on direct control of the required thermal energy inputs via available actuators on the heating/cooling elements like fans or valves. It is a necessary add-on to model predictive control algorithms that use thermal models of a building and compute the optimal energy inputs in different building zones for minimization of the overall energy cost required to keep the thermal comfort.

Revision history

Revision	Date	Description	Author (Organization)
v0.1	14 November 2017	First draft	Anita Martinčević (UNIZGFER)
v0.1	15 November 2017	First draft	Petar Marić (SVEMOFSR)
v0.1	15 November 2017	First draft	Ivan Bevanda (SVEMOFSR)
v1.0	31 December 2017	Finalized the corresponding D4.4.1 document	Petar Marić, Ivan Bevanda (SVEMOFSR), Mario Vašak (UNIZGFER)
V1.1	15 January 2019	Updated version based on feedback from pilots operation	Mario Vašak, Anita Martinčević (UNIZGFER)
V1.2	7 June 2019	Updated version based on feedback from pilots	Mario Vašak, Anita Martinčević (UNIZGFER)
V1.3	26 June 2019	Updated version based on feedback from pilots	Mario Vašak, Anita Martinčević (UNIZGFER), Ivan Bevanda, Petar Marić (SVEMOFSR)
v2.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

This annex to D4.5.3 is focused on implementation of the energy management in building zones with fan coil units, radiators or floor heating. The implementation is based on direct control of the required thermal energy inputs via available actuators on elements like fans or valves. It is developed as a necessary add-on to model predictive control algorithms that use thermal models of a building and compute the optimal energy inputs in different building zones for minimization of the overall energy cost required to keep the thermal comfort. The presented approach is substantially different to the generally accepted temperature control via local reactive control that only uses local measurements in deciding the control actions for zones thermal actuators. We demonstrate that closed loop performance, spent energy and the users comfort, is significantly improved over conventional control approaches. The model of a fan coil unit is identified and validated based on real data gathered from the living lab of University of Zagreb Faculty of Electrical Engineering and Computing. This model is then used for the subsequent control development. Similar approach is also followed for radiators and floor heating/cooling units.

1 Submodule for fan coils module interface (Z.I.1)

1.1 Introduction

Minimization of energy dissipation in buildings, due to the relatively large share of their consumption in the overall global consumption [1], [2], has been set as one of the main priorities for improvement of the building sector energy efficiency. Due to a relatively long life span of buildings, it is essential to make significant efforts to increase the energy efficiency of the existing Heating, Ventilation and Air Conditioning (HVAC) systems, i.e., to reduce energy consumption, cut down the maintenance cost and improve comfort conditions. The way most building HVAC systems are operated is undoubtedly one of the most prominent sources of unnecessary energy dissipation. The recent studies have shown that building consumption can be reduced up to 30% just by ensuring proper operation of its subsystems [3], [4].

Due to their improved performance over classic radiators, Fan Coil Units (FCUs) are widely used for localized heating and cooling. Traditional FCU control strategies include fuzzy, hysteresis and PID controllers [5], [6]. All the mentioned control approaches switch between fan speeds based on temperature difference between setpoint value and current zone temperature. This mostly results with zone temperature oscillations of constant, predefined amplitude, leading to unnecessary energy consumption. Energy Management System (EMS) that acts by adjusting the optimal temperature setpoint values usually neglects the performance of the local thermal actuators in the zone. Such systems are also practically inapplicable in tight comfort bounds with simple hysteresis controllers used locally for FCU control.

In this deliverable an MPC-based energy management of thermal actuators (heating/cooling devices) in zones through direct control of the required thermal energy inputs is developed. As such, it is an extension to Centralized MPC (CMPC) algorithms that calculate optimal thermal energy profiles per zones based on thermodynamic model of the building [7], [8]. The developed algorithm is a link between the commanded variables from the CMPC and the actual actuation profile on heating/cooling devices required for these commands to be realized. The significance of the proposed approach is a direct control of thermal energy inputs per zone rather than generally accepted temperature control [9]. By doing so, a high level of modularity and flexibility for different types of thermal actuators is gained, offering a fast replicability of the method, and this is also demonstrated by extending the principle to radiators and floor heating/cooling elements.

1.2 Interface for the fan coils interfacing module

Table 1.1.1: Required inputs for the fan coils energy input control submodule

Variable name	Notation	Description
Energy input references for fan coils (one or several, depending whether more rooms are handled at once for coordination reasons)	$E_{\mathrm{ref}}^{\mathrm{T}}$ (noted herein as $u_{t\|t}^{*}$)	Energy input command for fan coils that needs to be followed, computed by MPC module on the zone level
Parameters of the **simplified**	$A_{room}, B_{room},$	Model obtained through the

building thermal dynamics model (one or several, depending whether more rooms are handled at once for coordination reasons)	C_{room}, D_{room} (noted herein as $A_b, [B_d, B_u]$ C_b, D_b)	identification procedure above
Parameters of the **fan coil model** that relates fan coil actuation, room temperature and medium conditions registered on a calorimeter to fan coil energy transmitted to room air in a defined time period (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{\mathrm{fc}}(Q_{\mathrm{w}})$, $B_{\mathrm{fc}}(Q_{\mathrm{w}})$, $C_{\mathrm{fc}}(Q_{\mathrm{w}})$, $D_{\mathrm{fc}}(Q_{\mathrm{w}})$	Model obtained through the identification procedure above
Current **setpoint temperature** / comfort setpoint (one or several, depending whether more rooms are handled at once for coordination reasons)	SP_0	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Setpoint used for the particular time and zone/zones in last MPC computation of the required thermal input from heating/cooling elements	$\mathrm{SP}_{\mathrm{MPC}}$	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Currently estimated heat disturbance for the zone (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^{D} (noted herein as $E_{\mathrm{d},0}$)	Needed to correct the required thermal heat input from actuators if the estimated disturbance has changed from the time of MPC computation
Heat disturbance for the zone used by the central zone MPC algorithm	$E_{d,t\|t}^{\mathrm{CMPC}}$	Needed to correct the required thermal heat input from actuators when heat disturbance for the zone used by the central MPC algorithm deviates from the real currently estimated heat disturbance.

Table 1.1.2: Outputs of the fan coils energy input control submodule

Variable name	Notation	Description
Computed current **commands to fan coils actuators**	FS_0 (can be also fan coil valve command if both can be actuated)	To be applied to the fan coil / fan coils
Computed **future planned actuations of the fan coils actuators**	FS	Possibly needed to better estimate electricity load and heating/cooling profile in the building in near future

1.3 Methodology

The proposed zone level optimal energy management consists of two hierarchical control levels. Higher optimization level consists of CMPC for calculation of optimal thermal energy profile per zone for all observed zones. Lower hierarchical level consists of locally-distributed controllers (LMPC), one per each zone, employed to control thermal actuators in an optimal way by respecting the commands given by the higher control level (Figure 1.3.1). A more complex control configuration with single low level controller and zones organized in groups with a goal to smoothen the heat demand for the group can be found in [11]. Here we focus on a fully decentralized configuration that enables adherence to the thermal energy commands from the higher level.

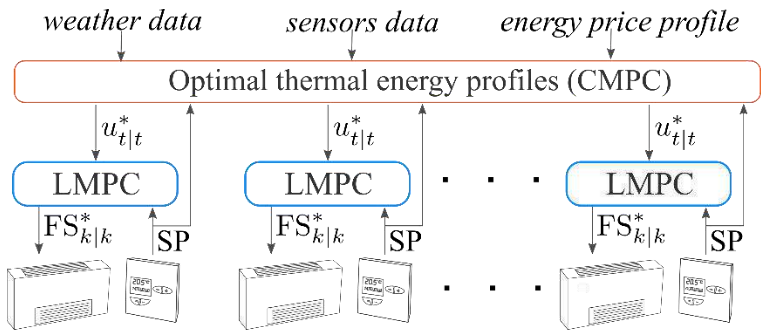


Figure 1.1.1: Hierarchical MPC for zones comfort control.

The CMPC algorithm for assessment of optimal thermal energies per zones is developed in [7]. It has been shown that the derived optimal control formulation outperforms classic zone temperature control algorithms both in energy consumption and achieved comfort, even for very strict comfort constraints. The nature of thermal actuators in the referenced study was intentionally left out to estimate the upper limit of energy savings achievable by implementation of such a set-up. In terms of building climate control, the CMPC calculates an optimal plan of heating and/or cooling with sampling time T_s^c for all included zones based on weather prediction, disturbance prediction, energy price prediction (in cases with volatile prices) and constraints such as temperature constraint or physical limitations of thermal actuators. The first control action per zone $u^*_{t|t}$ is then forwarded to low level controllers and the procedure is repeated at the next CMPC time step.

The goals of LMPC energy management of FCU are (*i*) to ensure that the temperature profile remains in the comfort limits, (*ii*) to assure realization of energy input set by the CMPC and (*iii*) to guarantee the minimal disruption of the users and the minimum energy consumed by the fan by preferring lower fan speeds and minimizing the number of fan speed switching. To accomplish all the goals, LMPC needs to operate on significantly lower time scales than the ones used for assessment of optimal thermal energy flows into the zone. It turns out that a minute scale of T_s is a good choice for reasonable data transfer between the FCU and the central control/data acquisition system and low enough scale for reducing the zone temperature oscillations, which are unavoidable in FCU operation, especially for FCUs without the possibility of the medium mass flow control. At the beginning of every T_s^c-long time-interval, LMPC receives an energy command from the CMPC $u^*_{t|t}$ and distributes it into $H = T_s^c / T_s$ equal shares. After receiving the energy command, LMPC calculates an optimal fan speed trajectory N steps into the future at every T_s:

$$\mathrm{FS}_k^* = \left[\mathrm{FS}_{k|k}^* \quad \mathrm{FS}_{k+1|k}^* \quad \dots \quad \mathrm{FS}_{k+(N-1)|k}^*\right], \tag{1-1}$$

where $N \leq H$ is the control horizon length of LMPC controller, and notation $\mathrm{FS}_{k+1|k}^*$ stands for predicted optimal fan speed at time step $k+1$ calculated at time step k. In accordance with receding horizon principle, only the first control action $\mathrm{FS}_{k|k}^*$ is forwarded to the FCU and procedure is repeated at the next time interval. If the prediction horizon is outside the interval $[t, \quad t+T_s^c]$, the horizon is shortened so that $k+N \leq H$ is satisfied. To be able to predict future FCU behaviour, the FCU model identified in the first part of deliverable D441 concerning identification is discretized with sampling time $T_d = 1$ s to preserve the model accuracy. The resulting discrete-time system equations are:

$$T_{\mathrm{w}}^{\mathrm{out}}(k+1) = (A_d)^M T_{\mathrm{w}}^{\mathrm{out}}(k) + \sum_{j=0}^{M-1}(A_d^j B_d)\begin{bmatrix} T_{\mathrm{w}}^{\mathrm{in}}(k) \\ T_{\mathrm{a}}^{\mathrm{in}}(k)\end{bmatrix}, \tag{1-2}$$

where A_d and B_d are discrete-time counterparts of continous-time fan coil unit model [16] and $M = T_s/T_d$. Analogously, energy inserted into the zone within the time interval $[k \quad k+1]T_s$ is defined as:

$$E_{\mathrm{t},k} = \left(T_d \sum_{j=0}^{M-1} C_d A_d^j\right) T_{\mathrm{w},k}^{\mathrm{out}} + \left(D_d T_s + C_d \sum_{j=0}^{M-1} \sum_{l=0}^{j-1} A_d^l B_d T_d\right)\begin{bmatrix} T_{\mathrm{w},k}^{\mathrm{in}} \\ T_{\mathrm{a},k}^{\mathrm{in}}\end{bmatrix}, \tag{1-3}$$

where C_d and D_d are discrete-time counterparts of the same matrices of continuous time model.

Goal (*i*)

Zone dynamics can be described with linear state space model of the following form [16]:

$$\begin{gathered} x_k = A_b x_{k-1} + B_u u_{k-1} + B_b d_{k-1}, \\ T_{\mathrm{a},k}^{\mathrm{in}} = C_b x_k \end{gathered} \tag{1-4}$$

where $x_k \in \mathbb{R}^n$ is the system state vector, $u_k \in \mathbb{R}^1$ is the thermal energy input, $d_k \in \mathbb{R}^p$ is the disturbance input (outdoor temperature, solar irradiance, internal gains, temperatures of neighboring rooms, etc.). Matrices A_b, B_u, B_b and C_b are of appropriate dimensions and are obtained either based on first principles modeling or by use of identification methods [12],[13],[14]. To limit the zone temperature oscillations and to enforce the temperature trajectory to be within a user-defined interval the following constraints need to be respected:

$$\mathrm{SP}_k - \Delta_k \leq T_{\mathrm{a},k+i|k}^{\mathrm{in}} \leq \mathrm{SP}_k + \Delta_k, \qquad \forall\, i = 1, \dots, N, \tag{1-5}$$

where Δ_k is the allowed deviation from SP_k at time $t + kT_s$. If set point SP_k changes within the CMPC calculation interval Δ_k are automatically set to 0 on the rest of the CMPC interval.

Goal (*ii*)

To assure realization of the energy request set by the CMPC, difference between the realized and requested energy so far:

$$\mathit{\Delta}\, E_{\mathrm{t},k} = \frac{k}{H} u^*_{t|t} - \int_t^{t+kT_s} P^*_{\mathrm{t}} dt + \int_t^{t+kT_s} (E^{\mathrm{CMPC}}_{d,t|t} - E_{\mathrm{d}}) dt\,, \tag{1-6}$$

is calculated at every time step $k \geq 1$ as:

$$\mathit{\Delta}\, E_{\mathrm{t},k} = \mathit{\Delta}\, E_{\mathrm{t},k-1} + \left(\frac{u^*_{t|t}}{H} - \int_{t+(k-1)T_s}^{t+kT_s} P^*_{\mathrm{t}} dt + (E^{\mathrm{CMPC}}_{d,t|t} - E_{d,k}) T_s \right), \tag{1-7}$$

where P^*_{t} is the thermal power from FCU into zone calculated from the FCU measurements [16], $E^{\mathrm{CMPC}}_{d,t|t}$ is predicted disturbance heat flux used in the CMPC calculation and $E_{d,k}$ is currently estimated heat flux which may differ from the one used in CMPC calculation. The realization of the energy requested by the CMPC is then enforced by minimizing the difference between energy to be realized with FCU $E_{\mathrm{t},k+i|k}$ and the requested one increased for the backlogs defined with (3-6):

$$J_1(\mathrm{FS}) = \left| \left(\tfrac{N}{H} u^*_{t|t} + \mathit{\Delta}\, E_{\mathrm{t},k} \right) - \textstyle\sum_{i=0}^{N-1} E_{\mathrm{t},k+i|k} \right|. \tag{1-8}$$

Iterative update of $\mathit{\Delta}\, E_{\mathrm{t},k}$ assures the offset-free input energy control as it emulates the integrator behavior.

Goal (*iii*)

Although one FCU consumes a small amount of electricity when its fan is on (~50 W), due to the large number of FCUs in the whole building and long working hours, the total electric power consumption occupies a large share of central HVAC system electricity consumption. Therefore, optimizing the FCU performance improves thermal comfort but also potentially contributes to the electric energy savings:

$$J_2(\mathrm{FS}) = \textstyle\sum_{i=0}^{N-1} E_{\mathrm{el}}(\mathrm{FS}_{k+i|k}), \tag{1-9}$$

where E_{el} denotes electrical energy consumption of a FCU. By minimizing the electricity consumption, lower fan speeds are favoured thus minimizing also the noise. Since switching between fan speeds is the noisiest part of FCU operation, the following penalty function is introduced to reduce it:

$$J_3(\mathrm{FS}) = \textstyle\sum_{i=0}^{N-1} \Delta^{\mathrm{FS}}_{k+i|k}, \tag{1-10}$$

$$\Delta_{k+i|k}^{\mathrm{FS}} = \begin{cases} 0, & \text{if} \quad \mathrm{FS}_{k+i|k} = \mathrm{FS}_{k+i-1|k} \\ 1, & \text{if} \quad \mathrm{FS}_{k+i|k} \neq \mathrm{FS}_{k+i-1|k} \end{cases}, \tag{1-11}$$

$\forall i = 0, \dots, N-1$, with $\mathrm{FS}_{k-1|k} = \mathrm{FS}^*_{k-1|k-1}$ where $\mathrm{FS}^*_{k-1|k-1}$ is the optimal fan speed calculated and applied to the FCU in the previous time step.

The final consolidated LMPC optimization problem for FCU control, written in compact form, is as follows:

$$\begin{aligned} \mathrm{FS}^*_{\mathrm{k}} = \mathrm{argmin}_{FS} \quad & J_1(\mathrm{FS}) + \sigma_1 J_2(\mathrm{FS}) + \sigma_2 J_3(\mathrm{FS}) \\ \text{s.t.} \quad & (3-2), (3-3), (3-4), (3-5) \\ & \mathrm{FS}_{k-1|k} \in \{0,\mathrm{L},\mathrm{M},\mathrm{H}\} \;\; \forall i = 0,1, \dots N-1 \end{aligned}, \tag{1-12}$$

whereas, to enable implementation, constraints defined with $(1-5)$ are included as soft constraints. The preferred behaviour is enforced by changing the weights denoted with σ. The optimization problem (3-12) belongs to a class of Mixed Integer Linear Programs (MILPs) which can be efficiently solved with e.g. CPLEX [15]. The overall algorithm for the MPC energy management of FCUs is given in Algorithm 1.

Algorithm 1: The LMPC algorithm for a FCU control

```
collect new measurements Q_w, T_w^in, T_a^in, T_w^out
check for SP and Δ updates,
if SP has been changed within the one CMPC interval then
    set Δ to zero on the rest of the CMPC interval
end if
if k = H then
    receive u_{t|t} from CMPC,
    initialize k = 0, ΔE_k = 0,
else
    if k + N > H then
        reduce control horizon to N = H − k,
    end if
    update ΔE_k via (1-7)
end if
solve the FCU optimization problem (1-12),
Forward FS*_{k|k} to the FCU,
k = k + 1;
```

1.4 Results

The Algorithm 1 is realized and tested first within MATLAB environment [10], before implementation on different pilot buildings. Data used as external conditions for dynamic building simulation are data from 13th to 20th March 2014 gathered on meteorological station close to the UNIZGFER's 3Smart pilot building. The weather conditions in the selected week (Figure 1.4.1) are chosen as representative variable conditions.

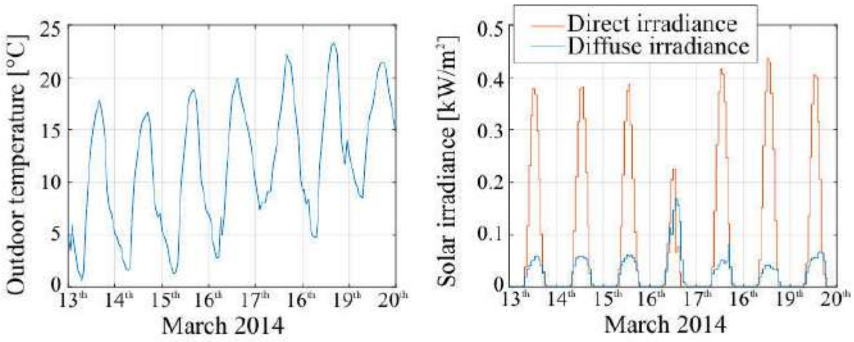


Figure 1.1.1: Weather conditions from 13th to 20th March 2014.

All disturbances on CMPC level are assumed to be perfectly known. Simulations are performed with the following parameters: $\mathrm{SP} = 24$ °C, $\Delta = 0.5$ °C, $N = 10$, $T_s^c = 3600$ s and $T_s = 60$ s. The temperature is regulated only during working hours, from 6:00 to 18:00. Figure 4.2 shows a comparison of temperatures for a typical south-oriented office equipped with FCC06 and the FCU thermal consumption for the three control approaches: continuous hysteresis control and two approaches based on CMPC, the LMPC and the idealized algorithm with uniform power tracking of thermal energy references. For a fair comparison, the hysteresis controller is switched on at 5:00 to meet the requirements of working hours in time.

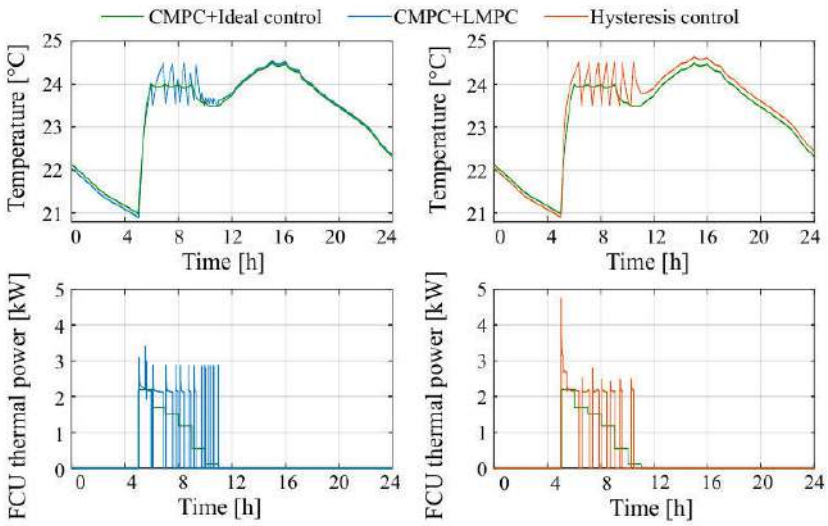


Figure 1.1.2: Zone temperature response and thermal FCU power with different types of control and $\boldsymbol{\sigma_1} = \boldsymbol{\sigma_2} = \mathbf{100}$ for March 17th, 2014.

Preheating, as a well-known advantage of the MPC, suppresses the need for instant zone heating and thus flattens the energy consumption profile by reducing the peak power loads. Figure 1.4.3 gives performance comparison of the developed control algorithm and the hysteresis one for the selected period in terms of the objective goals J_2 and J_3, overall thermal energy consumption, and average deviation from SP for different weights σ_1 and σ_2.

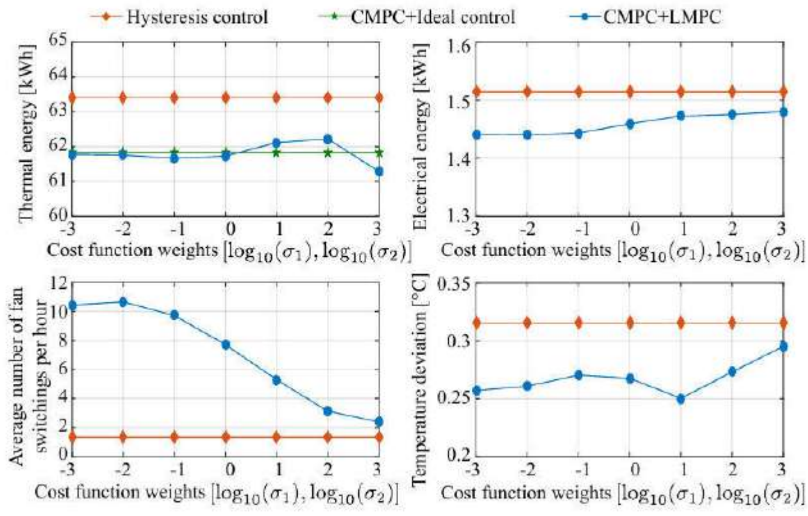


Figure 1.1.3: Performance comparison of the developed control algorithms for the period from 13th to 20th March 2014.

For appropriately selected weights the developed algorithm outperforms the hysteresis one both in energy consumption and comfort with average number of switching per hour within the acceptable range.

Fan coil units (FCUs), due to their inherent non-linearity and limited choice of fan speeds, represent a serious challenge for implementation of real-time offset-free MPC that ensures adherence to required thermal energy inputs and comfort constraints. Despite complexity, energy management based on adherence to the commanded thermal energies, opens the space for cooperation between the building zone level and other building subsystems (e.g., smart grid or central HVAC system) through communication of energy consumptions and internal prices for it. This also enables the coordination of all observed building subsystems and shifting the possible energy cost saving percentage beyond the sum of the individual saving potentials.

2 Submodule for floor heating/cooling module interface (Z.I.3)

2.1 Introduction

During the past decade, the various attempts and studies have been conducted in order to minimize the overall energy consumption in the residential and office buildings. These attempts came as a result of a relatively high costs of energy demand in buildings over a long life span.

In this deliverable the model predictive control approach will be presented for the actuation and usage of the floor heating/cooling system units for the purposes of cutting the energy demands in residential objects. This approach can be seen as an extension to central model predictive algorithms that calculate thermal energy profiles based on the thermodynamical model of a building. The main advantage of this methodology is direct control of the thermal energies in the building which is more effective from the temperature control of separate zones or rooms.

2.2 Module interface

Table 2.2.1. Required inputs for the floor heating energy input control submodule

Variable name	Notation	Description
Energy input references for floor heating/cooling (one or several, depending whether more rooms are handled at once for coordination reasons)	$E_{\mathrm{ref}}^{\mathrm{T}}$ (noted herein as $u_{t\|t}^{*}$)	Energy input command for floor heating/cooling that needs to be followed, computed by MPC module on the zone level
Parameters of the **simplified building thermal dynamics model** (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{room}, B_{room}, C_{room}, D_{room}$ (noted herein as $A_b, [B_d, B_u]$ C_b, D_b)	Model obtained through the identification procedure of the building.
Parameters of the **floor heating model** that relates valve actuation, room temperature and medium conditions registered on a calorimeter to heating/cooling energy transmitted to room air in a defined time period (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{\mathrm{fhc}}(V_{\mathrm{x}})$, $B_{\mathrm{fhc}}(V_{\mathrm{x}})$, $C_{\mathrm{fhc}}(V_{\mathrm{x}})$, $D_{\mathrm{fhc}}(V_{\mathrm{x}})$	Model obtained through the identification procedure of the floor heating/cooling unis system.
Current **set point temperature** / comfort set point (one or several, depending whether more rooms are handled at once for coordination reasons)	SP_0	Needed to check whether the user has changed a set point in order to quickly adapt to the new set point (on a sampling time lower than the sampling time of MPC)
Set point used for the particular time and zone/zones in last MPC computation of the required thermal input from heating/cooling elements	$\mathrm{SP}_{\mathrm{MPC}}$	Needed to check whether the user has changed a set point in order to quickly adapt to the new set point (on a sampling time lower than the sampling time of MPC)
Currently estimated heat disturbance for the zone (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^{D} (noted herein as $E_{\mathrm{d},0}$)	Needed to correct the required thermal heat input from actuators if the estimated disturbance has changed from the time of MPC computation
Heat disturbance for the zone used by the central zone MPC algorithm	$E_{d,t\|t}^{\mathrm{CMPC}}$	Needed to correct the required thermal heat input from actuators when heat disturbance for the zone used by the central MPC algorithm deviates from the real currently estimated heat

		disturbance.

Table 2.2.2: Outputs of the floor heating/cooling energy input control submodule

Submodule outputs		
Computed current commands to valve actuators	V_{x0}	Command to be applied to the valve/valves of floor heating/cooling (one or several, depending whether more rooms are handled at once for coordination reasons)

2.3 MPC approach and methodology

The main idea behind this model predictive control approach for the particular zone in the building is to follow the central model predictive control that calculates the optimal thermal energies per zone for the whole building. In other words our locally distributed MPC controllers are used as a relationship between the commanded variables from the central MPC controller and thermal actuation profile on a floor heating/cooling system. If our central MPC controller sets a command to generate a particular quantity of thermal energy into the zone, our local MPC controller needs to find the most effective way to set the floor heating/cooling valves needed to generate the commanded thermal energy.

When designing the local MPC controller it is crucial that it needs to operate on a scale that is lower than a scale of a central MPC controller. The problem that can happen with the floor heating/cooling system is related to the slow dynamic properties of a system. In the case of floor heating/cooling the process is relatively long in comparison with the quarter-hourly scale of the central MPC controller. The idea is to set the scale of the local MPC controller to 15 minutes, same as the hourly scale of the central MPC.

In the deliverable for the identification of the floor heating/cooling unit system the mathematical model of the floor heating/cooling system has been developed based on the inputs such as inlet water temperature, outgoing water temperature, valve actuation and air temperature in the room. The output of the mathematical model is given in a form of thermal power affecting the zone (room).

The algorithm for the local MPC controller needs to predict the future system behaviour based on the given mathematical model. The control actions for the floor heating/cooling have to be decided upon the predicted behaviour of the system.

There are some crucial tasks that we expect from our local MPC controller to satisfy. The first condition to fulfil is to maintain the temperature inside the temperature comfort limits which are given as an input from the end user (3-1).

$$SP_k - \Delta_k \leq T_{a,k+i}^{i} \leq SP_k + \Delta_k \tag{2-1}$$

Second condition for locally distributed MPC controller is to follow the energy profile requested by the central MPC controller during its hourly scale. It would be ideal situation to follow the energy profile reference perfectly.

The central MPC controller should be aware what is the minimal and maximal amount of energy that floor heating/cooling system can transmit in to the zone in one central MPC interval.

At the beginning of 15 minute interval the central MPC should set the energy profile reference according to the energy limitations of the floor heating/cooling system. Since the valves installed on to the pipes of the system are PWM actuators, local MPC controller will calculate the best possible degree of valve actuation for the following 15 minutes (central MPC interval). This calculation will be carried out based on the mathematical model of the floor heating/cooling system.

The mathematical model of the floor heating/cooling system will first be calculated for the medium mass flow that is at 50 % of the maximal mass flow that can be used within the system based on the equation (3-2).

$$m_w c_w \frac{dT_{w,out}}{dt} = \begin{cases} Q_w c_w (T_{w,in} - T_{w,out}) - U_0 (T_{w,out} - T_{air})^n & Q_w \neq 0, \\ U_0 (T_{w,out} - T_{air})^n & Q_w = 0, \end{cases} \tag{2-2}$$

Then the predicted energy transmitted into the zone is going to be calculated based on the equation (3-3).

$$\int_0^{Td} U_0 \left(T_{w,out} - T_{air}\right)^n dt. \tag{2-3}$$

The local MPC algorithm should in the first moment as it gets the energy profile command from central MPC controller compare it with the energy profile predicted for the interval with the valve opening set at 50 % of the maximum. Based on the comparison the algorithm is going to decide whether the valve opening is going to get into the interval (50%-100%) or (0%-50%). Depending on the chosen interval of medium mass flow percentage the best possible command is going to be calculated so that the predicted energy profile transmitted into the zone is as close as possible to the one requested from the central MPC controller. So initially the interval [0%-100%] is halved and then the counter goes through the first [0%-50%] or the second [50%-100%] to find the optimal value of the valve openness.

After the value of the valve openness for the following 15 minutes has been calculated in the first few second of the CMPC interval the local MPC algorithm is going to send the actuation command to the valve. In order to avoid potential disturbances to the energy profile in the zone the feedback loop is going to be used in the local MPC algorithm. This feedback loop is going to be called every minute so the algorithm will have the possibility to send 15 different valve actuation commands during one central MPC interval.

3 Submodule for radiators module interface (Z.I.2)

3.1 Introduction

Panel radiators are water to air heat exchangers, designed to satisfy requirements of heating demand typically for rooms. The radiator heats up the surrounding environment by following the mechanism of convection and radiation. Model predictive control of HVAC systems implies a group of advanced control methods that optimize the heating and/or cooling heating system parameters (usually every hour) for the selected time period (typically 24 hours), depending on predicted weather conditions and the way of using the building, primarily with the aim of reducing total consumption or energy costs while meeting the conditions of thermal comfort.

3.2 Module interface

Table 3.3.1: Required inputs for the radiators energy input control submodule

Radiators energy input control submodule		
Frequency of submodule calls: every minute		
Variable name	**Variable annotation**	**Variable description**
Submodule inputs		
Energy input references for radiators (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^T	Energy input command for radiators that needs to be followed, computed by MPC module on the zone level
Parameters of the simplified building thermal dynamics model (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{room}, B_{room}, C_{room}, D_{room}$	Model obtained through the identification procedure
Parameters of the radiators model that relates radiators actuation, room temperature and medium conditions registered on a calorimeter to radiators energy transmitted to room air in a defined time period (one or several, depending whether more rooms are handled at once for coordination reasons)	$A_{rad}(V_x)$ $B_{rad}(V_x)$ $C_{rad}(V_x)$ $D_{rad}(V_x)$	Model obtained through the identification procedure
Current setpoint temperature / comfort setpoint (one or several, depending whether more rooms are handled at once for coordination reasons)	SP_0	Needed to check whether the user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Setpoint used for the particular	SP_{MPC}	Needed to check whether the

time and zone/zones in last MPC computation of the required thermal input from heating/cooling elements		user has changed a setpoint in order to quickly adapt to the new setpoint (on a sampling time lower than the sampling time of MPC)
Currently estimated heat disturbance for the zone (one or several, depending whether more rooms are handled at once for coordination reasons)	E_0^D	Needed to correct the required thermal heat input from actuators if the estimated disturbance has changed from the time of MPC computation
Heat disturbance for the zone used by the central zone MPC algorithm	$E_{d,t\|t}^{\mathrm{CMPC}}$	Needed to correct the required thermal heat input from actuators when heat disturbance for the zone used by the central MPC algorithm deviates from the real currently estimated heat disturbance.

Table 3.3.2: Outputs of the radiators energy input control submodule

Submodule outputs		
Computed current commands to valve actuators	V_{x0}	Command to be applied to the valve/valves of radiators (one or several, depending whether more rooms are handled at once for coordination reasons)

3.3 Methodology

The proposed zone level optimal energy management consists of two hierarchical control levels. Higher optimization level consists of CMPC for calculation of optimal thermal energy profile per zone for all observed zones. Lower hierarchical level consists of locally-distributed controllers (LMPC), one per each zone, employed to control thermal actuators in an optimal way by respecting the commands given by the higher control level [1].

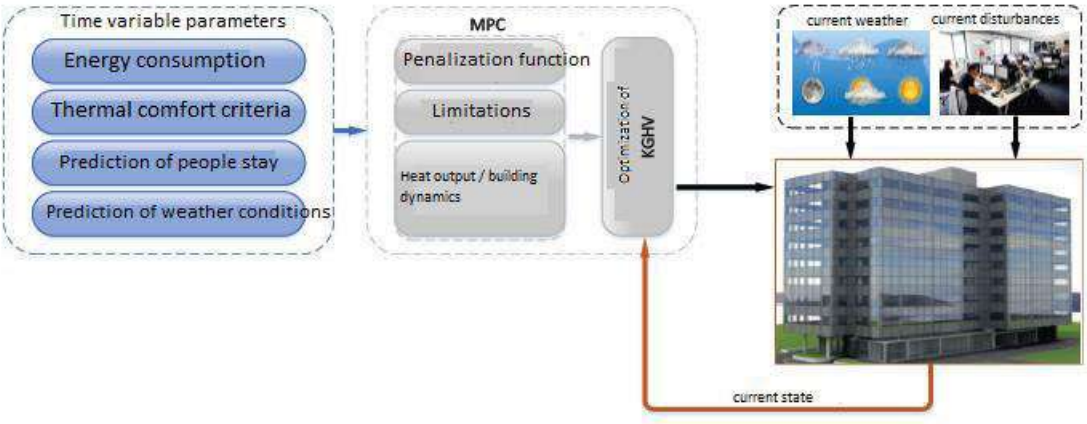


Figure 3.3.1: Concept of MPC

The algorithm for the LMPC needs to predict the future system behaviour based on the given mathematical model. The control actions for the radiators have to be decided upon the predicted behaviour of the system. The LMPC has to keep the zone temperature in the limits which are given by the user:

$$SP_k - \Delta_k \leq T_{a,k+i}^{i} \leq SP_k + \Delta_k \qquad (3-1)$$

Second condition for LMPC is to follow the energy profile requested by the CMPC during its 15 minute scale. The CMPC every 15 minutes calculates the needed energy profile for each zone and sends it to the LMPC. So at the beginning of every T_S^c-long time-interval, LMPC receives an energy command from the CMPC and the information about the valve position at the beginning of the interval (x_V= 0 or 1) from the sensor. LMPC needs to calculate how long the valve has to remain open or closed (regarding the valve position at the beginning) during the T_S^c-long time-interval to satisfy the energy request given by CMPC. The bisection method is used by LMPC in order to find the needed time. The interval is repeatedly being bisected and then the interval in which energy must lie for further processing is selected. The calculation continues until the appropriate time is found for which the valve has to stay in on or off position. LMPC operates on a minute time scale T_S= 1 min so in every minute during the 15 minutes interval it calculates how much energy has been transfered from the radiator to the zone since the beginning of the T_S^c-long time-interval. LMPC makes a subtraction of overall energy request and transfered energy until that moment and calculates how long the valve needs to remain open or closed to see if there is difference from the previous calculations during the interval. The overall algorithm for the MPC energy management of radiators is given in Algorithm 1.

It is expected that a narrow band of energies is feasible for the energy input control. Boundaries are represented with X_v=0 the entire sample and X_v=1 the entire sample while taking into account the current state of the radiator actuator.

Algorithm 1: The LMPC algorithm for the radiator energy input control

collect new measurements $Q_{\mathrm{w}}, T_{\mathrm{w}}^{\mathrm{in}}, T_{\mathrm{a}}^{\mathrm{in}}, T_{\mathrm{w}}^{\mathrm{out}}$,
define sensibility region (s.r.) around the energy input from the CMPC,

receive energy command from the CMPC, receive starting valve position (x_v= 0 or 1), set the first and the last moment of prediction horizon (t1=0 and t2=15 min)

if x_V==1

while ((t2-t1)>1) calculate predicted $T_{\mathrm{w}}^{\mathrm{out}}$ and energy $E_{\mathrm{t,rad}}$ transmitted in to the zone in the current CMPC 15-minutes and calculate $tint = \frac{t1+t2}{2}$ where Q_w=1 for the interval $[0, t_{int}]$ and Q_w=0 for the interval $[t_{int}, 15]$

if $E_{\mathrm{t,rad}} > energy\ command\ from\ CMPC + s.r.$ **then**
set $t2 = tint$,

else if $E_{\mathrm{t,fh}} < energy\ command\ from\ CMPC - s.r.$ **then**
set $t1 = tint$,

else
break

end if
end while

else

while ((t2-t1)>1) calculate predicted $T_{\mathrm{w}}^{\mathrm{out}}$ and energy $E_{\mathrm{t,rad}}$ transmitted in to the zone in the current CMPC 15-minutes and calculate $tint = \frac{t1+t2}{2}$ where Q_w=1 for the interval $[0, t_{int}]$ and Q_w=0 for the interval $[t_{int}, 15]$

if $E_{\mathrm{t,rad}} > energy\ command\ from\ CMPC + s.r.$ **then**
set $t1 = tint$,

else if $E_{\mathrm{t,fh}} < energy\ command\ from\ CMPC - s.r.$ **then**
set $t2 = tint$,

else
break

end if
end while
end if

Bibliography

[1] Renewable Energy Policy Network for the 21st Century (REN21). Renewables 2014, Global status report, 2014.

[2] Europe’s buildings under the microscope. Buildings Performance Institute Europe, 2011.

[3] National Renewable Energy Laboratory, Office of Energy Efficiency and Renewable Energy, Saving energy in commercial buildings, 2011.

[4] Carbon Trust, Heating, ventilation and air conditioning (HVAC) equipment - A guide to equipment eligible for enhanced capital allowances, 2012.

[5] Y. Li, J. Bian, N. Zhang, and J. Li, “Experimental study on intelligent control scheme for fan coil air-conditioning system,” Advances in Mechanical Engineering, vol. 2013, 2013.

[6] C.-M. Chu, T.-L. Jong, and Y.-W. Huang, “Thermal comfort control on multi-room fan coil unit system using lee-based fuzzy logic,” Energy Conversion and Management, vol. 46, no. 9-10, pp. 1579–1593, 2005.

[7] A. Martinčević, M. Vašak, and V. Lešić, “Model predictive control for energy-saving and comfortable temperature control in buildings,” in 24th Mediterranean Conference on Control and Automation, MED 2016, 2016, pp. 298–303.

[8] V. Lešić, A. Martinčević, and M. Vašak, “Modular energy cost optimization for buildings with integrated microgrid,” Applied Energy, vol. 197, pp. 14–28, 2017.

[9] D. Sturzenegger, D. Gyalistras, M. Morari, and R. Smith, “Model predictive climate control of a Swiss office building: Implementation, results, and cost-benefit analysis,” IEEE Transactions on Control Systems Technology, vol. 24, no. 1, pp. 1–12, 2016.

[10] MATLAB, version 8.5.0 (R2015a). The MathWorks Inc., 2015.

[11] A. Starčić, V. Lešić, and M. Vašak, “Predictive control for heating power variance and peak reduction in buildings,” in IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Oct 2016, pp. 7089–7094.

[12] A. Martinčević, A. Starčić, and M. Vašak, “Parameter estimation for low-order models of complex buildings,” in IEEE PES Innovative Smart Grid Technologies Conference Europe, Oct 2014, pp. 1–6.

[13] J. Seem, Modeling of heat transfer in buildings. University of Wisconsin–Madison, 1987.

[14] M. Vašak, A. Starčić, and A. Martinčević, “Model predictive control of heating and cooling in a family house,” in MIPRO 2011 - 34th International Convention on Information and Communication Technology, Electronics and Microelectronics - Proceedings, 2011, pp. 739–743.

[15] IBM, “ILOG CPLEX: High-performance software for mathematical programming and optimization,” 2013.

[16] 3Smart Deliverable 4.4.1 Open software module for zone consumption management – Estimation and Prediction

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Estimation and prediction submodules for central HVAC level management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Belgrade Faculty of Mechanical Engineering
Authors (Partners)	Hrvoje Novak, Mario Vašak (UNIZGFER), Vladimir Jovanović, Nebojša Manić, Mirko Komatina (UNIBGFME)
Contact person	Hrvoje Novak (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of estimation and prediction submodules on the central HVAC system level for hierarchical management of building subsystems. More detailed logic behind the individual modules is provided in the annex.
Keyword List	Heat Pump; Heat Exchanger; Circulation Pump; Mathematical Model; Estimation; Non-controllable Consumption; Prediction

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.3.1	Mario Vašak (UNIZGFER)
V0.2	15 January 2019	Updated version	Mario Vašak (UNIZGFER)
V0.3	7 June 2019	Updated version	Mario Vašak (UNIZGFER), Vladimir Jovanović, Nebojša Manić (UNIBGFME)
V0.4	26 June 2019	Updated version	Mario Vašak (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of submodules.

The submodules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on central HVAC system level prediction and estimation submodules.

Each submodule is presented via an interfacing table that explains what data are used by the submodules as inputs and what are the final output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following estimation and prediction submodules are designed, commissioned and tested on the central HVAC system level:

HVAC.PE.1 – submodule for estimation of a heat pump model in off-line operation mode and for estimation of energy consumption of the heat pump in on-line operation mode (tested in UNIZGFER, HEP, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

HVAC.PE.2 -- submodule for estimation of a thermal losses/gains model and a flow shares model of a piping system in off-line operation mode and for estimation of thermal losses/gains and flows in on-line operation mode (tested in UNIZGFER, HEP, IDRIJA school and sports centre, STREM school, STREM retirement and care centre, EPHZHB building and EON pilot buildings within 3Smart);

HVAC.PE.3 -- submodule for estimation of an energy consumption model of a circulation pump in the central HVAC system in off-line operation mode and for estimation of circulation pump electricity consumed in on-line operation mode (tested in UNIZGFER pilot building within 3Smart);

HVAC.PE.4 – submodule for estimation of a prediction model for non-controllable heat consumption in the building in off-line operation and for prediction of non-controllable heat consumption in on-line operation (tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart).

In the following chapters the mentioned submodules are presented with their interface tables showing which data they use as inputs and which data they provide as outputs to be at the disposal to other submodules. Detailed explanations of algorithms behind each of the submodules are provided in the previously delivered 3Smart document D4.3.1 (related to prediction and estimation). Based on feedback from pilots, D4.3.1 is further updated and here provided as Annex 1.

Source and sink for the data used by submodules is a properly structured 3Smart database. Its structure in the part concerned by the central HVAC system level prediction and estimation submodules is provided in Annex 2.

2 HVAC.PE.1 submodule

HVAC.PE.1 submodule is used for estimation of a heat pump model in off-line operation mode and for estimation of energy consumption of the heat pump in on-line operation mode. Within 3Smart it is tested in UNIZGFER, HEP, STREM retirement and care centre, EPHZHB and EON pilot buildings.

The submodule interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Required inputs for the HVAC.PE.1 submodule.

Parameter	Description	Format
$T_{\mathrm{w}}^{\mathrm{out}}$ [°C]	Temperature of the medium	Historical sequence of data with

	coming out of the heat pump	time-stamps, minutely sampled
$T_{\mathrm{w}}^{\mathrm{in}}$ [°C]	Temperature of the medium coming into the heat pump	Historical sequence of data with time-stamps, minutely sampled
$E_{\mathrm{e,hp}}[\mathrm{kWh}]$	Electrical energy consumption of the heat pump	Historical sequence of data with time-stamps, minutely sampled
Q_{s} $[\mathrm{m}^3/\mathrm{h}]$	Medium flow through the heat pump	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{env}}[°\mathrm{C}]$	Temperature of the environment	Historical sequence of data with time-stamps, minutely sampled

Outputs:

Table 2.2: Outputs for the HVAC.PE.1 submodule.

Parameter	Description	Format
$\gamma_{\mathrm{p}}[-]$	Factor-obtained from equation 2.9	seasonal averaged value

On-line operation of the module (compute the estimated electricity consumption of the heat pump by applying the identified COP model with γ_p, for monitoring purposes, runs each minute)

Inputs:

Parameter	Description	Format
$T_{\mathrm{s}}[°\mathrm{C}]$	Temperature of the medium coming out of the heat pump	current, minutely sampled
$T_{\mathrm{w,hp}}^{\mathrm{in}}[°\mathrm{C}]$	Temperature of the medium coming into the heat pump	current, minutely sampled
γ_{p}	Factor	The value obtained by the off-line operation of module
Q_{s} $[\mathrm{m}^3/\mathrm{h}]$	Medium flow through the heat pump	current, minutely sampled
$T_{\mathrm{env}}[°\mathrm{C}]$	Temperature of the environment	current, minutely sampled

Outputs:

Parameter	Description	Format
$E_{\mathrm{e,hp}}[\mathrm{kWh}]$	Estimated electricity consumption of the heat pump	current

3 HVAC.PE.2 submodule

HVAC.PE.2 submodule is used for estimation of a thermal losses/gains model and a flow shares model of a piping system in off-line operation mode and for estimation of thermal losses/gains and flows in on-line operation mode. Within 3Smart it is tested in UNIZGFER, HEP, IDRIJA school and sports centre, STREM school, STREM retirement and care centre, EPHZHB building and EON pilot buildings.

The submodule interface is defined in Table 3.1 and Table 3.2.

Table 3.1: Inputs for HVAC.PE.2 submodule

Parameter	Description	Format
$T_{\mathrm{s}}[°\mathrm{C}]$	Temperature of the medium coming out of the heat pump	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w,hp}}^{\mathrm{in}}[°\mathrm{C}]$	Temperature of the medium coming into the heat pump	Historical sequence of data with time-stamps, minutely sampled
$Q_{\mathrm{s}}\ [\mathrm{m}^3/\mathrm{h}]$	Medium flow through the heat pump	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w,cal,i}}^{\mathrm{in}}[°\mathrm{C}]$	Measured temperature at the medium entrance into the floor segment *i*	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w,cal,i}}^{\mathrm{out}}[°\mathrm{C}]$	Measured temperature at the medium exit from the floor segment *i*	Historical sequence of data with time-stamps, minutely sampled
$Q_{\mathrm{w,cal,i}}\ [m^3/\mathrm{h}]$	Measured input/output flow for the floor segment *i*	Historical sequence of data with time-stamps, minutely sampled
Piping data		
Roughly estimated temperature of the space through which the pipes run	Roughly estimated c through which the pipes run	

Table 3.2: Outputs of the HVAC.PE.2 submodule

Parameter	Description	Format
Parameters of the temperature drop/rise model on the supply line for each of the floor segments i (one model for each i)	The parameters calculated based on equation 3.18 from Annex 1	Output for the 3Smart database
Parameters of the temperature model for return temperature into the central system based on floor segments outgoing temperatures and flows	The parameters calculated based on equation 3.18 from Annex 1	Output for the 3Smart database
α_i	Parameters for flow shares	Output for the 3Smart database

4 HVAC.PE.3 submodule

HVAC.PE.3 submodule is used for estimation of an energy consumption model of a circulation pump in the central HVAC system in off-line operation mode and for estimation of circulation pump electricity consumed in on-line operation mode (tested in UNIZGFER pilot building within 3Smart).

The submodule interface is defined in **Table 4.1** and **Table 4.2**.

Table 4.1: Inputs for the HVAC.PE.3 module

INPUT		FORMAT
$\eta_p = f(Q_s)$	total pump efficiency obtained by fitting the curve which is provided by the pump manufacturer	Pump data
$p_p\,[\mathrm{Pa}]$	Measurement of pressure drop;	Historical data, minutely sampled
$Q_s\,[\mathrm{m}^3/\mathrm{h}]$	Measurement of flow	Historical data, minutely sampled

Table 4.2: Ouptuts of the HVAC.PE.3 module

OUTPUT		FORMAT
$f(Q_s)$ proportional to Q_s^3	Parameters of the electrical energy consumption model	The procedure for parameters identification is implemented in Python

5 HVAC.PE.4 submodule

HVAC.PE.4 is a submodule used for estimation of a prediction model for non-controllable heat consumption in the building in off-line operation and for prediction of non-controllable heat consumption in on-line operation. Within 3Smart it is tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings.

The module interface is provided with the following tables.

Table 5.1: Required inputs for non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Historical profile of the non-controllable energy consumption on the central HVAC unit	$E_{\mathrm{t,nc}}$	Non-controllable thermal energy consumption on the HVAC level
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I_{\mathrm{diff}}^{h}, I_{\mathrm{dir}}^{\mathrm{n}}$ Remaining pilot sites: $T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

Table 5.2: Outputs of the non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	$\theta_{\mathrm{t,nc}}$	Needed for on-line operation of the submodule.
Predicted non-controllable heating/cooling energy consumption evolution (for on-line operation of the submodule)	$(E_{\mathrm{t,nc}})_{\mathrm{N}}$	Needed for the MPC module on the central HVAC level

Bibliography

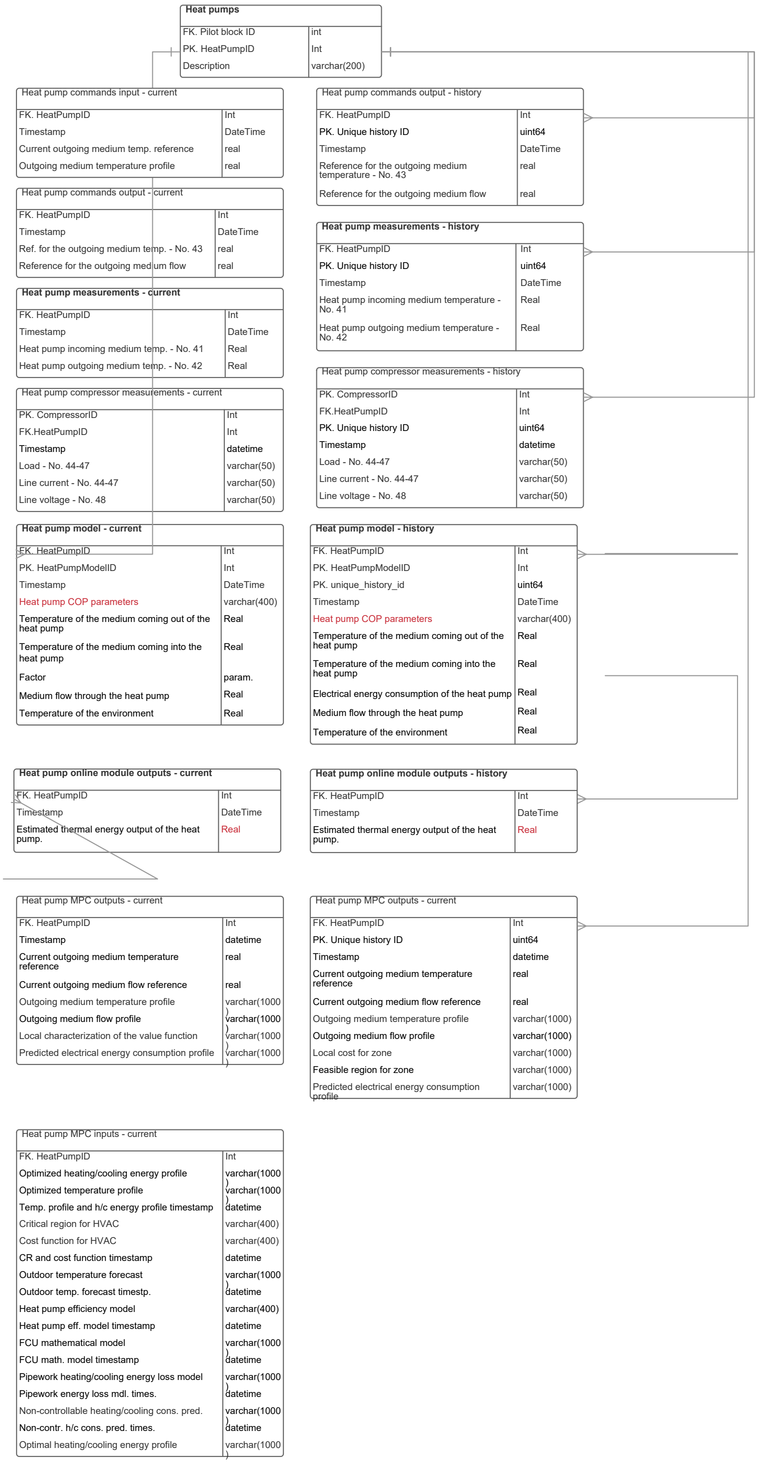
[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 – Open software module for central HVAC system level consumption management – Estimation and prediction submodules

Provided as a separate document.

Annex 2 – 3Smart database organization for open software module for the central HVAC system level management – Estimation and prediction submodules

HVAC.PE.1 Input/output data database structure:



HVAC.PE.2 Input/output data database structure:

pipework	
FK. Pilot block ID	**int**
PK. PipeworkID	**Int**
Description	varchar(200)

pipework model inputs	
FK. PipeworkID	**Int**
Timestamp	DateTime
Temperature of the medium coming out of the heat pump	Real
Temperature of the medium coming into the heat pump	Real
Medium flow through the heat pump	Real
Measured temperature at the medium entrance into the floor segment i	Real
Measured temperature at the medium exit from the floor segment i	Real
Measured input/output flow for the floor segment i	Real

pipework model	
FK. PipeworkID	**Int**
FK. CalorimeterID	**Int**
PK. PipeworkModelID	**Int**
Timestamp	DateTime
The parameters calculated based on presentation in file 3Smart_HEP_TT_March_HVAC.PE.2.pptx	Param.
Parameters for flow shares	Param.

pipework model outputs	
FK. PipeworkID	**Int**
Timestamp	DateTime
Estimated (based on the model) temperatures (calorimeters/zone elements)	Real
Estimated return medium temperature (at the heat pump input)	Real
Estimated flows (calorimeters)	Real

pipework model outputs history	
FK. PipeworkID	**Int**
PK. unique_history_id	**uint64**
Timestamp	DateTime
Estimated (based on the model) temperatures (calorimeters/zone elements)	Real
Estimated (based on the model) temperatures (calorimeters/zone elements)	Real
Estimated flows (calorimeters)	Real

HVAC.PE.3 Input/output data database structure:

pipework	
FK. Pilot block ID	**int**
PK. PipeworkID	**Int**
Description	varchar(200)

pipework model inputs	
FK. PipeworkID	**Int**
Timestamp	DateTime
Temperature of the medium coming out of the heat pump	Real
Temperature of the medium coming into the heat pump	Real
Medium flow through the heat pump	Real
Measured temperature at the medium entrance into the floor segment i	Real
Measured temperature at the medium exit from the floor segment i	Real
Measured input/output flow for the floor segment i	Real

pipework model	
FK. PipeworkID	**Int**
FK. CalorimeterID	**Int**
PK. PipeworkModelID	**Int**
Timestamp	DateTime
The parameters calculated based on presentation in file 3Smart_HEP_TT_March_HVAC.PE.2.pptx	Param.
Parameters for flow shares	Param.

pipework model outputs	
FK. PipeworkID	**Int**
Timestamp	DateTime
Estimated (based on the model) temperatures (calorimeters/zone elements)	Real
Estimated return medium temperature (at the heat pump input)	Real
Estimated flows (calorimeters)	Real

pipework model outputs history	
FK. PipeworkID	**Int**
PK. unique_history_id	**uint64**
Timestamp	DateTime
Estimated (based on the model) temperatures (calorimeters/zone elements)	Real
Estimated (based on the model) temperatures (calorimeters/zone elements)	Real
Estimated flows (calorimeters)	Real

HVAC.PE.4. input/output data database structure

Input data database structure:

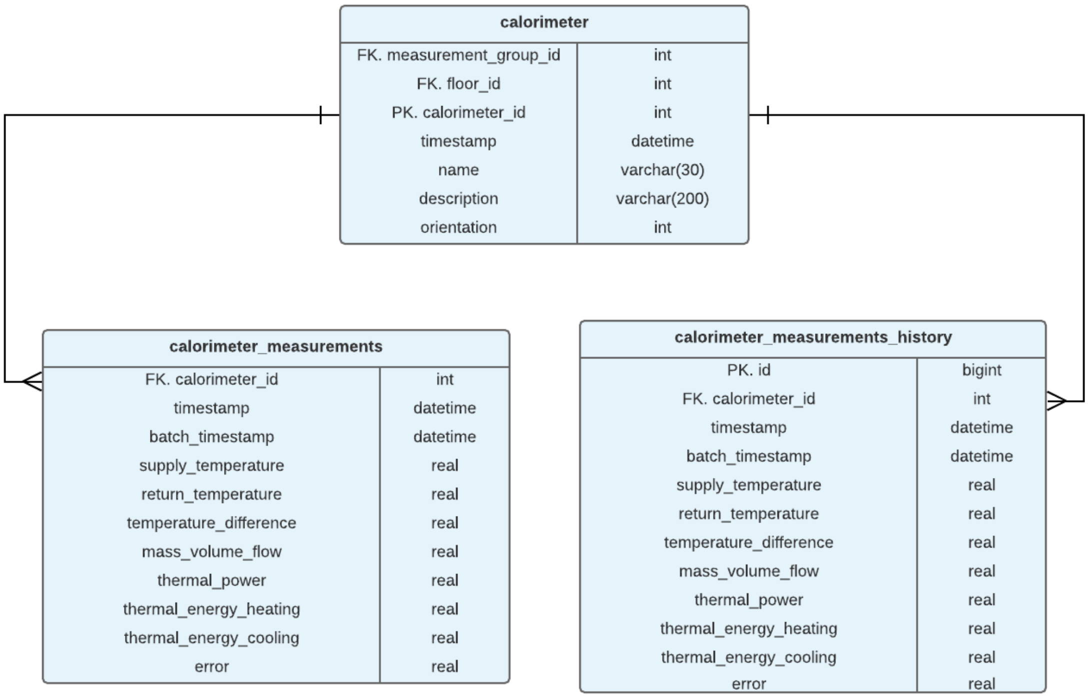


Figure 1. Current and historical non-controllable thermal energy consumption data database structure.

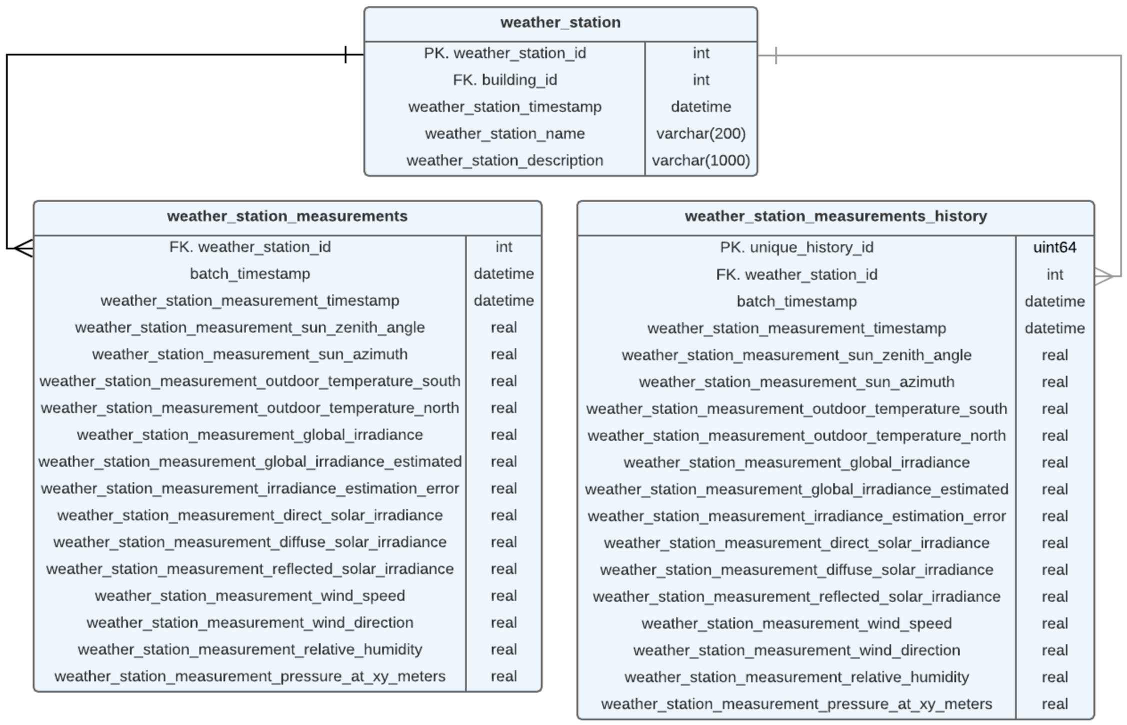


Figure 2. Weather measurements data database structure.

weather_predictor	
PK. weather_predictor_id	int
weather_predictor_timestamp	datetime
weather_predictor_name	varchar(100)
weather_predictor_description	varchar(100)
weather_predictor_sample_time	int

weather_prediction	
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

weather_prediction_history	
PK. id	bigint
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

Figure 3. Weather forecast data database structure.

Output data database structure:

hvac_pe4_outputs	
FK. heating_substation_id	int
timestamp	datetime
nctrl_heat_consumption_pred	varchar(2000)
nctrl_cool_consumption_pred	varchar(2000)

hvac_pe4_outputs_history	
FK. heating_substation_id	int
PK. id	uint64
timestamp	datetime
nctrl_heat_consumption_pred	varchar(2000)
nctrl_cool_consumption_pred	varchar(2000)

Figure 4. Current and historical non-controllable thermal energy predictions data database tables.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 PREDICTION AND ESTIMATION IN CENTRAL HVAC SYSTEM LEVEL

Open software module for central heating/cooling system management – Prediction and estimation module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	UNIBGFME, UNIZGFER
Authors (Partners)	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), Mihailo Milanović (UNIBGFME), Milijana Paprika (UNIBGFME), Hrvoje Novak (UNIZGFER)
Contact person	Mirko Komatina (UNIBGFME)
Abstract (for dissemination)	This annex provides information on the background logic for functioning of prediction and estimation submodules on the central HVAC system level.
Keyword List	3Smart, open software module, central heating/cooling management, heat pump, heat losses/gains, pipework

Revision history

Revision	Date	Description	Author (Organization)
0.1	14 November 2017	First draft	Hrvoje Novak (UNIZGFER)
0.2	15 November 2017	First draft (minor corrections)	Mario Vašak (UNIZGFER)
0.3	15 November 2017	Alpha version	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME), Milijana Paprika (UNIBGFME), Hrvoje Novak (UNIZGFER)
0.4	15 December 2017	Beta version	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME), Milijana Paprika (UNIBGFME), Hrvoje Novak (UNIZGFER)
0.5	20 December 2017	Draft final	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME), Milijana Paprika (UNIBGFME), Hrvoje Novak (UNIZGFER)
0.6	29 December 2017	Final	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME), Milijana Paprika (UNIBGFME), Hrvoje Novak (UNIZGFER)
1.0	15 January 2019	Taken from D4.3.1	Mario Vašak (UNIZGFER)
1.1	7 June 2019	Updated based on feedback from pilots	Mario Vašak (UNIZGFER)
1.2	26 June 2019	Updated based on feedback from pilots	Mario Vašak, Hrvoje Novak, Nikola Hure (UNIZGFER), Vladimir Jovanović, Nebojša Manić, Mirko Komatina (UNIBGFME)
2.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of contents

Executive summary

This document as annex to D4.5.3 of central HVAC system prediction and estimation submodules focusses on the background logic of individual modules.

Heat pump system module was developed for the air/water heat pump which is used in the pilots of this project. Basic equations for heat pump system operating according to ideal Carnot cycle are presented for both heating and cooling mode.

Pipework heat losses/gains module was developed to provide precise regulation of heating/cooling units within buildings of the pilots from two different aspects, theoretical and practical. Conceptual project is based on theoretical equations for heat exchange between the fluid in the pipe and surrounding atmosphere, while the practical approach uses empirical equations provided for the heat losses of insulated and non-insulated pipes in the central heating systems.

Variable speed pump module was developed to ensure energy savings in central heating systems of the pilot buildings. Circulation pump was considered for two different cases (operation modes): Case1 - with fan coils (FCs) as heating/cooling zone devices and Case 2 - with radiators or floor heating/cooling zone devices. Mathematical equations for these cases are provided.

Prediction of the total non-controllable energy consumption submodule provides assessment of heating and cooling loads based on historical time series of consumption data and other influential variables.

1 Introduction

This report focusses on central HVAC system improvement through smart metering and control of water flow rate and supply temperature. Mathematical and numerical models are developed to assess the behavior of the system. Sub models for heat pump system, thermal losses in piping and variable frequency drive (VFD) for circulation pump are considered as vital for efficient operation of the central HVAC system within a holistic building energy management.

2 Heat pump system (HVAC.PE.1)

Heat pump system is one part of the central HVAC system providing heating and cooling of the building depending on its design and implementation.

2.1 Theoretical background

Heat pump is used to extract energy from the cold outdoors and carry it into the warm indoors as presented in Figure 2.1. The measure of performance of a heat pump is expressed by the coefficient of performance (COP_{HP}).

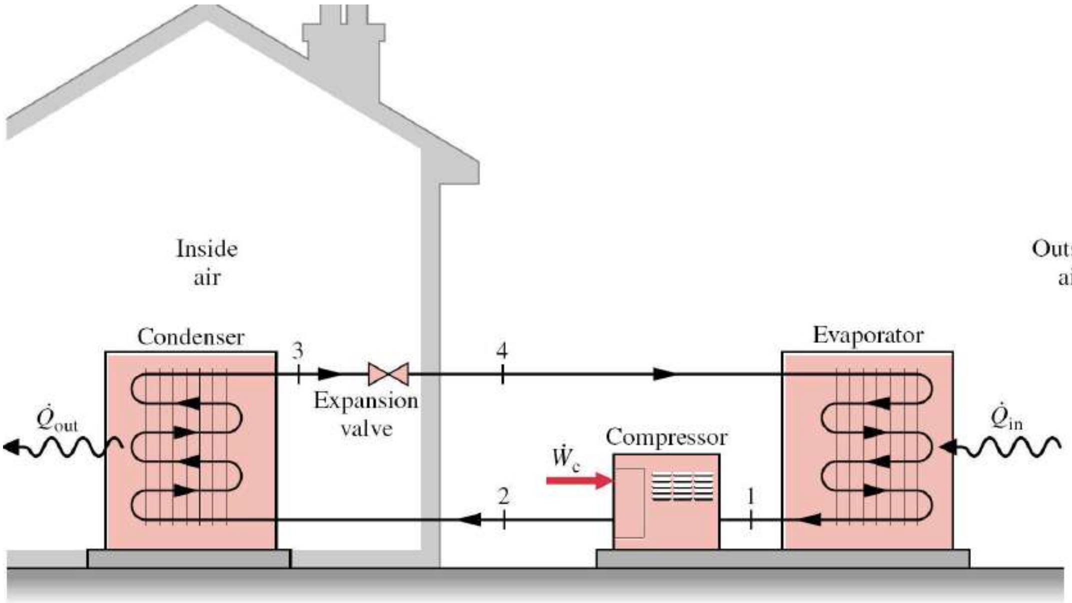


Figure 2.1: General concept of a heat pump system

Each heat pump system (Figure 2.2 and Figure 2.3) consists of the following main parts:

1. compressor;
2. condenser;
3. expansion valve;
4. evaporator.

The most widely used type of heat pumps is **air/water** heat pump which is used in the pilots of the 3Smart project. General layout of a heat pump system operating in the heating cycle is presented in Figure 2.2. The heat pump transfers heat energy from the heat source (outdoor air) to the heat sink (water from heating system of the building). Indoor unit is operating as a condenser and outdoor as an evaporator.

More detailed description of the heat pump system operating in heating mode is presented in Figure 2.3.

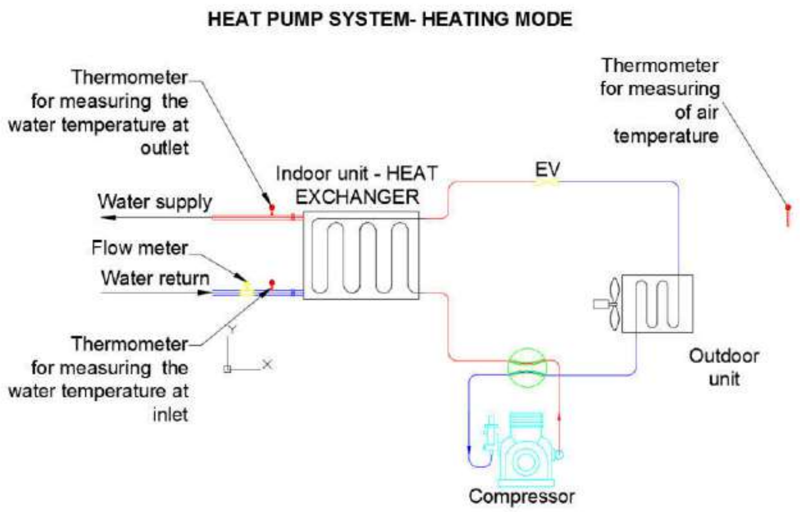


Figure 2.2: A general layout of a heat pump system operating in the heating cycle

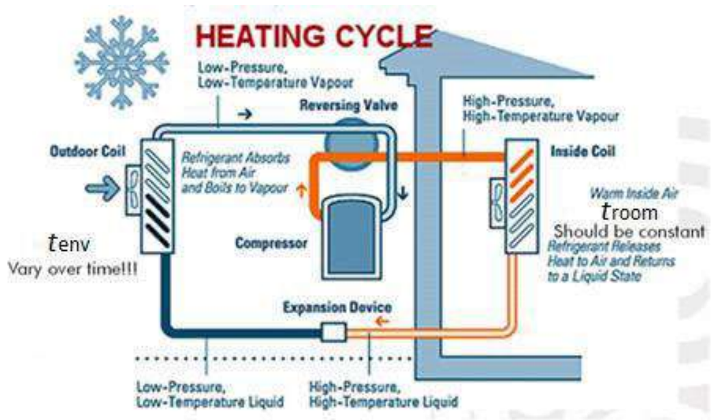


Figure 2.3: Detailed description of a heat pump system operating in the heating mode [1]

A general layout of a heat pump system operating in the cooling cycle is presented in Figure 2.4. The heat pump transfers the heat from the building (heat source) to the ambient air (heat sink). The evaporator is located at the building side (indoor unit), while the condenser is located at the outdoor side (outdoor unit).

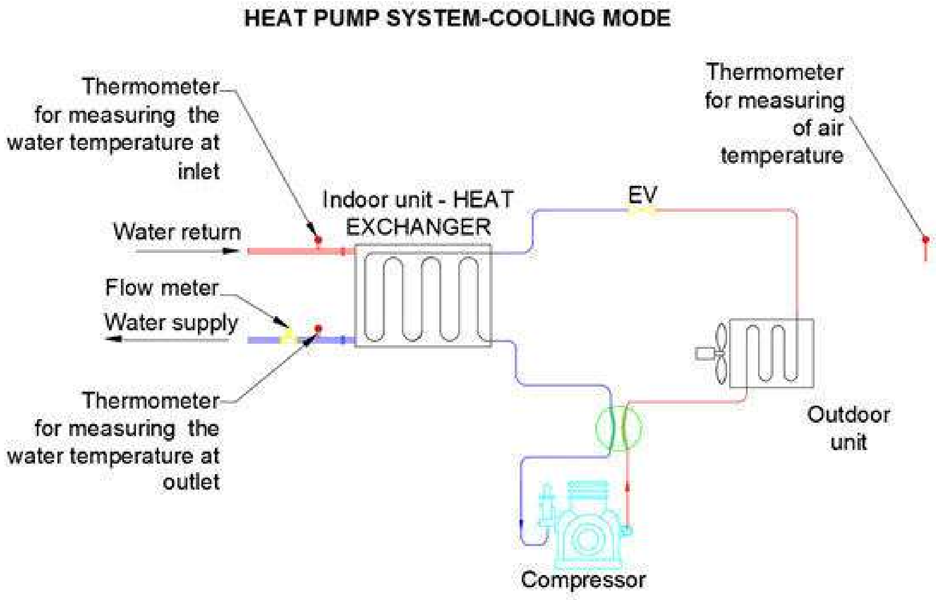


Figure 2.4: A general layout of a heat pump system operating in the cooling cycle

More detailed description of a heat pump system operating in the cooling mode is presented in Figure 2.5.

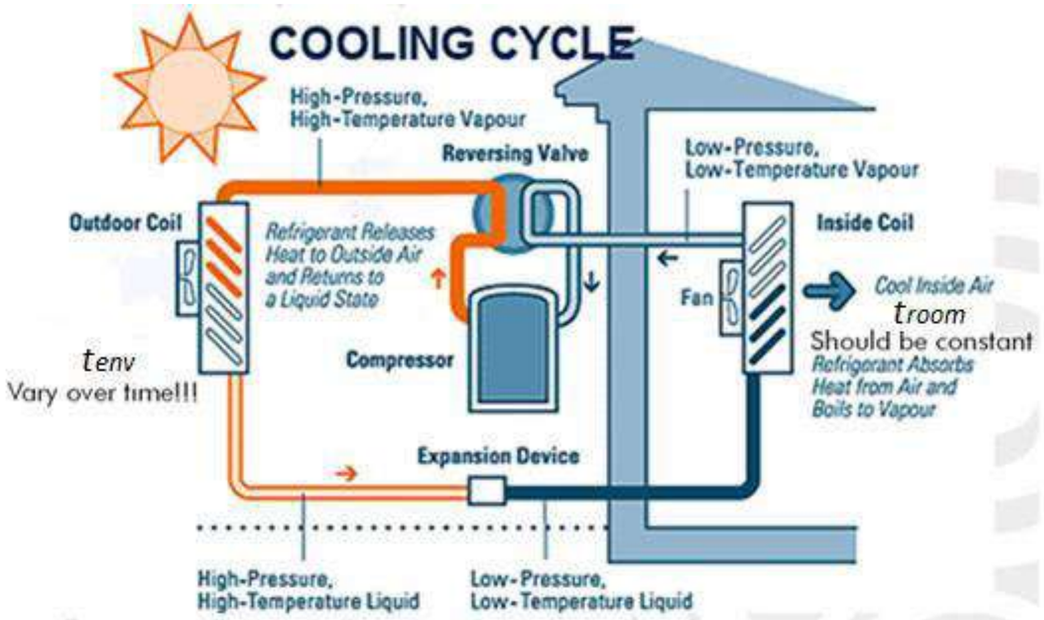


Figure 2.5: Detailed description of a heat pump system operating in the cooling mode [1]

There are numerous standards developed for testing and rating heat pumps, ISO and EN. Base EN standard covering heat pump with electrically driven compressors for space heating and cooling is EN 14511 series of standards (4 parts). Base ISO standard for heat pumps is ISO 13612 (2 parts).

Heat pump systems dimensioning according to ISO 13612 is presented in Figure 2.6.

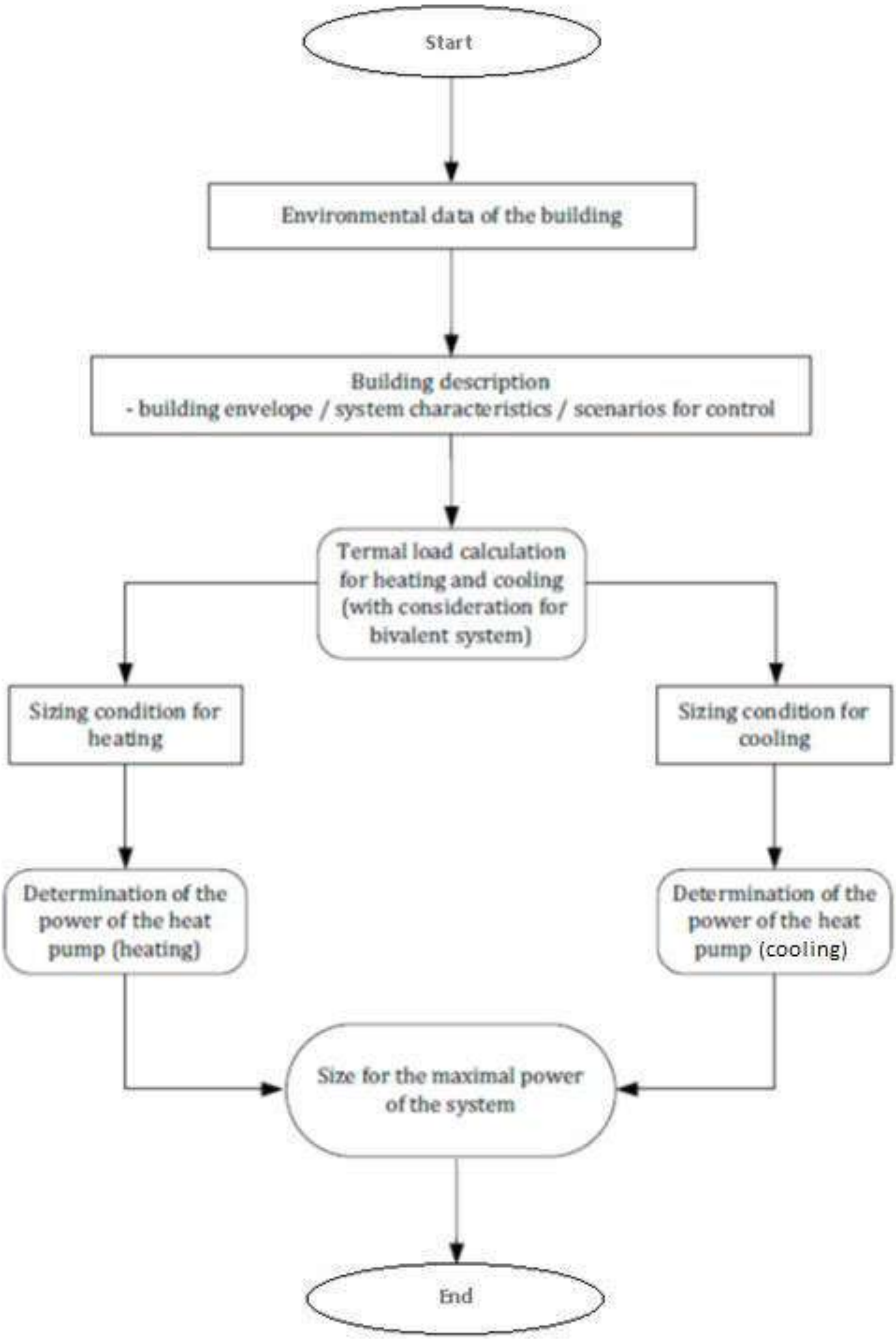


Figure 2.6: Algorithm for dimensioning of a heat pump systems according ISO 13612

For operation in both modes the compressor uses electric energy ($E_{e,hp}$) to provide mechanical work for transfer of the heat from the heat source to the heat sink. The heat exchanged between the heat pump and the working medium is denoted as $E_{t,hp}$.

The coefficient of performance of heat pump (COP) is defined as the ratio of the desired transferred heat and the electrical energy consumption of the heat pump by the heat pump compressor. In the analyzed case the useful heat effect is $E_{t,hp}$, thus COP is defined as:

$$COP = \frac{E_{t,hp}}{E_{e,hp}} \tag{2.1}$$

There are many types of heat pumps differed by the heat transfer medium. Good general approximation of a heat pump, independent from the heat transfer medium can be obtained using the theoretical Carnot cycle which operates between the corresponding temperatures (heat source/heat sink). This approximation is presented below.

2.2 Carnot Cycle heat pump model and algorithm

In the project case the corresponding temperatures are the mean temperature of water (T_m) that flows through the heat exchanger and the temperature of the environment (T_{env}) (air temperature). Carnot cycle in *T-S* diagram, where *T* is temperature and *S* is entropy, is presented in Figure 2.7 and Figure 2.8.

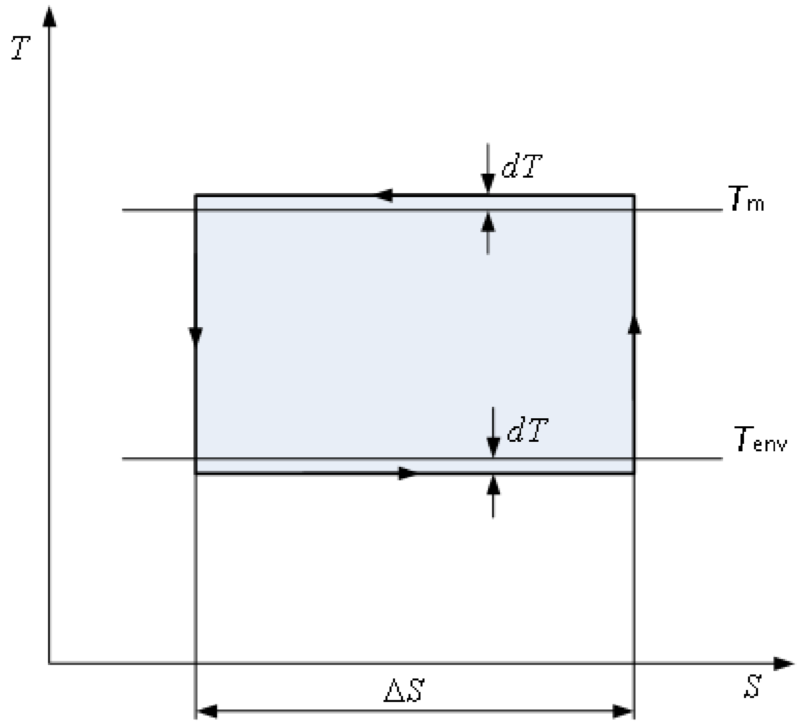


Figure 2.7: Carnot cycle in *T-S* diagram -Heating

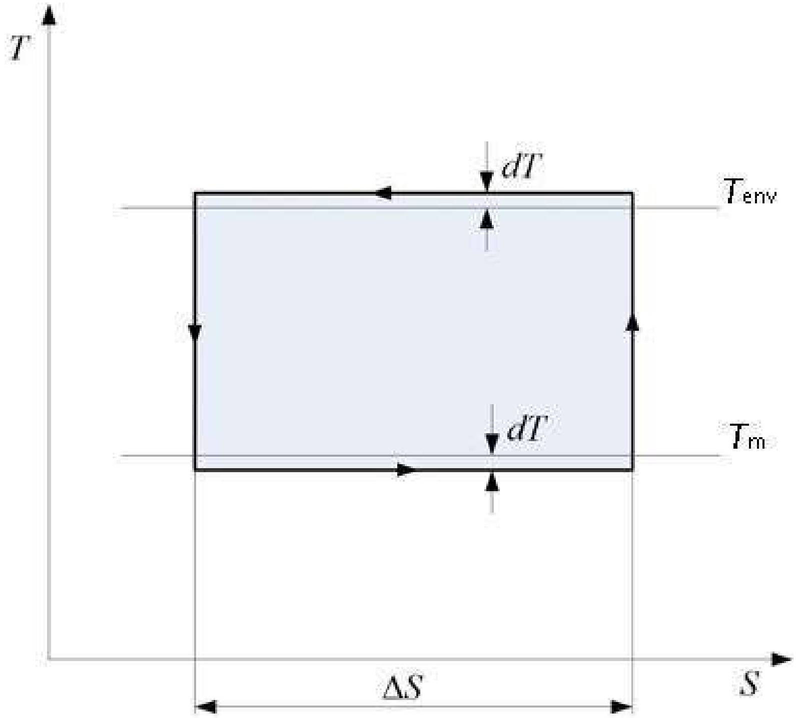


Figure 2.8: Carnot cycle in *T-S* diagram - Cooling

For heating mode the inlet water temperature is calculated as:

$$T_{\mathrm{w,hp}}^{\mathrm{in}} = T_s - \frac{3.6 \cdot 10^6 E_{\mathrm{t,hp}}}{4186 \cdot \frac{Q_{\mathrm{s}}}{3.6} \cdot T_{\mathrm{d}}} \ [°\mathrm{C}] \tag{2.2}$$

For cooling mode the inlet water temperature is calculated as:

$$T_{\mathrm{w,hp}}^{\mathrm{in}} = T_{\mathrm{s}} + \frac{3.6 \cdot 10^6 E_{\mathrm{t,hp}}}{4186 \cdot \frac{Q_{\mathrm{s}}}{3.6} \cdot T_{\mathrm{d}}} \ [°\mathrm{C}] \tag{2.3}$$

The mean water temperature of the heat exchanger is calculated as:

$$T_{\mathrm{m}} = \frac{T_{\mathrm{w,hp}}^{\mathrm{in}} + T_{\mathrm{s}}}{2} \ [°\mathrm{C}] \tag{2.4}$$

Where:

T_{m} [°C] – temperature of the medium coming out of the heat pump

$T_{\mathrm{w,hp}}^{\mathrm{in}}$ [°C] – temperature of the medium coming into the heat pump

T_{s} [°C] – is the water temperature at the outlet of the heat exchanger

$E_{\mathrm{t,hp}}$ [kWh] – is delivered heat to the water in the heat exchanger

T_{d}[s]-sampling time
Q_{s} [m^3/h]- volume flow of the supplied medium

The theoretical maximum efficiency of the heat pump is described by the Carnot efficiency [2]:

- Carnot heat pump system – heating mode:

$$COP_{\mathrm{HP}} = \frac{Q_{\mathrm{out}}}{W_{in}} = \frac{Q_{\mathrm{out}}}{Q_{\mathrm{out}} - Q_{\mathrm{in}}} = \frac{(T_{\mathrm{m}} + 273.15)\Delta S}{(T_{\mathrm{m}} + 273.15)\Delta S - (T_{\mathrm{env}} + 273.15)\Delta S} \tag{2.5}$$

$$COP_{\mathrm{HP}} = \frac{T_{\mathrm{m}} + 273.15}{T_{\mathrm{m}} - T_{\mathrm{env}}} \tag{2.6}$$

- Carnot heat pump system – cooling mode:

$$COP_{\mathrm{R}} = \frac{Q_{\mathrm{in}}}{W_{\mathrm{in}}} = \frac{Q_{\mathrm{in}}}{Q_{\mathrm{out}} - Q_{\mathrm{in}}} = \frac{(T_{\mathrm{m}} + 273.15)\Delta S}{(T_{\mathrm{env}} + 273.15)\Delta S - (T_{\mathrm{m}} + 273.15)\Delta S} \tag{2.7}$$

$$COP_{\mathrm{R}} = \frac{T_{\mathrm{m}} + 273.15}{T_{\mathrm{env}} - T_{\mathrm{m}}} \tag{2.8}$$

where T_{env} [°C] is the measured environment temperature.

The COP value of the real heat pump system can be estimated as a product of the *COP* of Carnot cycle and the factor γ_p:

$$COP_{\mathrm{real}} = \gamma_{\mathrm{p}} COP \tag{2.9}$$

where the factor γ_{p} is about 0.5 [3] (the most efficient heat pump has the value of factor $\gamma_{\mathrm{p}} = 0.7$ [4]).

On the other hand, the heat exchanged between the medium and the heat pump (the energy balance of the heat pump's exchanger on the medium side):

$$E_{\mathrm{t,hp}} = \frac{Q_{\mathrm{s}}}{3.6 \cdot 10^{6}} c_{\mathrm{w}} \rho_{\mathrm{w}} T_{\mathrm{d}} \left| T_{\mathrm{w,hp}}^{\mathrm{in}} - T_{\mathrm{s}} \right| \;[\mathrm{kWh}] \tag{2.10}$$

where:

$c_{\mathrm{w}} \left[\frac{\mathrm{kJ}}{\mathrm{kgK}}\right]$ – is the specific heat capacity of the medium

$\rho_{\mathrm{w}} \left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$ – is the density of the medium

Finally, the electrical energy consumption of the heat pump can be obtained using *COP* of the real cycle:

$$E_{\mathrm{e,hp}} = \frac{E_{\mathrm{t,hp}}}{COP_{\mathrm{real}}} \;[\mathrm{kWh}] \tag{2.11}$$

Graphical presentation of Carnot and real cycle COPs vs. environment temperature for both, heating and cooling mode is given in Figure 2.9. Figure 2.10 represents the electric power of the heat pump system and heat transfer rate across the heat exchanger vs. environment temperature for both, heating and cooling mode.

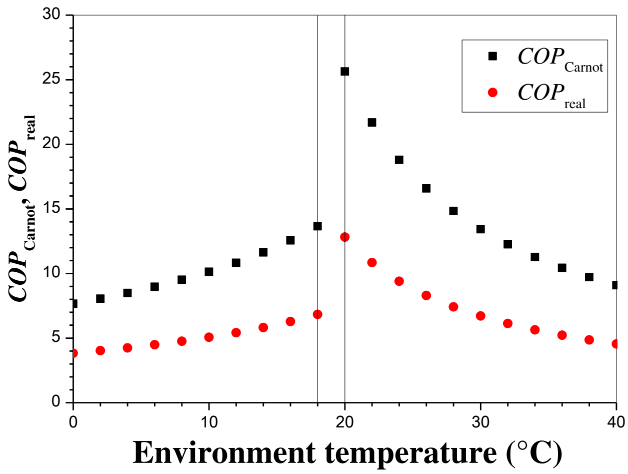


Figure 2.9: COP of Carnot and real cycle for heating and cooling mode

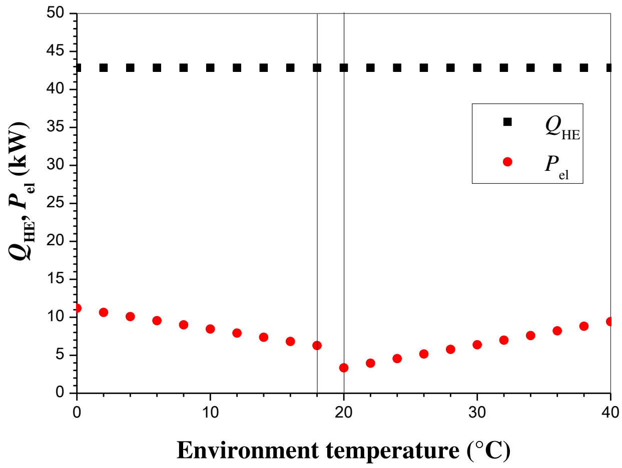


Figure 2.10: Electric power of the heat pump system and heat transfer rate across the heat exchanger as a function of environment temperature for heating and cooling mode

2.3 Algorithm for Carnot Cycle heat pump model

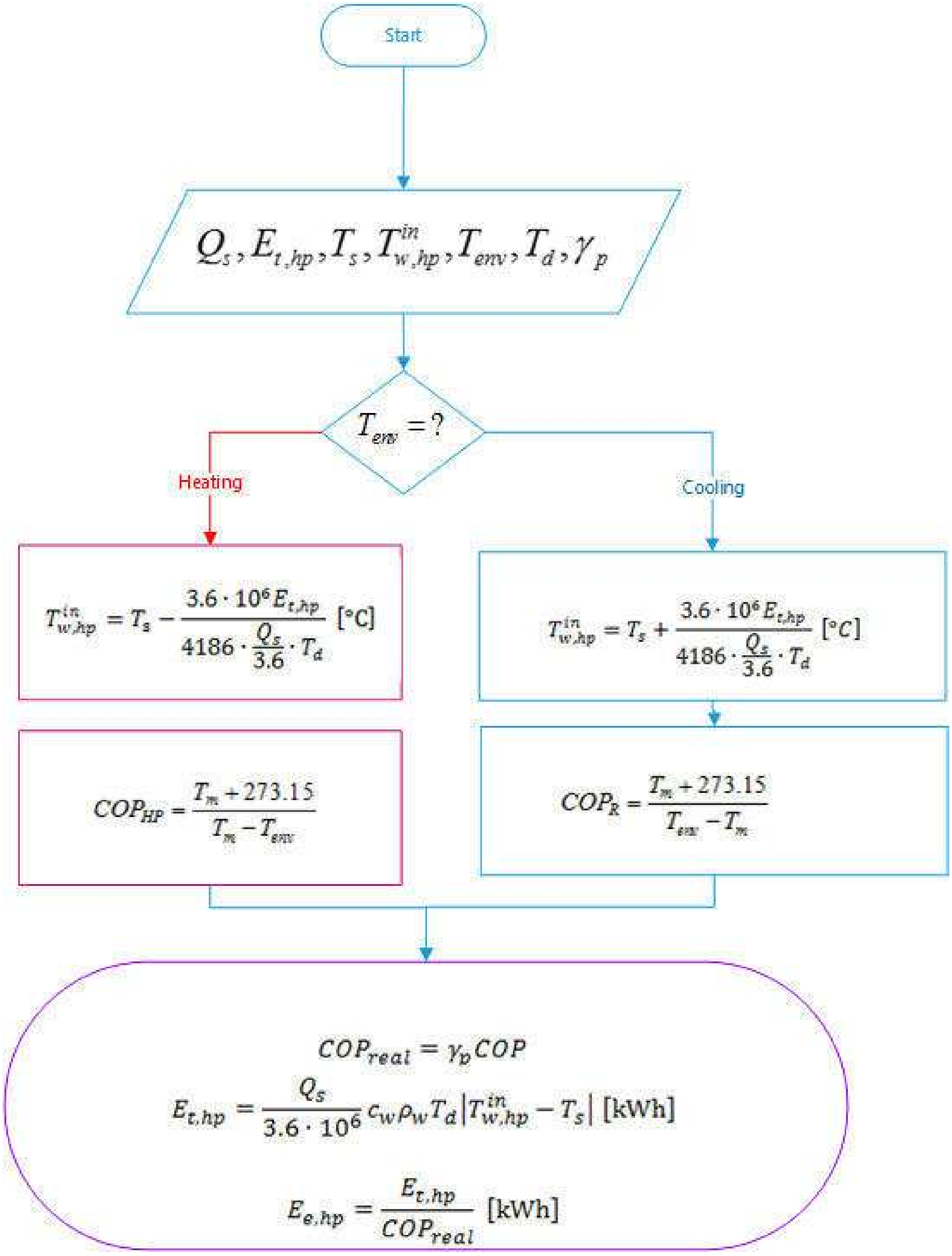


Figure 2.11: Algorithm for heat pump

Off-line operation of the module (linear regression which runs periodically using a historical sequence of data in order to compute the coefficient γ_{p}):

Inputs:

Parameter	Description	Format
$T_{\mathrm{w}}^{\mathrm{out}}$ [°C]	Temperature of the medium coming out of the heat pump	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w}}^{\mathrm{in}}$ [°C]	Temperature of the medium coming into the heat pump	Historical sequence of data with time-stamps, minutely sampled
$E_{\mathrm{e,hp}}$[kWh]	Electrical energy consumption of the heat pump	Historical sequence of data with time-stamps, minutely sampled
Q_{s} [m^3/h]	Medium flow through the heat pump	Historical sequence of data with time-stamps, minutely sampled
T_{env}[°C]	Temperature of the environment	Historical sequence of data with time-stamps, minutely sampled

Outputs:

Parameter	Description	Format
$\gamma_{\mathrm{p}}[-]$	Factor-obtained from equation 2.9	seasonal averaged value

On-line operation of the module (compute the estimated electricity consumption of the heat pump by applying the identified COP model with γ_p, for monitoring purposes, runs each minute)

Inputs:

Parameter	Description	Format
T_{s}[°C]	Temperature of the medium coming out of the heat pump	current, minutely sampled
$T_{\mathrm{w,hp}}^{\mathrm{in}}$[°C]	Temperature of the medium coming into the heat pump	current, minutely sampled
γ_{p}	Factor	The value obtained by the off-line operation of module
Q_{s} [m^3/h]	Medium flow through the heat pump	current, minutely sampled
T_{env}[°C]	Temperature of the environment	current, minutely sampled

Outputs:

Parameter	Description	Format
$E_{\mathrm{e,hp}}[\mathrm{kWh}]$	Estimated electricity consumption of the heat pump	current

3 Heat losses/gains of the pipework (HVAC.PE.2)

Heat losses/gains of the pipework are needed for exact control and regulation in 3Smart project. Depending on insulation and pipelines length there could be temperature drop or rise of the heating/cooling fluid at the inlet of a heating/cooling element in a building zone related to the initial temperature generated in the central heating/cooling unit (heat pump or heat exchanger). Prediction of this temperature drop/rise is vital for accurate regulation of the heating/cooling power of the final consumers (fan coils or similar units). Further it is necessary to predict the thermal losses in the piping segment from the central unit to the input of all elements and from the output of the elements back to the central unit.

3.1 Theoretical background

The model contains iterative procedure for estimating the water outlet temperature. The description consists of the modelling procedure with main equations used in the model, the results obtained from the model for the "measured" dummy inputs and references.

The process of heat transfer between fluid that flows through the pipework and surrounding (air) consists of:

1. Heat transfer by convection between the fluid and the pipe wall
2. Heat transfer by conduction through the pipe wall
3. Heat transfer by conduction through the insulation (if exists)
4. Heat transfer by convection at outside surface of pipework
5. Heat exchange between pipework to surrounding.

As a result of the fact that the outside surface temperature of the pipework is relatively low, heat transfer by radiation can be neglected. Furthermore, the pipe is thin and drop of the temperature is also small thus heat transfer by conduction through the pipe can be neglected.

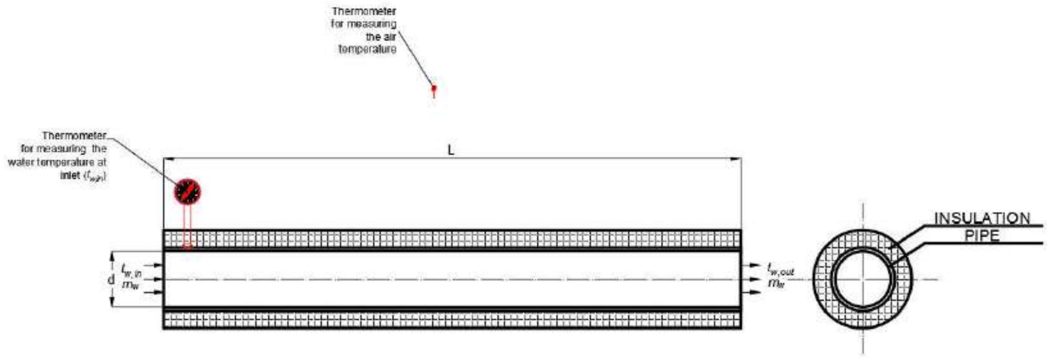


Figure 3.1: Physical model

The input parameters of the model are:

1. t_{env} –the measured environment (air) temperature, [°C]
2. δ –the insulation thickness, [mm]
3. k_{ins} –thermal conductivity of the insulation , [W/mK]
4. d –inner diameter of the pipe, [mm]
5. L –the length of the pipe, [m]
6. $\dot{m}_{\mathrm{w}}$ –the measured mass flow rate of water through the pipe, [kg/s]
7. $t_{\mathrm{w,in}}$ –the measured water temperature at the inlet, [°C]

The procedure is iterative, thus in the first step the outlet water temperature is assumed to be the same as the inlet water temperature, $t_{\mathrm{w,out}} = \mathrm{t}_{\mathrm{w,in}}$.

The mean water temperature is:

$$t_{\mathrm{w,mean}} = \frac{t_{\mathrm{w,in}} + t_{\mathrm{w,out}}}{2} \; [°\mathrm{C}] \tag{3.1}$$

The water density at the mean water temperature [4]:

$$\rho = \frac{A}{B^{1+\left(1-\frac{t_{\mathrm{w,mean}}+273.15}{C}\right)^{D}}} \left[\frac{kg}{m^3}\right] \tag{3.2}$$

The equation coefficients are: *A=0.14395; B=0.0112; C=649.727; D=0.05107.*

The dynamic viscosity of water at the mean water temperature [6]:

$$\mu = 2.414 \cdot 10^{-5} \cdot 10^{\frac{247.8}{t_{\mathrm{w,mean}}+273.15-140}} \; [\mathrm{Pas}] \tag{3.3}$$

The thermal conductivity of water at the mean temperature [7]:

$$k = -0.5752 + 6.397 \cdot 10^{-3}(t_{\mathrm{w,mean}} + 273.15) - 8.151 \cdot 10^{-6}(t_{\mathrm{w,mean}} + 273.15)^2 \left[\frac{\mathrm{W}}{\mathrm{mK}}\right] \quad (3.4)$$

The specific heat capacity of water at the mean temperature [7]:

$$c_{\mathrm{pw}} = 28.07 - 0.2817(t_{\mathrm{w,mean}} + 273.15) + 1.25 \cdot 10^{-3}(t_{\mathrm{w,mean}} + 273.15)^2 - 2.48 \cdot 10^{-6}(t_{\mathrm{w,mean}} + 273.15)^3 + 1.857 \cdot 10^{-9}(t_{\mathrm{w,mean}} + 273.15)^4 \left[\frac{\mathrm{kJ}}{\mathrm{kgK}}\right] \quad (3.5)$$

The mean velocity of water:

$$w = \frac{4\dot{m}_{\mathrm{w}}}{\rho \cdot (d \cdot 10^{-3})^2 \pi} \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \quad (3.6)$$

The Reynolds number:

$$\mathrm{Re} = \frac{w \cdot (d \cdot 10^{-3}) \cdot \rho}{\mu} \quad (3.7)$$

The Prandtl number:

$$\mathrm{Pr} = \frac{c_{pw} \cdot 10^3 \cdot \mu}{k} \quad (3.8)$$

The Nusselt number valid for smooth tubes over a large Reynolds number range including the transition region [8]:

$$\mathrm{Nu} = \frac{(\frac{f}{8}) \cdot (\mathrm{Re} - 1000)\mathrm{Pr}}{1 + 12.7\left(\frac{f}{8}\right)^{\frac{1}{2}} (\mathrm{Pr}^{\frac{2}{3}} - 1)} \quad (3.9)$$

The correlation is valid for $0.5 < Pr < 2000 \; and \; 3000 < Re < 5 \cdot 10^6$.

The friction factor f (the Darcy friction factor) is determined from Petukhov's formula:

$$f = \frac{1}{[0.79 \cdot \ln(\mathrm{Re}) - 1.64]^2} \quad (3.10)$$

The heat transfer coefficient at the contact surface between water and pipe:

$$h_{\mathrm{in}} = \frac{\mathrm{Nu} \cdot k}{d \cdot 10^3} \left[\frac{\mathrm{W}}{\mathrm{m^2K}}\right] \tag{3.11}$$

The heat transfer coefficient at the contact surface between air and pipework can be assumed:

$$h_{\mathrm{out}} = 10 \left[\frac{\mathrm{W}}{\mathrm{m^2K}}\right] \tag{3.12}$$

The overall heat transfer coefficient:

$$U = \frac{1}{\frac{1}{d \cdot 10^{-3}\pi h_{\mathrm{in}}} + \frac{1}{2\pi k_{\mathrm{ins}}} ln \frac{d + 2\delta}{d} + \frac{1}{(d + 2\delta)10^{-3}\pi h_{\mathrm{out}}}} \left[\frac{\mathrm{W}}{\mathrm{mK}}\right] \tag{3.13}$$

Heat losses/gains per unit length of pipe:

$$\dot{q}_{\mathrm{l}} = U \cdot \left|t_{\mathrm{w,mean}} - t_{\mathrm{env}}\right| \; \left[\frac{\mathrm{W}}{\mathrm{m}}\right] \tag{3.14}$$

Total heat losses/gains of the pipe:

$$\dot{Q} = U \cdot \left|t_{\mathrm{w,mean}} - t_{\mathrm{env}}\right| \cdot L \;\; [\mathrm{W}] \tag{3.15}$$

The estimated water temperature at the end of the pipe:

$$t_{\mathrm{w,out}} = t_{\mathrm{w,in}} \pm \frac{\dot{Q}}{\dot{m}_{\mathrm{w}} c_p} \;\; [°\mathrm{C}] \tag{3.16}$$

sign "+" is used when the environment (air) temperature is higher than the inlet water temperature while the sign "– " is used when the environment (air) temperature is lower than the inlet water temperature.

The obtained (new) water temperature at the outlet will be used for next iteration process in which the mean water temperature is:

$$t_{\mathrm{w,mean}} = \frac{t_{\mathrm{w,in}} + t_{\mathrm{w,out,LAST}}}{2} \;\; [°\mathrm{C}] \tag{3.17}$$

The convergence criteria uses the temperature difference of the $t_{\mathrm{w,out}}$ from last iteration step and $t_{\mathrm{w,out}}$ from previous iteration step and if this temperature difference is smaller than the requested error the iteration procedure will be stopped.

3.2 Results of the introduced iterative procedure

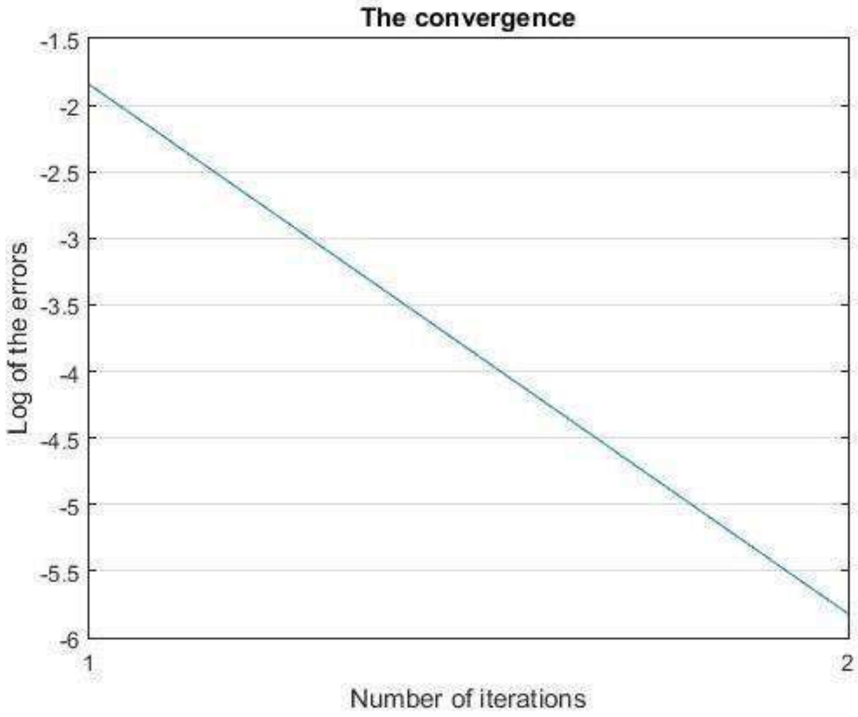


Figure 3.2: Convergence graph

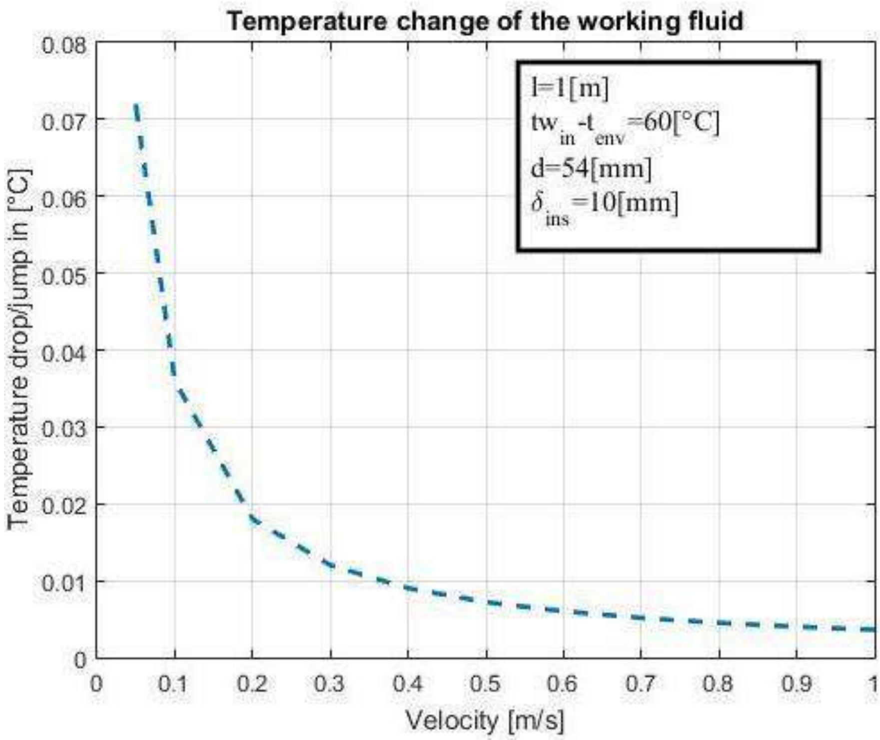


Figure 3.3: Temperature change of the working fluid (with insulation) graph

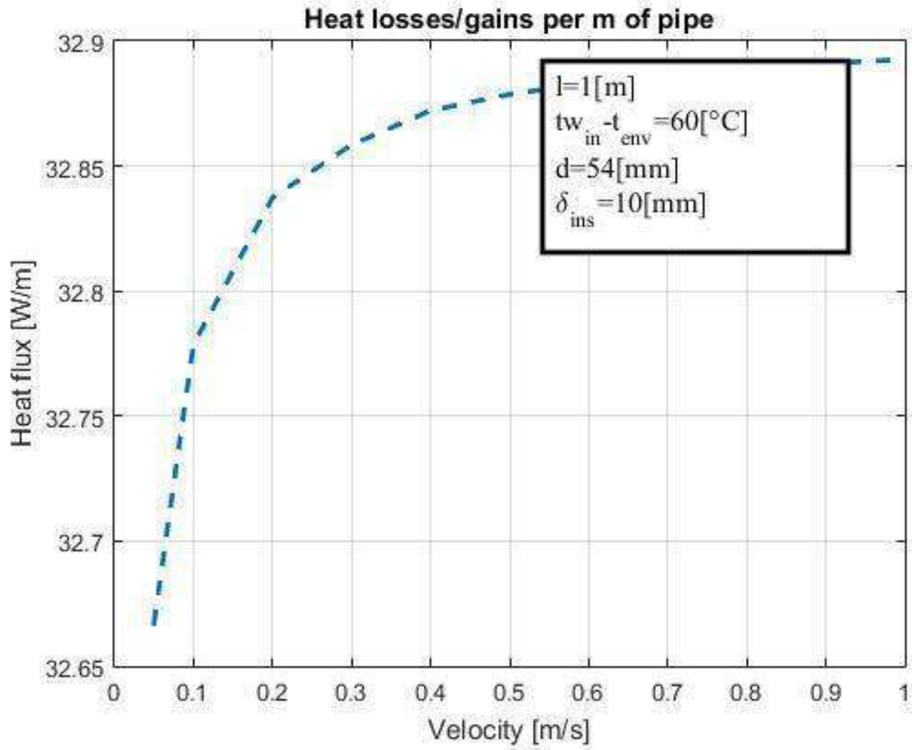


Figure 3.4: Heat losses/gains per m of pipe with insulation

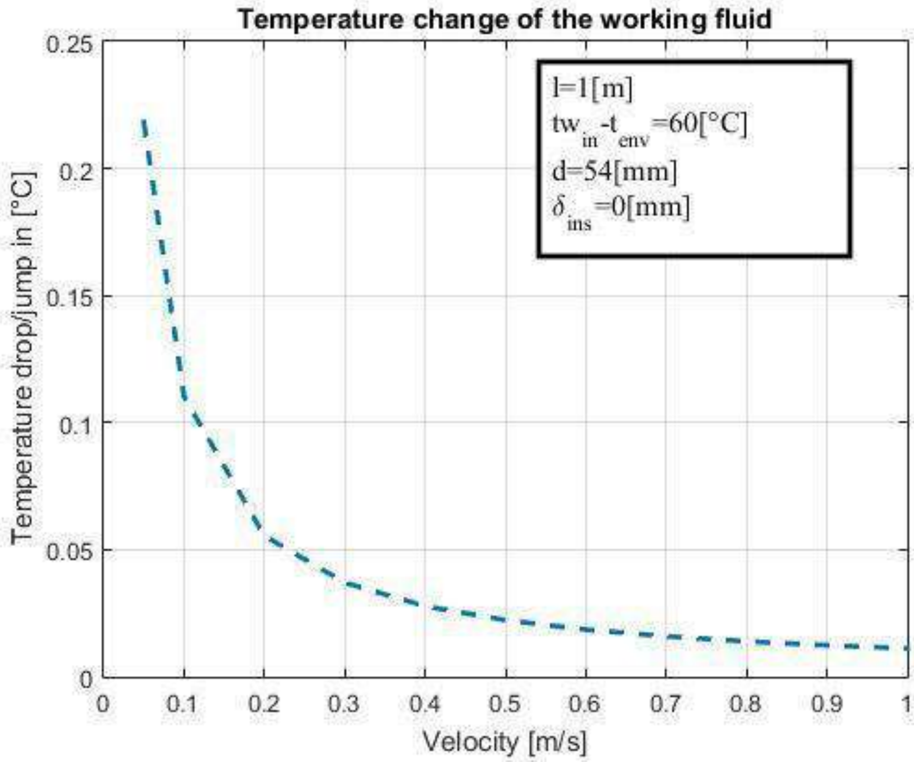


Figure 3.5: Temperature change of the working fluid (without insulation) graph

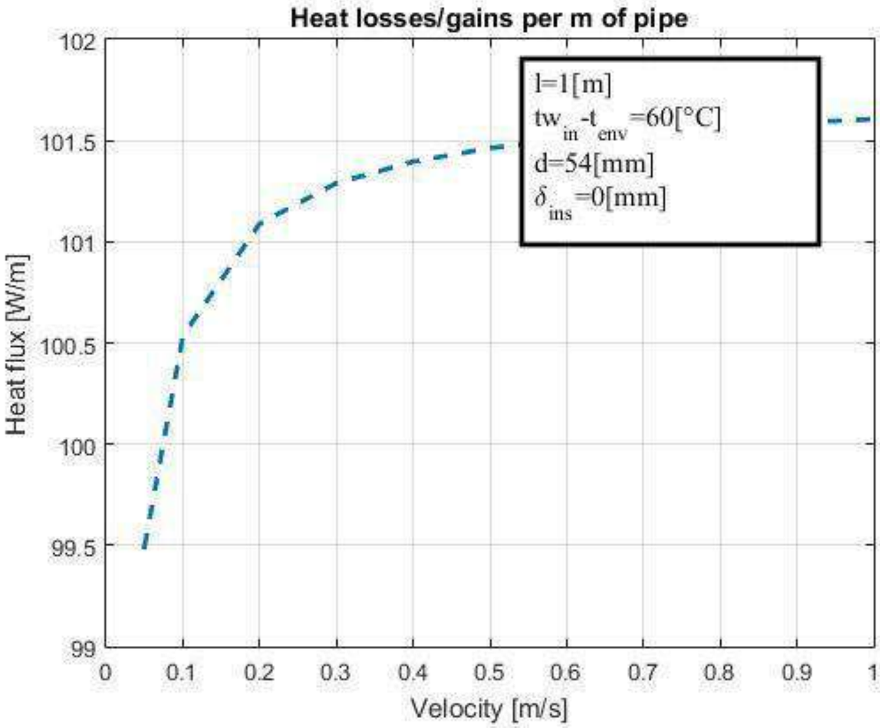


Figure 3.6: Heat losses/gains per m of pipe without insulation

3.3 Practical approach for heat losses/gains of the pipework model

Heat losses/gains of the pipework can be estimated/adopted as suggested by BSRIA Limited and The Chartered Institution of Building Services Engineers (CIBSE) in A guide to HVAC Building Services Calculations Second edition [9].

Rule of the thumb for design data is:

Water flow temperature – heating mode:

- LTHW: 70-95 °C
- MTHW: 100-120 °C
- HTHW: over 120 °C

Estimate the heat loss for each section (often taken as a percentage of unit load such as 5-10 %, or as a typical value such as 25 W/m run for insulated heating pipes, 100 W/m run for un-insulated pipes) to give the total load for each section. The heat loss will depend on pipe orientation (vertical/horizontal) and the quality and installation of insulation. Heat losses can be less than 5% if the pipes are well insulated.

More precise dependence of temperature change is given in Figure 3.7 for the case of uninsulated pipe.

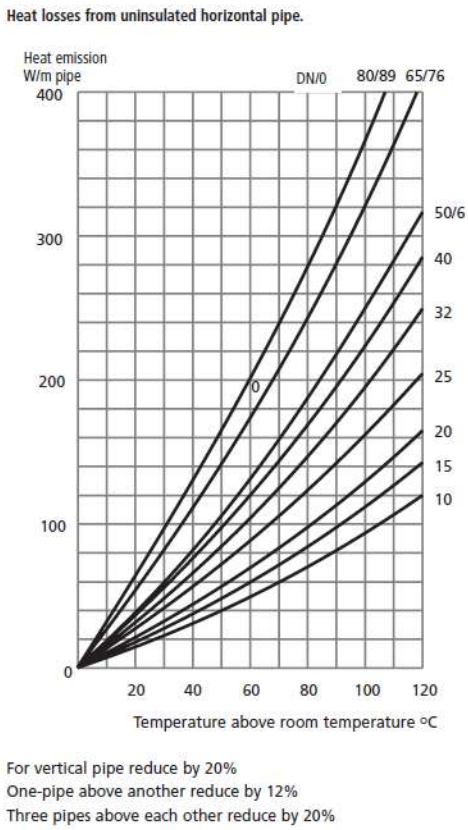


Figure 3.7: Heat losses from uninsulated horizontal pipe depending on the temperature difference of fluid and ambient temperature

For the verification of the measurements, the theoretical model described in 3.1 will be used, which will be compared to the results of the measurements on the real objects.

3.4 Algorithm for heat losses/gains of the pipework model

Algorithm for heat losses/gains of the pipework was developed based on the model described in 3.1.

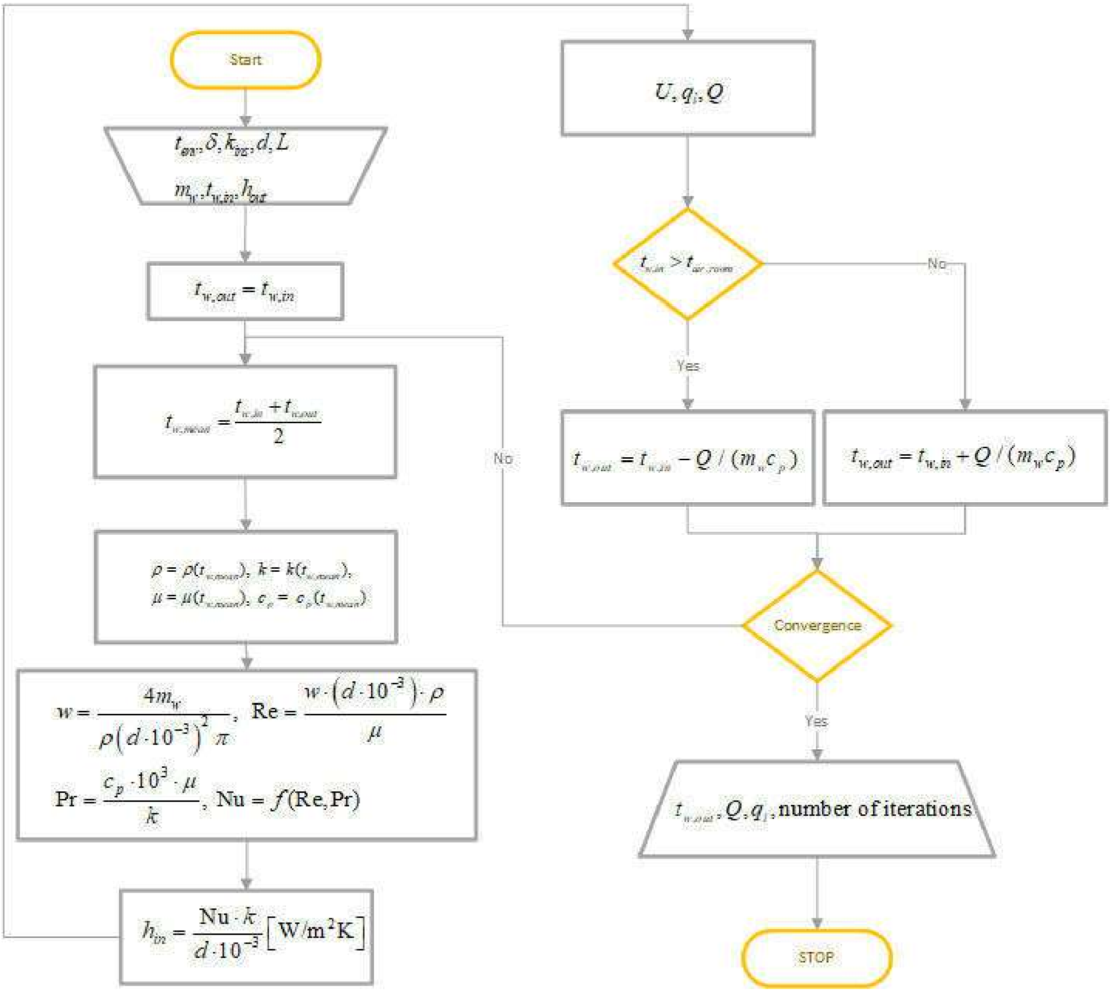


Figure 3.8: Algorithm for heat losses/gains of the pipework model

3.5 Approximate function for temperature change evaluation

With iterative model as a starting point, the simplification has been made in order to make the import of the model with other models easier.

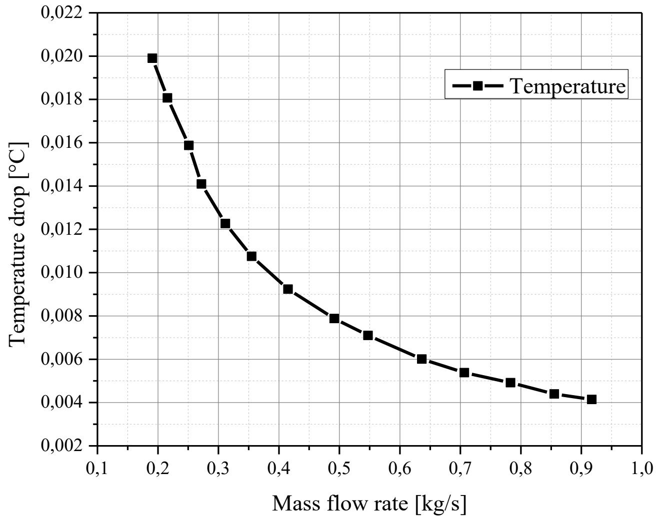


Figure 3.9: Diagram of temperature change with the variation of mass flow rate per unit pipe length

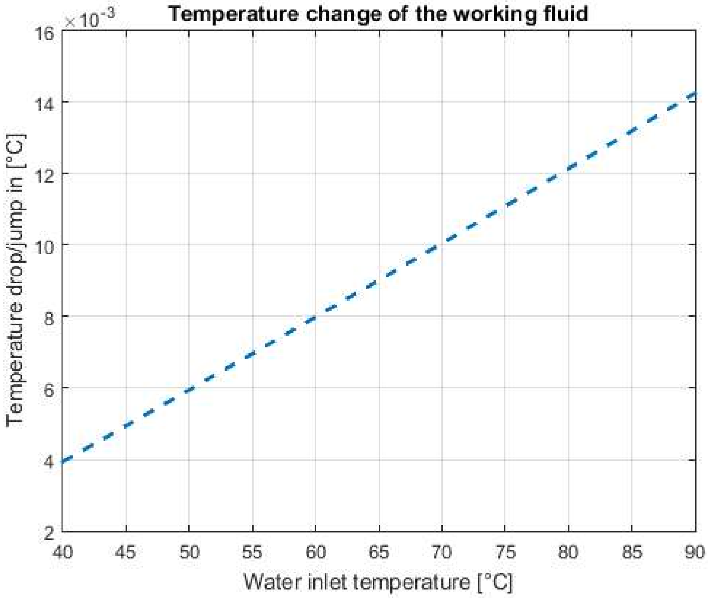


Figure 3.10: Diagram of temperature change with the variation of water inlet temperature per unit pipe length

The fluid temperature change along the section „*i*" of the length l_i :

$$\Delta T_{\mathrm{i}} = \xi_1 (a + b\frac{Q_{\mathrm{s,i}}}{3.6} + c(\frac{Q_{\mathrm{s,i}}}{3.6})^2) l_{\mathrm{i}} = \xi_1 (0.0251 + 2 \cdot 10^{-4} T_{\mathrm{w}}^{\mathrm{in}} - 0.062803 \cdot \frac{Q_{\mathrm{s,i}}}{3.6} + \\ + 0.0398896 \cdot (\frac{Q_{\mathrm{s,i}}}{3.6})^2) \cdot l_{\mathrm{i}} \tag{3.18}$$

where ξ_1 is the tuning factor, $Q_{s,i}$ is fluid mass flow [m3/h], and l_i represents the section "*i*" length expressed in meters. The initial value of the factor ξ_1 is $\xi_1 = 1$.

This model refers to an insulated pipe with 10 mm insulation thickness. For uninsulated pipes, the factor ξ_1 has higher values.

3.6 Approximate function for heat gain/loss evaluation

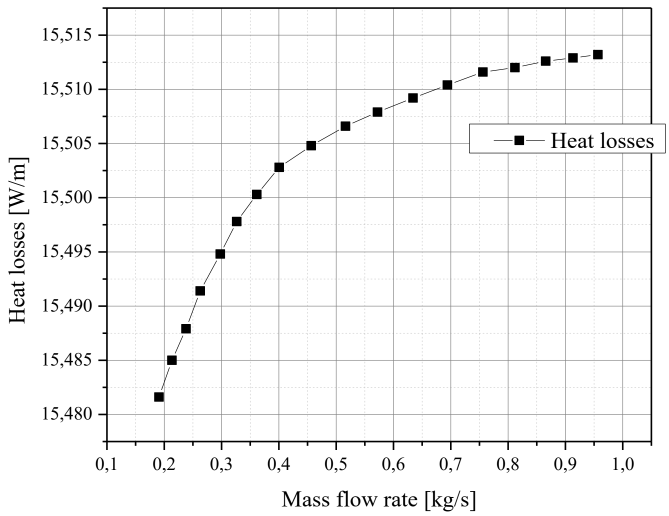


Figure 3.11: Diagram of heat losses with the variation of mass flow rate

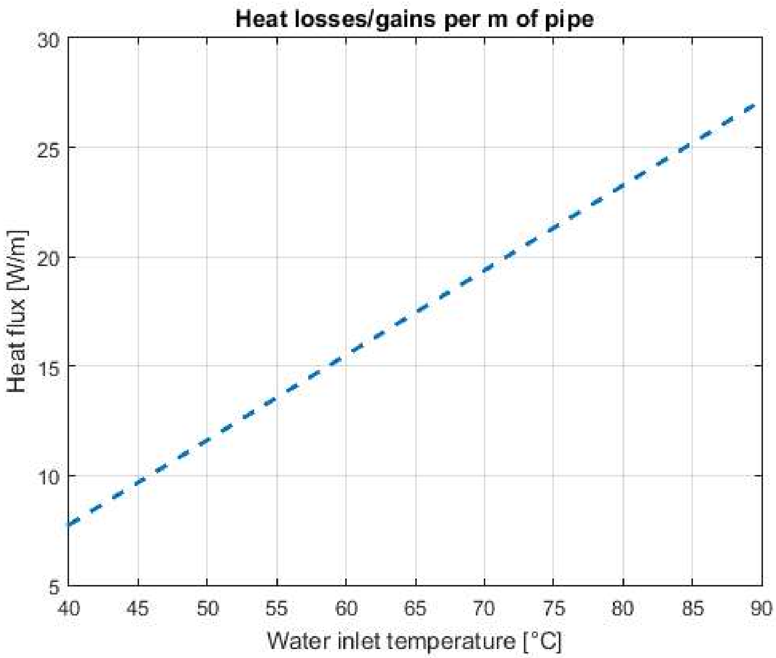


Figure 3.12: Diagram of heat losses with the variation of water inlet temperature

Heat loss along the section "i" can be determined according to the following formula:

$$\dot{Q}_{\mathrm{i}} = (0.4115 T_{\mathrm{w}}^{\mathrm{in}} + 6.553)\xi_2 l_{\mathrm{i}}\ [\mathrm{W}] \tag{3.19}$$

where ξ_2 is a tuning factor, l_i represents the length of the section i expressed in meters. The initial value of the factor ξ_2 is $\xi_2 = 1$.

This model refers to an insulated pipe with the 10 mm insulation thickness. For uninsulated pipes, the factor ξ_2 has higher values.

Off-line operation of the module –heat losses/gains on supply line

- compute parameters of the temperature drop/rise models from the central system to the entrance into each of the floor segments *i* which uses as inputs the medium flow at the output of the central system Q_{s} and the medium temperature at the output of the central system towards the building T_{s}
- for heat losses on the return line: compute parameters of the temperature model for temperature at the return of the central system $T_{\mathrm{w,hp}}^{\mathrm{in}}$ which uses as inputs the return flows $Q_{\mathrm{w,cal,i}}$ and return temperatures $T_{\mathrm{w,cal,i}}^{\mathrm{out}}$ coming from each of the floor segments denoted with *i*
- for flow shares: compute ratio parameter $\alpha_i = \frac{Q_{\mathrm{w,cal,i}}}{Q_{\mathrm{s}}}$ for each of the floor segments *i*

Inputs:

Parameter	Description	Format
$T_{\mathrm{s}}[°\mathrm{C}]$	Temperature of the medium coming out of the heat pump	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w,hp}}^{\mathrm{in}}[°\mathrm{C}]$	Temperature of the medium coming into the heat pump	Historical sequence of data with time-stamps, minutely sampled
$Q_{\mathrm{s}}\ [\mathrm{m}^3/\mathrm{h}]$	Medium flow through the heat pump	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w,cal,i}}^{\mathrm{in}}[°\mathrm{C}]$	Measured temperature at the medium entrance into the floor segment *i*	Historical sequence of data with time-stamps, minutely sampled
$T_{\mathrm{w,cal,i}}^{\mathrm{out}}[°\mathrm{C}]$	Measured temperature at the medium exit from the floor segment *i*	Historical sequence of data with time-stamps, minutely sampled
$Q_{\mathrm{w,cal,i}}\ [m^3/\mathrm{h}]$	Measured input/output flow for the floor segment *i*	Historical sequence of data with time-stamps, minutely sampled
Piping data		
Roughly estimated temperature of the space through which the pipes run	Roughly estimated c through which the pipes run	

Outputs:

Parameter	Description	Format
Parameters of the temperature drop/rise model on the supply line for each of the floor segments i (one model for each i)	The parameters calculated based on equation 3.18	
Parameters of the temperature model for return temperature into the central system based on floor segments outgoing temperatures and flows	The parameters calculated based on equation 3.18	
α_i	Parameters for flow shares	

On-line operation of the module: Evaluate the estimated models in off-line operation such that their outputs are computed and that they can be checked with actual measurements.

Estimated temperature drop based on equation 3.18 as well as the parameters calculated in off-line operation regime.

4 Submodule for estimation of the energy consumption of the central hydraulic pump (HVAC.PE.3)

Modulation of water flow rate is usually achieved by means of a two-port valve which throttles the water flow rate in response to a room temperature sensor. For radiators and radiant panels, thermostatic radiator valves (TRVs) are invariably used. These are low cost, self-actuating valves requiring no wiring: one TRV per emitter is used. For natural convectors a two-port valve connected to a room temperature sensor is common. Occasionally, three port valves are used to divert water away from a group of emitters, controlled from a room sensor. Regardless of how the flow modulation is achieved, the response of the emitter is the same. Figure 4.1 shows how heat output varies with water flow rate. The axes used represent fractional heat output and flow rate, i.e. the ratio of actual heat output (or flow rate) to the design value at full duty. The curve is based on a flow temperature of 80°C and a design return water temperature of 70°C. As can be seen, the response curve is highly non-linear: a large reduction in water flow rate results in only a small reduction in heat output rate. The curve becomes more linear as the design flow and return temperature difference are increased. Such non-linearity requires a valve characteristic which will produce a large reduction in flow rate for a small valve stem movement. Such valves are available but are relatively expensive. They might be used for controlling a group of emitters but are not economically attractive propositions for individual emitters. Less expensive valves need to be virtually closed to give a significant reduction in heat output rate. Furthermore, as the two-port valve closes, the pressure drop across it increases and so the valve has to close further. This results in poor controllability and possibly a valve that is unable to shut off completely [9].

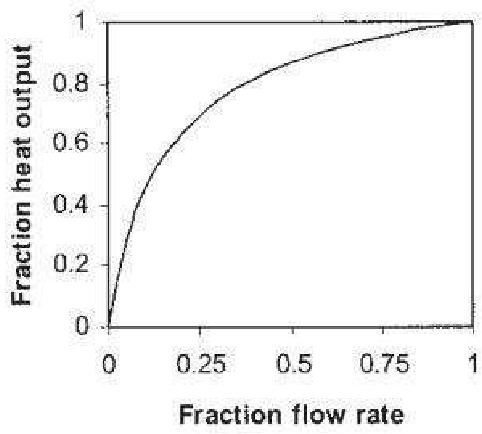


Figure 4.1: Variation of heat output rate with water flow rate.

Flow temperature modulation is achieved by blending of return water with the flow using a three-port 'mixing' valve. The response of the heat emitter to a change in flow water temperature is shown in Figure 4.2. As can be seen, unlike the two-port throttling valve, the response is very nearly linear so good control is achieved. Unfortunately, such valves and control systems are relatively expensive. As such, this system is used for controlling large groups of emitters [9].

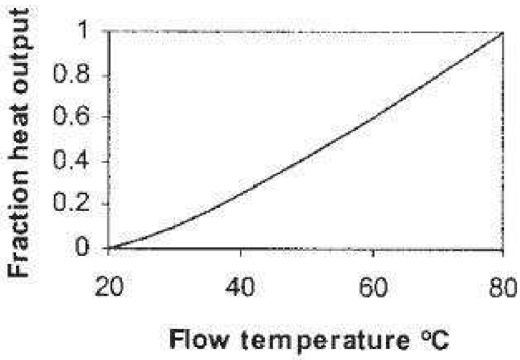


Figure 4.2: Variation of heat output rate with water flow temperature.

4.1 Theoretical background

Development of the module for Variable Frequency Drive (VFD) control of the circulation pump was carried out in two parts. The first part deals with the impeller characteristic curve. This curve is plotted on the pump manufacturer's performance specification for a given pump and it shows the relationship between the head (pressure loss) and capacity (flow rate) of a given pump/impeller combination. In addition to capacity and head data, information for efficiency, brake horsepower, and net positive suction head (NPSH) required are also generally plotted on the performance curve. These hydraulic data all simultaneously displayed on the same coordinate plane can be used for HVAC system analysis. The second part deals with what is commonly referred to as the system curve. It depicts a system's piping circuit resistance to flow at various flow rates. This curve is not presented on the pump manufacturer's performance curve because each piping system is unique and totally independent of the pump's manufactured performance. In brief, the characteristic curve is equipment dependent while the system curve is equipment independent. For mathematical modelling all mentioned curves must be digitized and presented as mathematical expressions with certain limitations which should be also provided.

Regarding the simplifications necessary for implementation of the developed module for VFD of the circulation pump presented in this report several assumptions were made:

- System curve is assumed to be constant despite changes of pressure loss from friction in pipeline due to flow rate change;
- HVAC system is assumed to be equipped with one circulation pump;
- VFD control of the circulation pump is made according to differential pressure at the pump inlet and outlet;

According to model description and presented assumptions for the particular case, overall methodology for HVAC.PE.3 submodule development is presented in Figure 4.3.

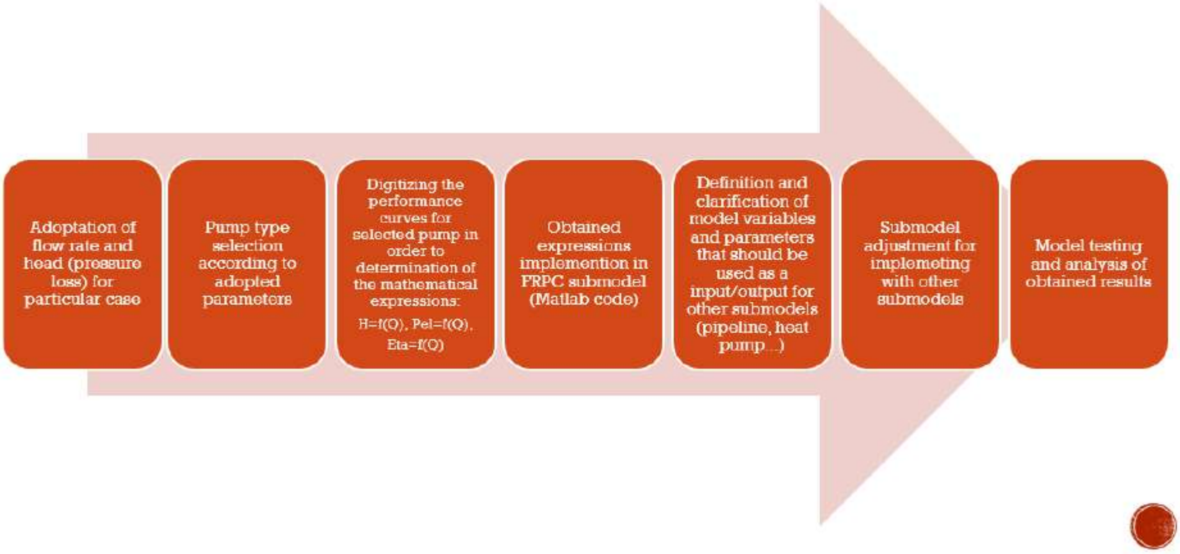


Figure 4.3: Submodule development methodology

Circulation pump selection and performance curves

First steps in the methodology to be presented consist in the adaptation of pump parameters due to heat demands for heating/cooling operation which should result in the appropriate pump model selection. Model selection diagrams provided by pump manufacturers will be used for obtaining the pump model. The performance curves obtained from the model will then be used for submodule development. Example for pump selection according to the adopted flow rate (up to 16 m^3/h) and adopted head (up to 7 m) are presented in Figure 4.4.

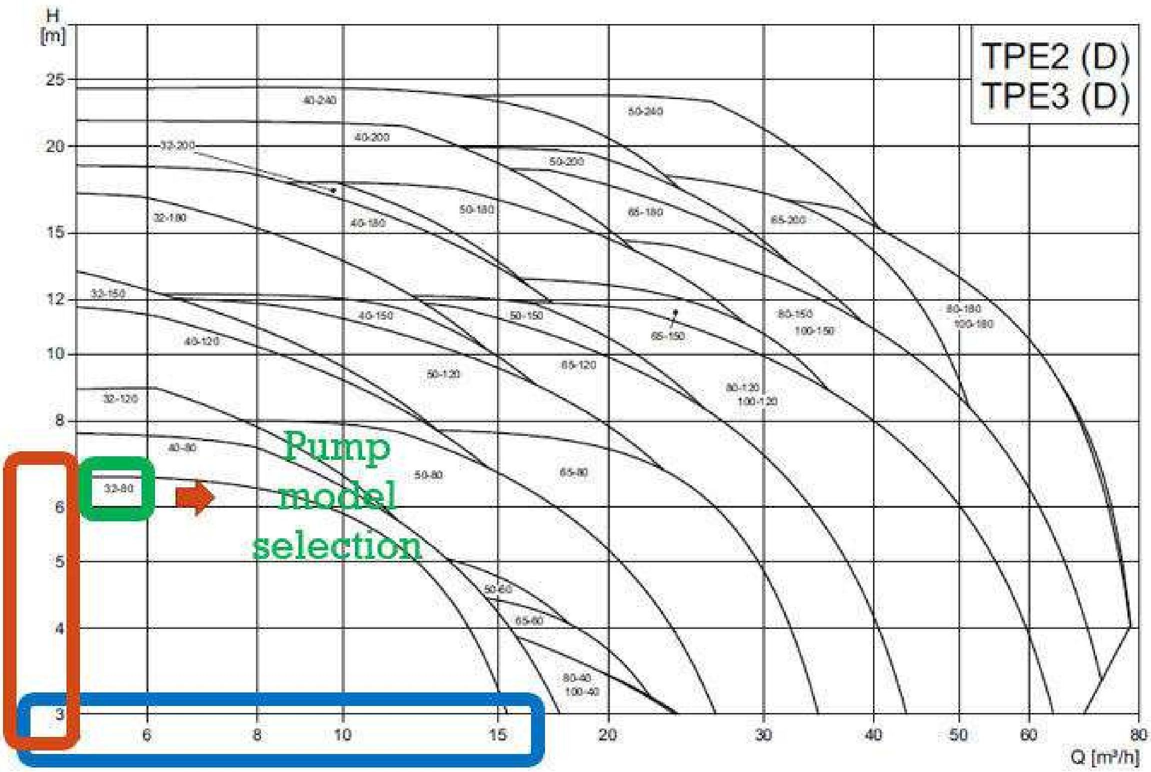


Figure 4.4: Example of pump model selection

Performance curves for the selected pump model were obtained also according to the provided manufactures data from pump specifications and characteristics. Performance curves consist of flow rate dependencies of head, brake horsepower, efficiency and NPSH, and for selected pump model in the previous example are presented in Figure 4.5.

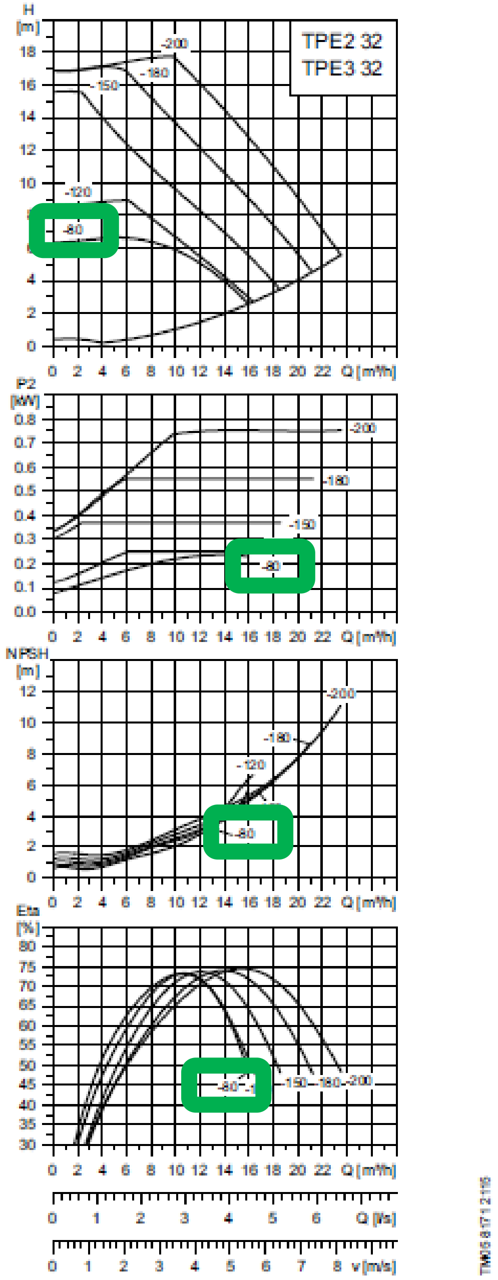


Figure 4.5: Performance curves for the selected pump model [9]

Formation of mathematical expressions

It has been shown that the typical pump performance curve takes the form of a parabolic-type power function in the form,

$$y = k \cdot x^n \text{ , where n > 0 .....(4.1)}$$

For all practical purposes, the head, brake horsepower, efficiency and capacity of a centrifugal pump can be considered to vary parabolically and usually can be represented by a quadratic expression in the standard form,

$$y = a \cdot x^2 + b \cdot x + c \ldots\ldots\ldots..(4.2)$$

Rewriting in the hydraulic form for all data presented on pump performance curves yields,

$$\begin{aligned} H &= a_1 \cdot Q^2 + b_1 \cdot Q + c_1 \\ P_{\mathrm{el}} &= a_2 \cdot Q^2 + b_2 \cdot Q + c_2 \ldots\ldots\ldots.(4.3) \\ \eta &= a_3 \cdot Q^2 + b_3 \cdot Q + c_3 \end{aligned}$$

Coefficients a_i, b_i *and* c_i (i = 1, 2, 3) in previous equations are calculated according to least squares data approximation which are obtained by digitizing performance curves of the selected pump. Examples of the digitizing and approximation procedure for determination of coefficients for P_{el} are presented in Figure 4.6.

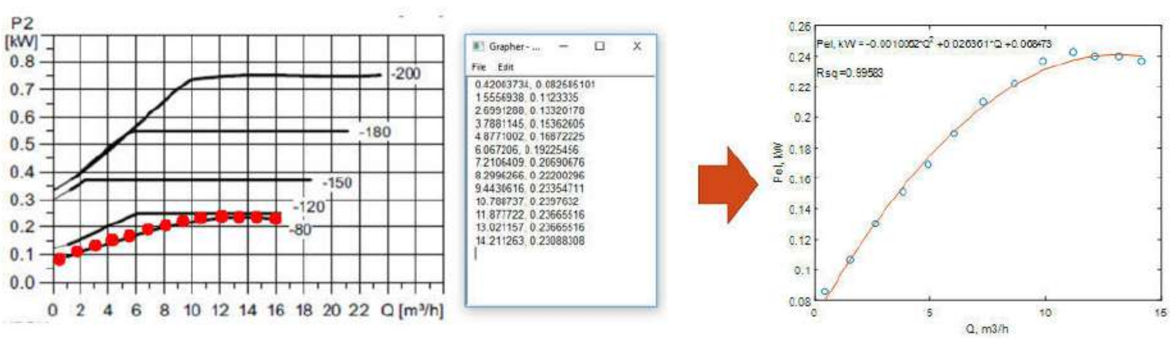
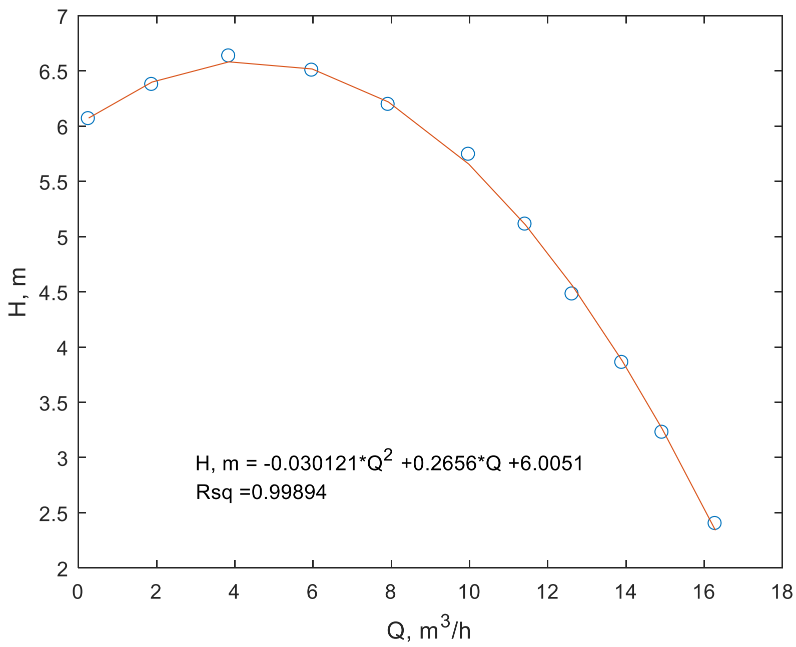


Figure 4.6: Examples of the digitizing and approximation procedure for determination of coefficients for P_{el}

Results for determination of coefficients for all pump performance curves are presented in Figure 4.7.



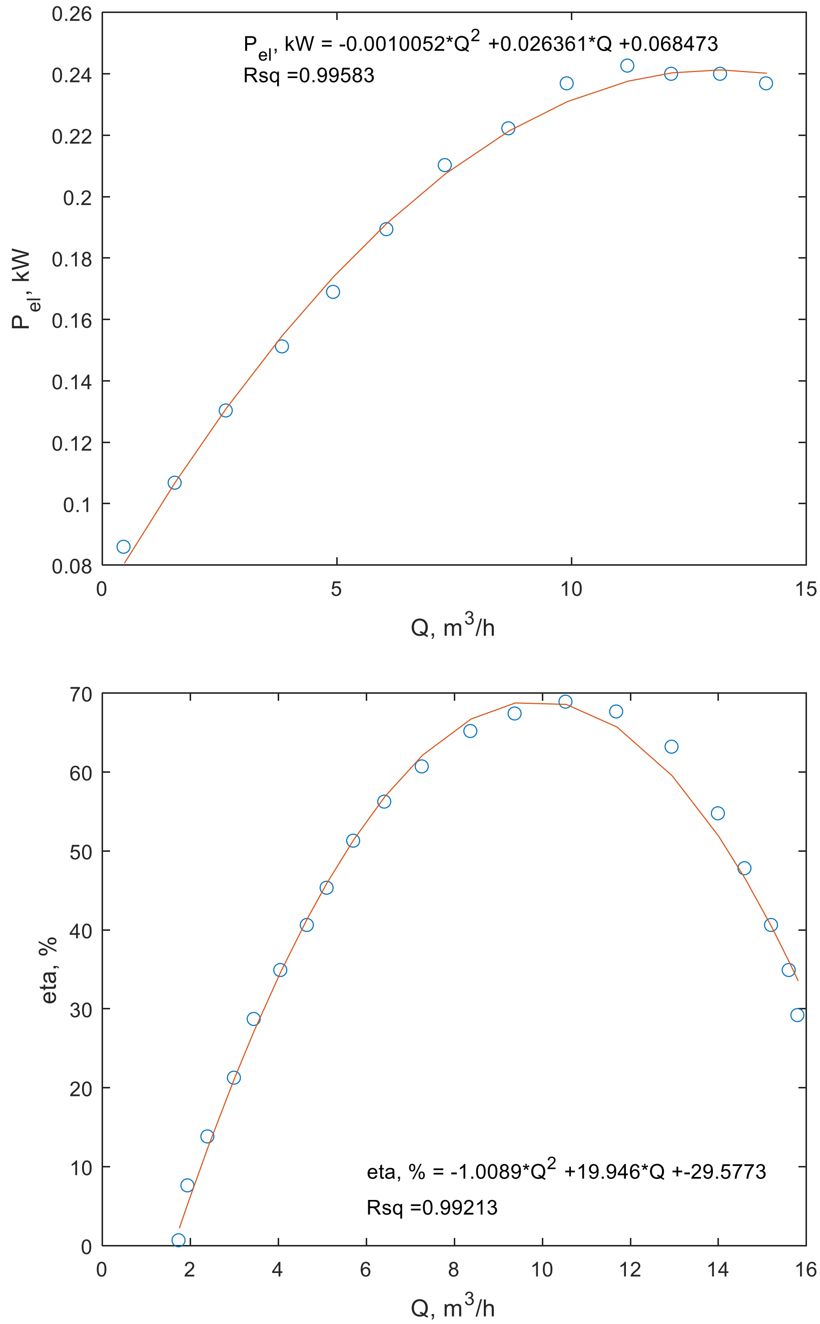


Figure 4.7: Results for determination of coefficients for all pump performance curves

Results of mathematical expressions formation for the selected pump model obtained by performance curves presented in the previous example, which can be used in developing FRPC submodule are given as:

$$H = -0.0301 \cdot Q^2 + 0.2656 \cdot Q + 6.0051; (R^2 = 0.99894)$$
$$P_{\mathrm{el}} = -0.0010052 \cdot Q^2 + 0.026361 \cdot Q + 0.068473; (R^2 = 0.99583) \dots\dots\dots (4.4)$$
$$\eta = -1.0089 \cdot Q^2 + 19.946 \cdot Q - 29.5773;\ (R^2 = 0.99213)$$

Where, H [m] is head or pump pressure loss, P_{el} [kW] pump brake horsepower, η [%] pump efficiency and Q [m^3/h] pump capacity or flow rate. Limitations for presented mathematical expressions are that it can be used only for pump flow rates from 0 up to 16 m^3/h, due to selection limits of the pump model. Detailed VFD model description is presented in the following chapter.

4.2 HVAC.PE.3 model and algorithm

HVAC.PE.3 model for HVAC systems are defined according to algorithm presented in the Figure 4.8. Primary aims of the model are prediction and estimation of energy consumption of the hydraulic pump with VFD due to change of flows as an issue of different heat demands in the HVAC systems. Additionally, HVAC.PE.3 model should provide key parameters for controlling the circulation pump by central control unit. Also during the operational phase in order to obtain necessary data and input parameters for the model function, the calibration process should be performed.

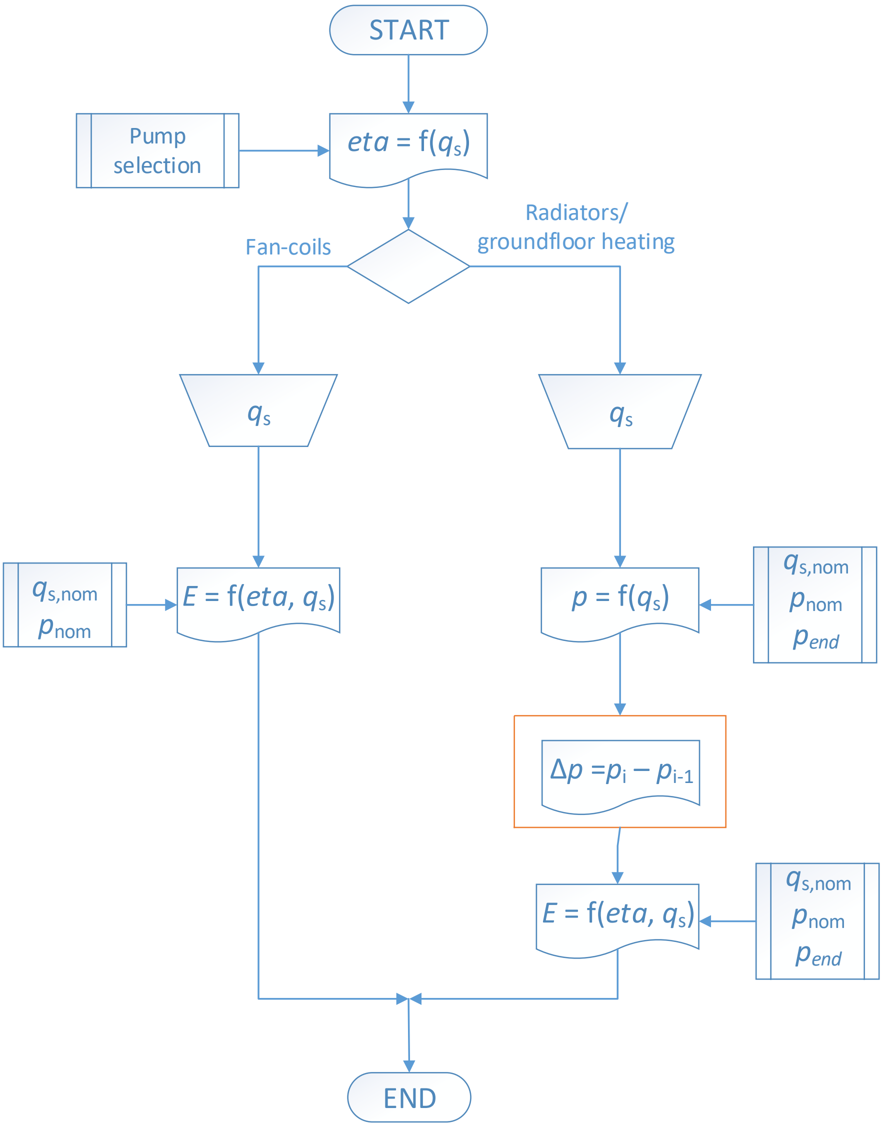


Figure 4.8: Algorithm for variable frequency drive for circulation pump

HVAC.PE.3 model should be considered in order to comprehend HVAC systems with fan coils (FC) and with radiators/ground floor heating elements (RG). Also during calibration process determination of the nominal pump flow ($Q_{s,nom}$) and nominal differential pressure on the inlet/outlet of the pump (p_{nom}) is necessary with the HVAC systems with FC. For the HVAC systems with RG additionally should be carried out determination of (p_{end}) differential pressure measured on the „last" radiator/ground floor heating element in the RG system longest line. These parameters obtained during calibration process are used as constant in the model equations for prediction and estimation of the circulation pump energy consumption in the HVAC systems. But fine tuning of the model should be considered and adaption of the model parameters should be performed for each test site in particular. Detailed formulation of the HVAC.PE.3 model will be presented in the following chapter.

4.3 HVAC.PE.3 Model

Regarding to project description HVAC.PE.3 model should be considered in two cases:

- CASE 1 - with fan coils (FC) as heating/cooling elements in the system
- CASE 2 - with radiators or floor heating elements (RG) in the system

4.3.1 Fan coils as heating/cooling elements in the system

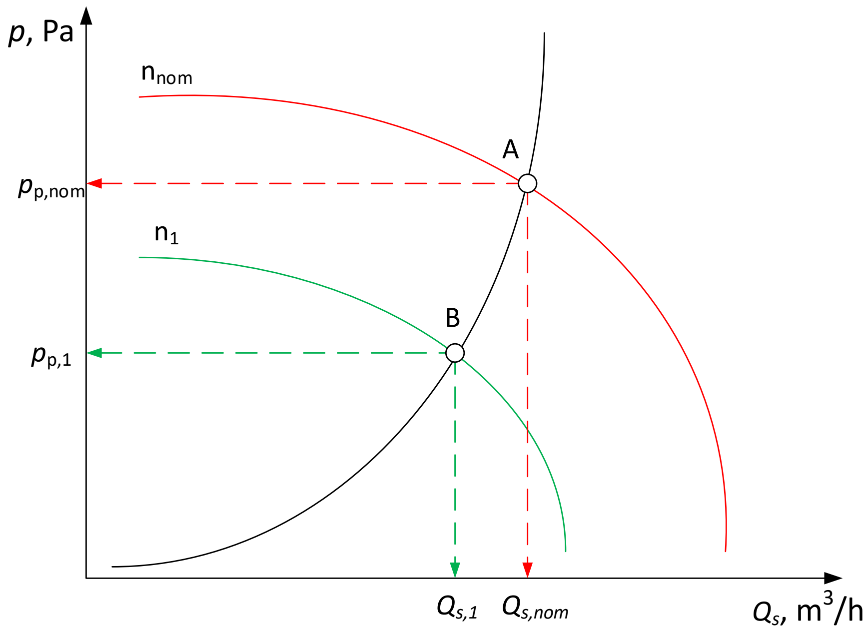


Figure 4.9: Pump diagram

Point A represents a pump working point at nominal power with maximum flow through fan coils (FCs). Point B represents a pump operating point when changing the operational mode due to flow reduction through FC.

The following symbols are used:

$Q_{\mathrm{s,nom}}$, m^3/h – volume flow at nominal working regime

H_{nom}, m – head (pressure drop) at nominal working regime.

n_{nom}, rpm – pump rpm at nominal working regime.

Q_{s} , m^3/h – volume flow required by FC

Power of the pump at nominal working regime can be calculated as:

$$P_{\mathrm{p,nom}} = \frac{Q_{\mathrm{s,nom}} \cdot H_{\mathrm{nom}} \cdot 9.81 \cdot \rho}{\eta_{\mathrm{uk}}}, \text{ kW} \quad \ldots\ldots\ldots\ldots\ldots\ldots(4.5)$$

$\rho = 1000 \frac{\mathrm{kg}}{\mathrm{m}^3}$ - water density (adopted value)

According to affinity laws:

$$\frac{Q_{\mathrm{s,nom}}}{Q_{\mathrm{s,1}}} = \frac{n_{\mathrm{nom}}}{n_1} \Rightarrow n_1 = n_{\mathrm{nom}} \cdot \frac{Q_{\mathrm{s,1}}}{Q_{\mathrm{s,nom}}} \quad \ldots\ldots\ldots(4.6)$$

$$\frac{P_{\mathrm{p,nom}}}{P_{\mathrm{p}}} = \frac{n_{\mathrm{nom}}^3}{n_1^3} \Rightarrow P_{\mathrm{p}} = P_{\mathrm{p,nom}} \cdot \frac{n_1^3}{n_{\mathrm{nom}}^3} = P_{\mathrm{p,nom}} \cdot \frac{n_{\mathrm{nom}}^3 \cdot Q_{\mathrm{s}}^3}{n_{\mathrm{nom}}^3 \cdot Q_{\mathrm{s,nom}}^3} \quad \ldots\ldots\ldots\ldots(4.7)$$

Power in kW at flow $Q_1\left(q_s\right)$ for constant value of total efficiency.

$$P_{\mathrm{p}} = P_{\mathrm{p,nom}} \cdot \frac{Q_{\mathrm{s}}^3}{Q_{\mathrm{s,nom}}^3} \quad \ldots\ldots\ldots\ldots\ldots\ldots(4.8)$$

Where is power for nominal pump regime given by:

$$P_{\mathrm{p,nom}} = \frac{9.81 \cdot Q_{\mathrm{s,nom}} \cdot H_{\mathrm{nom}}}{\eta_{\mathrm{uk}}} \quad \ldots\ldots\ldots\ldots\ldots(4.9)$$

Head at nominal working regime could be expressed as:

$$H_{\mathrm{nom}} = 10.2 \cdot \frac{p_{\mathrm{p,nom}}}{10^5 \cdot SG} \quad \ldots\ldots\ldots\ldots\ldots\ldots(4.10)$$

Total efficiency of the pump should be determined according to the digitizing procedure of performance curves for the specific pump and for this case (regarding to maximum flows and head) it can be adopted from (4.4) for flow range:

$$Q_s = 0 - 15\frac{\mathrm{m}^3}{\mathrm{h}} \rightarrow \eta_{\mathrm{uk}} = \eta_{\mathrm{p}} = -1.0089 \cdot Q_s^2 + 19.946 \cdot Q_s - 29.5773 \text{.......(4.11)}$$

$$Q_s = 15 - 70\frac{\mathrm{m}^3}{\mathrm{h}} \rightarrow \eta_{\mathrm{uk}} = \eta_{\mathrm{p}} = -0.0592 \cdot Q_s^2 + 5.2014 \cdot Q_s - 39.889 \text{ ....(4.12)}$$

Finally (4.9) with consideration of eqs. (4.10) and (4.12) given the nonlinear hydraulic pump model as:

$$P_{\mathrm{p}} = \frac{9.81 \cdot \frac{Q_{\mathrm{s,nom}}}{3.6} \cdot (10.2 \cdot \frac{p_{\mathrm{p,nom}}}{10^5 \cdot SG})}{\eta_{\mathrm{p}}} \cdot \frac{Q_s^3}{Q_{\mathrm{s,nom}}^3} \text{ ....(4.13)}$$

Where following annotations are used:

Q_s , m^3/h – pump flow

$Q_{s,nom}$, m^3/h – nominal pump flow (when maximum flow throw fan-coils / all radiators are opened)

$p_{p,nom}$, Pa – nominal differential pressure measured on pump inlet/outlet (when maximum flow throw fan-coils / all radiators are opened)

SG = 1, - specific gravity of water

η_p – total pump efficiency obtained by fitting the curve ($\eta_p = f(Q_s)$) which is provided by the manufacturer of the pump.

4.3.2 Radiators or floor heating elements in the system

Regarding to literature review this case should have considered different hydraulic pump control modes in order to reduce energy consumption in the system. Due to proper selection of the hydraulic pump operational mode in the following chapters the behavior of uncontrolled and controlled pumps will be shown.

The operation of uncontrolled pumps

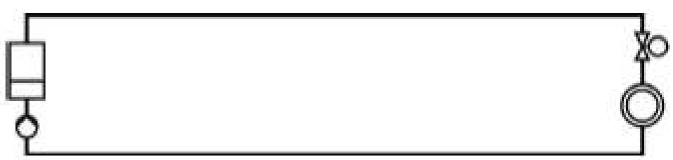


Figure 4.10: The operation of uncontrolled pumps scheme

If the valve closes, the resistance increases and the volumetric flow decreases. Hence, the plant characteristic becomes steeper. Due to the higher resistance in the piping network the pump needs to provide a higher pressure.

With uncontrolled pumps the speed n remains constant and the operating point follows the pump characteristic to the left.

The example shown below demonstrates the shifting of the operating point at part load 50 % and as a result, the related changes in the energy consumption of the pump.

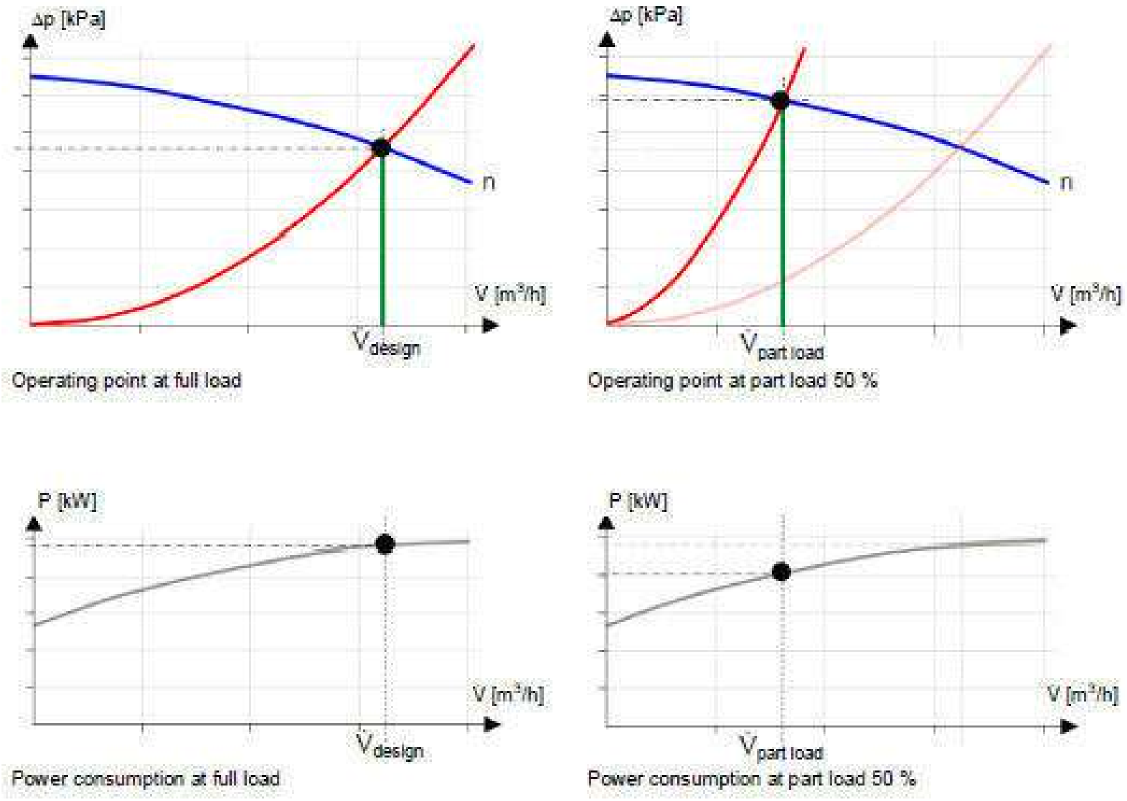


Figure 4.11: Operating point and power consumption at full and part load of uncontrolled pumps

The operation of controlled pumps can be carried out with:

1. Constant pump pressure

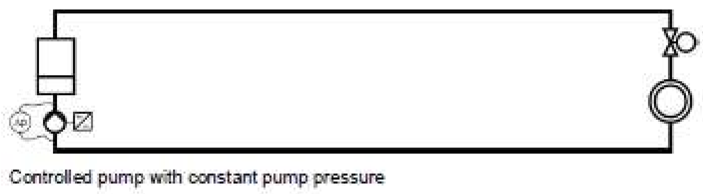


Figure 4.12: Controlled pump with constant pump pressure scheme

At part load the pressure across the pump is kept constant. This can be controlled either electronically in the pump itself or with a pressure dependent control and a variable speed drive at the pump. The operating point follows the line of constant pressure horizontally to the left.

The example shown below demonstrates the shifting of the operating point at part load 50 %

and as a result, the altered energy consumption of the pump.

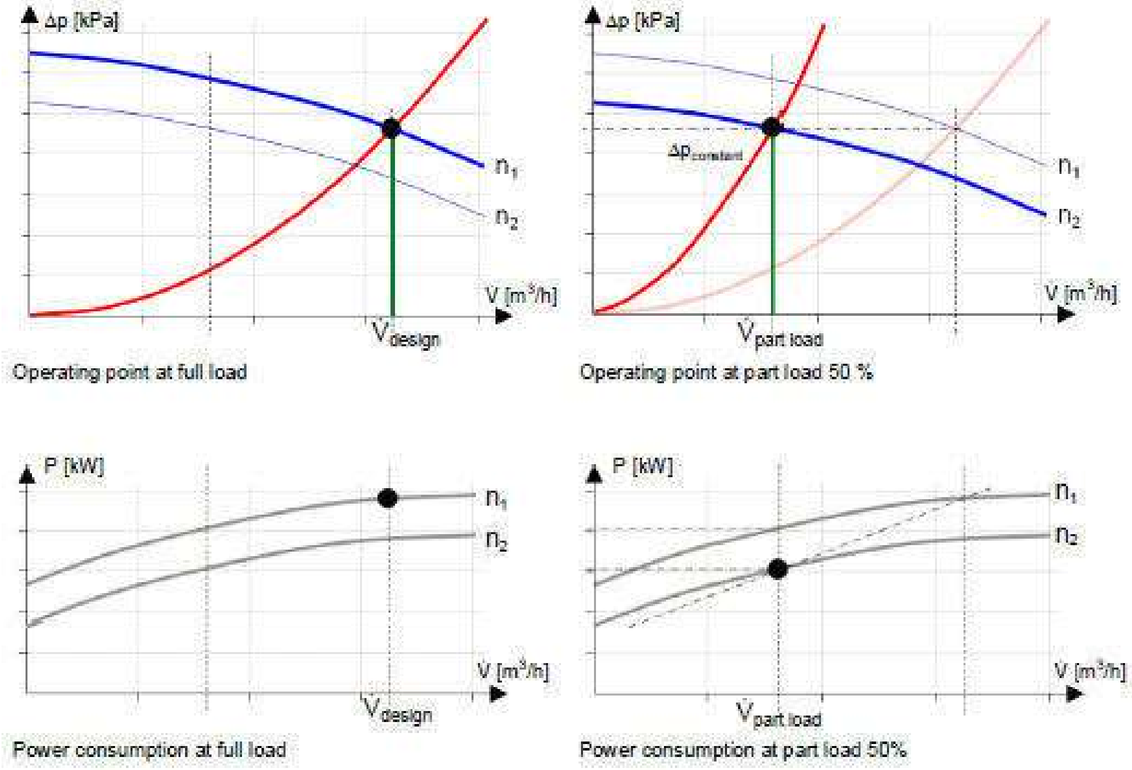


Figure 4.13: Operating point and power consumption at full and part load of constant pressure controlled pumps

2. Constant differential pressure across the end of the plant

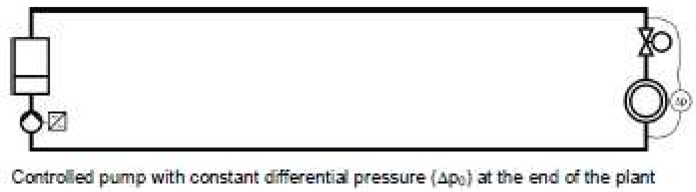


Figure 4.14: Controlled pump with constant differential pressure scheme

The differential pressure Δp_0 is held constant across the end of the plant. There are two possibilities to achieve this constant pressure at the end:

- a measuring point at the end of the plant, connected to a pressure controller and a variable frequency drive (VFD) at the pump
- an electronic control in the pump itself ("Δp variable" control)

The operating point follows the control slope that runs towards Δp_0 near $\dot{V} = 0\ m^3/h$
The example shown below demonstrates the shifting of the operating point at part load 50% and as a result, the related changes in the energy consumption of the pump.

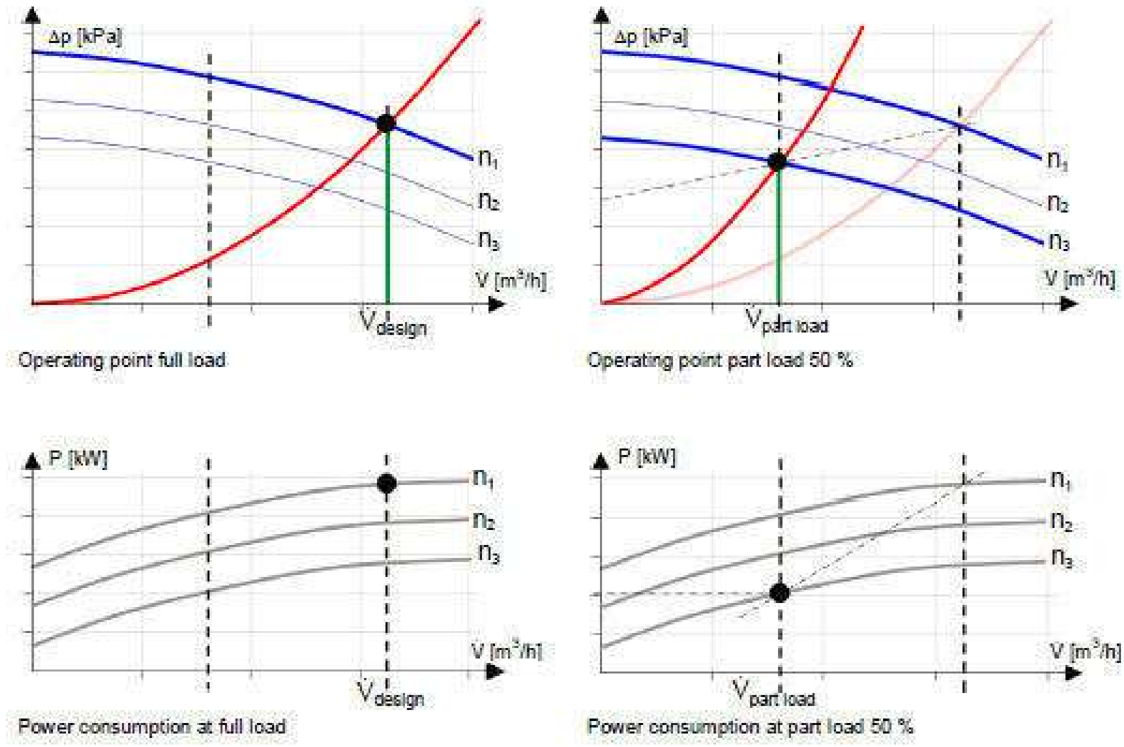


Figure 4.15: Operating point and power consumption at full and part load of constant differential pressure controlled pumps

The plant characteristic is steeper at part load (50 %). Due to the reduced volumetric flow the resistance in the plant is reduced as well. The controlling across the end of a plant ensures that the necessary differential pressure there is still maintained.
With a controlled pump with the measuring point at the end, the energy consumption of the pump is even further reduced [10].

Energy savings with controlled pumps

For plants with variable volumetric flows, controlled pumps save energy very efficiently. The selection of the control system depends on the situation on site (distances, investments, etc.) As shown in the chart below, controlled pumps consume less power. Thus, energy and costs can be saved. A pump with constant differential pressure across the end of the plant is more efficient than a controlled pump with a constant pump pressure.
The chart below shows the saving capacity on the basis of a data sheet of a pump.

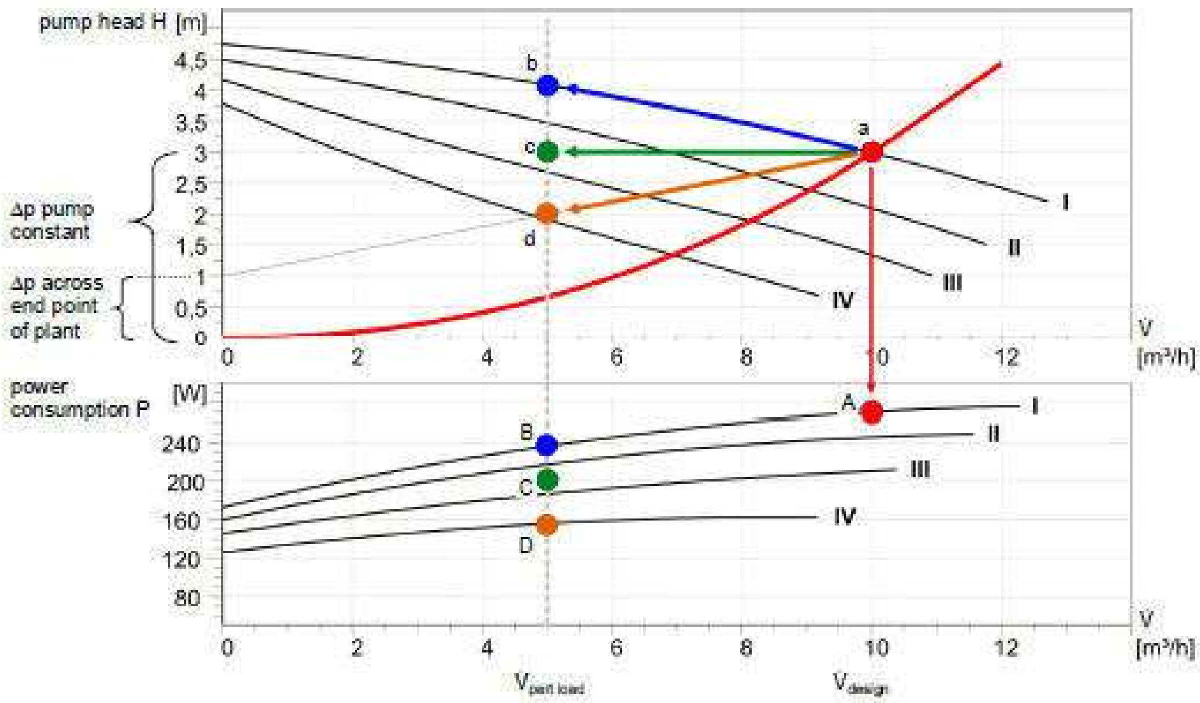


Figure 4.16: Operating point and power consumption for controlled pumps

Operating points and power consumption in comparison

a: operating point, design

b: operating point, part load, uncontrolled

c: operating point, part load, controlled Δp_{pump}

d: operating point, part load, controlled Δp_{end}

A: power consumption, design

B: power consumption, part load, uncontrolled

C: power consumption, part load, controlled Δp_{pump}

D: power consumption, part load, controlled Δp_{end}

According to presented control modes HVAC.PE.3 model is defined for constant differential pressure across the end of the plant as follows:

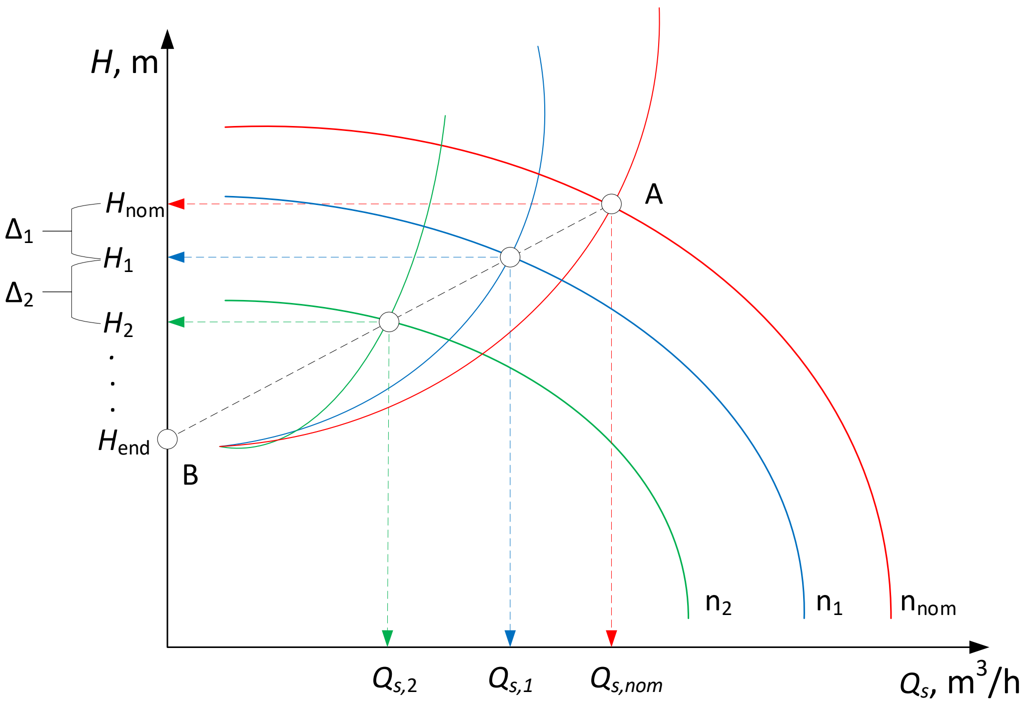


Figure 4.17: Pump diagram

Point A represents a pump working point at nominal power with maximum flow through radiators/floor heating elements. Point B considers theoretical pump operating point when all radiators/floor heating elements are closed and the flow through the pump is equal 0. The operational modes of the pump due to flow reduction by exclusion of particular radiator or floor heating element from the system are presented by points between.

Where following annotations are used:

$Q_{s,nom}$, [m^3/h] – volume flow at nominal working regime

H_{nom} , [m] – head (pressure drop) at nominal working regime.

n_{nom} , rpm – pump rpm at nominal working regime.

$H_1, H_2, \ldots$,[m] – Head (or differential pressure) on the pump when particular radiator/floor heating elements is closed

$Q_{s1}, Q_{s2}, \ldots$, [m^3/h] – reduced volume flow through the system caused with closing the radiators/heating elements

H_{end} ,[m] – Head (or differential pressure) by theoretical case when all radiators/floor heating elements are closed and volume flow through the pump is zero. It can be measured on the “last” radiator/floor heating element on the system and it is used for regulating the flow.

In order to define nonlinear VFD hydraulic pump model for this case the equation for the line through points A and B should be determined by

$$H = H_{\mathrm{end}} + (H_{\mathrm{nom}} - H_{\mathrm{end}}) \cdot \frac{Q_{\mathrm{s}}}{Q_{\mathrm{s,nom}}} \quad .....(4.14)$$

Possible cases:

1. All radiators are open:

$$Q_{\mathrm{s}} = Q_{\mathrm{s,nom}} \to H = H_{\mathrm{nom}} \quad .....(4.15)$$

2. Radiator no.1 is closed:

$$Q_{\mathrm{s}} = Q_{\mathrm{s,1}} \to H_1 = H_{\mathrm{end}} + (H_{\mathrm{nom}} - H_{\mathrm{end}}) \cdot \frac{Q_{\mathrm{s,1}}}{Q_{\mathrm{s,nom}}} \quad .....(4.16)$$

then $\Delta_1 = H_{nom} - H_1$ is pressure drop on Radiator no.1

3. Radiators no.1 and no.2 are closed:

$$Q_{\mathrm{s}} = Q_{\mathrm{s,2}} \to H_2 = H_{\mathrm{end}} + (H_{\mathrm{nom}} - H_{\mathrm{end}}) \cdot \frac{Q_{\mathrm{s,2}}}{Q_{\mathrm{s,nom}}} \quad .....(4.17)$$

then $\Delta_2 = H_1 - H_2$ is pressure drop on Radiator no.2

4. All radiators are closed (theoretical case):

$$Q_{\mathrm{s}} = 0 \to H = H_{\mathrm{end}} \quad .....(4.18)$$

According to equation 4.14 with consideration of equations 4.16, 4.17 and 4.18, VFD hydraulic pump model for case with radiators/floor heating elements can be given as

$$P_{\mathrm{p}} = \frac{9.81 \cdot \frac{Q_{\mathrm{s}}}{3.6} \cdot (10.2 \cdot \frac{p_{\mathrm{end}}}{10^5 \cdot SG} + (10.2 \cdot \frac{p_{\mathrm{p,nom}}}{10^5 \cdot SG} - 10.2 \cdot \frac{p_{\mathrm{end}}}{10^5 \cdot SG}) \cdot \frac{Q_{\mathrm{s}}}{Q_{\mathrm{s,nom}}}}{\eta_{\mathrm{p}}} \quad ..... (4.19)$$

Where following annotations are used:

Q_{s} , [m3/h] – pump flow

$Q_{\mathrm{s,nom}}$, [m3/h] – nominal pump flow (when maximum flow throw fan-coils / all radiators are opened)

$p_{\mathrm{p,nom}}$, [Pa] – nominal differential pressure measured on pump inlet/outlet (when maximum flow through fan-coils / all radiators are opened)

p_{end}, [Pa] – differential pressure at inlet/outlet measured on the „last" radiator in the longest line - only for radiators/ground floor heating (when all radiators are opened)

SG = 1, - specific gravity of water

η_{p} – total pump efficiency obtained by fitting the curve ($\eta_{\mathrm{p}} = f(Q_{\mathrm{s}})$) which is provided by the manufacturer of the pump.

4.3.3 Analysis of application on particular test-sites

Application of the HVAC.PE.3 module as a part of the overall central HVAC module design on particular test-sites should consider the following:

- For selected circulation pump from all test-sites should be provided pump characteristics efficiency vs flow ($\eta_\mathrm{p} = f(Q_\mathrm{s})$);
- output for the hydraulic pump submodule will be pump parameters for electricity consumption for different operating regimes and for different heating/cooling element configurations.
- Additional demands for input variables which will be used in hydraulic pump submodule (i.e. parameters which should be measured on particular test-site parameters identification) are pump flow and differential pressure on the pump ($Q_\mathrm{s,nom}$; p_nom) at nominal regime. For radiators/ground floor heating systems, input variables are differential pressure on the last element or differential pressure on the pump when all the elements were closed ($p_\mathrm{end}(Q_\mathrm{s}=0)$).

4.3.4 HVAC.PE.3 model parameters

Main purpose of the module HVAC.PE.3 – compute parameters of the electrical energy consumption model for the circulation pump, in two configurations:

- Case a - static hydraulic situation with fan coils
- Case b - configuration with radiators or floor heating/cooling

<u>Off-line module operation for Case a</u>: (model of the energy consumption that is a function of Q_s, identification performed e.g. once a month based on collected data)

INPUT		FORMAT
$\eta_\mathrm{p} = f(Q_\mathrm{s})$	total pump efficiency obtained by fitting the curve which is provided by the pump manufacturer	Pump data
$p_\mathrm{p}\left[\mathrm{Pa}\right]$	Measurement of pressure drop;	Historical data, minutely sampled
$Q_\mathrm{s}\left[\mathrm{m}^3/\mathrm{h}\right]$	Measurement of flow	Historical data, minutely sampled

OUTPUT		FORMAT
$f\left(Q_\mathrm{s}\right)$ proportional to Q_s^3	Parameters of the electrical energy consumption model for eq. 4.13	The procedure for parameters identification is implemented in Python

On-line operation for Case a consider:

OUTPUT		FORMAT
$P_{\mathrm{p}} \cdot \Delta\tau = f\left(Q_{\mathrm{s}}\right)$	Energy consumption for the pump based on currently sampled data of flow Q_{s} eq. 4.13	The procedure for on-line evaluation is implemented in Python

Off-line module operation for Case b: (model of the energy consumption that is a function of $Q_{\mathrm{s,nom}}, Q_{\mathrm{s}}, p_{\mathrm{end}}$, performed e.g. once a month based on collected data)

INPUT		FORMAT
$\eta_{\mathrm{p}} = f(Q_{\mathrm{s}})$	total pump efficiency obtained by fitting the curve which is provided by the pump manufacturer	Pump data
$p_{\mathrm{p}}\left[\mathrm{Pa}\right]$	Measurement of pressure drop;	Historical data, minutely sampled
$Q_{\mathrm{s}}\left[\mathrm{m}^3/\mathrm{h}\right]$	Measurement of flow	Historical data, minutely sampled

OUTPUT		FORMAT
$f\left(Q_{\mathrm{s,nom}}, Q_{\mathrm{s}}, p_{\mathrm{end}}\right)$	Parameters of the electrical energy consumption model for eq. 4.19	The procedure for parameters identification is implemented in Python

On-line operation for Case b consider:

OUTPUT		FORMAT
$P_{\mathrm{p}} \cdot \Delta\tau = f\left(Q_{\mathrm{s}}\right)$ and set $Q_{\mathrm{s,nom}}, p_{\mathrm{end}}$	Energy consumption for the pump based on currently sampled data of flow Q_{s} eq. 4.19	The procedure for on-line evaluation is implemented in Python

5 Non-controllable consumption prediction submodule (HVAC.PE.4)

Submodule for prediction of the total non-controllable energy consumption on the central HVAC unit.

5.1 Submodule inputs

Table 5.1: Required inputs for non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Historical profile of the non-controllable energy consumption on the central HVAC unit	$E_{\mathrm{t,nc}}$	Non-controllable thermal energy consumption on the HVAC level
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I_{diff}^{h}, I_{dir}^{\mathrm{n}}$ Remaining pilot sites: $T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

5.1.1 Non-controllable thermal energy consumption

Non-controllable thermal energy consumption on the HVAC level represents the HVAC level thermal energy consumption that is not controlled by zones or HVAC MPC modules. It is measured/calculated differently on each considered pilot site depending on the configuration of the pilot site HVAC level and available measurement equipment present on the site. Since very often different systems are utilized during heating and cooling operation regimes, the non-controllable consumption is measured/calculated differently depending on the current operation regime. Determination of the non-controllable thermal energy consumption for different pilot sites is presented in the following Table 5.2.

Table 5.2 Non-controllable thermal energy consumption determination.

Pilot site	Heating regime	Cooling regime
UNIZG-FER	Consumed thermal energy on the heating substation primary side meter (billing meter) - consumed thermal energy on all fan coils	Consumed thermal energy on the calorimeter for cold medium supply towards B buildings climatization
HEP	Consumed heat on the billing meter - calculated consumption on all controllable zone elements - calculated heat loss on the horizontal ducts - calculated heat loss on the supply and return vertical	Cooling energy produced by the water chiller - data from the main calorimeter entering the pilot building
Idrija (school building)	Consumed heat on the billing meter - calculated consumption on all controllable zone elements - calculated heat loss on the supply and return verticals	
Idrija (sports centre building)	Consumed heat on the billing meter - calculated consumption on all controllable zone elements - calculated heat loss on the supply and return verticals	
EON	Consumed heat on the central calorimeter - (consumed heat in zones with fan coils measurements + calculated energy loss on the vertical supply lines)	Consumed heat on the central calorimeter - (consumed heat in zones with fan coils measurements + calculated energy loss on the vertical supply lines)
STREM (school building)	Consumed heat on the non-controllable thermal circuit calorimeter	Consumed heat on the non-controllable thermal circuit calorimeter
STREM (retirement care centre)	Output of the cooling machine - calorimeter measurement from the controlled supply line	Consumed heat on the central calorimeter - calorimeter measurement from the controlled supply line

EPHZHB	Measured heating energy at two separate AHU units	Measured cooling energy at two separate AHU units

5.1.2 Solar irradiance data

Depending on the availability of solar irradiance measurements on different pilot sites throughout the project, two separate sets of weather measurements inputs are used.

On the UNIZG-FER pilot site, where direct normal and diffuse horizontal irradiance measurements are available, they are used as submodule inputs and paired with the same forecasted variables during submodule operation.

Due to high costs of direct and diffuse irradiance sensors other pilot sites provide measurements of global horizontal and tilted global irradiations which are then used as submodule inputs. Since measured and forecasted irradiances are now different, during submodule operation, forecasted direct and diffuse irradiance, solar angles (obtained through the use of Pysolar python library), geographical pilot site data and current datetime, are used for calculation of global horizontal and tilted global irradiances thus matching the measured and forecasted irradiance variables.

5.2 Submodule outputs

Table 5.3: Outputs of the non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	$\theta_{\mathrm{t,nc}}$	Needed for on-line operation of the submodule.
Predicted non-controllable heating/cooling energy consumption evolution (for on-line operation of the submodule)	$(E_{\mathrm{t,nc}})_{\mathrm{N}}$	Needed for the MPC module on the central HVAC level

Since the non-controllable consumption model is based on artificial neural networks (as presented in the following section 5.3) the predictions sometimes tend to reach impossible values, e.g. negative thermal energy values during heating period when the module output is actually fluctuating around 0. The structure of neural networks prohibits the model to incorporate exact boundaries on the model outputs, therefor all generated predictions are post-processed in order to avoid such unexpected values. E.g. during heating season, the non-controllable energy on the UNIZG-FER pilot represents the thermal energy consumption of the radiators and therefor cannot be negative. Therefore all the generated negative prediction values are set to 0.

5.3 Methodology

Based on a detailed description of artificial neural networks (ANN) given in [12], in the following sections a condensed description of ANNs structures and learning algorithms is given, together with a description of prediction module structure and operation schemes.

5.3.1. Artificial neural networks

Understanding of the human brain functioning and its learning and adaptation abilities made researchers try imitating its structure in order to imitate its capabilities in the computer systems. The basic element of the brain is a neural cell or neuron. Human brain contains 10^{11} neurons interconnected in the network with more than 10^{15} links. Although the neuron structure is rather simple, because of the immense number of links among them, a brain can perform the most complex operations. Schematic representation of a biological neuron is shown in Figure 5.1.

Neuron is composed of the cell body (soma), axon and a number of dendrites. Front end of an axon is connected to the cell body and its back end is split in a large number of branches. These branches are terminated by telodendria with their terminal buttons that touch dendrites of the other neurons. The terminal buttons contain numerous small bags with transmitters. A small distance between a telodendron of one neuron and a dendrite of another is called a synapse. Axon of one neuron forms synaptic interactions with many other neurons. Impulses generated in the cell body travel through an axon to a synapse. Depending on the efficiency of each synaptic transfer, action potentials of different intensity come over dendrites to the cell body where they are then collected and processed. If their cumulative value is greater than the neuron sensitivity threshold, a cell body generates an action potential which is spread over the axon to the other neurons, and if it is lower, the neuron remains inactive and does not generate an action potential. From the signal processing perspective, neuron operation can be divided in *synaptic operation* which gives a certain relevance (weight) to each input signal and *somatic operation* which collects all the "weighted" input signals, and due to their cumulative values, generates or does not generate a signal which is transferred towards other neurons.

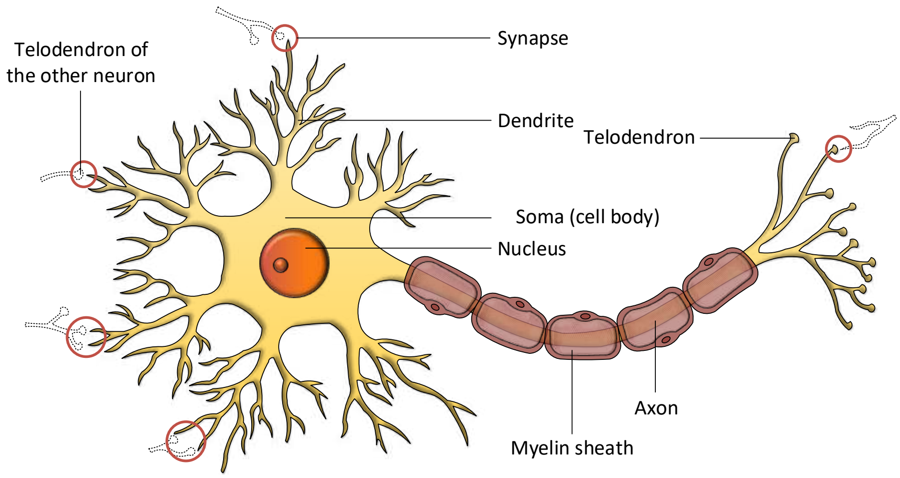


Figure 5.1: Schematic representation of a biological neuron.

5.3.1.1. Artificial neuron model

Early research in the field of artificial neurons was published by McCulloh and Pitts in 1943 and 1947 [13], [14]. Their model was based on a simple implementation of synaptic and somatic operations and was called a perceptron. Schematic representation of a perceptron is shown in Figure 5.2.

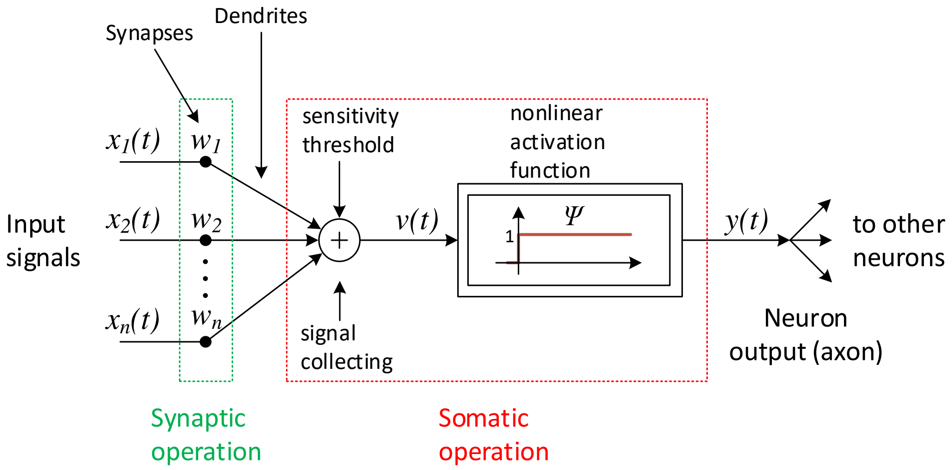


Figure 5.2: Schematic representation of a perceptron.

Synaptic operation is performed by multiplying input signals x_i with their weight coefficients w_i. Sum of all weighted signals is compared to a neuron sensitivity threshold w_{n+1}. If this sum is greater than a sensitivity threshold, nonlinear activation function ψ generates an output signal y equal to 1, and if it is less, neuron output is zero.

Mathematically, a perceptron can be described using these relations:

$$v(t) = \sum_{t=1}^{n} w_i(t)\, x_i(t) - w_{n+1},$$

$$y(t) = \psi(v),$$

where:

$\boldsymbol{x_u} = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is a vector of neuron input signals;

$\boldsymbol{w_s} = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is a vector of neuron input signals;

$\boldsymbol{w_{n+1}}$ is a neuron sensitivity threshold;

$\boldsymbol{v(t)}$ is a similarity measure between input signals and synaptic weight coefficients (result of the confluence operation);

$\boldsymbol{\psi(t)}$ is a nonlinear activation function;

$\boldsymbol{y(t)}$ is a neuron output.

However, because of the too simple model of a neuron, especially because of the discontinuity in nonlinear activation function, perceptron is not able to solve some simple operations. These constraints of the perceptron can be overcome by applying a continuous differentiable activation function. Sigmoid functions are commonly used as activation functions because it was proved that the ANNs composed of at least three layers of neurons with sigmoid functions can represent any continuous function. One of the most commonly used activation functions is *tansig* defined by the following expression:

$$\psi(v) = \frac{2}{1 + e^{-2g_0 v}} - 1,$$

where g_o is an activation gain and it is usually set to 1. Because of an extension of the initial model, in literature neurons with sigmoid activation functions are also referred to as perceptrons.

Neuron models can be divided in two groups: static and dynamic models. Static neuron models, as opposed to dynamic ones, do not contain dynamic elements and their output depends exclusively on current values of input signals and weight coefficients. In this deliverable only ANNs with static neuron models are analyzed.

5.3.1.2. Multilayer perceptron

Static neural networks are most commonly used ANNs, especially in identification and control applications. A basic element of the static ANN is a static neuron. In static ANNs neurons are organized in a feedforward way, i.e.: each neuron can be connected to the network inputs and/or to other neurons, but in the way that no feedback connections are formed. Therefore, static ANNs do not contain any dynamic elements and that makes them statically stable which is their most important advantage in relation to dynamic ANNs. However, in order to model a dynamic system, delayed input and output signals have to be explicitly included in the vector of input signals of the static ANN. The most commonly used static ANNs are multilayer perceptrons (MLP) whose structure is presented in Figure 5.3. MLPs consist of perceptrons organized in serially connected layers. Layers are often labelled with numbers $0, 1, 2, \cdots, L$, while for the number of nodes in the l-th layer we use label $n(l)$. The zeroth layer only transfers the input vector to an input of the first layer, L-th layer is an output layer, while layers between them are called hidden layers. Every neuron in a hidden layer is connected to all the neurons in two neighboring layers with unidirectional feedforward connections. Connections between neurons of the neighboring layers are represented by synaptic weight coefficients which act as signal gains on the corresponding connections. Values of the synaptic weight coefficients determine the network behavior, i.e.: its ability of approximating a nonlinear function.

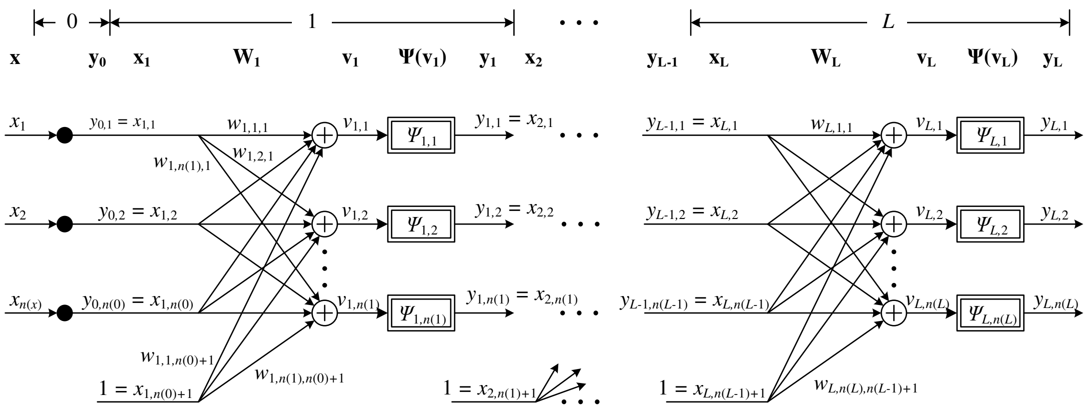


Figure 5.3: Schematic representation of a multilayer perceptron.

Mathematically, MLPs can be described by the following relations:

$$
\begin{aligned}
y_0 &= x, \\
x_l &= [y_{l-1}^T, 1]^T, \qquad 1 \le l \le L, \\
v_l &= W_l \cdot x_l, \qquad 1 \le l \le L, \\
y_l &= \psi(v_l), \qquad 1 \le l \le L,
\end{aligned}
$$

where:

$\boldsymbol{x} = \left[x_1, x_2, \cdots, x_{n(x)}\right]^T$ is a vector of the network input od dimension $n(x)$;

$\boldsymbol{y_0} = \left[y_{0,1}, y_{0,2}, \cdots, y_{0,n(0)}\right]^T$ is an output vector of the 0-th layer of dimension $n(0)$;

$\boldsymbol{x_l} = \left[x_{l,1}, x_{l,2}, \cdots, x_{l,n(l-1)}, x_{l,n(l-1)+1}\right]^T$ is an input vector to the l-th layer (input $x_{l,n(l-1)+1} = 1$ multiplied by corresponding weight coefficient gives a scalar bias to neurons of the l-th layer);

$\boldsymbol{v_l} = \left[v_{l,1}, v_{l,2}, \cdots, v_{l,n(l)}\right]^T$ is an output vector of the confluence operation of the l-th layer;

$\boldsymbol{y_l} = \left[y_{l,1}, y_{l,2}, \cdots, y_{l,n(l)}\right]^T$ is an output vector of the l-th layer;

$\boldsymbol{W_l} = \begin{bmatrix} \boldsymbol{w_{l,1,1}} & \cdots & \boldsymbol{w_{l,1,j}} & \cdots & \boldsymbol{w_{l,1,n(l-1)}} & \boldsymbol{w_{l,1,n(l-1)+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{w_{l,i,1}} & \cdots & \boldsymbol{w_{l,i,j}} & \cdots & \boldsymbol{w_{l,i,n(l-1)}} & \boldsymbol{w_{l,i,n(l-1)+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{w_{l,n(l),1}} & \cdots & \boldsymbol{w_{l,n(l),j}} & \cdots & \boldsymbol{w_{l,n(l),n(l-1)}} & \boldsymbol{w_{l,n(l),n(l-1)+1}} \end{bmatrix}$ is a weight coefficient matrix of the synaptic connections of the l-th layer, dimension of which is $n(l) \times (n(l-1)+1)$;

$\boldsymbol{\Psi_l(v_l)} = \left[\Psi_{l,1}(v_{l,1}), \Psi_{l,2}(v_{l,2}), \cdots, \Psi_{l,n(l)}(v_{l,n(l)})\right]^T$ is an activation function vector of the l-th layer (usually $\Psi_{l,1} = \Psi_{l,2} = \cdots \Psi_{l,n(l)}$).

The most commonly used activation function in the hidden layer is *tansig*, while in the output layer linear activation function is used. The activation gain is usually set to one.

The most important properties of the ANNs are universal approximation, learning and adaptation. ANN property of approximating any continuous function to an arbitrary accuracy is its most important property from the perspective of modelling, identification and control of nonlinear processes. Learning and adaptation properties enable that an adequately calibrated ANN has the generalization ability when the data that was not present in the calibrating data set comes to its input.

5.3.1.3. *Neural network learning algorithms*

Learning algorithm tunes network parameters in order to achieve its desired behavior. In identification and control of nonlinear dynamic systems desired behavior of a neural network is usually known, so error-based algorithms are used for the learning/calibrating procedure. Schematic representation of the error-based algorithm for neural network learning is shown in Figure 5.4.

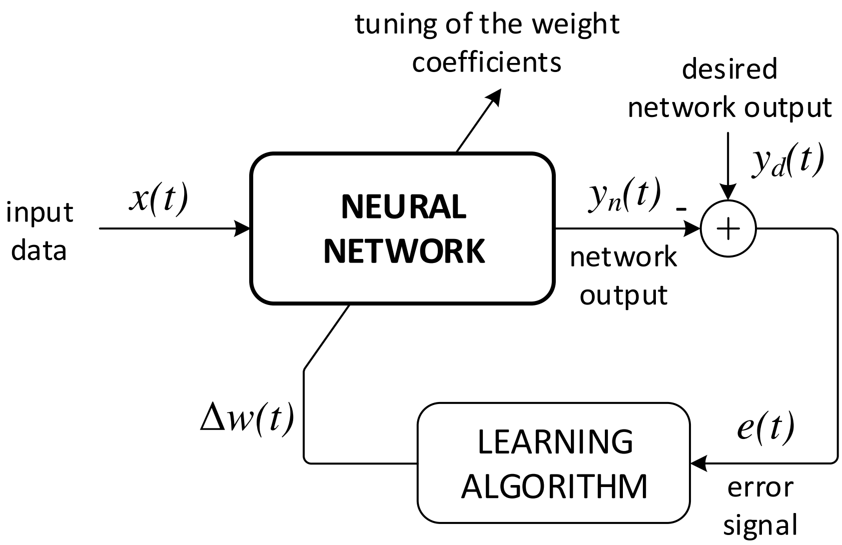


Figure 5.4: Schematic representation of the error-based algorithm for neural network learning.

Resulting neural network response y_n to the input data is compared to the external reference signal y_d, which represents desired network behaviour, generating error signal e based on which the learning algorithm changes synaptic weight coefficients of the network in order to improve its behaviour, i.e.: to decrease the error. As an error measure a criterion function $\Im(\Theta)$ is used and it can be any positive scalar function dependent on ANN parameters $\mathbf{\Theta}$. The most commonly used criterion function is defined as:

$$\Im(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{v=1}^{N} e(v,\boldsymbol{\Theta}) \cdot e^T(v,\boldsymbol{\Theta}) = \frac{1}{2}\sum_{v=1}^{N}\sum_{i=1}^{n(L)} e_i^2(v,\boldsymbol{\Theta}) = \frac{1}{2} e^{*T}(\boldsymbol{\Theta}) \cdot e^*(\boldsymbol{\Theta}),$$

where ν is a number of the measured sample, N is an overall number of measured samples, $e^*(\Theta)$ is the error vector of the whole measured data set, which is of dimension $N_e = N \cdot n(L)$.

There are two basic approaches in minimizing the criterion function $\Im(\Theta)$: non-recursive and recursive. According to the non-recursive approach, function $\Im(\Theta)$is minimized such that network parameter changes are determined based on the complete set of N measured samples. According to the recursive approach, function $\Im(\Theta)$ is minimized based on a local criterion function $\Im_v(\Theta)$, i.e. network parameters are changed after each measured sample.

Learning algorithm tunes network parameters until the criterion function reaches its minimum. Minimum of the criterion function $\Im(\Theta)$ can be formally defined by its Taylor series expansion in vicinity of the parameter vector Θ^0 for which the minimum is obtained, and by ignoring its third and higher order terms:

$$\Im(\boldsymbol{\Theta}) \cong \Im(\boldsymbol{\Theta^0}) = \nabla\Im^T(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}^0} \cdot \Delta\boldsymbol{\Theta} + \frac{\mathbf{1}}{\mathbf{2}}\Delta\boldsymbol{\Theta}^{\boldsymbol{T}} \cdot \boldsymbol{H}(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}^0} \cdot \Delta\boldsymbol{\Theta},$$

where:

$\Delta\boldsymbol{\Theta} = \boldsymbol{\Theta} - \boldsymbol{\Theta}^{\mathbf{0}}$;

$\nabla\Im(\boldsymbol{\Theta})$ is a gradient vector of the criterion function:

$$\nabla\Im(\boldsymbol{\Theta}) = \left[\frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}}, \frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}}, \cdots, \frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}}\right];$$

$\boldsymbol{H}(\boldsymbol{\Theta}) = \nabla^2\Im(\boldsymbol{\Theta})$ is a Hessian matrix of the criterion function:

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \begin{bmatrix} \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_1^2} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}\partial\theta_{\boldsymbol{n}(\theta)}} \\ \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}\partial\theta_{\mathbf{1}}} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_2^2} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}\partial\theta_{\boldsymbol{n}(\theta)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}\partial\theta_{\mathbf{1}}} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}^2} \end{bmatrix}.$$

For the previously defined criterion, gradient vector and Hessian matrix become:

$$\nabla\Im(\boldsymbol{\Theta}) = \boldsymbol{J}^T(\boldsymbol{\Theta}) \cdot e^*(\boldsymbol{\Theta}),$$

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \nabla^2\Im(\boldsymbol{\Theta}) = \boldsymbol{J}^T(\boldsymbol{\Theta}) \cdot \boldsymbol{J}(\boldsymbol{\Theta}) + \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N_e}} e_i^*(\boldsymbol{\Theta})\nabla^2 e_i^*(\boldsymbol{\Theta}),$$

where $\boldsymbol{J}(\boldsymbol{\Theta})$ is a Jacobian matrix:

$$\boldsymbol{J}(\boldsymbol{\Theta}) = \begin{bmatrix} \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}} & \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}} \\ \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}} & \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}} & \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}} \end{bmatrix}.$$

Parameter vector $\Theta = \Theta^*$ will be the minimum argument of the function $\Im(\Theta)$ if the following conditions are fulfilled:

$$\nabla\Im(\Theta^*) = 0,$$
$$\Delta\Theta^{\mathrm{T}} \cdot \mathrm{H}(\Theta^*) \cdot \Delta\Theta > 0.$$

Therefore, tuning of the ANN parameters Θ is in fact a *nonlinear optimization* problem where the criterion function $\Im(\Theta)$ is the objective function of the optimisation problem. *Gradient methods* are most commonly used nonlinear optimization techniques. The main problem in applying gradient methods in ANN learning procedure is calculating a gradient vector of the criterion function over the network parameters. This problem has slowed research and application of ANNs for a while, but was successfully solved using the backpropagation algorithm. More details can be found in [12].

Tuning of the ANN parameter vector Θ is based on an iterative procedure:

$$\Theta(k+1) = \Theta(k) + \Delta\Theta(k) = \Theta(k) + \alpha(k)s_d(k),$$

where:

$s_d(k)$ is the minimum searching direction in the k-th iteration of the optimisation procedure (it is based on an information on a function $\Im(\Theta)$);

$\alpha(k)$ is the learning coefficient in the k-th iteration of the optimisation procedure (it determines the step size in the searching direction).

Depending on the procedure of determining the minimum searching direction $s_d(k)$, gradient methods can be divided into four groups:

- *Steepest descent methods*: $s_d(k) := -\nabla\Im(\Theta(k))$;
- *Conjugate gradient methods:* $s_d(k) := -\nabla\Im(\Theta(k)) + \beta(k) \cdot s_d(k-1)$, where $\beta(k)$ is a scalar parameter which ensures conjugacy;
- *Newton methods:* $s_d(k) := -\left[\nabla^{2\Im}(\Theta(k))\right]^{-1}\nabla\Im(\Theta(k))$;
- *Quasi-Newton methods* [15], [16]*:* $s_d(k) := -S(k)\nabla\Im(\Theta(k))$ where $S(k) \cong \left[\nabla^{2\Im}(\Theta(k))\right]^{-1}$.

ANN learning algorithms are named based on the corresponding nonlinear optimization methods which are used: steepest descent algorithms, conjugate gradient algorithms etc.

5.3.2 Applying neural networks to system modelling

In the last 20 years neural network applications for predicting variables in ecological and technical systems have become a well-known procedure in a research community [17]. In the early phases of their applications, ANNs were considered as a novel approach in system modelling and the majority of published papers in that period were related to applying ANNs in different systems and exploring their advantages in relation to the well-known statistic approaches [18]. Many review papers in this research area did not only affirm a potential of using the ANNs in prediction systems, but they also noted an importance of developing a standard methodology in the model development procedure using ANNs. Clearly defined methodology is an important procedure for all modelling methods, but especially in ANN modelling because models are developed based on the available data and they are not explicitly based on the physical system that is modelled, therefore, a possibility of developing a model which is not very meaningful is increased.

Main steps in developing the prediction model using ANNs are shown in Figure 5.5. Flow of data and outcomes for each step are also shown. First step in model development process is a choice of appropriate model outputs (variables which are going to be predicted) and potential inputs. A choice of potential inputs is based on *a priori* knowledge on the modelled process and on data availability. Selected data have to be processed (scaled, filtered, lagged) for being in an appropriate form for the next model development steps.

A general ANN prediction model can be expressed in the following form:

$$\mathrm{Y} = \mathrm{f}(\mathrm{X}, \mathrm{W}) + \mathrm{e},$$

where Y is a model output vector, X is a model input vector, W is a model parameter vector (weight coefficients), f is a function which defines input-output relationship and e is a model error vector. Therefore, in model development process we need to define model inputs X, a functional relationship f defined by the ANN structure and ANN parameter vector W. Model inputs are determined using the so called Input Variable Selection (IVS) procedures which are described in subsection 5.3.3. Result of this step are model development data which are then divided in calibration and validation data sets. Calibration data are used in ANN learning algorithms for determining the optimal model parameters, while validation data are used for validating the calibrated model on the independent data set. If implicit regularization is used as a stopping criterion of the learning algorithm, calibration data are divided in training and testing data sets.

The main objective of the ANN learning process is to find the global minimum of the criterion function $\mathfrak{I}(\Theta)$. However, in modelling of dynamic systems which inherently contain noise, the global minimum of the criterion function is not the optimal solution because the obtained model does not assure the best generalization properties. In the first phase of the ANN learning process a decrease of the criterion function $\mathfrak{I}(\Theta)$ on the training data leads to a decrease of the criterion function $\mathfrak{I}^t(\Theta)$ on the testing data. However, after certain number of iterations, value of the criterion function $\mathfrak{I}^t(\Theta)$ starts increasing although $\mathfrak{I}(\Theta)$ is still decreasing and, therefore, further adjusting of the ANN parameters leads to a deterioration of its generalization properties. This problem can be solved by early stopping of the learning process when a criterion function value on the testing data starts increasing. This procedure is called an *implicit regularization*.

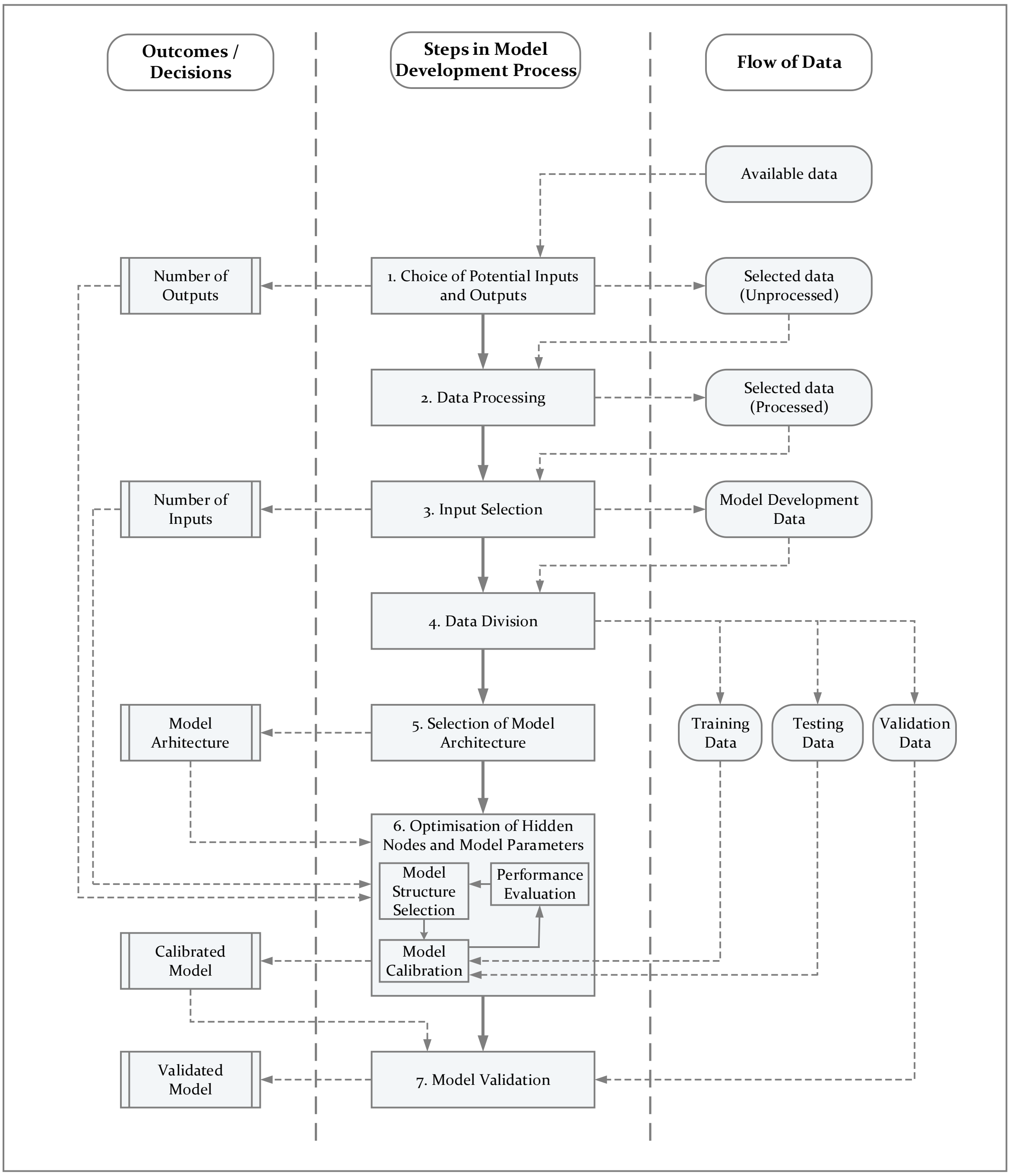


Figure 5.5: Main steps in the model development process using artificial neural networks [18].

Next step implies choosing a number of hidden layers and a number of neurons in each layer. The optimal structure of ANN is usually determined iteratively [18]. For a fixed structure, optimal parameters of the ANN are determined using learning procedure and they depend on the choice of learning algorithm and on initial ANN parameters. In general case criterion function is nonconvex and applying gradient methods can trap model parameter vector in a local minimum of the criterion function which is not the optimal solution. Therefore, a calibration process implies a number of calibration instances for different initial values of model parameters. ANN, defined by its structure and parameters, which has the minimal criterion function value on the calibration data is then validated on the validation data set. To ensure that a model development process results in the best possible model, it is required that training, testing and validation data sets have the same statistical properties [19].

5.3.3 Input variable selection procedure

One of the most important steps in modelling of complex systems is selection of the appropriate input variables. However, this step is usually not concerned to be of an extreme importance and most of the input variables are determined heuristically or based on *a priori* knowledge of the system which can result in including too many or too little input variables [20].

As a consequence of omitting one or more relevant input variables, model will not be able to describe the whole dynamics and phenomena of the system. Possibility of omitting relevant input variables is much greater for time series in which input candidates are not only different variables, but also their lagged values (unless dynamic ANNs are used) which significantly increases the number of potential input variables. Including too many input variables can be caused by poorly assessed relevance of an input variable or by existence of a redundancy among them, where some of the chosen variables contain some useful information, but are interdependent, so they contain a redundant information. This case leads to an increase in a number of local minima in the criterion function [18] and makes it harder to determine the optimal model parameters if a gradient method is used for ANN learning. On the other hand, with an increase of input variables, a number of model parameters is also increased which, as a consequence, leads to decreased speed and quality of the learning procedure. Furthermore, existence of an input variable which does not affect the output variable can lead to a deterioration of ANN generalization properties, i.e. the model will perform poorly on data that were not used during model calibration procedure.

These considerations indicate that the optimal ANN input variable set consist of the minimal set of variables which can describe the system behavior well enough. A number of IVS algorithms were developed and they can be classified in *wrapper* and *filter* algorithms [21].

5.3.3.1 Wrapper algorithms

IVS using wrapper algorithms is based on developing a number of ANNs with different input vectors and the choice of an appropriate input set is determined based on performance of the corresponding ANN. The main drawback of this approach is that such a procedure can last very long because it is required to develop a large number of ANNs whereas the development of each implies an appropriate choice of the ANN structure and the learning algorithm. Additionally, appropriateness of the input variables chosen for a certain ANN architecture is not guaranteed for another architecture, so the application of the obtained input set is rather limited [20].

For d potential input variables, a number of possible input subsets is $2^d - 1$. Therefore, because of the large computational and time requirements, all possible input variable combinations are almost never tested. The most commonly used wrapper algorithms are forward selection, backward elimination and genetic algorithms [21].

Forward selection is an incremental procedure for forming the optimal input variable set in which a number of variables is incrementally increased. In the beginning, one out of d variables, for which an ANN with the best performance is obtained, is chosen. Then, the input set is enlarged by the next one out of $d-1$ remained variables for which an ANN performance is most improved. A procedure is repeated until adding a new variable to the input set does not lead to a significant improvement of the ANN performance.

Backward elimination is a procedure inverse to a forward selection, i.e. the input variable set is incrementally reduced. The procedure starts with an input set which contains all the potential input variables and the least relevant variables are progressively eliminated from the input set. This procedure is computationally more intensive than the forward selection because a large number of inputs requires learning an ANN with much larger number of parameters.

Genetic algorithms introduce stochastic elements in the procedure of selecting the optimal input variable set, increasing a possibility of finding the optimal set. Genetic algorithms show their advantages in relation to forward selection and backward elimination when the candidate set contains variables which only combined with other variables show their relevance to an output variable, while taken separately, do not have an excessive importance.

5.3.3.2 Filter algorithms

Unlike wrapper, filter algorithms use statistical measure of dependence between an output variable and potential inputs as a criterion for input selection. Uncoupling IVS procedure and model calibration does not only increase the modelling efficiency, but also extends possible applications of the obtained input set. However, efficiency of a filter algorithm is highly dependent on the statistical measure employed [20].

The most commonly used statistical measure of dependence is a linear correlation coefficient whose main drawback is that it only determines the linear dependence between variables which is particularly problematic in the model development using ANNs because they are used as an alternative to linear regression when a dependence between model inputs and output is nonlinear. Therefore, it is more meaningful to use an appropriate nonlinear statistical measure of dependence, like mutual information [18]. Unlike linear correlation coefficient, mutual information is also sensitive to dependences which are reflected in higher input-output correlation moments – mutual information is equal to zero if and only if two variables are strictly independent [22].

Apart from inputs relevance, IVS procedures should also consider redundancy of the input variables. In order to do so, a suitable algorithm based on partial mutual information (PMI) was developed and it is described in the next subsection.

5.3.3.3 Input variable selection algorithm based on partial mutual information

For a given continuous random variable X with a codomain $C(X)$, Shannon entropy is defined as:

$$\mathrm{H(X)} = -\int_{\mathrm{C(X)}} \mathrm{f(x)} \ln \mathrm{f(x)}\ \mathrm{dx},$$

where x is an outcome of random variable X and $f(x)$ is its probability density function (pdf). Entropy is a term well-known in the information theory and it represents an informational description of random events and defines a measure of the information content, i.e. random variable uncertainty. Mutual information of two random variables, X and Y, is defined as:

$$\mathrm{I(X;Y)} = \int_{\mathrm{C(Y)}} \int_{\mathrm{C(X)}} \mathrm{f(x,y)} \ln\left(\frac{\mathrm{f(x,y)}}{\mathrm{f(x)f(y)}}\right) \mathrm{dxdy},$$

where $f(x)$ and $f(y)$ are pdfs of the variables X and Y, respectively, and $f(x,y)$ is a joint pdf of the random vector (X,Y). Mutual information can be expressed using entropies as:

$$\mathrm{I(X;Y)} = \mathrm{H(X)} + \mathrm{H(Y)} - \mathrm{H(X,Y)},$$

where $H(X)$ and $H(Y)$ are entropies of the random variables X and Y, respectively, and $H(X,Y)$ is a joint entropy of the random vector (X,Y). Mutual information represents a reduction in uncertainty of the random variable Y knowing the random variable X and *vice versa*. Figure 5.6 depicts the dependency among mutual information and entropies of the random variables X and Y.

Here, $H(Y|X)$ is conditional entropy of Y given X, that is, the amount of uncertainty in the random variable Y when the value of X is known, and it is formally defined as:

$$\mathrm{H(Y|X)} = \int_{\mathrm{C(Y)}} \int_{\mathrm{C(X)}} \mathrm{f(x,y)} \ln\left(\frac{\mathrm{f(x)}}{\mathrm{f(x,y)}}\right) \mathrm{dxdy}. \qquad (1\text{-}22)$$

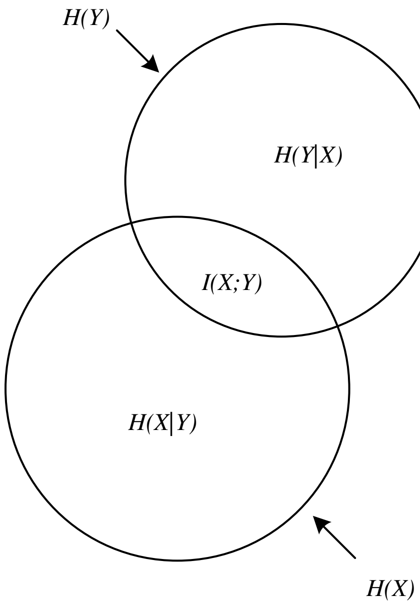


Figure 5.6: Venn diagram showing a relationship among mutual information and entropies of random variables X and Y.

Let us now consider the third random variable, Z. A part of a mutual information $I(Z;Y)$ which is not contained in X, $I(Z;Y|X)$, is called a partial mutual information and it is determined using the following expression:

$$\mathrm{I(Z;Y|X) = H(X,Z) + H(X,Y) - H(X) - H(X,Y,Z).}$$

Given X and the already reduced uncertainty $H(Y|X)$ shown in Figure 5.6, the PMI $I(Z;Y|X)$ is defined as the further reduction in uncertainty of the random variable Y that is gained by the additional mutual observation of the random variable Z.

Figure 5.7 depicts the dependence among PMI, individual and joint entropies of the random variables X, Y and Z. PMI is invariant under strictly monotonic transformations which makes it robust against possibly nonlinear distortions among random variables [23] and this is one of its most important advantages in relation to the linear correlation. However, a problem in determining a mutual information is that pdfs of the random variables have to be known. In practice, the real pdfs are not known and it is needed to estimate them. This topic is covered in the next subsection.

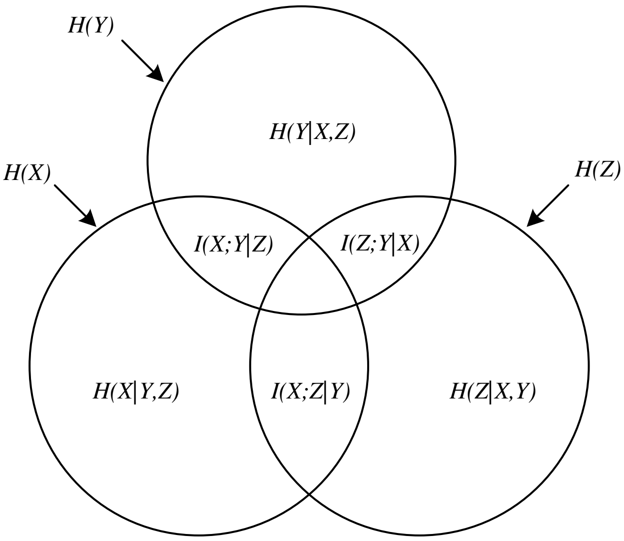


Figure 5.7: Venn diagram showing a relationship among partial mutual information and entropies of the random variables X, Y and Z.

PMI-based IVS algorithm is presented in [24]. Details of the algorithm are presented here:

Algorithm 1: Partial mutual information-based input variable selection

Input: output variable Y, potential input variables C
Result: chosen input variables X

Initialize $X \leftarrow \emptyset$
while $C \neq \emptyset$ **do**
  for each $c \in C$
    Estimate $I(c, Y|X)$
  Determine $c_s \in C$ that maximises $I(c, Y|X)$
  if *algorithm termination criterion is satisfied* **then**
    Stop running the algorithm
  Move c_s to X

In [20] a number of algorithm termination criteria are analyzed. In this work a predefined number of the most relevant input variables was used as a termination criterion.

5.3.3.4 Estimating partial mutual information

Considering the expression above it can be seen that for estimating an entropy of the random variable, it is first required to determine its pdf which is estimated from the available historical data, i.e. from the considered random variable outcomes. There are two main approaches in estimating a pdf: *parametric* and *non-parametric*.

The parametric approach assumes that data are drawn from a known parametric family of distributions, for example the normal distribution with mean μ and variance σ^2. Estimating the pdf then becomes a problem of estimating the parameters μ and σ^2. The non-parametric approach does not assume a form of the pdf, so non-parametric methods are usually much more robust and accurate than the parametric ones. A review of the most commonly used non-parametric estimation methods can be found in [25].

One of the most commonly used non-parametric pdf estimation methods is *kernel density estimation* and this method is proposed in [24] in the original version of Algorithm 1. However, this approach has some drawbacks -- apart from the fact that it is computationally very intensive and that it requires relatively large number of data samples for an accurate estimation, its behavior is dependent on the kernel function parameters. This problem becomes even harder when a dimension of the random variable is increased [26]. Much more accurate and computationally less intensive pdf estimation method is *k-th nearest neighbor method*. The method in which an entropy of the random variable is directly determined is presented in [23] and it is described here.

Let us consider three continuous time series, $\{x_t\}$, $\{y_t\}$ and $\{z_t\}$, which represent the outcomes of random processes $\{X_t\}$, $\{Y_t\}$ and $\{Z_t\}$, respectively. For each vector $v_t \equiv \{x_t, y_t, z_t\}$, $t = 1,2,\cdots,N$ and a fixed integer k, $1 \leq k \ll N$, a distance $\varepsilon_k(t)$ to its k-th neighbour is defined. It means that a set $\{v_{t^*}\}$, where $t^* = 1,2,\cdots,N$, $t^* \neq t$, contains $k-1$ vectors with distances from v_t less than $\varepsilon_k(t)$ and $N-k-1$ vectors with the distance greater than $\varepsilon_k(t)$.

Therefore, for each t distance of v_t to each element of $\{v_{t^*}\}$ is determined:

$$\varepsilon(\mathrm{t}) = \{||\mathrm{v}_{\mathrm{t}^*} - \mathrm{v}_\mathrm{t}||\}.$$

This set is then sorted and distance $\varepsilon_k(t)$ is determined by selecting the k-th element of the sorted set. The distance is determined using max norm, i.e. $||\cdot|| = \max\{||\cdot||_x, ||\cdot||_y, ||\cdot||_z\}$, where $||\cdot||_x$, $||\cdot||_y$ and $||\cdot||_z$ can be any norm, but this algorithm suggests using max norm as well. Let us now define a vector $w_t \equiv \{x_t, z_t\}$, $t = 1,2,\cdots,N$.

For each t a number of vectors in $\{w_{t^*}\}$ with distances strictly less than $\varepsilon_k(t)$ is determined:

$$\mathrm{N}_{\mathrm{xz}}(\mathrm{t}) = \#\{\mathrm{t}^* \neq \mathrm{t};\ ||\mathrm{w}_{\mathrm{t}^*} - \mathrm{w}_\mathrm{t}|| < \varepsilon_\mathrm{k}(\mathrm{t})\}.$$

where # denotes a number of elements in the set. In a similar way $N_{xy}(t)$ and $N_x(t)$ are defined, for which w_t is defined using vectors $\{x_t, y_t\}$ and $\{x_t\}$, respectively. PMI is estimated using the following expression:

$$\hat{I}(Z;Y|X) = \frac{1}{N}\sum_{t=1}^{N}\left[h_{N_{xz}(t)} + h_{N_{xy}(t)} - h_{N_x(t)}\right] - h_{k-1},$$

where h_n is the n-th negative harmonic number defined as $h_n = -\sum_{i=1}^{n} i^{-1}$ [23].

The k-th nearest neighbor method is computationally much faster than kernel methods are and, regardless of a number of considered variables dimension, it requires defining only one scalar parameter, k.

Here, we analyses the properties of the PMI estimator in case of the normal distribution for which PMI can be determined analytically, as shown in [23]. Multivariate normal distribution of the random vector $X \in R^n$ with mean $a \in R^n$ and covariance matrix $R \in R^{n\times n}$ is defined by its pdf:

$$\mathrm{f}(X) = \frac{1}{(2\pi)^{n/2}\sqrt{R}} exp\left(\frac{1}{2}(x-a)^T R^{-1}(x-a)\right),$$

and it is denoted as $X \sim \mathcal{N}_n(a, R)$ where $|R|$ denotes a determinant of the covariance matrix R. For n-dimensional normal distribution $\mathcal{N}_n(a, R)$ entropy is determined using the following expression:

$$H(X) = \frac{n}{2}(1 + ln2\pi) + \frac{1}{2} ln|R|.$$

5.3.4 Structure of the prediction model

This section analyses an identification procedure for prediction models with time horizon of 12-36 hours. One of the main issues in developing such a multiple-output system is how to assess its performance, i.e.: how to define a criterion which will tell us if one model is better than the other. The response is trivial if each output of one model outperforms the corresponding output of the other model, but generally it is not the case. The simplest approach is to define a local criterion function for each output and a global criterion function could be e.g.: a sum of the local criterion functions. The first drawback of this approach is that we are usually more concerned about sooner prediction hours than about hours at the end of a prediction horizon, so we do not want to give the same weight to each local criterion function. An alternative is to use weighted sum of the local criterion functions as a global criterion, but a question of how to choose these weights remains open. The second drawback is that such a model has the same input vector which is used for describing input-output relationship for each output, which generally does not have to be the optimal choice. Certainly, developing a separate model for each output can at least perform as well as one model with multiple outputs. The first advantage of this approach is that defining a criterion function is trivial because for single-output models the local criterion corresponds to the global criterion. The second advantage is that such an approach does not necessarily imply a unique input vector for each model. The main drawback of this approach is that the whole developing process, including IVS, defining the optimal model structure and model calibration has to be carried out multiple times which can be computationally very intensive for a large prediction horizon. The concept of this approach is depicted in Figure 5.8.

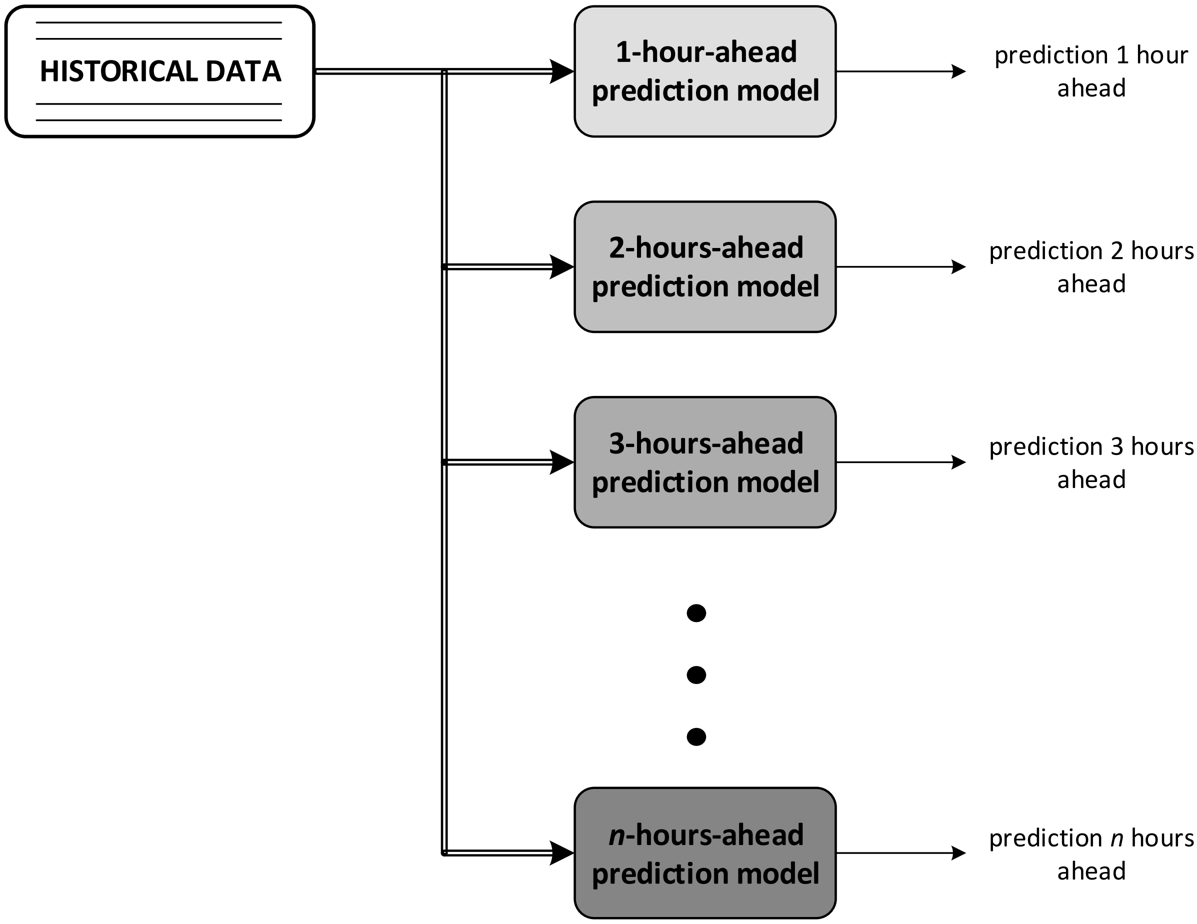


Figure 5.8: A static approach of the prediction system which uses a separate model for each system output.

Unlike the above-mentioned *static* approaches, the third approach uses the fact that the prediction system is considered as dynamic, i.e.: its output depends on past outputs. This *dynamic* approach is depicted in Figure 5.9. The main idea behind this approach is that the model does not have to use all the actual data, but also the provisional data, e.g.: output of the 1-hour-ahead model is a prediction for one hour ahead and this value can be used by the same model for predicting for two hours ahead. Analogously, this procedure can be repeated for obtaining the prediction for k hours ahead. It is expected that this approach will be less accurate than the one shown in Figure 5.8 because in this case a prediction error of the model is accumulated over the whole prediction horizon. However, if the performance of such an approach is not much worse than the one of the static approach, from the computational point of view, applying dynamic approach is much more efficient and contains significantly less parameters. Additionally, in some applications a larger prediction horizon may be required. Extension of the existing prediction model to a larger prediction horizon using dynamic approach is trivial; for the static approach this is not the case. Therefore, a dynamic approach is chosen for the prediction system.

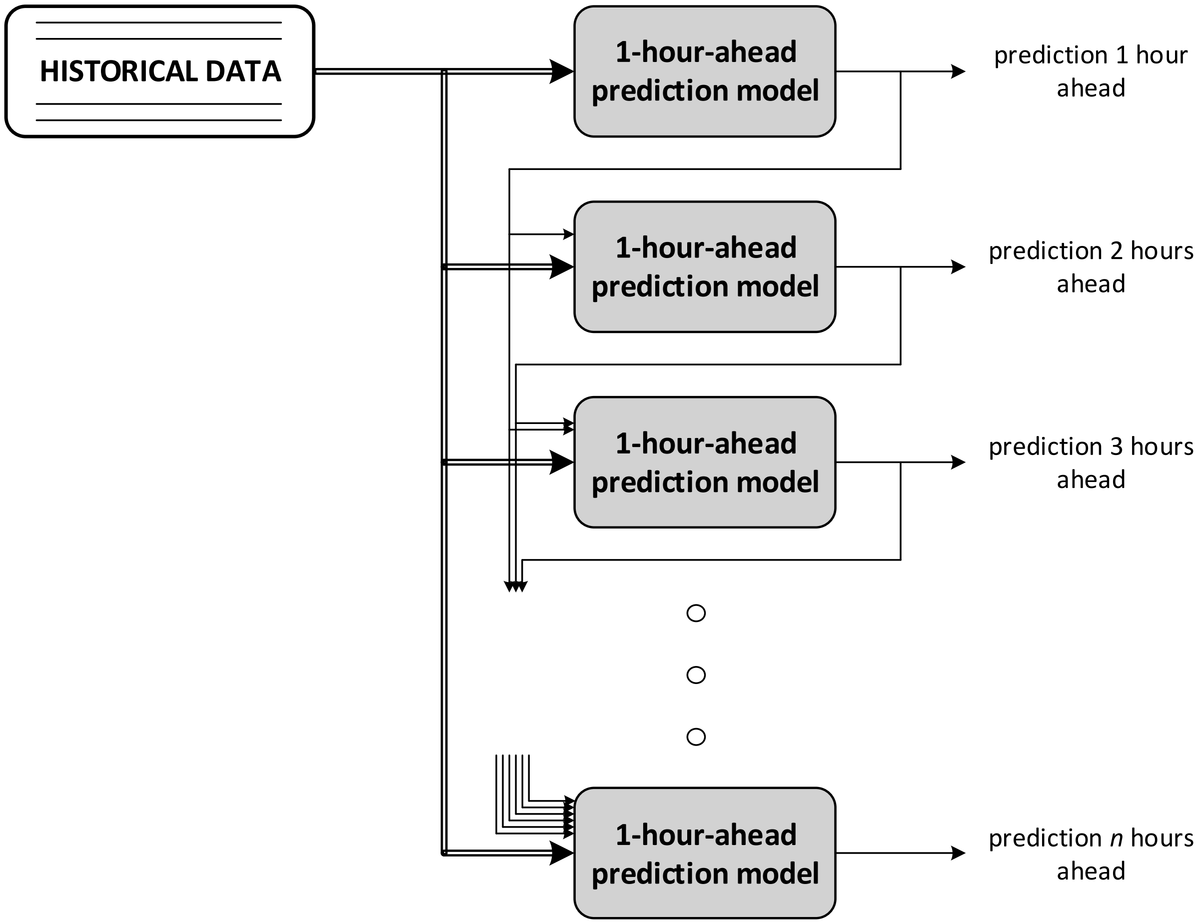


Figure 5.9: A dynamic approach of the prediction system which uses a single model for estimating system outputs for the whole prediction horizon.

5.3.4.1 Adaptive structure of the prediction system

It is often the case that historical data used for calibrating the prediction model do not cover the complete set of possible input-output vectors or that predicted variable values that occurred in past differs from values for the coming period due to factors which were not considered or did not have a significant impact on the variable during model calibration process. Occurrence of these factors can lead to poor predicting abilities of the existing prediction model. Therefore, for robust operation of the prediction system the model should be able to adapt to possible changes in the system.

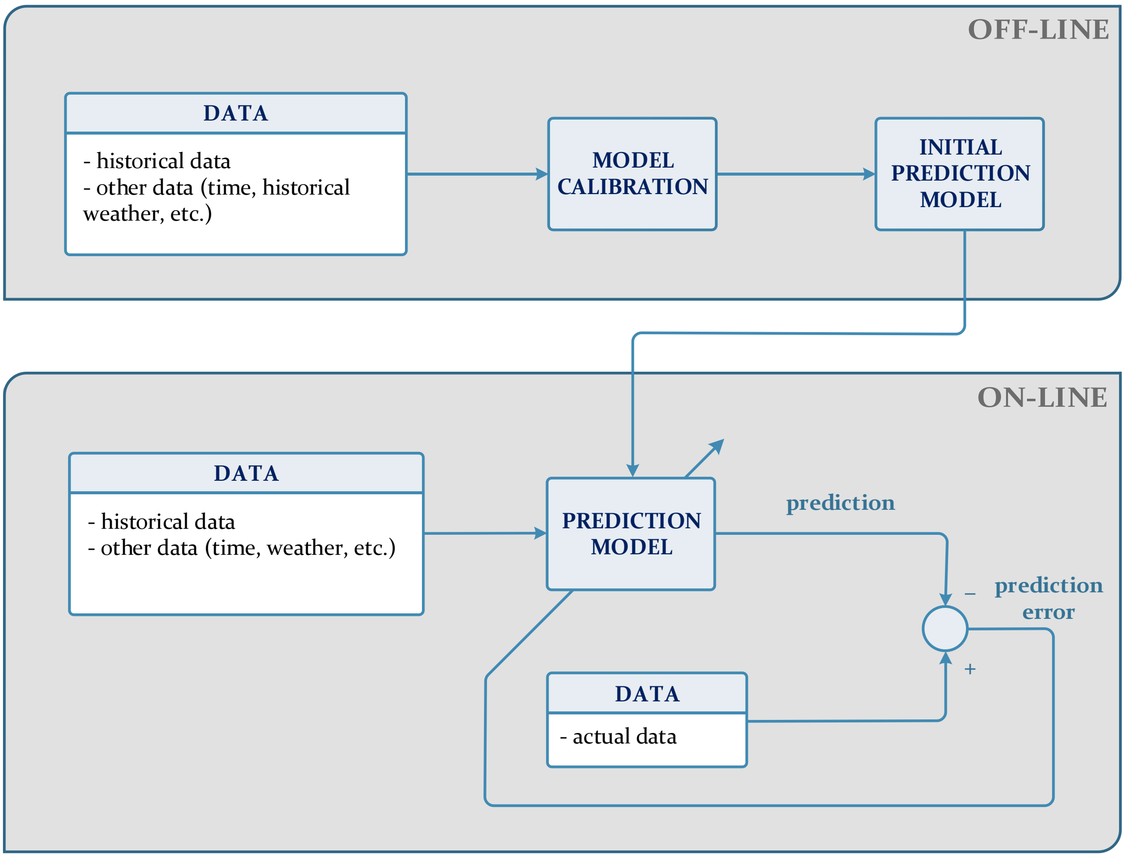


Figure 5.10: Adaptive module structure / a principle overview.

Modified structure of the prediction module is shown in Figure 5.10. The system is composed of two parts: *off-line* and *on-line*. In the off-line part historical data are used for obtaining the initial prediction model. The on-line part of the module uses the initial model developed in the off-line part in order to generate predictions. When the data are available, they are compared to the corresponding predictions which results in the prediction error for the certain time instant. Model parameters are then tuned such that the prediction error is decreased. The presented procedure of using the feedback information on prediction accuracy for model parameters tuning introduces an adaptation ability to module.

5.3.4.2 Possible approaches to the on-line tuning of model parameters

Most real systems are time-variant. In order to track changes in the system, its model parameters should be continuously estimated. The on-line part of the prediction system, mentioned in the previous section, is the tool for continuous tuning of the model parameters such that the model tracks the actual predicted variable evolution as accurately as possible.

Artificial neural network (ANN) is a flexible model structure that can be easily and systematically calibrated and adapted. There is a large number of methods suggested in literature for the so called *recursive* neural network learning. Some of them are based on the recursive approximation of typical gradient methods [12], [27]. On the other hand, some recursive methods are based on the methodology for dynamic system state estimation [28]-[31]. These methods are based on the state-space representation of the ANN model [32]:

$$\begin{gathered} \mathrm{w_{k+1}} = \mathrm{w_k} + \mathrm{r_k}, \\ \mathrm{d_k} = \mathrm{G}(\mathrm{x_k}, \mathrm{w_k}) + \mathrm{e_k}, \end{gathered}$$

where G is a function which defines the input-output mapping and is determined by the ANN structure, x_k is an input vector, w_k is a vector of ANN parameters and e_k is an error vector. In the above equation a vector of parameters w_k corresponds to a stationary process with identity state matrix, driven by process noise r_k. ANN model written in this form enables using extended Kalman filter (EKF) or unscented Kalman filter (UKF) for the ANN parameter estimation. However, the ANN models with relatively large number of inputs and nodes in the hidden layer result in a large number of parameters, and applying EKF or UKF becomes intractable due to numerical stability issues [33]. On the other hand, recursive gradient methods for ANN learning are quite robust and their application is not limited to ANNs with a small number of parameters. Therefore, this approach in recursive ANN learning is analyzed hereinafter.

5.3.4.3 Applying the on-line tuning procedure in normal operation

We use the initially developed prediction model as an initial prediction model for the on-line part of prediction system (see Figure 5.10). Gradient descent method with momentum term is used for the recursive ANN learning. ANN parameters Θ are updated based on the following relation:

$$\Delta\Theta(\mathrm{k}) = -\alpha\nabla\mathfrak{I}_\nu\big(\Theta(\mathrm{k})\big) + \gamma_\mathrm{m}\Delta\Theta(\mathrm{k}-1),$$

where $\Delta\Theta(k) = \Theta(k+1) - \Theta(k)$, α is the learning coefficient, $\nabla\mathfrak{I}_\nu\big(\Theta(k)\big)$ is the gradient of local criterion function on the corresponding data set and γ_m is the nonnegative momentum term which speeds up the learning convergence while attenuating the parasitic oscillations [12]. If the parameter vector Θ is to be updated using more than one data sample, we consider two different learning styles: (i) *incremental learning* in which the model parameters are updated consecutively after each data sample is presented to the model; and (ii) *batch learning* in which the parameters are updated once after all the data samples are presented. The recursive ANN learning is performed using MATLAB® Neural Network Toolbox [34].

The on-line tuning parameters, learning coefficient α and momentum term γ_m can be determined based on the initial set of data that were used for obtaining the initial model. However, those data might not contain an evident variation in predicted variable, thus no significant difference in the performance of off-line and on-line model would be observed. Therefore, on-line tuning parameters can be determined based on the performance of on-line prediction model on the modified testing data – e.g. a linear trend is added to the original data such that predicted variable mean increases by 50% of the initial mean per month.

5.3.5 Concept of conditional adaptation (outliers handling)

In addition to the normal operation, another possible scenarios which affect the prediction system can occur. In the normal operation scenario we assumed that data do not contain potentially irregular or corrupted data samples (referred to as *outliers*). However, it is often the case that data on actual data are corrupted -- using these data samples within the on-line tuning procedure could cause an undesirable model behavior. Instantaneous change in mean may be a result of many different external factors that influence the predicted variable, but it may also be caused by a meter problem – in the latter case data are characterized as corrupted.

The basic idea in avoiding the on-line tuning procedure using corrupted data is by marking those data, i.e. if a data sample is suspected to be an outlier, it is marked and that data sample will not be used in the on-line tuning procedure. In order to recognize an outlier occurrence, min/max values of the model inputs are used as boundaries for filtering the outliers.

Bibliography

[1] Deliverable D4.1.1: Vašak et al., - Building-side EMS concept and information exchange interfaces definition, University of Zagreb, Faculty of Electrical Engineering and Computing, 2017.

[2] Y.A. Çengel, M.A. Boles, Thermodynamics: An Engineering Approach, McGraw-Hill Higher Education, 2008. https://books.google.rs/books?id=hmkdogEACAAJ.

[3] G.F. Hundy, A.R. Trott, T.C. Welch, Chapter 2 - The Refrigeration Cycle BT - Refrigeration and Air-Conditioning (Fourth Edition), in: Butterworth-Heinemann, Oxford, 2008: pp. 15–29. doi:https://doi.org/10.1016/B978-0-7506-8519-1.00002-5.

[4] Cooling Equipment, in: Energy Audit of Building Systems, CRC Press, 2010: pp. 9–22. doi:doi:10.1201/b10342-10.

[5] http://www.ddbst.com/online.html, (n.d.).

[6] T.T. Al-Shemmeri, Engineering Fluid Mechanics, 2012.

[7] https://syeilendrapramuditya.wordpress.com/2011/08/20/water-thermodynamic-properties/, (n.d.).

[8] V. Gnielinski, New Equations for Heat and Mass Transfer in Turbulent Pipe and Channel Flow, International Chemical Engineering. 16 (1976) 359–368.

[9] A.R. Day, M.S. Ratcliffe, K.J. Shepherd, Heating Systems, Plant and Control, Blackwell Science, 2003.

[10] GRUNDFOS DATA BOOKLET, In-line circulator pumps, TP, TPD, TPE, TPED, TPE2, TPE2 D, TPE3, TPE3 D, V7124417 0117, ECM: 1201316, GRUNDFOS A/S, DK-8850 Bjerringbro. Denmark

[11] Hydraulics in building systems, Siemens Ltd, Building Technologies division, Switzerland, http://www.siemens.com/buildingtechnologies.

[12] I. Petrović, M. Baotić, and N. Perić, Inteligentni sustavi upravljanja: Neuronske mreže, evolucijski i genetički algoritmi. University in Zagreb Faculty of Electrical Engineering and Computing, 2011.

[13] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133, dec 1943.

[14] W. Pitts andW. S. McCulloch, “How we know universals: The perception of auditory and visual forms,” Bulletin of Mathematical Biophysics, vol. 9, pp. 127–147, 1947.

[15] K. Levenberg, “A method for the solution of certain problems in least squares,” Quart. Applied Math., vol. 2, pp. 164–168, 1944.

[16] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” SIAM Journal on Applied Mathematics, vol. 11, no. 2, pp. 431–441, 1963.

[17] H. R. Maier and G. C. Dandy, “Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications,” Environmental Modelling & Software, vol. 15, no. 1, pp. 101 – 124, 2000.

[18] H. R. Maier, A. Jain, G. C. Dandy, and K. Sudheer, “Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions,” Environmental Modelling & Software, vol. 25, no. 8, pp. 891 – 909, 2010.

[19] R. May, H. Maier, and G. Dandy, “Data splitting for artificial neural networks using SOM based stratified sampling,” Neural Networks, vol. 23, no. 2, pp. 283 – 294, 2010.

[20] R. J. May, H. R. Maier, G. C. Dandy, and T. G. Fernando, “Non-linear variable selection for artificial neural networks using partial mutual information,” Environmental Modelling & Software, vol. 23, no. 10–11, pp. 1312 – 1326, 2008.

[21] K. Suzuki, Artificial Neural Networks - Methodological Advances and Biomedical Applications. InTech, 2011.

[22] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” Phys. Rev. E, vol. 69, p. 066138, Jun 2004.

[23] S. Frenzel and B. Pompe, “Partial mutual information for coupling analysis of multivariate time series,” Phys. Rev. Lett., vol. 99, p. 204101, Nov 2007.

[24] A. Sharma, “Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — A strategy for system predictor identification,” Journal of Hydrology, vol. 239, no. 1–4, pp. 232 – 239, 2000.

[25] B.W. Silverman and P. J. Green, Density Estimation for Statistics and Data Analysis. London: Chapman and Hall, 1986.

[26] T. Duong and M. L. Hazelton, “Cross-validation bandwidth matrices for multivariate kernel density estimation,” Scandinavian Journal of Statistics, vol. 32, no. 3, pp. 485–506, 2005.

[27] L. Ngia and J. Sjoberg, “Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm,” IEEE Transactions on Signal Processing, vol. 48, no. 7, pp. 1915–1927, 2000.

[28] I. Rivals and L. Personnaz, “A recursive algorithm based on the extended Kalman filter for the training of feedforward neural models,” Neurocomputing, vol. 20, no. 1–3, pp. 279–294, 1998.

[29] X.Wu and Y.Wang, “Extended and Unscented Kalman filtering based feedforward neural networks for time series prediction,” Applied Mathematical Modelling, vol. 36, no. 3, pp. 1123–1131, 2012.

[30] R. Zhan and J.Wan, “Neural network-aided adaptive unscented Kalman filter for nonlinear state estimation,” IEEE Signal Processing Letters, vol. 13, no. 7, pp. 445–448, 2006.

[31] R. Van der Merwe and E. Wan, “The square-root unscented Kalman filter for state and parameter-estimation,” in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 6, pp. 3461–3464, 2001.

[32] S. Haykin, Kalman Filtering and Neural Networks. Adaptive and Learning Systems for Signal Processing, Communications and Control Series,Wiley, 2004.

[33] L. Padilla and C. Rowley, “An adaptive-covariance-rank algorithm for the unscented Kalman filter,” in Proceedings of the IEEE Conference on Decision and Control, pp. 1324–1329, 2010.

[34] MATLAB, version 8.1.0 (R2013a). Natick, Massachusetts: The MathWorks Inc., 2013.

6 Nomenclature

COP	-	Coefficient of Performance
t_{room}	K	Room temperature
t_{env}	K	Environment temperature
Q_{room}	kW	Heat flux to a heated space
Q_{env}	kW	Heat flux from cold environment to a heat pump
$W_{net,in}$	kW	Input mechanical power provided to a heat pump
P_{el}	kW	Electric power
$\dot{Q}_{HE}$	kW	Heat transfer rate across the heat exchanger
t_m	°C	Mean temperature of the water
$t_{w,in}$	°C	Inlet water temperature
$t_{w,out}$	°C	Outlet water temperature
$\Delta\tau$	s	Sampling time
$E_{t,HE}$	kWh	Delivered heat to the water in the heat exchanger
COP_H	-	Carnot Coefficient of performance for heating mode
$\dot{Q}_{out}$	kW	Output heat power
W_{in}	kW	Input power
ΔS	kJ/K	Entropy change
COP_C	-	Carnot Coefficient of performance for cooling mode
$\dot{Q}_{in}$	kW	Input heat power
COP_{Real}	-	Real coefficient of Performance
ΔT_i	°C	The fluid temperature change along the section „i“
f	-	Pump efficiency factor
c_{pw}	J/kg K	Specific heat capacity of water
δ	mm	the insulation thickness
k_{ins}	W/mK	thermal conductivity of the insulation
D	mm	inner diameter of the pipe
L	m	the length of the pipe
$t_{w,mean}$	°C	Mean water temperature
ρ	kg/m^3	Density
A,B,C,D	-	The equation coefficients
μ	Pas	Dynamic viscosity
k	W/mK	Thermal conductivity
ω	m/s	Water velocity
Re	-	Reynolds number
Pr	-	Prandtl number
Nu	-	Nusselt number
F	-	Friction factor
h_{in}	W/m^2K	The heat transfer coefficient at the contact surface between water and pipe
h_{out}	W/m^2K	Heat transfer coefficient at the contact surface between air and pipework

U	W/m^2K	Overall heat transfer coefficient
$\dot{q}_l$	W/m	Heat losses/gains per unit length of pipe
$\dot{Q}$	W	Heat gains/losses of the pipe
$t_{w,out,LAST}$	°C	Temperature used in iteration process
LTHW	-	Low temperature hot water
MTHW	-	Medium temperature hot water
HTHW	-	High temperature hot water
η	%	Pump efficiency
Q	m^3/s	Pump water flow rate
H_{nom}	m	Nominal pump head
q_{nom}	m^3/s	volume (or mass) flow at nominal working regime
n_{nom}	rpm	pump rpm at nominal working regime
H_{uk}	-	Total efficiency of pump and engine
Qs	m^3/s	volume (or mass) flow required by FC
$Qs_{,nom}$	m^3/s	volume (or mass) flow at nominal working regime
Δp	Pa	Pressure differential
$\dot{V}$	m^3/s	Volume flow rate
Ns	-	Pump specific speed
BEP	-	Best efficiency point pump
p_{nom}	Pa	nominal differential pressure measured on pump inlet/outlet
p_{end}	Pa	differential pressure at inlet/outlet measured on the „last“ radiator in the longest line - only for radiators/ground floor heating (when all radiators are opened)
$E = f(Q_s, Q_{s,\text{nom}}, p_{\text{nom}}, p_{\text{end}}, \eta_p)$	kW	circulation pump energy consumption

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Model predictive control module for central HVAC system management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER)
Authors (Partners)	Nikola Hure, Mario Vašak (UNIZGFER)
Contact person	Nikola Hure (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of model predictive control module on the level of the central HVAC system for hierarchical management of building subsystems. The module logic is provided in more detail in the annexed document.
Keyword List	Medium Temperature, Medium Flow, Heat Pump, Heat Exchanger, Energy, Cost, Thermal Energy, Electricity, Model Predictive Control

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.3.1	Mario Vašak (UNIZGFER)
V0.2	15 January 2019	Updated version	Mario Vašak, Nikola Hure (UNIZGFER)
V0.3	7 June 2019	Updated version	Mario Vašak, Nikola Hure (UNIZGFER)
V1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing – and the algorithms are implemented via a sequence of modules.

The modules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on central HVAC system level model predictive control.

The central HVAC system level model predictive control modules are presented via corresponding interfacing tables that explain what data are used by them as inputs and what are their output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following model predictive control submodules are designed, commissioned and tested on the central HVAC system level:

HVAC.MPC.1 – module for model predictive control that decides on the starting temperature and possibly flow of the medium for maintaining the building climate coming from a heat exchanger (tested in UNIZGFER, HEP, IDRIJA buildings, STREM school, STREM retirement and care centre and EON buildings within 3Smart);

HVAC.MPC.2 – module for model predictive control that decides on the starting temperature of the medium for maintaining the building climate coming from a heat pump (tested in UNIZGFER, HEP, STREM retirement and care centre, EPHZHB and EON buildings within 3Smart).

In the following chapter the modules are presented with their interface tables showing which data they use as inputs and which data they provide as outputs to be at the disposal to submodules and building actuation elements. Detailed explanations of algorithms behind are provided in the previously delivered 3Smart document D4.3.1 (related to model predictive control). D4.3.1 model predictive control part is updated based on feedback from pilot sites and provided as Annex 1 to this document.

Source and sink for the data used by the module is a properly structured 3Smart database. Its structure in the part concerned by the module is provided in Annex 2.

2 HVAC.MPC.1 and HVAC.MPC.2 submodules

HVAC.MPC.1 submodule is used for model predictive control that decides on the starting temperature and flow of the medium for maintaining the building climate coming from a heat exchanger. Within 3Smart it is tested in UNIZGFER, HEP, IDRIJA buildings, STREM school, STREM retirement and care centre and EON buildings.

HVAC.MPC.2 submodule is used for model predictive control that decides on the starting temperature and flow of the medium for maintaining the building climate coming from a heat pump. Within 3Smart it is tested in UNIZGFER, HEP, STREM retirement and care centre, EPHZHB building and EON buildings.

The module interface is defined in Table 2.1.

Table 2.1: Input-output variables list of the heating substation/heat pump control module.

	Source/destination submodule	Variable
Inputs	*Zone-level submodules*	- temperature predictions in the zones T_z
		- predicted required thermal energy input in the zones $E_{t,z}$
		Parameters of the heating/cooling zone elements

		model (for maximum attainable heating energy and electricity consumption): - fan coils: $a_{\mathrm{fc},v}$, $b_{\mathrm{fc},v}$, $l_{\mathrm{fc},v}$, $\bar{E}_{\mathrm{e,fc},j}$, $E_{t,\mathrm{fc},0}$ (please refer to Section 3.5 of Annex 1) - radiators: $a, b, n_{\mathrm{r}}, m_{\mathrm{r}}, c_{\mathrm{r}}, U_0$ (please refer to Section 3.6) - underfloor heating: $m_{\mathrm{fh}}, c_{\mathrm{fh}}, U_0$ (please refer to Section 3.6)
		- measured medium outlet temperatures on the radiator/underfloor heating/cooling units $T_{w,\cdot}^{out}$ (please refer to Section 3.6)
	Central HVAC system level submodules	- predicted non-controllable thermal energy loads $E_{\mathrm{t,nc}}$
		Hydraulic model of the plant: - parameters of the Q-p characteristics, C_{Q-p} (please refer to Section 3.2) - parameters of the medium flow model (nominal conditions) $C_{q,j}, j \in \mathcal{J}$
		- parameters of the heat pump coefficient of performance COP, γ_{p}
		- parameters of the temperature model at the zone element inlets $C_{T,j}, j \in \mathcal{J}$
		- parameters of the heat losses model in the pipework (defined with $C_{T,i}$ and $C_{q,i}$)
		- parameters of the hydraulic pump model electricity consumption V_{q}, p_{end}
	Microgrid level submodules	- electricity cost function and constraints on the prediction horizon $J_M^*(E_e)$
	Other	- air ambient temperature T_{env} (please refer to Section 3.4), the medium heat capacity c_w - cost of heating energy c_{t}
Outputs	*Zone level submodules*	- heating cost and constraints for different zones on the prediction horizon $J_H^*(E_{t,z})$
	Zone level submodules /Central HVAC system level submodules	- predicted temperatures and flows of the supplied medium over the prediction horizon T_{s}, $Q_{\mathrm{s,nom}}$ or Q_{s}
	Microgrid level submodules	- predicted electrical energy consumption E_{e} of the heating/cooling system on the prediction horizon
	District heating system operator (optional)	- predicted thermal energy consumption E_t of the heating/cooling system on the prediction horizon

Bibliography

[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 – Open software module for the central HVAC system level consumption management – Model predictive control module

Provided as a separate document.

Annex 2 – 3Smart database organization for open software module for the central HVAC system level consumption management – Model predictive control module

hvac_mpc1	
PK. hvac_mpc1_id	Int
description	varchar(250)
timestamp	timestamp without time zone
solver	Int

hvac_mpc1_inputs	
FK. hvac_mpc1_id	Int
heat_price_profile	varchar(2000)

hvac_mpc1_outputs	
FK. hvac_mpc1_id	Int
FK. heating_substation_id	Int
timestamp	timestamp without time zone
supply_medium_nominal_flow_ref	real
supply_medium_temperature_ref	real
mixing_valve_medium_temperature_ref	real
supply_medium_nominal_flow_ref_profile	varchar(2000)
supply_medium_temperature_ref_profile	varchar(2000)
mixing_valve_medium_temperature_ref_profile	varchar(2000)

Comments:
1. heat_price_profile in inputs table should be in the form:
Format: {"resolution:": "15T", "starttime": some_timestamp, "prediction": [0.1, 0.2, ...]} -- values denote the price in euros
If the prediction is not specified some predefault heat price will be used within model.
2. Database should propagate the values written to columns supply_medium_nominal_flow_ref, supply_medium_temperature_ref and mixing_valve_medium_temperature_ref towards the actuators when 3smart switch is enabled for the heating

hvac_mpc1_outputs_history	
PK. id	bigint
FK. hvac_mpc1_id	Int
FK. heating_substation_id	Int
timestamp	timestamp without time zone
supply_medium_nominal_ref	real
supply_medium_temperature_ref	real
mixing_valve_medium_temperature_ref	real
supply_medium_nominal_flow_ref_profile	varchar(2000)
supply_medium_temperature_ref_profile	varchar(2000)
mixing_valve_medium_temperature_ref_profile	varchar(2000)

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 MODEL PREDICTIVE CONTROL IN THE CENTRAL HVAC SYSTEM

Open software module for central heating/cooling system management – Model predictive control module

Project Acronym	3Smart			
Grant Agreement No.	DTP1-502-3.2-3Smart			
Funding Scheme	Interreg Danube Transnational Programme			
Project Start Date	1 January 2017			
Project Duration	30 months			
Work Package	4			
Task	4.5			
Date of delivery	**Contractual:** 30 June 2019		**Actual:** 30 June 2019	
Code name	**Version:** 2.0	Final ☒	Final draft ☐	Draft ☐
Type of deliverable	Report			
Security	Restricted			
Deliverable participants	UNIZGFER			
Authors (Partners)	Nikola Hure, Mario Vašak (UNIZGFER)			
Contact person	Nikola Hure (UNIZGFER)			

Abstract

(for dissemination)

This deliverable discusses the model predictive control (MPC) design procedure for the heating substation and heat pump control in presence of the time-varying prices for the heat and electricity. MPC module on each level enables coordination with neighbouring levels in the hierarchical organization, so here coordination with zone-level and microgrid-level MPC is considered.

Keyword List

MPC, heating substation, heat pump, energy losses, hydraulic pump, pipework, fan coil, radiator, underfloor heating

Revision history

Revision	Date	Description	Author (Organization)
v0.1	15 November 2017	Alpha version	Nikola Hure and Mario Vašak (UNIZGFER)
v0.2	22 November 2017	Beta version	Nikola Hure and Mario Vašak (UNIZGFER)
v1.0	29 November 2017	Final draft	Nikola Hure and Mario Vašak (UNIZGFER)
V1.1	08 December 2017	Revised final draft	Nikola Hure and Mario Vašak (UNIZGFER)
V1.2	21 December 2017	Final version of D4.3.1 MPC modules	Nikola Hure and Mario Vašak (UNIZGFER)
V1.3	15 January 2019	Taken over D4.3.1 for further development based on feedback from execution on pilots	Mario Vašak (UNIZGFER)
V1.4	7 June 2019	Updated version	Mario Vašak, Nikola Hure (UNIZGFER)
V2.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

The building management is achieved with the optimal control that considers the dynamical model of the building, its current conditions and available predictions of aimed system operation and disturbances to optimise the building performance index under provided constraints. Specifically, this annex to D4.5.3 document discusses the design of the model predictive control (MPC) module for the central heating, ventilation and air conditioning (HVAC) system level.

Objective of the designed HVAC MPC module is conditioning of the medium which is used for heating/cooling in the building, while considering the required thermal energy flows to the zones, thermal losses in the pipework, its temperature and hydraulic model, model of the hydraulic pump and models of the heating/cooling elements in the zones. Variables that are optimised by the module are the outgoing temperature of the medium and possibly also the flow of the medium. Objective of the derived MPC module is to ensure delivery of the required amount of thermal energy to the zone elements on the prediction horizon at the minimum cost.

1. Nomenclature

In the remainder of this deliverable is the following notation employed. Variable $T_s(k)$ denotes the temperature of the supplied medium at the time instant k. Discrete time indices are denoted with $k \in \{0,1,2,\dots,N\}$, zone indices are $i = \{1,2,\dots,n_z\}$, and element indices are $j \in \{1,2,\dots,n_e\}$, where N is the length of the prediction horizon, n_z is the number of zones and n_e is the overall number of heating/cooling elements throughout all zones.

1.1 Symbols

T_d	sampling time [s]
T_s	temperature of the medium supplied by the central HVAC system [°C]
T_z	temperature in zone(s) [°C]
$T_{w,e}^{av}$	mean medium temperature in the heating/cooling element [°C]
$T_{w,e}^{in}$	medium temperature at the inlet of the heating/cooling element [°C]
$T_{w,e}^{out}$	medium temperature at the outlet of the heating/cooling element [°C]
$T_{a,e}^{in}$	air temperature at the inlet of the heating/cooling element [°C]
T_{env}	temperature of the environment [°C]
Q_a	air volume flow [m^3/h]
Q_s	medium flow supplied by the hydraulic pump [m^3/h]
$Q_{s,nom}$	medium flow supplied by the hydraulic pump in the nominal conditions (all valves open) [m^3/h]
U_o	overall heat transfer coefficient of a system [W/K]
$E_{e,fc}$	electrical energy consumed by the fan coil unit in the discretisation interval [kWh]
$E_{e,p}$	electrical energy consumed by the pump in the discretisation interval [kWh]
$E_{t,l}$	pipework thermal losses in a single discretisation interval [kWh]
$E_{t,fc}$	thermal energy of a fan coil unit affecting the zone in a single discretisation interval [kWh]
$E_{t,fh}$	thermal energy of a floor heating affecting the zone in a single discretisation interval [kWh]
$E_{t,hp}$	thermal energy generated on a heat pump in a single discretisation interval [kWh]
$E_{t,r}$	thermal radiator energy affecting the zone in a single discretisation interval [kWh]
$E_{t,z}$	required thermal energy input to a zone in a single discretisation interval [kWh]
$E_{t,nc}$	non-controllable thermal energy loads in the discretisation interval [kWh]
P_p	consumed electrical power on the hydraulic pump [kW]
V_p	vector of parameters for a hydraulic pump efficiency model
COP	heat pump coefficient of performance

c_w specific heat capacity of a medium [J/kg/K]

c_a specific heat capacity of air [J/kg/K]

c_e cost of electrical energy [€/kWh]

c_t cost of thermal energy [[€/kWh]]

c_r specific heat capacity of a radiator system (medium + radiatior) [J/kg/K]

c_fh specific heat capacity of a floor heating system (concrete+medium) [J/kg/K]

m_w mass of a medium within a system [kg]

m_a air mass within a system [kg]

m_r effective mass of a radiator system [kg]

m_fh effective mass of a floor heating system [kg]

p_p differential pressure on a hydraulic pump [Pa]

$p_\mathrm{p,nom}$ nominal differential pressure on a hydraulic pump (all valves open) [Pa]

p_end pressure on a last branch of a pipework [Pa]

ρ_w density of a medium $[\mathrm{kg/m^3}]$

η_p hydraulic pump efficiency factor

γ_p heat pump efficiency factor

$\cdot$ general placeholder

1.2 Abbreviations

MPC model predictive control

SLP sequential linear programming

SQP sequential quadratic programming

HVAC heating, ventilation and air conditioning

2. General scheme

This deliverable discusses design of a model predictive controller (MPC) [1] for a heating substation and a heat pump on the central heating, ventilation and air conditioning (HVAC) system level of the hierarchy of the building-side EMS introduced in 3Smart via the concept document D4.1.1 [2]. Aim of the designed control module is to ensure that delivery of a required amount of thermal energy to the zone level $E_\mathrm{t,z}$ is possible, where the fan coils, radiators and underfloor heating are employed, and yet with a minimum cost. The required energy inputs are issued from the zone-level MPC module (see D4.5.3 – zone level MPC).

Variables that are optimised by the module are the outgoing temperature of the medium T_s and possibly also the flow of the medium Q_s within sampling periods of the prediction horizon. The formulated optimisation problem considers the models of the actuated elements in the zones, heat losses in the pipework, efficiencies of the central HVAC system units, predictions of the required thermal energy input to all zones $E_{\mathrm{t,z}}$ on the horizon, as well as the variable prices of the electrical and thermal energy on the prediction horizon, c_{e} and c_{t} respectively.

Intrinsically, the considered optimisation problem has non-convex constraints and the objective function. Herein, the proposed solution for solving the nonlinear MPC problem employs a sequential linear programming (SLP) method. Compared to the sequential quadratic programming (SQP) counterpart, SLP has a slower convergence but neither requires the convexity of the objective function nor the continuity of the corresponding Hessian matrix in the domain of the problem to be deployed.

3. Mathematical models

The control problem is stated with respect to all of the significant integrated subsystems. Without introducing any significant inaccuracy in the model, static models are assumed for certain subsystems, e.g. the electrical energy consumed on the heating elements, heat losses in the pipework etc. The former simplification is possible since the dynamics of respective subsystems are significantly faster compared to the discretisation interval of the heating substation or the heat pump T_{d}. Mathematical models of the subsystems employed for the design of the model predictive control problem are given in the following sections.

For a more detailed description of the heat pump model, hydraulic pump model and the heat losses in the pipework please refer to [3]. Technical report on the subject of the zone elements modelling and estimation is discussed in [4].

3.1 Medium flow through the pipework

From the mass medium flow through the controlled hydraulic pump of the heating/cooling system follow the medium flows through each segment of the pipework

$$Q_j = C_{q,j} Q_s, \#(3-1)$$

where Q_j is the flow through the j-th pipework segment, $C_{q,j}$ is the static coefficient associated to the j-th segment and Q_s is the realised medium flow through the hydraulic pump. By employing the mass conservativeness law, the realised mass flow Q_s is computed by considering the medium flows to the zones

$$Q_s = \sum_v Q_v \,, \#(3-2)$$

where v denotes the index from the set of zone branches. Herein, Q_v may denote the medium flow through the fan coils and radiators/underfloor heating units, depending on the configuration.

The medium flows through the system are influenced depending on the type of the employed zone actuator. Thus, if the heating is realised through the fan coils, a static hydraulic situation may be considered. In the respective scenario, the flow through each branch is determined solely by the supplied flow through the hydraulic pump, resulting with a single flow decision variable Q_s per time instant of the prediction horizon.

Conversely, by assuming that radiators/underfloor heating units are used as actuators in the zone level, the optimised flow through the hydraulic pump Q_s is determined by the flows through respective heating/cooling units, denoted with Q_q, as described in Section 3.6. Respective actuators are controlled by the installed valves that determine the effective flows Q_r that ensure required amount of thermal energy in the zones for a selected temperature of the medium.

Remark. The assumption of the derived model is that the state of the actuated valves in the zones does not influence the hydraulic conditions in surrounding environment of the pipework, i.e. the pressure drop on each of the parallel branches is approximately constant regardless of the locally applied controls to the valves, or the hydraulic pump is controlled in such a way that this influence is minimized.

3.2 Electric energy consumption of a hydraulic pump

The electric energy consumption of a hydraulic pump is incorporated in the electricity cost of the heating/cooling system. With regards to the employed actuators in the zones are two different scenarios employed:

i) Static hydraulic conditions with fan coils in the zones:

Consumed electrical energy on the pump is expressed as a function of the supplied medium flow and the differential pressure on the pump [3]

$$P_p = \frac{10.2}{10^5} g \cdot \eta_p^{-1} \cdot \frac{Q_s}{3.6} \cdot p_p, \#(3-3)$$

where g is the gravitational acceleration constant, ρ_w is the medium density, p_p is the differential pressure on the pump and η_p is the pump efficiency. The differential pressure on the pump is determined by the Q- p characteristics of the plant [3], which is given by

$$p_p = C_{Q-p}[Q_s^2, Q_s, 1]^\top, \#(3-4)$$

where $C_{Q-p} \in \mathbb{R}^3$ is the identified vector of parameters.

The pump efficiency can be accurately approximated by a second-degree polynomial

$$\eta_\mathrm{p} = V_\mathrm{p}[Q_s^2, Q_s, 1]^\top, \#(3-5)$$

where $V_p \in \mathbb{R}^3$ is the vector of identified model parameters.

ii) Dynamic hydraulic conditions with radiators/underfloor heating in the zones:

Model of the consumed electrical energy on the hydraulic pump is given by [3]

$$P_p = \frac{10.2}{10^5} \cdot \mathrm{g} \cdot \eta_{\mathrm{p}}^{-1} \cdot \frac{Q_s}{3.6}\left(p_{\mathrm{end}} + \left(p_{s,\mathrm{nom}} - p_{\mathrm{end}}\right)\frac{Q_{\mathrm{s}}}{Q_{\mathrm{s,nom}}}\right), \#(3-6)$$

where the index $_{\mathrm{nom}}$ in the variable symbol denotes the condition that all valves of the radiator/underfloor heating system are open. Thus, $p_{s,\mathrm{nom}}$ is the pressure on the pump and $Q_{s,\mathrm{nom}}$ the flow through the pump given that all valves on the radiator/underfloor heating actuators are open, whereas p_{end} is the predefined pressure on the farthest branch of the radiator/underfloor heating/cooling unit [3].

In this scenario, Q_s is determined by the medium flows through the pipework branches with heating elements Q_q (3-2). The nominal medium flow $Q_{s,\mathrm{nom}}$ is the decision variable in the optimisation problem, whereas the following relation applies between Q_q and $Q_{s,\mathrm{nom}}$

$$0 \le Q_q \le C_{q,j}\left[Q_{s,\mathrm{nom}}^2, Q_{s,\mathrm{nom}}, 1\right]^{\mathsf{T}}. \#(3-7)$$

3.3 Heat losses model and temperature model

Temperature difference between the medium in the pipework and the surroundings incurs the thermal energy losses. It is necessary to model respective losses to ensure that the required amount of thermal energy is attainable on the zone elements [3], which is achieved by designating the supplied thermal energy from the central HVAC system towards the building. Heat losses and the temperatures in the pipework are usually described with the coupled model resulting with the implicit relations [3].

Herein, the simplified and explicit model of the heat losses is considered that can be employed in the MPC design. Thus, employed temperature model is [3] given by

$$T_{w,i}^{in} = C_{T,i}[T_s, Q_s^2, Q_s, 1]^{\mathsf{T}}, \qquad i \in \mathcal{I}, \#(3-8)$$

where $T_{w,i}^{in}$ is the medium temperature at the beginning of the i-th segment and $C_{T,i} \in \mathbb{R}^4$ is the identified vector of model parameters. The temperature model together with the model of medium flows through the segments is used to derive the energy losses model in the pipework, whereas the model of the medium inlet temperatures at the zone elements $C_{T,j}, j \in \mathcal{J}$ is used to account for the attainable thermal energies in the zones.

The thermal energy losses model is given by

$$E_{t,l} = E_{t,l,s} + E_{t,l,r}, \#(3-9)$$

where $E_{t,l,s}$ denotes the thermal losses of the supply pipeworks (before the zone heating/cooling elements) and $E_{t,l,r}$ the thermal losses of the return pipework (after the zone heating/cooling elements). The thermal losses model of the supply pipework is obtained by employing the temperature drop model and hydraulic model of flows through the pipework segments

$$E_{t,l,s} = \frac{1}{12.96 \cdot 10^6}\left(T_d Q_s c_s\left(T_s - T_{w,2}^{in}\right) + \sum_{i \in \mathcal{I}} \sum_{l \in \mathcal{L}_i} T_d Q_i c_s\left(T_{w,i}^{in} - T_{w,l}^{in}\right) \right) = f_{Etls}(Q_s, T_s), \#(3-10)$$

where $\mathcal{L}_i$ denotes the set of pipework branches directly connected to the supply branch i. Similarly, the model of the thermal losses on the return pipework is given by

$$E_{t,l,r} = \frac{1}{12.96 \cdot 10^6} \sum_{i \in \mathcal{I}} \sum_{l \in \mathcal{L}_i} T_d Q_i c_s\left(T_{w,i}^{out} - T_{w,l}^{out}\right) = f_{Etlr}(Q_{\mathcal{J}}, T_{w,\mathcal{J}}^{out}), \#(3-11)$$

where $Q_{\mathcal{J}}$ denotes the vector of medium flows through the zone elements, which are indexed with the set $\mathcal{J}$, and $T_{w,\mathcal{J}}^{out}$ denotes the temperature at the outlet of the same elements. The outlet temperatures on the heating/cooling elements follow from their thermodynamic model, which are discussed in the following sections.

3.4 Heat pump model

The heat pump thermal model is derived from the characteristic equation of an ideal Carnot cycle [3]. Coefficient of performance of the real heat pump for heating mode is given by

$$COP = \gamma_{\mathrm{p}} \frac{T_{w,hp}^{av} + 273.15}{T_{w,hp}^{av} - T_{\mathrm{env}}}, \#(3-12)$$

where γ_{p} is the parameter of efficiency characterising a specific heat pump, T_{env} is the temperature of the ambient air and $T_{w,hp}^{av}$ is the mean temperature of the medium in the heat pump,

$$T_{w,hp}^{av} = \frac{T_{w,hp}^{in} + T_{w,hp}^{out}}{2}, \#(3-13)$$

where $T_{w,hp}^{in}$ denotes the temperature of the return medium from the heating/cooling circuit and $T_{w,hp}^{out} \equiv T_s$ is the temperature of the supplied medium.

Coefficient of performance is in the cooling mode given by

$$COP = \gamma_{\mathrm{p}} \frac{T_{w,hp}^{av} + 273.15}{T_{\mathrm{env}} - T_{w,hp}^{av}}. \#(3-14)$$

The energy balance equation on the heat exchanger in a single discretisation interval is given by

$$E_{t,\mathrm{hp}} = \frac{1}{12.96 \cdot 10^6} T_{\mathrm{d}} Q_s c_{\mathrm{w}}\left(T_s - T_{w,hp}^{in}\right), \#(3-15)$$

where c_{w} is the specific heat capacity of the medium for its mean temperature $T_{w,hp}^{av}$ and T_{d} is the discretisation interval. Heating energy generated on the heat pump is dissipated on the heating/cooling elements of the system, through the heat losses and other non-controllable thermal loads. Thus, the thermal energy balance equation in the system is determined with

$$E_{t,\mathrm{hp}} = E_{t,l} + E_{t,\mathrm{nc}} + \sum_i E_{t,z}(i), \#(3-16)$$

where $E_{t,l}$ are the energy losses, $E_{t,\mathrm{nc}}$ the non-controllable thermal loads and the sum on the right-hand side is the cumulative thermal energy output reference on the heating/cooling units in the zones. In (3-15), model of the energy losses $E_{t,l}$ is given by (3-8)-(3-10), non-controllable loads predictions $E_{t,\mathrm{nc}}$ in the horizon are inputs of the HVAC MPC module as well as the required thermal energy to be delivered to the zones $\sum_i E_{t,z}(i)$.

By combining (3-12)-(3-15), model of the heat pump is in the heating mode given by

$$\mathrm{COP} = \gamma_{\mathrm{p}} \frac{2T_{\mathrm{s}} - 12.96 \cdot 10^6 \frac{E_{\mathrm{t,hp}}}{T_{\mathrm{d}} Q_{\mathrm{s}} c_{\mathrm{w}}} + 2 \cdot 273.15}{\left(2T_{\mathrm{s}} - 12.96 \cdot 10^6 \frac{E_{\mathrm{t,hp}}}{T_{\mathrm{d}} Q_{\mathrm{s}} c_{\mathrm{w}}}\right) - 2T_{\mathrm{env}}}, \#(3-17)$$

whereas the heat pump model in the cooling mode is

$$\mathrm{COP} = \gamma_{\mathrm{p}} \frac{2T_{\mathrm{s}} - 12.96 \cdot 10^6 \frac{E_{\mathrm{t,hp}}}{T_{\mathrm{d}} Q_{\mathrm{s}} c_{\mathrm{w}}} + 2 \cdot 273.15}{2T_{\mathrm{env}} - \left(2T_{\mathrm{s}} - 12.96 \cdot 10^6 \frac{E_{\mathrm{t,hp}}}{T_{\mathrm{d}} Q_{\mathrm{s}} c_{\mathrm{w}}}\right)}. \#(3-18)$$

Final expressions for the heat pump are obtained by employing (3-9)-(3-11),(3-16) in (3-17)/(3-18).

3.5 Fan coil unit

Thermal model of the fan coil unit consists of the air and medium thermal balance equations [2]. By neglecting the dynamics of the water mass cooling/heating in the considered fan coil unit, which is here assumed due to a significantly slower sampling rate of the designed control system than is the value of the respective medium thermal time constant, the following equations are obtained

$$0 = \frac{1}{3.6} Q_j c_w \left(T_{w,j}^{in} - T_{w,j}^{out}\right) - U_{0,j}\left(T_{w,j}^{av} - T_{a,j}^{in}\right), \#(3-19)$$

$$0 = \frac{1}{3.6} Q_{a,j} c_a \left(T_{a,j}^{in} - T_{a,j}^{out}\right) + U_{0,j}\left(T_{w,j}^{av} - T_{a,j}^{in}\right), \#(3-20)$$

where $j \in \mathcal{F}$ denotes the index of the fan coil and $\mathcal{F} \subseteq \mathcal{J}$ is a set of all fan coils indices, $T_{w,j}^{in}$ and $T_{w,j}^{out}$ are the medium temperatures at the inlet and outlet of the considered fan coil respectively, $T_{w,j}^{av}$ is the mean medium temperature in the fan coil unit, $T_{a,j}^{in}$ and $T_{a,j}^{out}$ are the air temperatures in the inlet and outlet respectively, $Q_{a,j}$ is the air flow through the fan coil and Q_j the medium flow through the same fan coil unit, c_a and c_s are the specific heat capacities of air and medium respectively, U_0 is the corresponding heat transfer coefficient from water to air. Note that the air inlet temperature equals the temperature of the zone where the fan coil is situated,

$$T_{a,j}^{in} = T_{\mathrm{z},i}, \#(3-21)$$

where i denotes the index of the corresponding zone. The medium flow through the fan coil is obtained from (3-1) and the supply medium temperature at the fan coil inlet from (3-8).

The mean medium temperature in the fan coil unit is given by

$$T_{w,j}^{av} = \frac{T_{w,j}^{in} + T_{w,j}^{out}}{2}. \#(3-22)$$

Thermal energy supplied to the zone by the fan coil unit is determined with

$$E_{\mathrm{t,fc},j} = \frac{1}{12.96 \cdot 10^6} T_{\mathrm{d}} Q_j c_w \left(T_{w,j}^{in} - T_{w,j}^{out}\right) = \frac{1}{12.96 \cdot 10^6} T_{\mathrm{d}} Q_{a,j} c_a \left(T_{a,j}^{in} - T_{a,j}^{out}\right). \#(3-23)$$

By combining equations (3-18) and (3-22) is the dependence of the water outlet temperature on the fan coil derived, whereas by including the equation (3-21), the thermal energy attainable on the fan coil is obtained,

$$E_{t,\mathrm{fc},j} = \frac{1}{3.6 \cdot 10^6} \mathrm{T_d} \frac{2\frac{1}{3.6} Q_j c_w U_{0,j,3}\left(T_{w,j}^{in} - T_{\mathrm{z},i}\right)}{2\frac{1}{3.6} Q_j c_w + U_{\mathrm{fc},j,3}}, \#(3-24)$$

where $T_{\mathrm{z},i}$ is the air temperature in the zone, $U_{0,j}$ is the heat transfer coefficient of the fan coil unit given by the static characteristics

$$U_{0,j} \equiv U_{0,j,v} = \frac{a_{\mathrm{fc},v}}{1 + b_{\mathrm{fc},v} Q_j^{-l_{\mathrm{fc},v}}}. \#(3-25)$$

In (3-25) $v = \{1,2,3\}$ is the speed of the fan and $a_{\mathrm{fc},v}$, $b_{\mathrm{fc},v}$ and $l_{\mathrm{fc},v}$ are corresponding parameters obtained from the technical datasheet and identification [3]. The requirement for the heating substation module controller is posed such that the maximum available heating energy on the fan coil should be at least as large as is the required energy by the zone-level MPC.

The attainable thermal energy constraint is given by

$$|E_{t,z,j}| \leq |E_{t,\mathrm{fc},j}|, \#(3-26)$$

where $E_{t,z,j}$ is the thermal energy request for the j-th actuator and the absolute function is used to cover both the heating and cooling modes.

The experimental tests have shown that the electrical energy consumption of the fan coil does not differ significantly between the different fan speeds whereas the highest difference is observed between the first speed and the turned off state of the fan operation. Therefore, a simple electrical energy consumption model is assumed herein that considers only a bimodal operation of the fan coil to estimate the consumed electrical energy. If the required thermal energy input to the zone at the time instants is larger than the energy that can be ensured with the first speed of the fan, the electrical equivalent consumption of the fan coil operating at the first speed $\bar{E}_{\mathrm{e,fc},j}$ is assumed. Otherwise, if the required thermal energy input at the time instants is lower than the one that can be acquired by the fan coil operating with the first speed, the consumed electrical energy is given by

$$E_{\mathrm{e,fc},j} = \bar{E}_{\mathrm{e,fc},j} \frac{(3.6 \cdot 10^6)^2 (E_{\mathrm{t,z},j} \pm E_{t,\mathrm{fc},0})}{T_d \frac{2 \frac{1}{3.6} Q_j c_w U_{0,j,1} \left(T_{w,j}^{in} - T_{\mathrm{z},i}\right)}{2 \frac{1}{3.6} Q_j c_w + U_{0,j,1}}}, \#$$

$$(3.6 \cdot 10^6)^2 (E_{\mathrm{t,z},j} \pm E_{t,\mathrm{fc},0}) < T_d \frac{2 \frac{1}{3.6} Q_j c_w U_{0,j,1} \left(T_{w,j}^{in} - T_{\mathrm{z},i}\right)}{2 \frac{1}{3.6} Q_j c_w + U_{0,j,1}}, \qquad (3-27)$$

where $E_{t,\mathrm{fc},0}$ is included to account for the delivered thermal energy while the fan is not operating.

The electrical energy consumption covering both described modes of operation can be derived by employing the saturation function

$$E_{\mathrm{e,fc},j} = \bar{E}_{\mathrm{e,fc},j} \operatorname{sat}\left(\frac{3.6 \cdot 10^6 (E_{t,\mathrm{z},j} \pm E_{t,\mathrm{fc},0})}{T_d \frac{2 \frac{1}{3.6} Q_j c_w U_{0,j,1} \left(T_{w,j}^{in} - T_{\mathrm{z},i}\right)}{2 \frac{1}{3.6} Q_j c_w + U_{0,j,1}}}, 0, 1 \right), \#(3-28)$$

where $\mathrm{sat}(x, a, b)$ is the saturation function of input x with parameters a and b as the lower and the upper saturation limit respectively.

3.6 Radiator and underfloor heating model

Dynamic equations of the radiator is given by [4]

$$\frac{dT_{w,q}^{out}}{dt} = \frac{a}{3.6} Q_q \left(T_{w,q}^{in} - T_{w,q}^{out}\right) - b\left(T_{w,q}^{av} - T_{z,i}\right)^{n_{r,q}}, T_{w,q}^{out} \geq T_{z,i}, \#(3-29)$$

where $q \in \mathcal{Q}$ is the index of the radiator heating unit and $\mathcal{Q} \subseteq \mathcal{J}$ is the set of all indices, Q_q is the effective flow through the radiator in a single discretisation period, $T_{w,q}^{in}$ and $T_{w,q}^{out}$ are the inlet and the outlet temperatures of the medium respectively, $T_{w,q}^{av}$ is the average temperature of the medium in the radiator, parameter a describes the heat power input from the medium, b and $n_{r,q} \geq 1$ describe the heat power transferred to the air in the considered zone.

The flow through the radiator/underfloor heating is changing within the time-instant along with the state of the valve. The effective flow through the radiator/underfloor heating Q_q is estimated as the value that ensures the delivery of the required thermal energy amount in the zones given the initial temperature of the radiator/underfloor heating, air temperature in the zones and the temperature at the inlets of the heating elements.

Simplified thermal model of the underfloor heating is described with [4]

$$m_{fh} c_{fh} \frac{dT_{w,g}^{out}}{dt} = \frac{Q_g}{3.6} c_w \left(T_{w,g}^{in} - T_{w,g}^{out}\right) - U_{0,g}\left(T_{w,g}^{out} - T_{z,i}\right), \#(3-30)$$

where $g \in \mathcal{G}$ is the index of the underfloor heating unit and $\mathcal{G} \subseteq \mathcal{J}$ is the set of all indices, m_{fh} is the effective mass of the underfloor heating and c_{fh} its specific heat capacity, $U_{0,g}$ is the heat transfer coefficient from the medium to air in the zone.

The second term of the right-hand side in (3-30) and scaled coefficient of the same term in (3-29) describes the heat power delivered to the zone. It is approximated by

$$U_{0,q}\left(T_{w,q}^{av} - T_{z,i}\right)^{n_{r,q}} \approx 3.6 \cdot 10^6 \frac{E_{t,z,q}}{T_d},$$
$$U_{0,g}\left(T_{w,g}^{out} - T_{z,i}\right) \approx 3.6 \cdot 10^6 \frac{E_{t,z,g}}{T_d}, \#(3-31)$$

in the discretisation interval, where $E_{t,z,\cdot}$ is the required heat energy of the corresponding heating element in a single time interval. Thus, nonlinear differential equations (3-29)/(3-30) can be approximated by a linear differential equation with parameters $(T_{w,\cdot}^{out}(0), Q_\cdot, E_{t,z,\cdot}, T_{w,\cdot}^{in})$ in solution $T_{w,\cdot}^{out}(t)$, where $\cdot$ denotes the general placeholder symbol. Thus, the recursion for the medium temperature at the outlet $T_{w,\cdot}^{out}$ can be easily derived,

$$T_{w,\cdot}^{out}(kT_d) = f\left(T_{w,\cdot}^{out}((k-1)T_d), Q_\cdot((k-1)T_d), E_{t,z,\cdot}((k-1)T_d), T_{w,\cdot}^{in}((k-1)T_d)\right). \#(3-32)$$

The attainable thermal energy on the radiator/underfloor heating is obtained by integrating the thermal energy power in the discretisation interval,

$$E_{t,\mathrm{r},q} = \frac{1}{3.6 \cdot 10^6} \int_{\mathrm{k}T_\mathrm{d}}^{(\mathrm{k}+1)T_\mathrm{d}} U_{0,q}\left(T_{w,q}^{out} - T_{\mathrm{z},i}\right)^{n_{r,q}} d\tau,$$

$$E_{t,\mathrm{fh},g} = \frac{1}{3.6 \cdot 10^6} \int_{\mathrm{k}T_\mathrm{d}}^{(\mathrm{k}+1)T_\mathrm{d}} U_{0,g}\left(T_{w,g}^{out} - T_{\mathrm{z},i}\right) d\tau. \#(3-33)$$

To allow the exact computation of the integral expression, the temperature of the medium at the outlet of the element is linearly interpolated,

$$T_{w,:}^{out}(\mathrm{t}) \approx \left(1 - \frac{\mathrm{t} - \mathrm{k}T_\mathrm{d}}{T_\mathrm{d}}\right) T_{w,:}^{out}(\mathrm{k}T_\mathrm{d}) + \frac{\mathrm{t} - \mathrm{k}T_\mathrm{d}}{T_\mathrm{d}} T_{w,:}^{out}\left((\mathrm{k}+1)T_\mathrm{d}\right), \mathrm{k}T_\mathrm{d} \leq \mathrm{t} \leq (\mathrm{k}+1)T_\mathrm{d}. \#(3-34)$$

Employed model considers the effective flow value within the discretisation period Q_q which influences the electrical energy consumption at the hydraulic pump. Thus, the medium flow at the central HVAC unit is given by

$$Q_\mathrm{s} = \sum_{q \in \mathcal{Q}} Q_q.$$

4. MPC formulation

Optimisation of the medium flow and temperature is performed to achieve the minimum cost of energy while acquiring the required heating energy outputs to the zones.

The cost function of the considered MPC problem is given by

$$J = \min_{T_\mathrm{s}, Q_{\mathrm{s,nom}}} J_t + J_e, \#(3-35)$$

where

for the heat pump the consumed thermal energy boils down to the cost of electricity, which is given by

$$J_t = \sum_{k=0}^{N-1} \frac{c_\mathrm{e}(k)}{COP(k)} \left(\left|E_{t,\mathrm{nc}}(k)\right| + \left|E_{t,\mathrm{l}}\left(T_\mathrm{s}(k), Q_\mathrm{s}(k)\right)\right| + \sum_{j \in \mathcal{J}} \left|E_{t,\mathrm{z},j}(k)\right| \right), \#(3-36)$$

and in case of the heating substation is the thermal cost described with

$$J_t = \sum_{k=0}^{N-1} c_h(k)\left(E_{t,\mathrm{nc}}(k) + E_{t,\mathrm{l}}\big(T_\mathrm{s}(k), Q_\mathrm{s}(k)\big) + \sum_{j\in\mathcal{J}} E_{t,\mathrm{z},j}(k)\right). \quad (3-37)$$

Consumed electrical energy J_e in (3-36) for the HVAC system which excludes the electrical energy consumption for the preparation of the medium itself by the heat pumps is given by

$$J_\mathrm{e} = \sum_{k=0}^{N-1} c_\mathrm{e}(k)\left(E_{e,p}\big(Q_\mathrm{s}(k)\big) + \sum_{j\in\mathcal{F}} E_{e,\mathrm{fc},j}(k)\right). \#(3-38)$$

Among the heating elements, only the fan coils are considered as the electricity consumers, with the electrical energy consumption model provided with (3-28).

Constraints of the optimisation problem are:

- the attainable energy constraint on the zone elements

$$|E_{t,\cdot,j}| \geq |E_{t,\mathrm{z},j}|, \#(3-39)$$

  here $E_{t,\cdot,j}$ is the attainable thermal energy on the j-th heating/cooling element, which is in case of fan coils given by $E_{t,\mathrm{fc},j}$ (3-24) and in case of radiators/underfloor heating given by $E_{t,\mathrm{r},j}$ and $E_{t,\mathrm{fh},j}$ (3-33),

- supply medium flow $Q_\mathrm{s,nom}$ and temperature T_s constraints.

5. Simulation results

Operation of the developed MPC controller is validated in a simulation scenario with fan coils and heat pump in the heating season. The considered model consists of 23 zones with overall 29 fan coils, a controlled hydraulic pump and a heat pump. Within the MATLAB Simulink environment is interaction between the HVAC controller and the zone controller established. Zone controller receives as inputs the predictions of the medium temperature and flow from the HVAC controller, whereas it delivers the zone temperature predictions and required thermal energies in the zones to the HVAC controller. Besides, variables that also form the inputs for the HVAC controller are forecasted environment temperatures, non-controllable loads and variable electricity prices predictions in the considered horizon. Since the zone controller inputs depend on the HVAC controller outputs and vice versa, sequence of execution should be defined. Specifically, the zone controller accepts the predictions of temperatures and flows from the last time instant of the HVAC controller optimisation, which is herein included to incorporate the existing transport delay in the medium propagation through the building.

Sequential linear programming is used to derive a solution to the nonlinear MPC problem from Section 4. In order to achieve a speed up of the optimisation process, the warm starting of the MP problem is achieved by employing the computed control input prediction from the previous time instant. Since the optimisation scheme is iterative, the maximum number of iterations is in the considered scenario upper bounded with 10 000.

Fig. 1 depicts the pricing of electrical energy (possibly received from the microgrid-level MPC) and ambient air temperatures which are considered in the simulation.

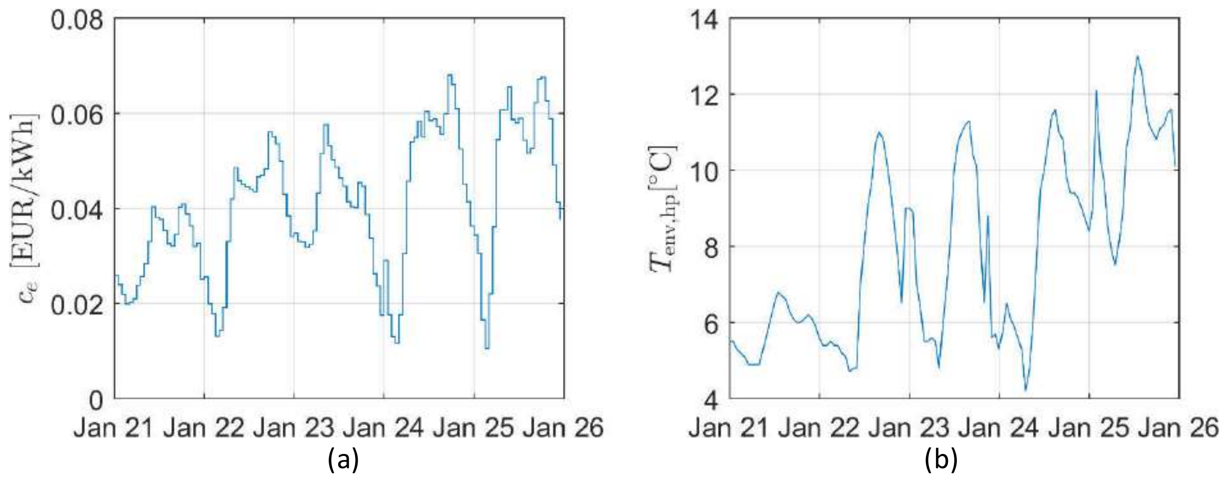


Fig. 1: a) Hourly electricity pricing and b) air ambient temperatures from January 21 to January 25, 2018

HVAC MPC controller inputs are depicted in Fig. 2, whereas on Subfigure 2-a is depicted the medium flow reference for the hydraulic pump and on Subfigure 2-b is depicted the commanded temperature of the supply medium. Imposed limits on the control variables are depicted with dashed lines. In the considered scenario the medium flow varies between the minimum and maximum values and the supply temperature has smaller variations. It is easily observed that peaks in the supplied medium flow coincide with the peaks of thermal energy demand in the zones, which is accompanied with the noticeable increase of the supply temperature. For example, if the price for the electric energy is low enough, the controller prefers higher rates of flow instead of the higher medium temperatures, since the latter one implies smaller thermal energy losses in the pipework.

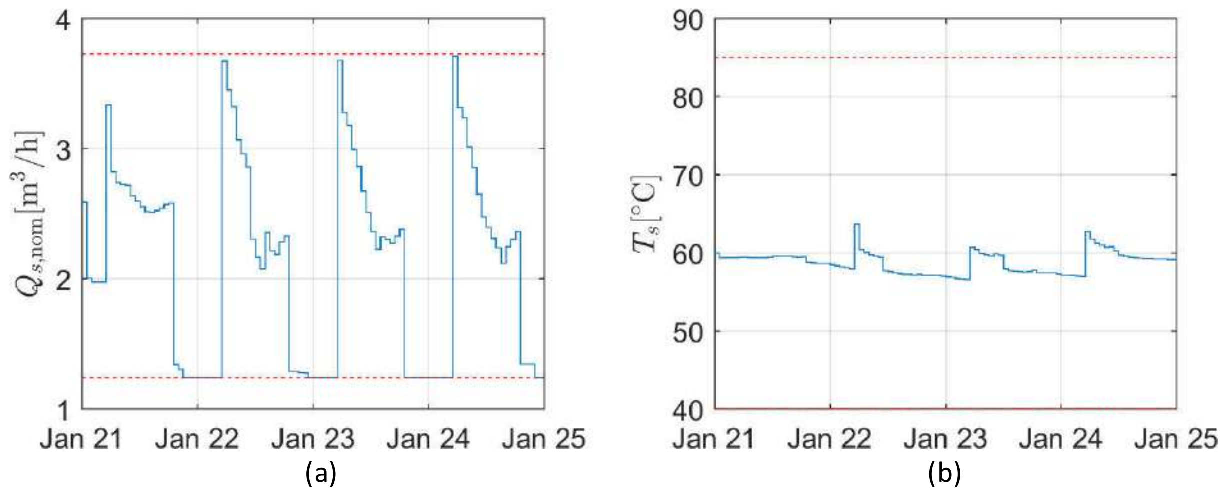
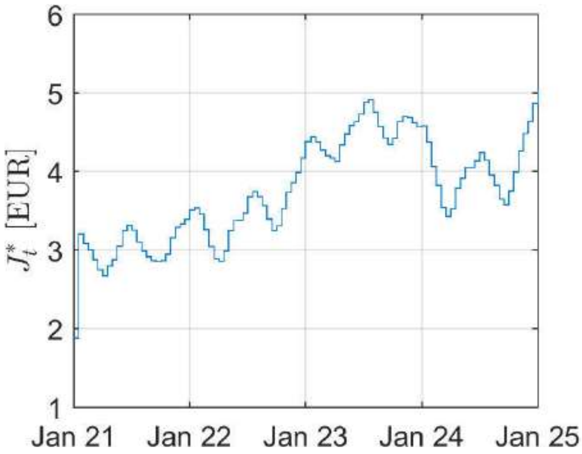
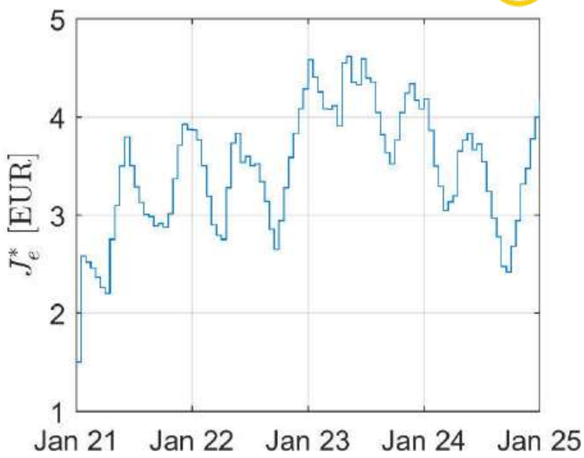


Fig. 2: a) Nominal medium flow and b) temperatures commanded by the HVAC controller

Cost of consumed thermal and electrical energy by the heating system is depicted in Fig. 3.



(a)



(b)

Fig. 3: Cost of a) thermal energy and b) electrical energy on the full prediction horizon obtained by solving the HVAC MPC problem

In Figs. 4-6 are displayed consumed and attainable thermal energies in each of the zone. Requirement that the attainable thermal energy always surpasses the energy required by the MPC zone controller is respected throughout the simulation.

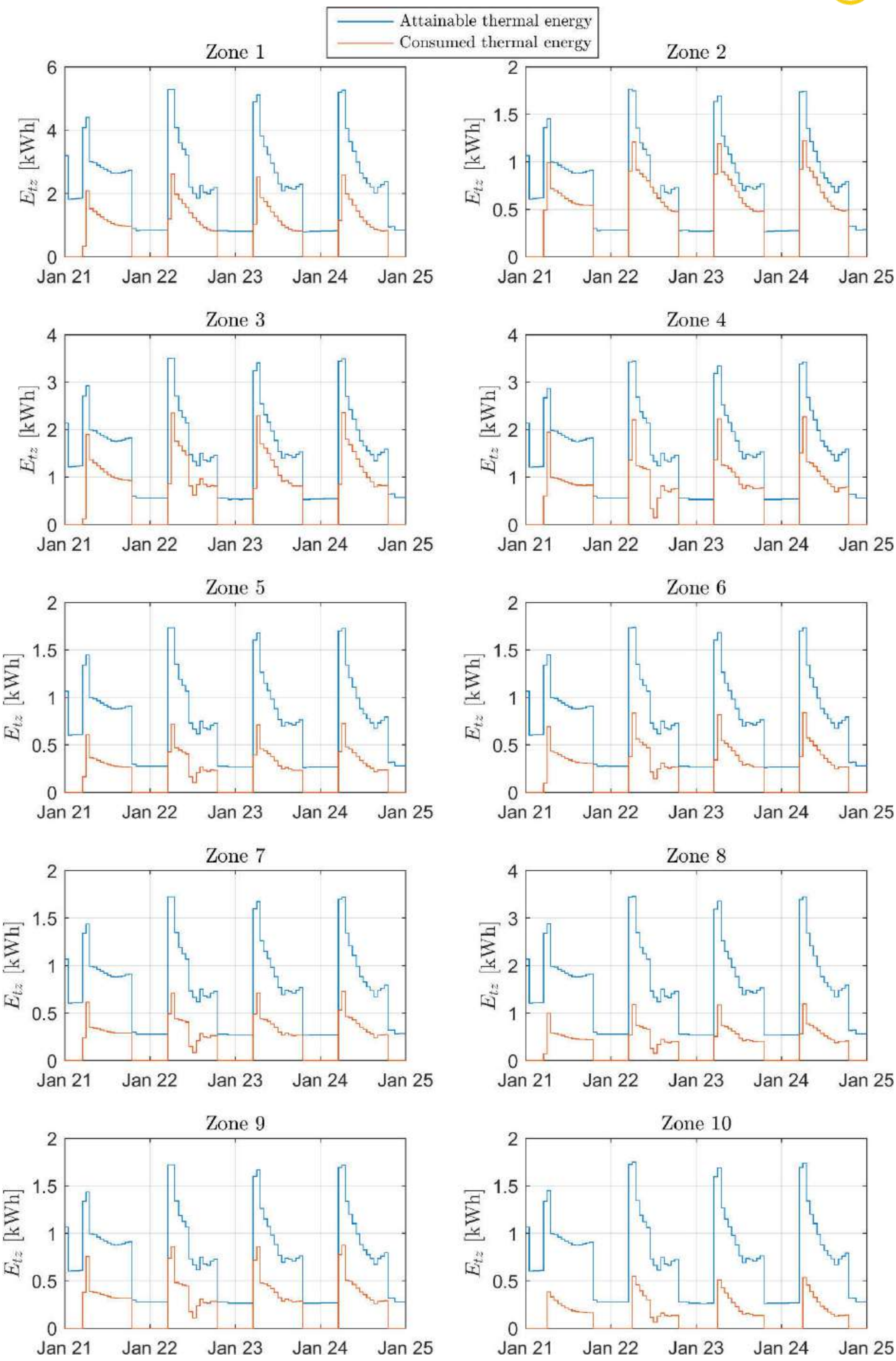


Fig. 4: Consumed (red) and attainable (blue) thermal energy in the zones

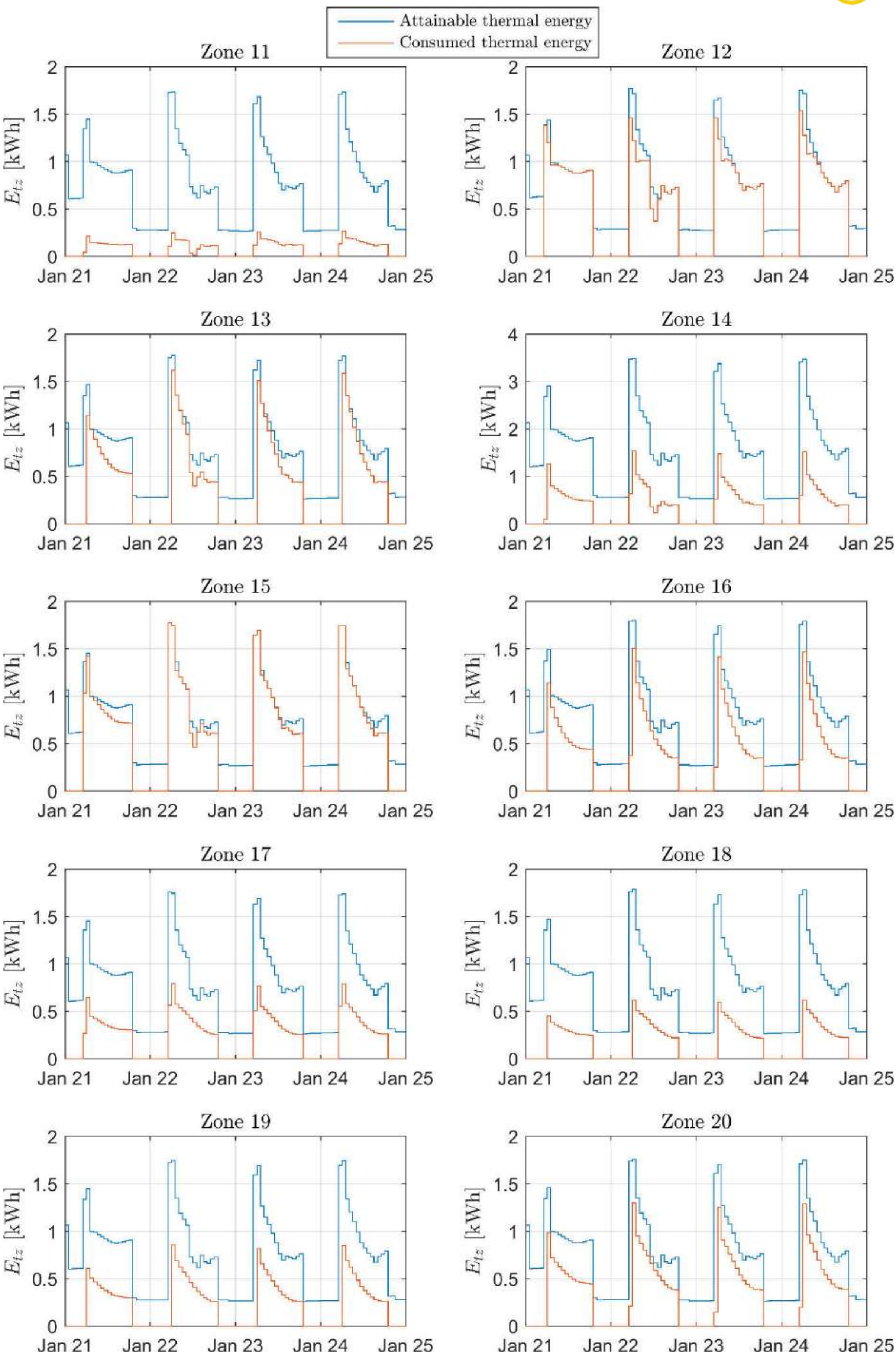


Fig. 5: Consumed (red) and attainable (blue) thermal energy in the zones

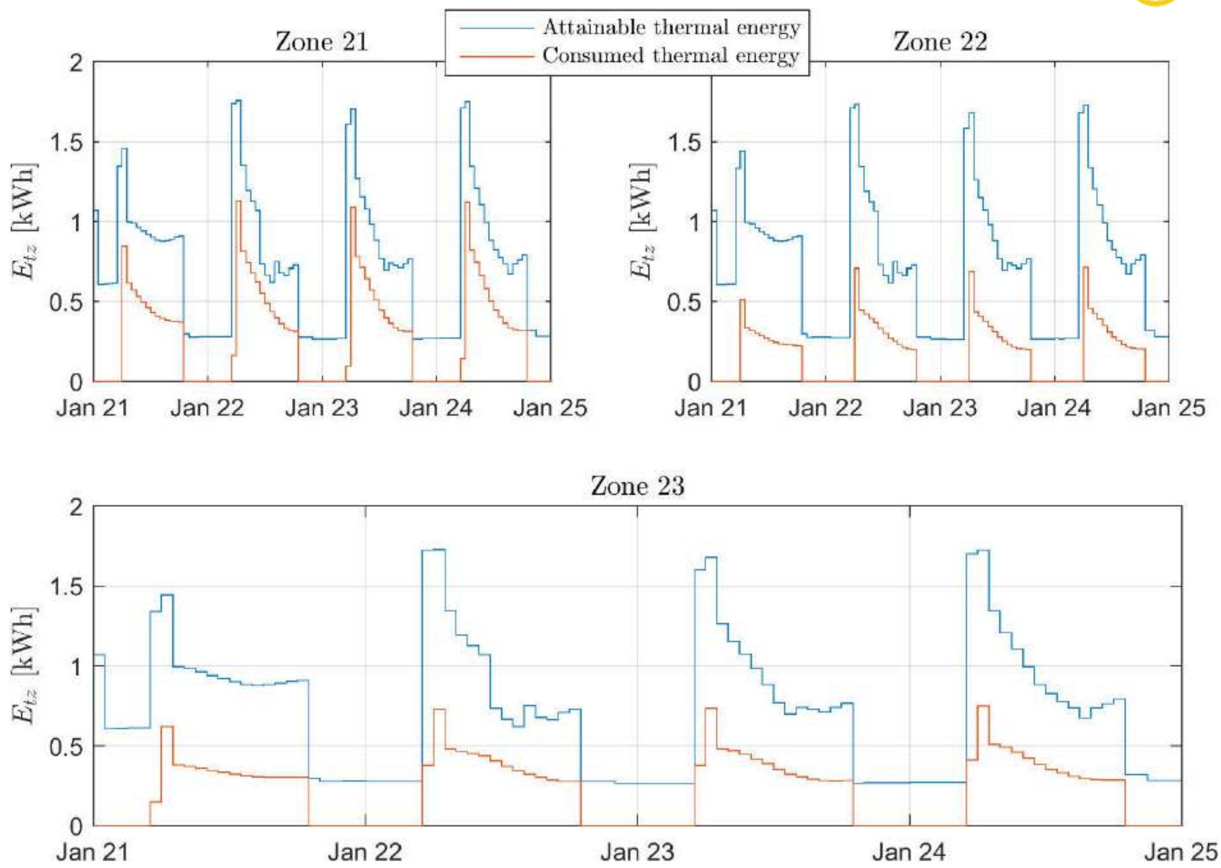


Fig. 6: Consumed (red) and attainable (blue) thermal energy in the zones

6. HVAC MPC module interface

Table 1: Input-output variables list of the heating substation/heat pump control module.

	Source/destination submodule	Variable
Inputs	*Zone-level submodules*	- temperature predictions in the zones T_z
		- predicted required thermal energy input in the zones $E_\mathrm{t,z}$
		Parameters of the heating/cooling zone elements model (for maximum attainable heating energy and electricity consumption): - fan coils: $a_{\mathrm{fc},v}$, $b_{\mathrm{fc},v}$, $l_{\mathrm{fc},v}$, $\bar{E}_{\mathrm{e,fc},j}$, $E_{t,\mathrm{fc},0}$ (please refer to Section 3.5) - radiators: $a, b, n_\mathrm{r}, m_\mathrm{r}, c_\mathrm{r}, U_0$ (please refer to Section 3.6) - underfloor heating: $m_\mathrm{fh}, c_\mathrm{fh}, U_0$ (please refer to Section 3.6)

		- measured medium outlet temperatures on the radiator/underfloor heating/cooling units $T_{w,\cdot}^{out}$ (please refer to Section 3.6)
	Central HVAC system level submodules	- predicted non-controllable thermal energy loads $E_{\mathrm{t,nc}}$
		Hydraulic model of the plant: - parameters of the Q-p characteristics, C_{Q-p} (please refer to Section 3.2) - parameters of the medium flow model (nominal conditions) $C_{q,j}, j \in \mathcal{J}$ (please refer to Section 3.1)
		- parameters of the heat pump coefficient of performance COP, γ_{p} (please refer to Section 3.4)
		- parameters of the temperature model at the zone element inlets $C_{T,j}, j \in \mathcal{J}$ (please refer to Section 3.3)
		- parameters of the heat losses model in the pipework (defined with $C_{T,i}$ and $C_{q,i}$, please refer to Section 3.3)
		- parameters of the hydraulic pump model electricity consumption V_{q}, p_{end} (please refer to Section 3.2)
	Microgrid level submodules	- electricity cost function and constraints on the prediction horizon $J_M^*(E_e)$
	Other	- air ambient temperature T_{env} (please refer to Section 3.4), the medium heat capacity c_w - cost of heating energy c_{t}
Outputs	*Zone level submodules*	- heating cost and constraints for different zones on the prediction horizon $J_H^*(E_{t,z})$
	Zone level submodules /Central HVAC system level submodules	- predicted temperatures and flows of the supplied medium over the prediction horizon T_{s}, $Q_{\mathrm{s,nom}}$ or Q_{s}
	Microgrid level submodules	- predicted electrical energy consumption E_{e} of the heating/cooling system on the prediction horizon
	District heating system operator (optional)	- predicted thermal energy consumption E_t of the heating/cooling system on the prediction horizon

7. Conclusion

The generalised mathematical description and control problem related to the optimal conditioning of the heating/cooling medium is discussed herein. It serves as a basis for the design of the HVAC control module for the heating substation and the heating pump, by considering different configurations of the employed heating/cooling units. Specifically, three distinct heating/cooling units are considered in this study: (i) fan coils, (ii) radiators and (iii) underfloor heating/cooling.

Operation of the proposed control scheme is validated first on a medium-scaled simulation model that includes the zone control MPC layer [4] and the HVAC MPC control layer, that are designed for heating of a building consisting of 23 zones with overall 29 fan coils. Obtained simulation results show that by the module issued supplied medium flow $Q_{\mathrm{s,nom}}$ and medium temperature T_{s} ensure the exact required amount of thermal energy in the zones, without any excess which would result with the higher losses in the system and consequently with the higher operational costs.

Bibliography

[1] Maciejowski, J. M., Predictive control: with constraints, Pearson education, 2002.

[2] M. Vašak et al., Deliverable D4.1.1 – Building-side EMS concept and information exchange interfaces definition, University of Zagreb, Faculty of Electrical Engineering and Computing, 2017.

[3] M. Komatina et al., Deliverable D4.5.3 – Final open software module for central heating/cooling system management – Prediction and estimation module, University of Belgrade, Faculty of Mechanical Engineering, 2019.

[4] M. Vašak et al., Deliverable D4.5.3 – Final open software module for zones consumption management, University of Zagreb, Faculty of Electrical Engineering and Computing, 2019.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Interfacing submodules for central HVAC system level management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Belgrade Faculty of Mechanical Engineering (UNIBGFME)
Authors (Partners)	Nebojša Manić (UNIBGFME)
Contact person	Mario Vašak (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an input/output data overview of the module that interfaces flow commands to the circulation pump actuating element on the central HVAC system level. More details on the module logic are provided in the annexed document.
Keyword List	Circulation Pump, Pressure, Flow

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.3.1	Mario Vašak (UNIZGFER)
0.2	15 January 2019	Updated version	Mario Vašak (UNIZGFER)
0.3	17 June 2019	Updated version	Mario Vašak (UNIZGFER), Nebojša Manić (UNIBGFME)
1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of modules.

The modules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on central HVAC system level interfacing module.

It is presented via an interfacing table that explains what data are used by the module as inputs and what are the final output data. The algorithm behind is in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following interfacing module is designed, commissioned and tested on the central HVAC system level:

HVAC.I.1 – module for interfacing flow commands to circulation pump (tested in UNIZGFER pilot building within 3Smart).

In the following chapter the mentioned module is presented with its interface tables showing which data it uses as inputs and which data it provides as outputs to be at the disposal to other submodules or to be used for the circulation pump. Detailed explanations of algorithms behind the module are provided in the previously delivered 3Smart document D4.3.1 (related to interfacing on the central HVAC system level). For completeness, D4.3.1 interfacing part is further improved based on feedback from pilot operation and annexed to this document (Annex 1).

Source and sink for the data used by the module is a properly structured 3Smart database. Its structure in the part concerned by the central HVAC system level interfacing module is provided in Annex 2.

2 HVAC.I.1 module

HVAC.I.1 module is used for interfacing flow commands to circulation pump. Within 3Smart it is tested in UNIZGFER pilot buildings.

The module interface is presented in Table 1.

Table 1: Input-output variables list of the heating substation/heat pump control module.

	Source/destination submodule	Variable
Inputs	*Central HVAC system level submodules*	– Flow measurements Q_s [m^3 h^{-1}](historical data, minutely sampled)
	HVAC.MPC module	– Flow set point Q_s^* [m^3 h^{-1}] (set by HVAC.MPC module)
Outputs	*HVAC.MPC module*	– Pump head (differential pressure) on the inlet and the outlet of the pump $p_p = f(Q_s)$ (compute pressure such that is followed by set point)

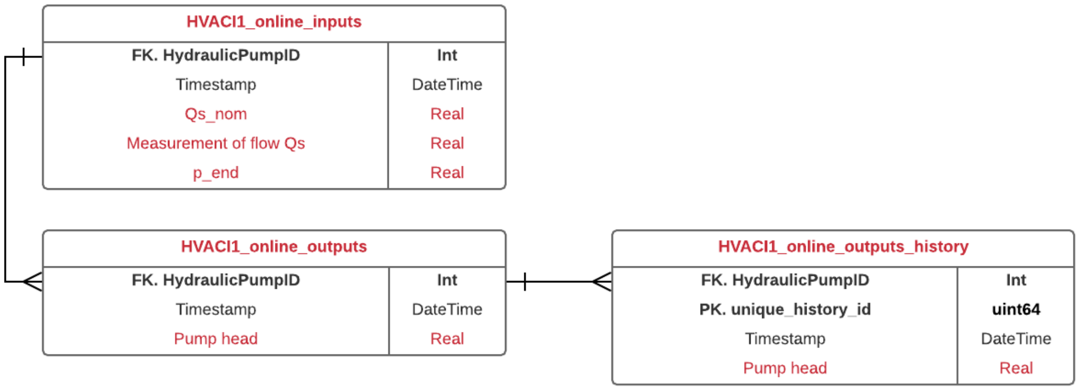
Bibliography

[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 -- Open software module for the central HVAC system level consumption management – Interfacing module

It is provided in a separate document.

Annex 2 -- 3Smart database organization for open software module for the central HVAC system level consumption management – Interfacing module



Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 CENTRAL HVAC SYSTEM INTERFACING

Open software module for central heating/cooling system management – Interface module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	UNIBGFME, UNIZGFER
Authors (Partners)	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME)
Contact person	Mirko Komatina (UNIBGFME)
Abstract (for dissemination)	This annex explains the logic of the central HVAC system interfacing module.
Keyword List	3Smart, open software module, central heating/cooling management, VFD circulation pump

Revision history

Revision	Date	Description	Author (Organization)
1.0	29 December 2017	Final draft of D4.3.1	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME).
1.1	30 March 2018	Taken over from D4.3.1 deliverable and updated	Mirko Komatina (UNIBGFME), Mario Vašak (UNIZGFER), Tatjana Lazović (UNIBGFME), Vladimir Jovanović (UNIBGFME), Nedžad Rudonja (UNIBGFME), Dimitrije Manić (UNIBGFME), Nebojsa Manić (UNIBGFME), Nikola Hure (UNIZGFER), MIhailo Milanović (UNIBGFME).
1.2	15 June 2019	Update	Nebojša Manić (UNIBGFME), Mario Vašak (UNIZGFER)
2.0	30 June 2019	Final quality-cjecked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of contents

Executive summary

This document focuses on interface module between central heating/cooling system management and field equipment on the central HVAC level.

Variable speed circulation pump module was developed to secure energy savings in central heating systems of the pilot buildings. Circulation pump was considered for two different cases (operation modes): Case1 - with fan coils (FCs) as zone heating/cooling devices and Case 2 - with radiators or floor radiant panels as zone heating/cooling devices. Mathematical equations for these cases are provided.

1 Interface module for central hydraulic pump (HVAC.I.1)

1.1 Introduction

Module for interfacing the computed commands towards the existing regulation systems in central HVAC units refers to the realization of the default temperature and flow at the exit from the central system to the building, where only temperature is regulated and no interference is required between the modules. This is due to directly forwarding the subordinate control circuit of the exit temperature and flow from the central HVAC system, which define module for setting the required pressure difference on the pump. This module was developed by UNIBGFME and it is presented in this document and referred to D4.1.1 and D4.3.1. Description of the parameters which the pump needs to manage based on model predictive control (MPC) output are explicitly stated.

1.2 Module interface

Theoretical background for this interface module is explained in detail in D4.3.1 for prediction and estimation modules and D4.1.1 and will not be described here again. Regarding to literature review also presented in D4.3.1, variable speed circulation pump should consider different hydraulic pump control modes to reduce energy consumption in the system and interfacing the computed commands towards the existing regulation systems. Due to the hydraulic pump operational mode the behavior of uncontrolled and controlled pumps will be shown.

The operation of uncontrolled pumps

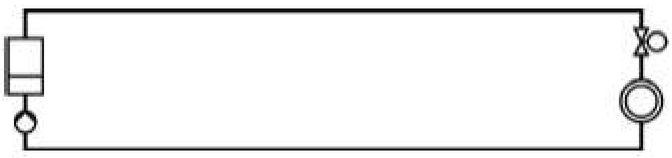


Figure 1.1: The operation of uncontrolled pumps scheme

If the valve closes, the resistance increases and the volumetric flow decreases. Hence, the plant characteristic becomes steeper. Due to the higher resistance in the piping network the pump needs to provide a higher pressure.
With uncontrolled pumps the speed n remains constant and the operating point follows the pump characteristic to the left.
The example shown below demonstrates the shifting of the operating point at part load 50 % and as a result, the related changes in the energy consumption of the pump.

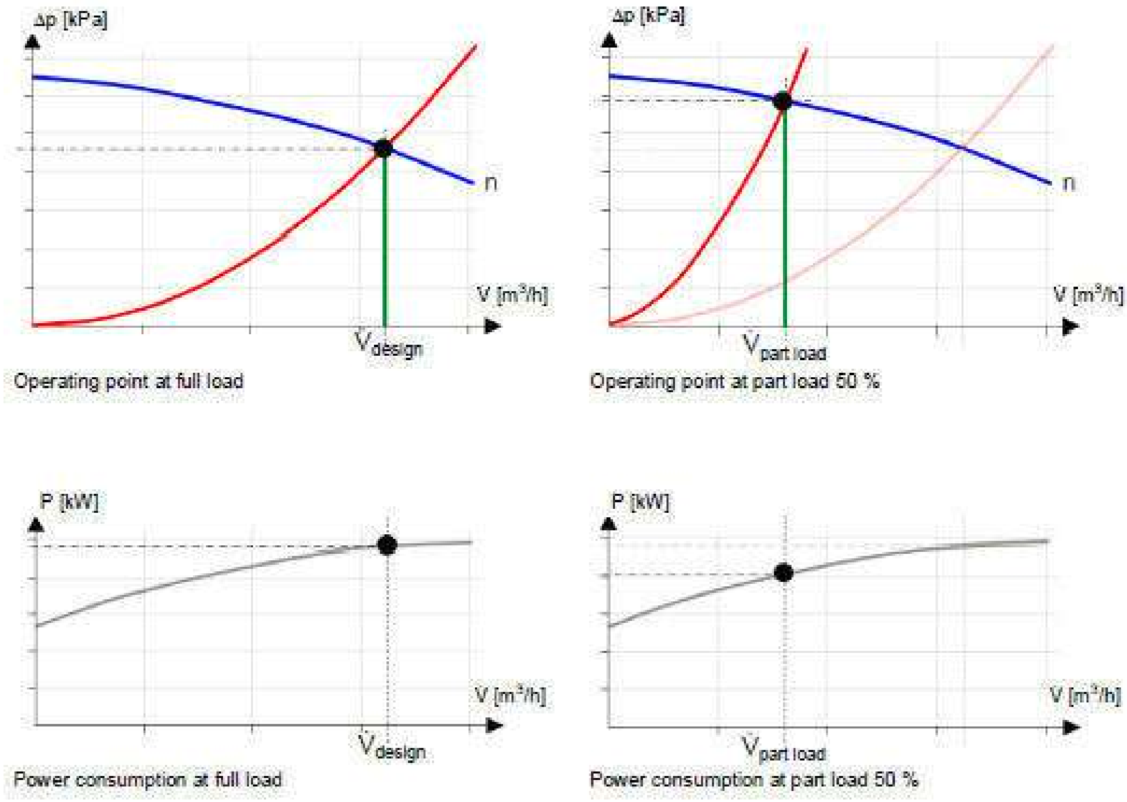


Figure 1.2: Operating point and power consumption at full and part load of uncontrolled pumps

The operation of controlled pumps can be carried out with:

1. Constant pump pressure

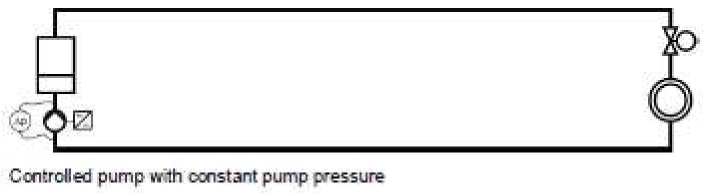


Figure 1.3: Controlled pump with constant pump pressure scheme

At part load the pressure difference across the pump is kept constant. This can be controlled either electronically in the pump itself or with a pressure dependent control and a variable speed drive at the pump. The operating point follows the line of constant pressure horizontally to the left.

The example shown below demonstrates the shifting of the operating point at part load 50 % and as a result, the altered energy consumption of the pump.

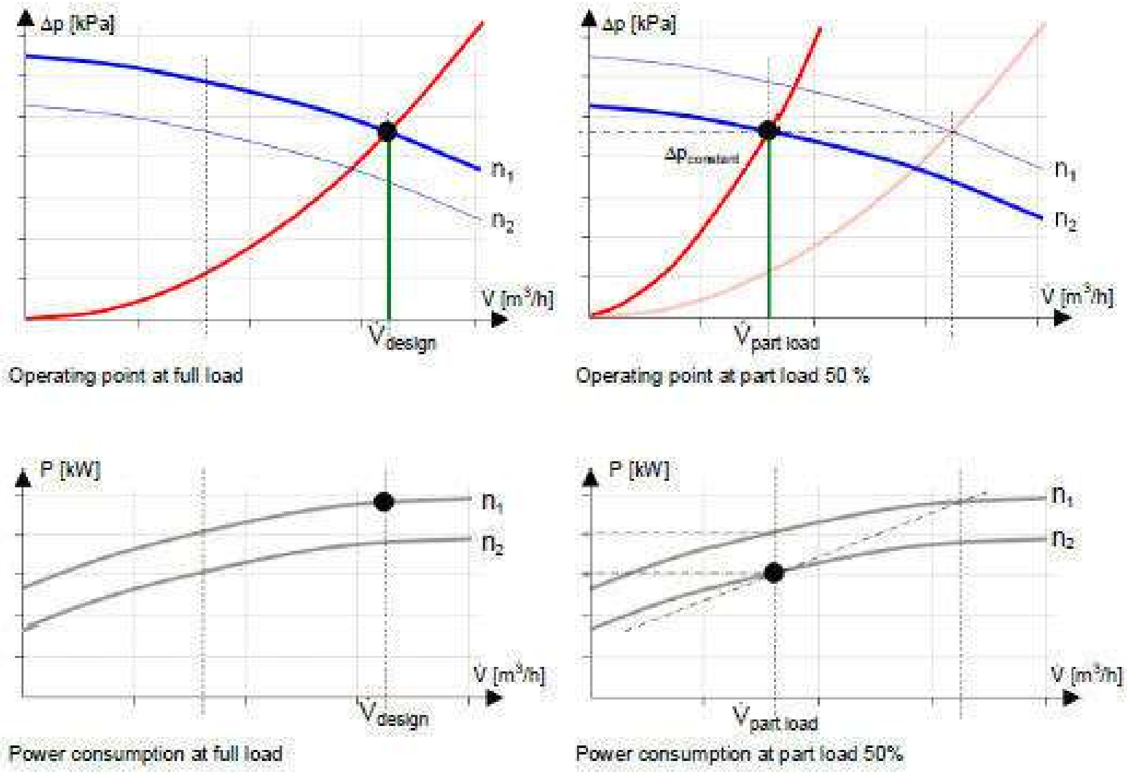


Figure 1.4: Operating point and power consumption at full and part load of constant pressure controlled pumps

2. Constant pressure difference across the end of the plant

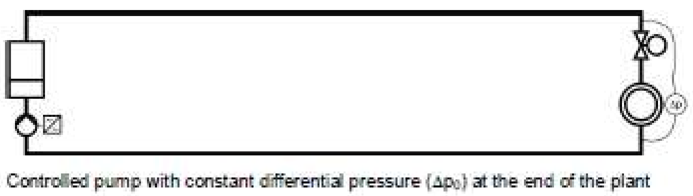


Figure 1.5: Controlled pump with constant differential pressure scheme

The pressure difference Δp_0 between central plant inlet and outlet is maintained at constant value. There are two different ways to achieve this:

- a measuring point at the plant outlet, connected to a pressure controller and a variable frequency drive (VFD) at the pump
- an electronic control in the pump itself ("Δp variable" control)

The operating point follows the control slope that runs towards Δp_0 near $\dot{V} = 0\ m^3/h$

The example shown below demonstrates the shifting of the operating point at part load 50% and as a result, the related changes in the energy consumption of the pump.

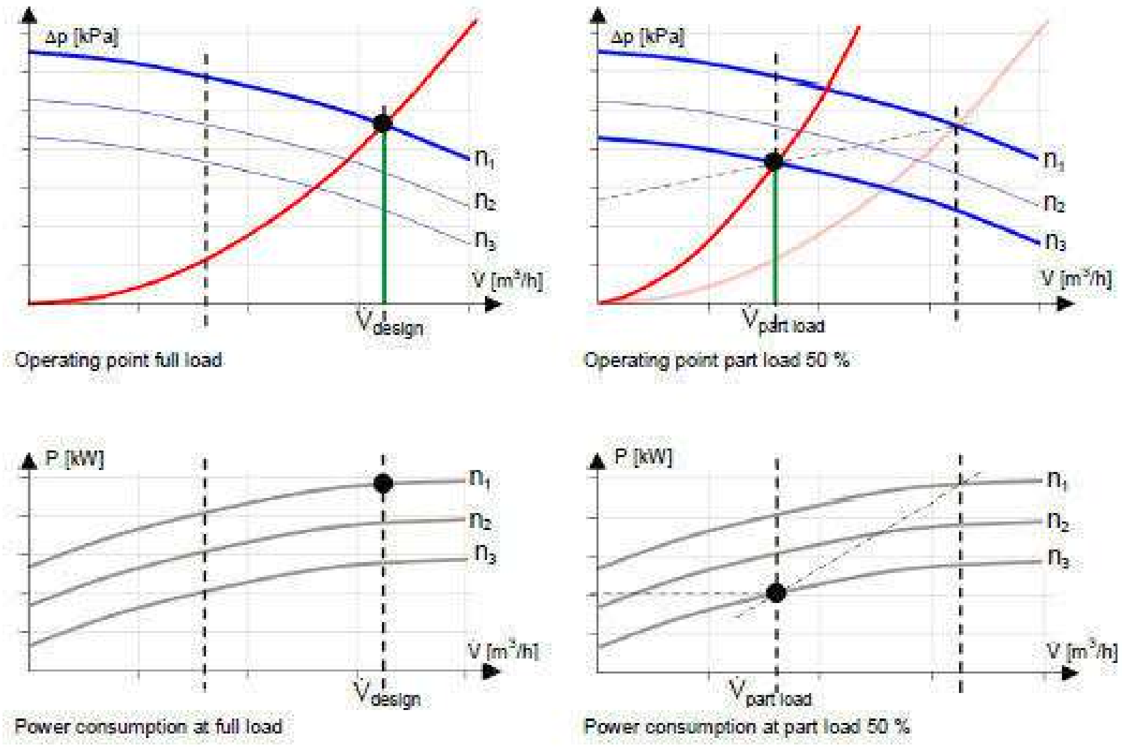


Figure 1.6: Operating point and power consumption at full and part load of constant differential pressure controlled pumps

The plant characteristic is steeper at part load (50 %). Due to the reduced volumetric flow the resistance in the plant is reduced as well. The controlling across the end of a plant ensures that the necessary differential pressure there is still maintained.
With a controlled pump with the measuring point at the end, the energy consumption of the pump is even further reduced (D4.3.1, prediction and estimation modules).

Energy savings with controlled pumps

Plants with variable volume flows, provide significant energy savings. The selection of the control system depends on the situation on site (distances, investments, etc.). As shown in the chart below, variable speed pumps consume less power. Thus, energy and costs can be saved. A pump with constant differential pressure across the end of the plant is more efficient than a controlled pump with a constant pump pressure.
The chart below shows the saving capacity based on a data sheet of a pump.

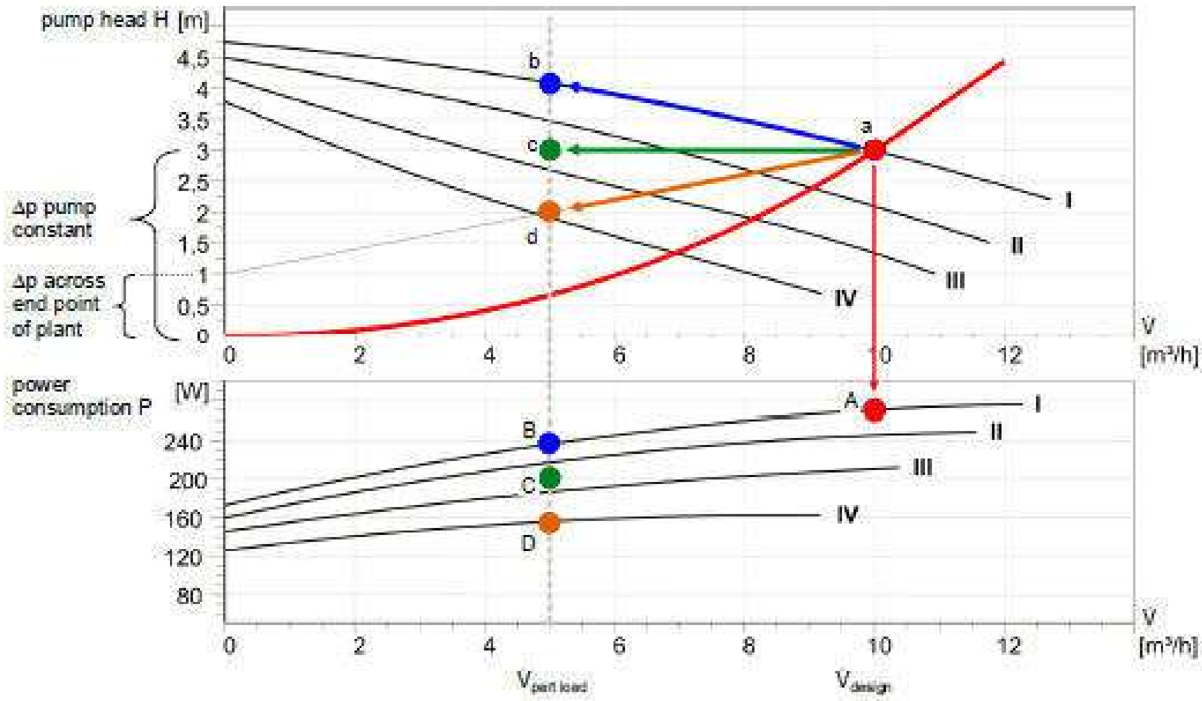


Figure 1.7: Operating point and power consumption for controlled pumps

Operating points and power consumption in comparison

a: operating point, design

b: operating point, part load, uncontrolled

c: operating point, part load, controlled Δp_{pump}

d: operating point, part load, controlled Δp_{end}

A: power consumption, design

B: power consumption, part load, uncontrolled

C: power consumption, part load, controlled Δp_{pump}

D: power consumption, part load, controlled Δp_{end}

According to prediction and estimation module defined in D4.3.1, constant pressure difference across the plant is selected as a pump control strategy. Detailed explanation for the selected control mode is presented in the following figure (1.8) and summary of the necessary equations are provided.

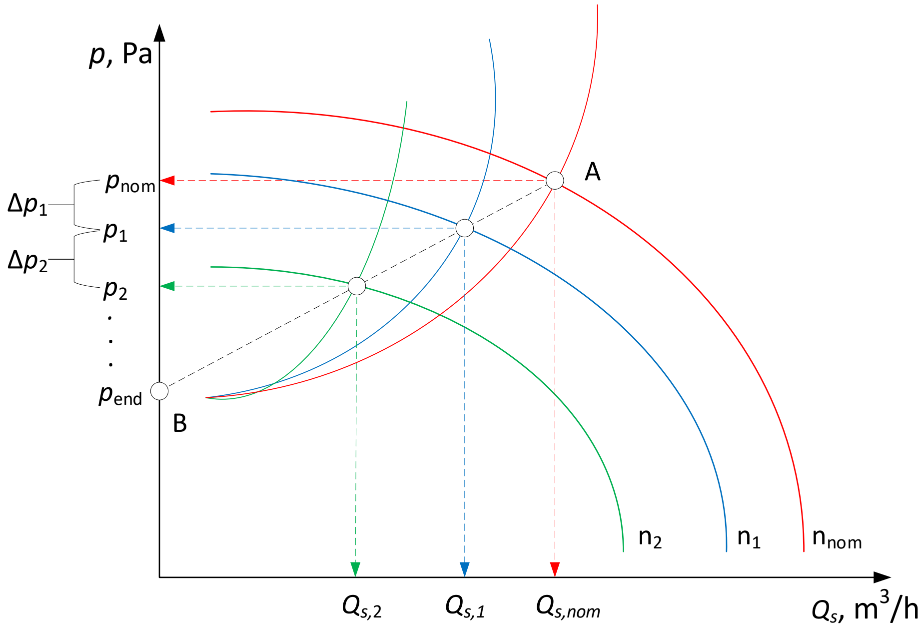


Figure 1.8: Pump diagram

Where following annotations are used:

$Q_{s,nom}$, [m^3/h] – volume flow at nominal working regime

$p_{p,nom}$, [m] – differential pressure at nominal working regime.

n_{nom} , rpm – pump rpm at nominal working regime.

$p_1, p_2, \dots$,[m] – differential pressure on the pump in the operational mode

$Q_{s1}, Q_{s2}, \dots$, [m^3/h] – reduced volume flow through the system caused by MPC output (change temperature)

p_{end} ,[m] – differential pressure by theoretical case measured on the "last" element on the system.

In Figure 1.8, point A represents a pump working point at nominal power with maximum flow through radiators/floor heating elements. Point B considers theoretical pump operating point when all radiators/floor heating elements are closed and the flow through the pump is equal 0. The operational modes of the pump due to flow reduction by exclusion of specific radiator or floor heating element from the system are presented by points between. According to MPC output (according to D4.3.1b – MPC part) setting the differential pressure on the variable speed pump outlet should be defined by the combining the equations for the head if the pump in this control mode and with consideration of the pressure which are provided in D4.3.1 prediction and estimation modules.

To define nonlinear variable speed pump model for this case the equation for the line through points A and B is determined by

$$p_p = p_{end} + (p_{p,nom} - p_{end}) \cdot \frac{Q_s}{Q_{s,nom}} \quad .....(1.1)$$

Equation 1.1 defines the required pressure difference on the VFD circulation pump based on the exit temperature and flow from the central HVAC system.

Possible cases:

1. All radiators are open:

$$Q_s = Q_{s,nom} \rightarrow p = p_{nom} \quad .....(1.2)$$

2. Radiator no.1 is closed:

$$Q_s = Q_{s,1} \rightarrow p_1 = p_{end} + (p_{nom} - p_{end}) \cdot \frac{Q_{s,1}}{Q_{s,nom}} \quad .....(1.3)$$

3. Radiators no.1 and no.2 are closed:

$$Q_s = Q_{s,2} \rightarrow p_2 = p_{end} + (p_{nom} - p_{end}) \cdot \frac{Q_{s,2}}{Q_{s,nom}} \quad .....(1.4)$$

4. All radiators are closed (theoretical case):

$$Q_s = 0 \rightarrow p = p_{end} \quad .....(1.5)$$

Where following annotations are used:

Q_s , [m3/h] – required pump flow provided by MPC output

$Q_{s,nom}$, [m3/h] – nominal pump flow

$p_{p,nom}$, [Pa] – nominal differential pressure measured on pump inlet/outlet

p_{end}, [Pa] – differential pressure by theoretical case measured on the “last” element on the system.

4.3.4 HVAC.I.1 model parameters

HVAC.I.1 interface model according to prediction and estimation model is also considered in two configurations:

- Case a - static hydraulic situation with fan coils
- Case b - configuration with radiators or floor heating/cooling

For Case a:

INPUT		FORMAT
$Q_s\left[\mathrm{m}^3/\mathrm{h}\right]$	Measurement of flow	Historical data, minutely sampled
$Q_s^*\left[\mathrm{m}^3/\mathrm{h}\right]$	Flow Set point	Set by HVAC.MPC module

OUTPUT		FORMAT
$p_p = f\left(Q_s\right)$	Differential pressure on the inlet and the outlet of the pump	Compute pressure such that Q_s is followed by set point Q_s^*

For Case b:

INPUT		FORMAT
$Q_s\left[\mathrm{m}^3/\mathrm{h}\right]$	Measurement of flow	Historical data, minutely sampled

OUTPUT		FORMAT
$p_p = f\left(Q_s, Q_{s,nom}, p_{p,nom}, p_{end}\right)$	Differential pressure on the inlet and the outlet of the pump	compute pressure such that (Q_s, p_p) is on the required line determined by $(0, p_{end})$ and ($Q_{s,nom}, p_{p,nom}$)

Application of the VFD hydraulic pump module as a part of the overall central HVAC module design on particular test-sites should consider the following:

- For selected circulation pump from all test-sites should be provided pump characteristics efficiency vs flow ($\eta_p = f(Q_s)$);
- output for the hydraulic pump submodule will be parameters for pump electricity consumption for different operating regimes and for different heating/cooling element configurations.
- Additional demands for input variables which will be used in hydraulic pump submodule (i.e. parameters which should be measured on particular test-site) are pump flow and differential pressure on the pump ($Q_{s,nom}$; p_{nom}) at nominal regime. For radiators/ground floor heating systems, input variables are differential pressure on the last element or differential pressure on the pump when all the elements were closed ($p_{end}(Q_s = 0)$).

Bibliography

[1] Deliverable D4.1.1: Vašak et al., - Building-side EMS concept and information exchange interfaces definition, University of Zagreb, Faculty of Electrical Engineering and Computing, 2017.

[2] Deliverable D4.5.3: Komatina et al., - Final open software module for central heating/cooling system management – Prediction and estimation module, University of Belgrade, Faculty of Mechanical Engineering, 2019.

[3] Deliverable D4.5.3: Hure, Vašak – Final open software module for central heating/cooling system management – Model predictive control module, University of Zagreb, Faculty of Electrical Engineering and Computing, 2019.

[4] Hydraulics in building systems, Siemens Ltd, Building Technologies division, Switzerland, http://www.siemens.com/buildingtechnologies.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Estimation and prediction submodules for microgrid management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Debrecen (UNIDEBTTK), Energetika, ekologija, ekonomija d.o.o. (E3)
Authors (Partners)	Hrvoje Novak, Mario Vašak (UNIZGFER), Arpad Racz (UNIDEBTTK), Marko Baša (E3)
Contact person	Mario Vašak (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of estimation and prediction submodules on the level of a building microgrid for hierarchical management of building subsystems -- input/output interfaces of different modules are presented. The logic behind each of the modules is provided in the annexed document.
Keyword List	Battery System Model, State of Charge, Photovoltaic System Production, Electricity Consumption, Domestic Hot Water Consumption, Estimation, Prediction

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.2.1	Mario Vašak (UNIZGFER)
v0.2	15 January 2019	Updated document	Mario Vašak (UNIZGFER)
v0.3	17 June 2019	Updated document	Hrvoje Novak (UNIZGFER), Arpad Racz (UNIDEBTTK), Marko Baša (E3)
v1.0	30 June 2019	Updated document	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of submodules.

The submodules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on microgrid level prediction and estimation submodules.

Each submodule is presented via an interfacing table that explains what data are used by the submodules as inputs and what are the final output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following estimation and prediction submodules are designed, commissioned and tested on the microgrid level:

M.PE.1 – submodule for estimation of a battery system model that relates charging and discharging energies of the system to state of charge of the battery (tested in UNIZGFER, HEP, STREM retirement and care centre and EPHZHB pilot buildings within 3Smart);

M.PE.2 -- submodule for estimation of battery state of charge (tested in UNIZGFER, HEP, STREM retirement and care centre and EPHZHB pilot buildings within 3Smart);

M.PE.3 -- submodule for estimation of a prediction model for photovoltaic system production in off-line operation and for prediction of a photovoltaic system production in on-line operation (tested in UNIZGFER, HEP, IDRIJA, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

M.PE.4 – submodule for estimation of a prediction model for non-controllable microgrid-relevant energy consumption in off-line operation and for prediction of non-controllable microgrid-relevant energy consumption in on-line operation (tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings within 3Smart);

M.PE.5 – submodule for estimation of a domestic hot water tank model that relates charging and discharging heat to the temperature of the water in the tank (tested in IDRIJA school and sports centre pilot buildings within 3Smart);

M.PE.6 – submodule for estimation of heating/cooling medium buffer tank model that relates charging and discharging heat to the temperature of the medium in the tank (tested in EPHZHB and EON pilot buildings within 3Smart);

In the following chapters the mentioned submodules are presented with their interface tables showing which data they use as inputs and which data they provide as outputs to be at the disposal to other submodules. Detailed explanations of algorithms behind each of the submodules are provided in the previously delivered 3Smart document D4.2.1 (related to prediction and estimation) which is inherited here as Annex 1 and additionally improved based on feedback from pilots.

Source and sink for the data used by submodules is a properly structured 3Smart database. Its structure in the part concerned by the microgrid level prediction and estimation submodules is provided in Annex 2.

2 M.PE.1 submodule

Z.PE.1 submodule is used for estimation of a battery system model that relates charging and discharging energies of the system to state of charge of the battery. Within 3Smart it is tested in UNIZGFER, HEP, STREM retirement and care centre and EPHZHB pilot buildings.

The submodule interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Inputs of the M.PE.1 submodule

Variable name	Variable annotation	Variable description
Historical profile of measured battery currents	I_{batt}	Historical profile of battery average currents measured by the battery pack controller
Historical profile of measured battery and battery cells voltages	U_{batt}	Historical profile of battery and battery cells voltages measured by the battery pack controller
Historical profile of measured battery cells temperatures	T_{batt}	Historical profile of battery cells temperatures measured by the battery pack controller
Historical profile of measured power for converter AC side (before the isolation transformer)	P_{convAC}	Historical profile of power for converter AC side measured by the power converter

Table 2.2: Outputs of the M.PE.1 submodule

Variable name	Variable annotation	Variable description
Battery model	θ_{batt}	Needed for the M.PE.2, MPC module on the microgrid level

3 M.PE.2 submodule

Z.PE.2 submodule is used for estimation of battery state of charge. Within 3Smart it is tested in UNIZGFER, HEP, STREM retirement and care centre and EPHZHB pilot buildings.

The submodule interface is defined in Table 3.1 and Table 3.2.

Table 3.1: Inputs of the M.PE.2 submodule

Variable name	Variable annotation	Variable description
Historical profile of measured battery currents	I_{batt}	Historical profile of battery average currents measured by the battery pack controller
Historical profile of measured battery and battery cells voltages	U_{batt}	Historical profile of battery and battery cells voltages measured by the battery pack controller
Historical profile of SoC from battery pack	SoC	Historical profile of calculated SoC calculated by the battery pack controller

Table 3.2: Outputs of the M.PE.2 submodule

Variable name	Variable annotation	Variable description
Current state of charge estimate in energy	$\mathrm{SoC_E}$	Needed for the MPC module on the microgrid level

4 M.PE.3 submodule

M.PE.3 submodule is used for estimation of a prediction model for photovoltaic system production in off-line operation and for prediction of a photovoltaic system production in on-line operation. Within 3Smart it is tested in UNIZG-FER, HEP, IDRIJA, STREM retirement and care centre, EPHZHB and EON pilot buildings.

The submodule interface is defined in Table 4.1 and Table 4.2

Table 4.1: Required inputs for the photovoltaic array production prediction submodule.

Variable name	Variable annotation	Variable description
Photovoltaic array power meters data	E_{pv}	Photovoltaic array energy production measured directly on the array power meter
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I_{\mathrm{diff}}^{h}, I_{\mathrm{dir}}^{\mathrm{n}}$ Remaining pilot sites: $T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

Table 4.2: Outputs of the photovoltaic array production prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{PV}	Needed for on-line operation of the submodule
Predicted profile of the photovoltaic array energy production (for on-line operation of the submodule)	$(E_{\mathrm{pv}})_{\mathrm{N}}$	Needed for the MPC module on the microgrid level

5 M.PE.4 submodule

M.PE.4 is a submodule used for estimation of a prediction model for non-controllable microgrid-relevant energy consumption in off-line operation and for prediction of non-controllable microgrid-relevant energy consumption in on-line operation. Within 3Smart it is tested in UNIZGFER, HEP, IDRIJA school and sports centre buildings, STREM school, STREM retirement and care centre, EPHZHB and EON pilot buildings.

The module interface is provided with the following Table 5.1 and Table 5.2.

Table 5.1: Required inputs for the microgrid level non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Historical profile of the non-controllable energy consumption on the microgrid level	$E_{e,\mathrm{nc}}$ $E_{t,\mathrm{nc}}$	Non-controllable energy consumption on the microgrid level, electrical energy for all pilot sites except Idrija pilot sites where both electrical and thermal energy on the microgrid level are considered
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I^{h}_{\mathrm{diff}}, I^{\mathrm{n}}_{\mathrm{dir}}$ Remaining pilot sites: $T_{\mathrm{env}}, I^{h}_{\mathrm{glo}}, I^{\mathrm{t}}_{\mathrm{glo}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I^{\mathrm{n}}_{\mathrm{dir}})_{\mathrm{N}}, (I^{\mathrm{h}}_{\mathrm{diff}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

Table 5.2: Outputs of the non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{ug}	Needed for on-line operation of the submodule.
Predicted profile of the non-controllable electricity consumption, predicted profile of the non-controllable thermal load (for on-line operation of the submodule)	$(E_{e,nc})_{\mathrm{N}}$ $(E_{t,nc})_{\mathrm{N}}$	Needed for the MPC module on the microgrid level

6 M.PE.5 submodule

M.PE.5 is a submodule used for estimation of a domestic hot water tank model that relates charging and discharging heat to the temperature of the water in the tank. Within 3Smart it is tested in IDRIJA school and sports centre pilot buildings.

The submodule interface is defined in the following tables (Table 6.1, Table 6.2).

Table 6.1: Required inputs for the M.PE.5 submodule

Variable name	Variable annotation	Variable description
Electric heaters data	heater_data	Electric heaters technical data
Heat exchangers data	exchanger_data	Heat exchangers technical data
Tank temperatures	dhw_temperatures	Temperatures of medium in tank, at diferent positions
Environment temperature	temperature_env	Room air temperature
Environment temperature-history	temperature_env_hist	Room air temperature, historical data
Imported heat	dhw_heat_import	Heat measured by heat meter on supply pipe from district heating.
Imported heat-history	dhw_heat_import _hist	Heat measured by heat meter on supply pipe from district heating, historical data.
Imported electric energy	dhw_el_import	Electricity consumed by electric heaters
Imported electric energy – history	dhw_el_import _hist	Electricity consumed by electric heaters, historical data
Flow measurement on exiting pipe	flow_export	Flow measured on exiting pipe. Exported energy is calculated from flow and temepratures.
Flow measurement on exiting pipe - history	flow_export _hist	Flow measured on exiting pipe, historical data.

Table 6.2: Outputs of the M.PE.5 submodule

Variable name	Variable annotation	Variable description
DHW tank state	dhw_state_pred	Current and predicted state of energy stored in tank.

7 M.PE.6 submodule

M.PE.6 is a submodule for estimation of heating/cooling medium buffer tank model that relates charging and discharging heat to the temperature of the medium in the tank. Within 3Smart it is tested in EPHZHB and EON pilot buildings.

The module input and output data are provided within the following tables.

Table 7.1: Inputs of the M.PE.6 submodule

Variable name	Variable annotation	Variable description
Historical controllable load measured state	$T_{heatstorage}$	Historical profile of measured temperature of the buffer tank
Historical controllable load energy input	E_{in}	Historical profile of measured energy input into the buffer tank measured by electricity meters
Historical controllable load energy output	E_{out}	Historical profile of measured energy output from the buffer tank measured by calorimeters
Room temperature	T_{env}	Historical profile of measured room temperature (room where the tank is located)
Parameter of the COP model of the chiller (if relevant)	COP	Calculated parameters of the chiller from HVAC.PE.1

Table 7.2: Outputs of the M.PE.6 submodule

Variable name	Variable annotation	Variable description
Simplified model of the controllable load	θ_l	Needed for the MPC module on the microgrid level

Bibliography

[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 – Open software module for microgrid level consumption management – Estimation and prediction submodules

It is provided as a separate document.

Annex 2 -- 3Smart database organization for open software module for the microgrid level management – Estimation and prediction submodules

M.PE.3.

Input data database structure:

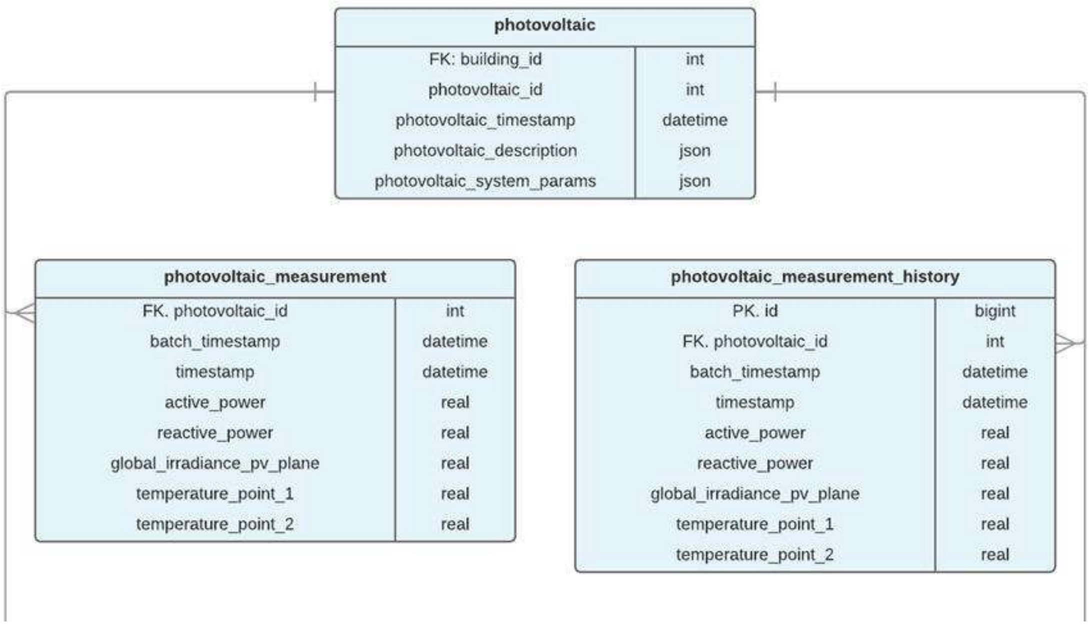


Figure 1. Current and historical photovoltaic array production data database structure.

Figure 2. Weather measurements data database structure.

weather_predictor	
PK. weather_predictor_id	int
weather_predictor_timestamp	datetime
weather_predictor_name	varchar(100)
weather_predictor_description	varchar(100)
weather_predictor_sample_time	int

weather_prediction	
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

weather_prediction_history	
PK. id	bigint
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

Figure 3. Weather forecast data database structure.

Output data database structure:

mgrid_pe3_outputs	
FK. microgrid_id	int
timestamp	datetime
pv_production_pred	json

mgrid_pe3_outputs_history	
FK. microgrid_id	int
PK. id	uint64
timestamp	datetime
pv_production_pred	json

Figure 4. Current and historical photovoltaic array production predictions data database tables.

M.PE.4.

Input data database structure:

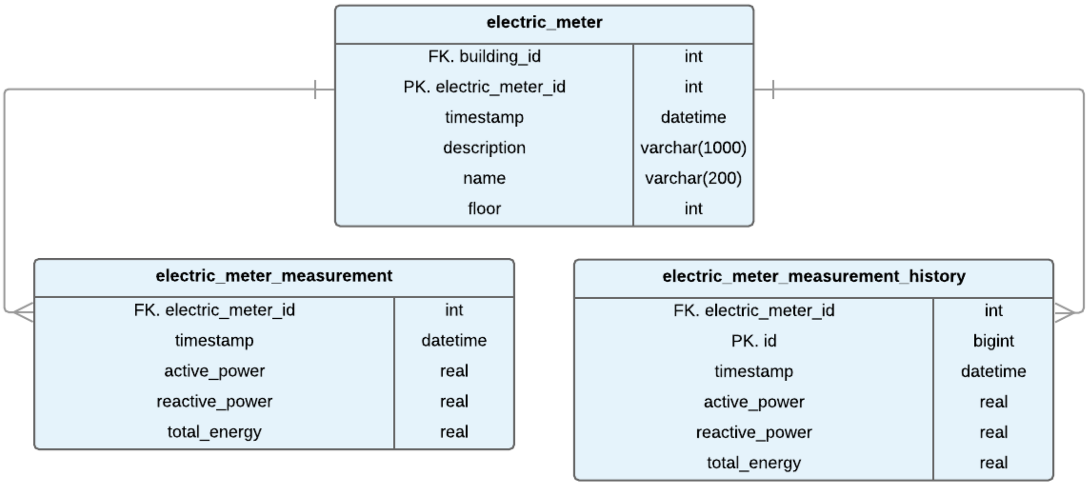


Figure 5. Current and historical microgrid level electrical non-controllable consumption data database structure (all pilots).

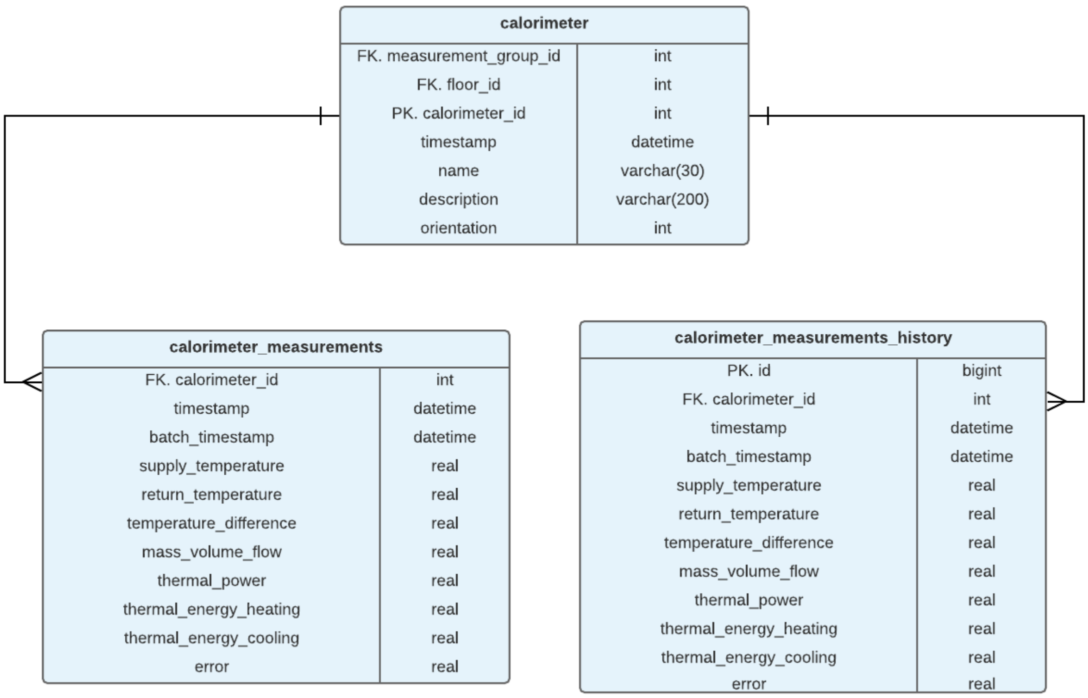


Figure 6. Current and historical microgrid level thermal non-controllable consumption data database structure (Idrija pilots).

Figure 7. Weather measurements data database structure.

weather_predictor	
PK. weather_predictor_id	int
weather_predictor_timestamp	datetime
weather_predictor_name	varchar(100)
weather_predictor_description	varchar(100)
weather_predictor_sample_time	int

weather_prediction	
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

weather_prediction_history	
PK. id	bigint
FK. weather_predictor_id	int
weather_prediction_timestamp	datetime
weather_prediction_start_timestamp	datetime
weather_prediction_temperature_at_2m	varchar(1000)
weather_prediction_dew_point_at_2m	varchar(1000)
weather_prediction_relative_humidity_at_2m	varchar(1000)
weather_prediction_mean_wind_speed_at_10m	varchar(1000)
weather_prediction_wind_direction_at_10m	varchar(1000)
weather_prediction_wind_gust_at_10m	varchar(1000)
weather_prediction_mean_wind_speed_at_bldg_top	varchar(1000)
weather_prediction_wind_direction_at_bldg_top	varchar(1000)
weather_prediction_mean_sea_level_pressure	varchar(1000)
weather_prediction_total_cloud_coverage	varchar(1000)
weather_prediction_high_cloud_coverage	varchar(1000)
weather_prediction_low_cloud_coverage	varchar(1000)
weather_prediction_mean_cloud_coverage	varchar(1000)
weather_prediction_total_precipitation	varchar(1000)
weather_prediction_total_snow	varchar(1000)
weather_prediction_direct_solar_irradiance	varchar(1000)
weather_prediction_diffuse_solar_irradiance	varchar(1000)
weather_prediction_total_solar_irradiance	varchar(1000)
weather_prediction_variance_of_the_2m_temperature	varchar(1000)
weather_prediction_variance_of_direct_solar_irradiance	varchar(1000)
weather_prediction_variance_of_diffuse_solar_irradiance	varchar(1000)
weather_prediction_variance_of_total_irradiance	varchar(1000)

Figure 8. Weather forecast data database structure.

Output data database structure:

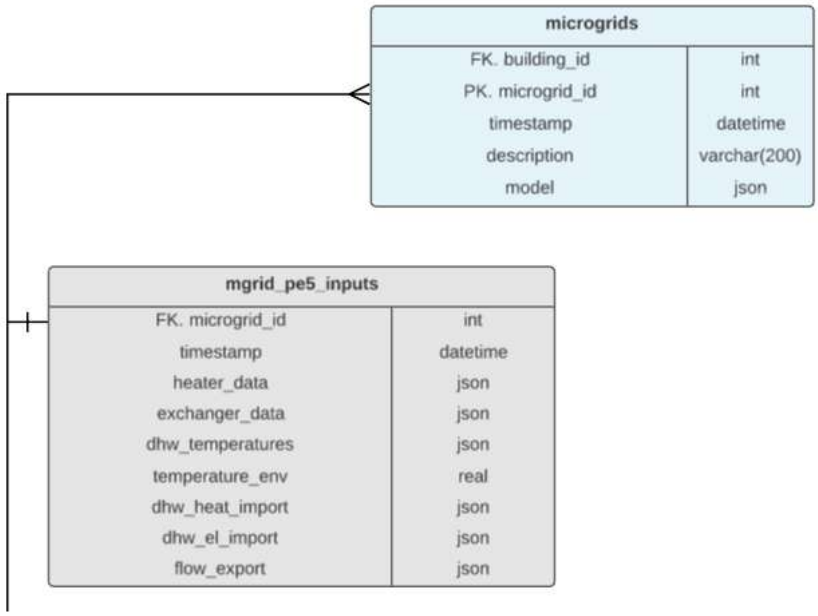
mgrid_pe4_outputs	
FK. microgrid_id	int
timestamp	datetime
nctrl_elec_consumption_pred	varchar(2000)
nctrl_ther_consumption_pred	varchar(2000)
nctrl_gas_consumption_pred	varchar(2000)

mgrid_pe4_outputs_history	
FK. microgrid_id	int
PK. id	uint64
timestamp	datetime
nctrl_elec_consumption_pred	varchar(2000)
nctrl_ther_consumption_pred	varchar(2000)
nctrl_gas_consumption_pred	varchar(2000)

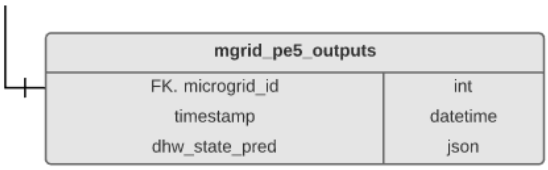
Figure 9. Current and historical microgrid level non-controllable consumption predictions data database tables.

M.PE.5.

Input data database structure:



Output data database structure:



Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX 1 TO D4.5.3 PREDICTION AND ESTIMATION ON THE MICROGRID LEVEL

Open software module for energy flows management in building's microgrid – Prediction and estimation modules

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Restricted
Deliverable participants	UNIDEBTTK, UNIZGFER, E3
Authors (Partners)	Árpád Rácz, Sándor Misák, István Szabó, Gergő Borbély, Réka Nagy-Szentesi, András Mucsi (UNIDEBTTK), Hrvoje Novak, Mario Vašak (UNIZGFER), Marko Baša (E3)
Contact person	Arpad Racz (UNIDEBTTK)
Abstract (for dissemination)	The annex provides details about the implementation logic for different prediction and estimation submodules on the microgrid level.
Keyword List	3Smart, open software module, building microgrid, battery system, renewable energy, controllable load

Revision history

Revision	Date	Description	Author (Organization)
0.1	14 July 2017	First draft	Árpád Rácz, Sándor Misák, István Szabó, Réka Nagy-Szentesi, András Mucsi (UNIDEBTTK), Hrvoje Novak, Mario Vašak (UNIZGFER), Marko Baša (E3)
0.2	15 November 2017	Alpha version	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK), Hrvoje Novak, Mario Vašak (UNIZGFER), Marko Baša (E3)
0.3	29 December 2017	Final draft	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK), Hrvoje Novak, Mario Vašak (UNIZGFER), Marko Baša (E3)
0.4	30 December 2017	Corrections	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK), Hrvoje Novak, Mario Vašak (UNIZGFER), Marko Baša (E3)
1.0	30 December 2017	Corrections	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK), Hrvoje Novak, Mario Vašak (UNIZGFER), Marko Baša (E3)
1.1	15 September 2018	Inherited from the deliverable D4.2.1	Mario Vašak (UNIZGFER)
1.2	15 January 2019	Updated document	Mario Vašak (UNIZGFER)
1.3	17 June 2019	Updated document	Mario Vašak (UNIZGFER)
1.4	26 June 2019	Final draft document	Mario Vašak, Hrvoje Novak (UNIZGFER), Arpad Racz (UNIDEBTTK), Marko Baša (E3)
2.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of contents

Executive summary

The main objective of the project Smart Building – Smart Grid – Smart City (3Smart) funded within the Interreg Danube Transnational Programme is to provide technology and legislative setup for cross-spanning energy management of buildings and utility grids, foremost electricity distribution grids.

One of the main pillars in reaching that objective is to derive a modular energy management tool for buildings, which can be easily adapted to different configurations of the building and adds upon the existing building automation system. This “3Smart” energy management concept consists of three modules put in the hierarchical organization: zones comfort module, central HVAC module and microgrid energy flows module. Further on, each of the modules incorporate three different submodules: prediction and estimation, optimal control and interfaces to the equipment. Each of the considered pilot locations has specific configuration where all modules are not necessarily present. High level of flexibility is therefore targeted to achieve easy modifications for particular pilots.

This annex to D4.5.3 describes the prediction and estimation submodules of the microgrid energy flows module. The module is positioned as the highest in the hierarchy of the building-side EMS and as a connection to the electricity distribution grid and energy markets. The core of the submodule is the model predictive control algorithm based on linear program. It minimises the cost of building operation in various time-variable market conditions known in advance. It manages energy storages operation, controllable loads and controllable energy production while respecting the system limitations and degradation in time. Non-controllable renewable energy sources production and non-controllable building consumption is also taken into account based on weather forecast data and historical measurements data. The outputs are consumption prices sent towards lower hierarchy modules as a basis for overall building consumption adjustment to current market conditions.

In this document prediction and estimation submodules are presented. These modules are essential for providing input information for the MPC submodule of the microgrid-level module.

Introduction

One of the main focuses of the 3Smart project is to derive a modular energy management tool for buildings that can be easily adapted to different configurations of the building and adds upon the existing building automation system. This “3Smart” energy management concept consists of three modules put in the hierarchical organization: zones comfort module, central HVAC module and microgrid energy flows module, as shown in Fig. 1. Further on, each of the modules incorporate three different submodules: prediction and estimation, optimal control and interfaces to the equipment (Fig. 2). Each of the considered pilot locations has specific configuration where all modules are not necessarily present. High level of flexibility is therefore targeted to achieve easy modifications for particular pilots.

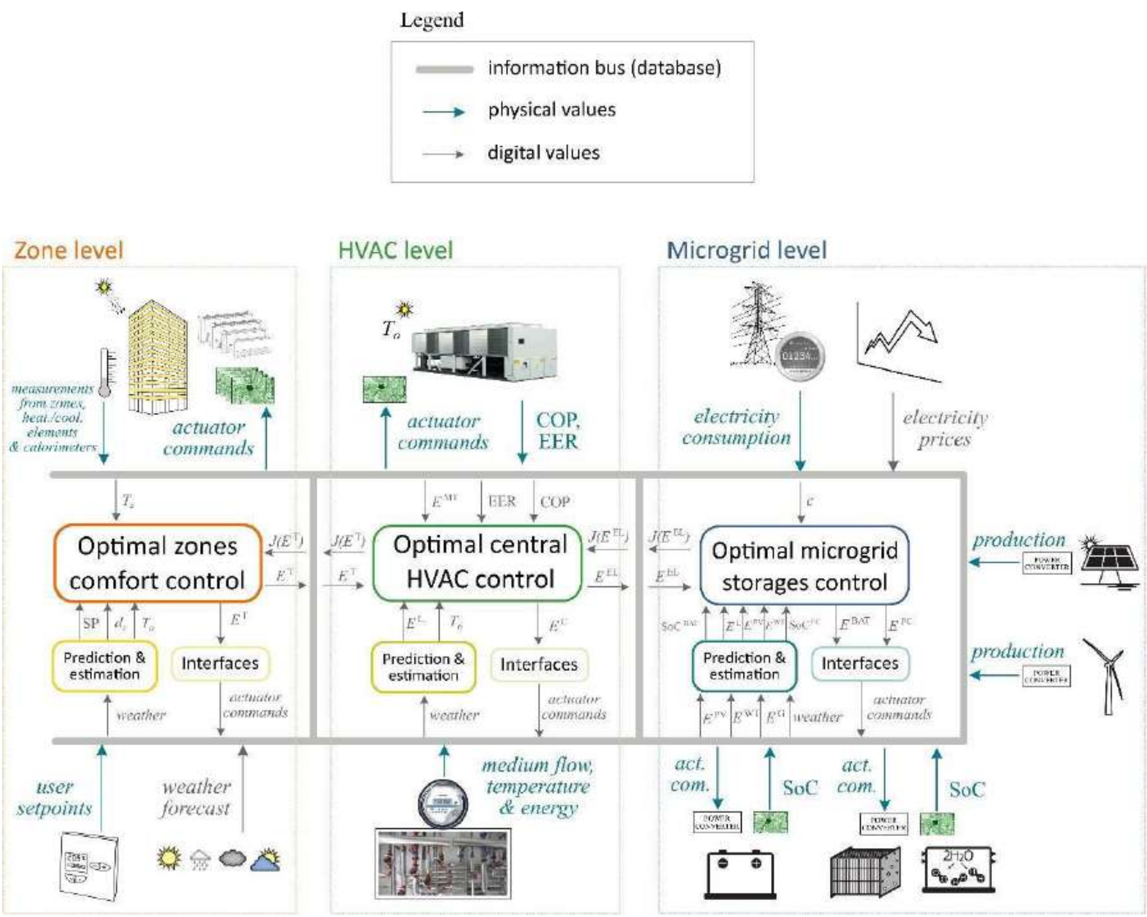


Figure 1: Functional diagram of the 3Smart EMS hierarchy on the building side.

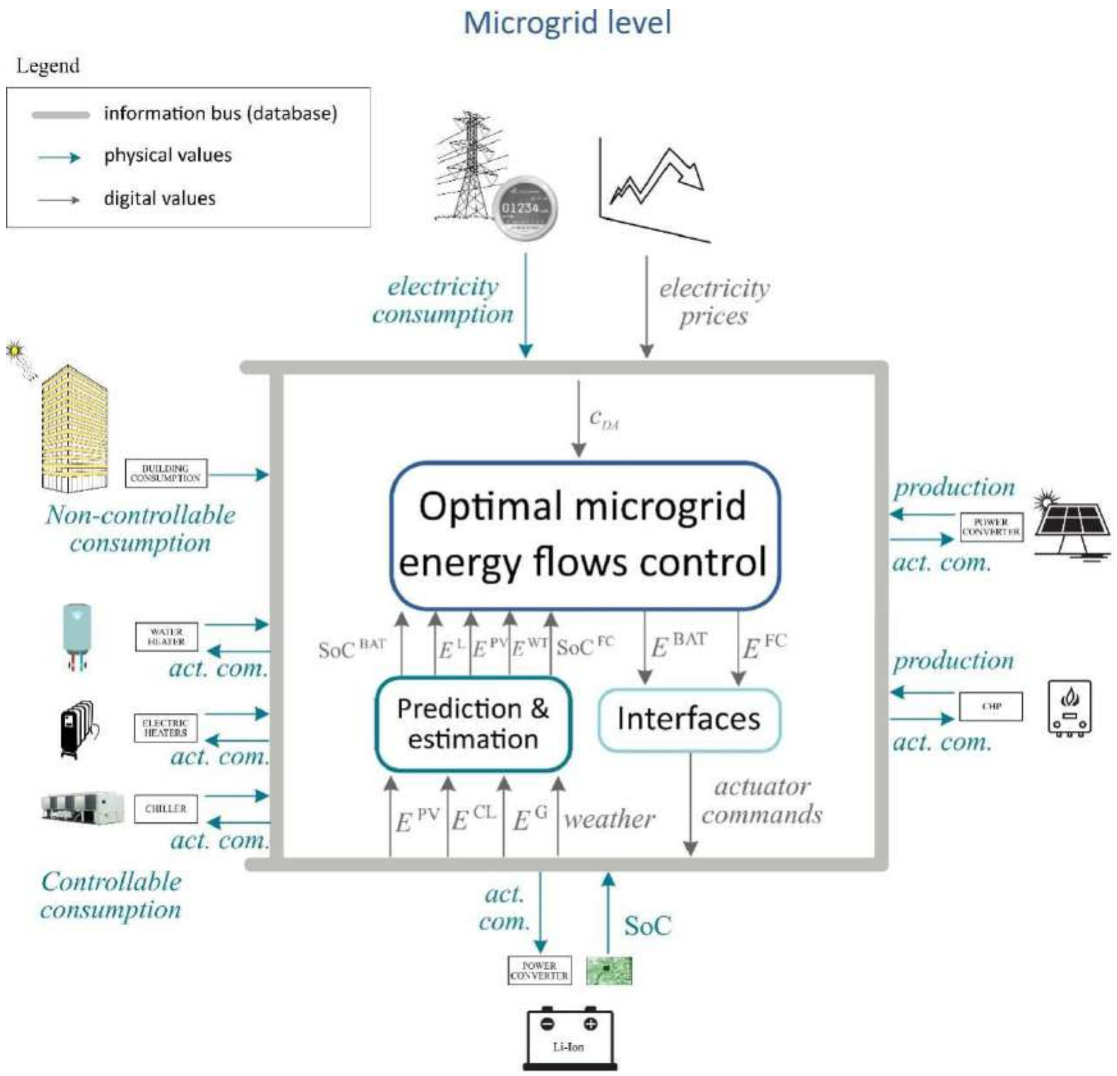


Figure 2: Modules schematics of the 3Smart EMS concept.

This document contains the following submodules:

- Submodule for identification of battery parameters
- Submodule for estimation of battery state of charge
- Submodule for prediction of the total non-controllable energy consumption on the microgrid level
- Submodule for prediction of the photovoltaic array production
- Submodule for estimation of the domestic hot water tank model
- Submodule for estimation of the heat buffer tank model

1 Submodule for identification of battery parameters (M.PE.1)

1.1 Theory

Battery Management System (BMS)

An effective BMS using the lithium-ion battery is compulsory so that battery can operate safely and reliably, prevent any physical damages, and handle thermal degradation and cell unbalancing. Moreover, different states of the battery, such as the SoC, state of health (SOH) can be assessed through an efficient battery management system, which can sense temperature, measure voltage and current, regulate safety alarm to avoid any overcharging/over discharging. Furthermore a BMS is essential for controlling and updating data, detecting faults, equalizing battery voltage that are the important factors for achieving a good accuracy of SoC and SoH. The components of BMS can be categorized into the hardware and software structure perspective, as shown in Figure 1.1 [1].

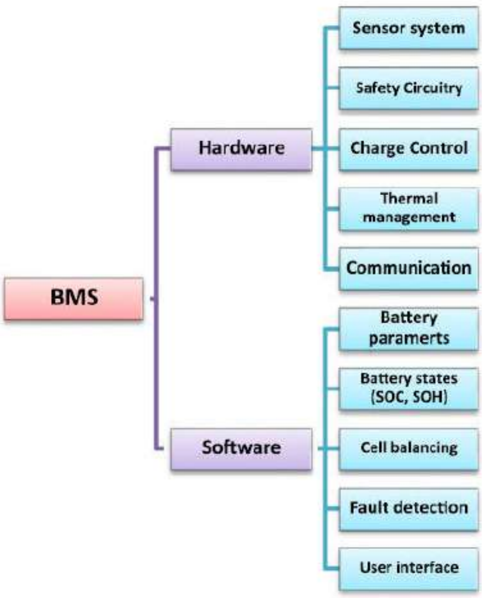


Figure 1.1: The components of BMS [1]

The function of estimating battery parameters and battery states will be handled by 3Smart software modules.

Equivalent circuit model of batteries

Battery models are typically classified as one of the following: electrochemical models, behavioral models or equivalent circuit-based models.

Electrochemical models are based on the battery physical construction and chemistry. While these models can be extremely accurate in describing the battery behavior, usually they are computationally time-consuming and not suitable for real-time control oriented applications. Different from the electrochemical approach, the equivalent circuit model describes the effect of the chemical processes of major interest using electrical circuit elements, such as voltage source, resistors and capacitors to approximate the battery dynamics [2].

Behavioral models are empirical and utilize functions to model battery dynamics. There are several models, namely: the combined model, the simple model, the zero-state hysteresis model, the one-state hysteresis model and the enhanced self-correcting model. These models can be easily optimized and they account for ohmic losses, hysteresis and polarization time constants [3].

The mostly used battery models are the equivalent circuit-based models, because of its simplicity and satisfactory performance [2].

Basic questions for high decision making (MPC):

- How much energy is available or can be stored?
- How much power can the battery system provide or absorb?

These parameters cannot be measured, they must be estimated from available data. The estimation process is shown in Figure 1.2 below. In Figure 1.2 *V* represent the voltage, *I* represent the current and *T* represent the temperature of each cell. From this set of data the estimator is able to determine the capacity (*Q*), the SoC value (*SoC*) and the internal resistance of each cell (*R*). After that it is possible to calculate with the whole battery pack. For the estimation of energy and power we must determine:

- Battery Capacity
- SoC: State of Charge
- Resistances and other battery model parameters

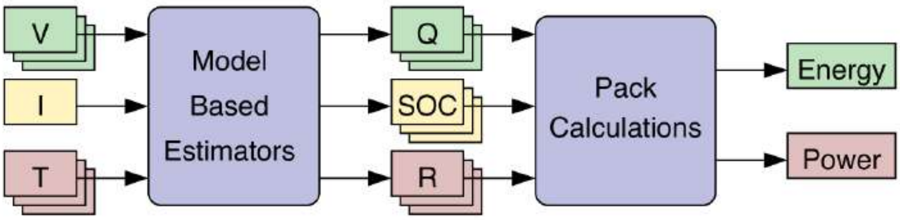


Figure 1.2: The process of the battery pack's parameters calculation [4]

State-of-charge (SoC) $z(t)$ **definition**

$$z(t) = z(0) - \frac{1}{Q}\int_0^t \eta i(\tau)d\tau \tag{1-1}$$

where Q is total capacity of the battery in As. The current is positive on discharge and negative on charge. The coulombic efficiency is approximately one. It is one for discharge, and smaller than one for charge. The total capacity is proportional to the number of positions capable of storing Li in the electrode structure. It is not a function of temperature, charging rate, etc.

Simplified cell model

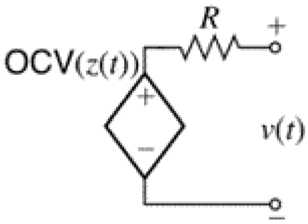


Figure 1.3: Simplified cell model [4]

$$v(t) = OCV\big(z(t)\big) - i(t)R \qquad (1\text{-}2)$$

$$i(t) = \frac{OCV\big(z(t)\big) - v(t)}{R} \qquad (1\text{-}3)$$

Here v is voltage, i current, *OCV* is open circuit voltage. Cell resistance can depend on *SoC*, temperature. It must be determined by a cell test and tabulated.

The determination of the OCV curve and the internal resistance is a difficult task, which is usually done off-line. Most of the proposed methods in the literature requires a long and tedious procedure involving charging and discharging pulse sequences followed by long resting periods. Higher sampling rates than the available one minute are required to capture the instantaneous change of voltage after the current pulse.

Under the present conditions, for online determination of the OCV curve a quasi-static approach will be applied. The battery is discharged from a fully charged initial state, discharged with a constant low current (0.02C) and then charged with the same low current [figure]. The average of the charge and discharge curves provides the OCV curve, and the difference is used to determine the internal resistance.

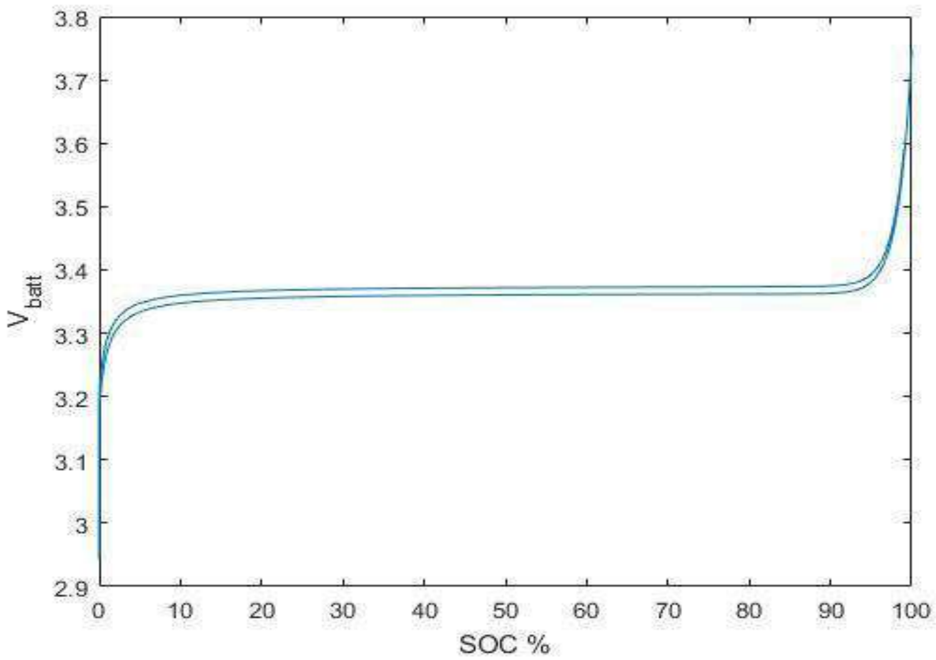


Figure 1.4: Charge and discharge voltages under constant current condition

Power estimate: We set a minimal and maximal terminal voltage as the limit of power input and output. Charge and discharge power is given as positive quantities. For a single cell:

$$P_{dis} = v(t)i(t) = v_{min}\frac{OCV(z(t)) - v_{min}}{R_{dis,\Delta T}} \quad (1\text{-}4)$$

$$P_{chg} = v_{max}\frac{v_{max} - OCV(z(t))}{R_{chg,\Delta T}} \quad (1\text{-}5)$$

The total cell energy is equal to:

$$E(t) = Q\int_{z_{min}}^{z(t)} OCV(\xi)d\xi \approx QV_{nom}\Delta z \quad (1\text{-}6)$$

In equation (1-6) E is the cell's actual stored energy and Q is the cell total capacity in ampere seconds (Coulombs) [4].

1. The pack power can be estimated as the lowest cell power multiplied by the number of cells.

2. Pack energy is calculated from the lowest cell discharge capacity, calculating the corresponding state of charge for all cells:

$$z_{low,k} = z_k(t) - \frac{Ah\ discharged}{Q_k} \quad (1\text{-}7)$$

$$E_{pack}(t) = \sum_k Q_k \int_{z_{low,k}}^{z_k(t)} OCV(\xi)d\xi \quad (1\text{-}8)$$

State-of-Health (SOH) estimation

Change of the parameters with time and number of cycles results in capacity loss, internal resistance change and increase of imbalance between cells. Periodic capacity measurement by discharging a fully charged battery can be used to determine the actual capacity (C=Q).

$$SoH = \frac{C}{C_{nom}} \tag{1-9}$$

In equation (1-9) C is the maximum capacity of the battery in its current state and C_{nom} is the initial nominal capacity of the fresh battery. Both values are in Ah.

From the State of health (SoH) estimation, the remaining useful lifetime (RUL) can be extrapolated as it is shown in Figure 1.5.

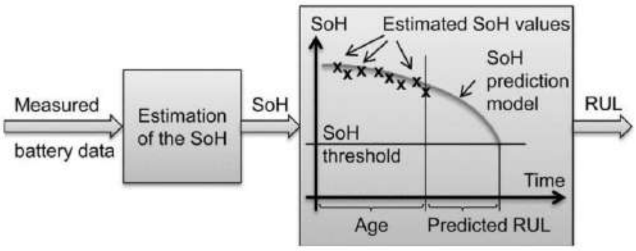


Figure 1.5: SoH and RUL estimation sequence [5]

Discharge curves of the fully charged battery pack can be used to determine capacity fading (see Figure 1.6).

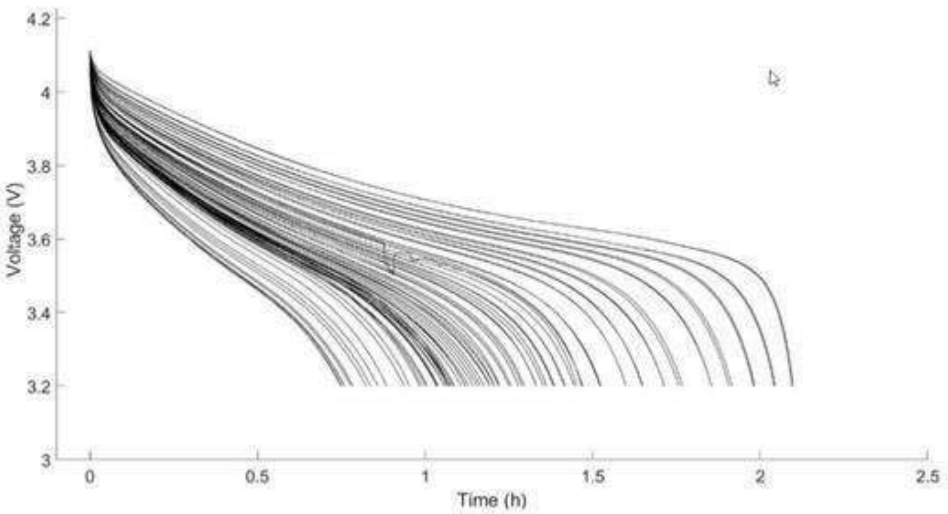


Figure 1.6: Reference discharge voltage profiles for determination of cell capacity changes over the course of battery testing [5]

In the minimalistic approach the battery pack is fully charged at regular intervals to determine the capacity.

1.2 Inputs

- Historical profile of measured battery currents
  - $[I_{batt}]$ = A

- Historical profile of measured battery and battery cells voltages
  - $[U_{batt}]$ = V
- Historical profile of measured battery cells temperatures
  - $[T_{batt}]$ = °C
- Historical profile of measured energy for converter AC side (before the isolation transformer)
  - $[E_{convAC}]$ = kWh

Resolution: 1/min

I_{batt}, U_{batt}, T_{batt} should be average for the time period.

E_{convAC} should be a sum for time period between the sampling instants, ending at the time stamp.

1.3 Outputs

- Battery/batteries model, θ_{batt}

Parameters of the battery/batteries model:

- $[E_{stored, max}]$ = kWh – constant
  - Maximum possible/allowed energy content of the battery system
- $[E_{stored, min}]$ = kWh – constant
  - Minimum possible/allowed energy content of the battery system (generally 0)
- $[P_{out, max}]$ = kW – constant
  - Maximum allowed output power of the battery system
- $[P_{in, max}]$ = kW – continuous function of E_{SoC} provided in a form of a polytope (conservative approximation)
  - Maximum allowed input power of the battery system
- η_{BatSys} – dimensionless, constant
  - Energy efficiency of the battery system (battery and converter together)

1.4 Internal parameters

- Nominal battery parameters
  - Original battery cell capacity in Ah
  - Number of battery cells

  - Discharging cut-off voltage
  - Charging cut-off voltage
  - Recommended charging - discharging current
  - Maximum short-time discharging current
  - Battery type: {LiFePO4, salt-water, lead-acid}
- Current battery pack capacity
  - $[C_{batt\_pack}] = Ah$
- OCV curve for battery cells
  - voltage vs. capacity
- Internal resistance of battery cells
  - $[R_{int}] = \Omega$
- Energy efficiency of battery system
  - $\eta_{Battery}$
- Energy efficiency of power converter
  - $\eta_{PowerConverter}$

1.5 Frequency of submodule calls

The submodule is envisioned to be called monthly.

1.6 Algorithm

The following steps need to be performed:

- Initial parameter setup
- Calibration cycle

Initial parameter setup

It should be done manually. Nominal battery parameters should have a default value, based on the datasheet. These values do not change during the lifetime of the system.

Calibration cycle

During a calibration cycle the following measurements are being performed:

- OCV measurement
- Internal resistance measurement
- Recalculation of maximum stored energy
- Recalculation of battery system efficiency

It should be done once a month. Weekends might be more suitable.

Steps:

1. Charge the battery to full capacity.

2. Fully discharge the battery with relatively low current and record the voltage values for each cell. Relative low current should be 0.05C. With this current the measurement takes approx. 20 hours.
3. Fully charge the battery with relatively low current and record the voltage values for each cell. Relative low current should be 0.05C. With this current the measurement takes approx. 20 hours.

The battery voltages are recorded during the full discharge/charge cycle with low current. The actual battery capacity is calculated by coulomb counting during the discharging processes. The state of charge is calculated based on the actual capacity and the Coulomb counting formula for charging and discharging stages. The average of the battery voltages at a given state of charge provides the OCV curve. The difference of the voltages divided by the double of the current provides the resistivity value.

The OCV curves is fitted with a polynomial function. This polynomial function together with its derivative and integral are used by the SoC estimation algorithm.

For the internal resistance a constant value is used. It is determined based on the average in the middle region of the curve.

The energy released during the discharge cycle gives maximum stored energy value. This value is measured on the AC side of the converter.

The efficiency of the battery system is calculated as the ratio of the ohmic loss to the stored energy. This value depends on the actual current.

Battery system efficiency consists of two components:

$$\eta_{BatSys} = \eta_{Battery} \cdot \eta_{PowerConverter} \qquad (1\text{-}10)$$

Each will be handled as constant using worst value occurred during the last month. The values measured during the calibration cycle are taken into account.

During the test sequence the battery system is turned offline (both $E_{stored,\,max}$ and $E_{stored,\,min}$ are set to zero). The time interval for the test is recorded in the database, and the measured values are copied to a table which always contains the latest calibration cycle.

2 Submodule for estimation of battery state of charge (M.PE.2)

2.1 Theory

Definition and classification of SoC estimation

The SoC is one of the most important variables for batteries, but its definition presents many different issues. In general, the SoC of a battery is defined as the ratio of its current charge

(*Q(t)*) to the actual maximum charge (Q_{act}). The starting nominal maximum charge is given by the manufacturer and represents the maximum amount of charge that can be stored in the battery. The actual charge is determined during the battery parameter estimation. The SoC can be defined as follows [6]:

$$SoC\ (t) = \frac{Q(t)}{Q_{act}} \tag{2-1}$$

Assuming that the initial value *SoC* (t_0) is known, instantaneous *SoC* is generally evaluated by integrating the battery current over time, as shown in the following equation:

$$SoC(t) = SoC(t_0) + \frac{\int_{t_0}^{t_0+\tau} I_{bat} d\tau}{Q_{act}} \times 100\% \tag{2-2}$$

where *SoC* is the current SoC value, I_{bat} is the battery current value and the Q_{act} is the actual charge [7].

The various mathematical methods of estimation are classified according to methodology. The classification of these SoC estimation methods is different in the various literatures. However, some literatures allow a division into the following four categories.

Direct measurement: these methods use physical battery properties, such as the voltage and impedance of the battery.

Book-keeping estimation: these methods use charging/discharging current as the input and integrated the charging/discharging current over time to calculate the SoC

Adaptive systems: the adaptive systems are self-designed and can automatically adjust the SoC for different charging/discharging conditions. Various new adaptive systems for SoC estimation have been developed.

Hybrid methods: the hybrid models benefit from the advantages of each SoC estimation method and allow a globally optimal estimation performance. The literature shows that the hybrid methods generally produce good estimation of SoC, compared to individual methods.

The table presents the specific SoC estimation methods in view of the methodology. The applications of specific SoC estimation methods in battery management systems are consequentially different [6].

Categories	**Estimation methods**
Direct measurement	**Open circuit voltage method**
	Terminal voltage method
	Impedance method
	Impedance spectroscopy method
Book-keeping estimation	**Coulomb counting method**
	Modified Coulomb counting method
Adaptive systems	**Back propagation (BP) neural network**

	Radial basis function (RBF) neural network
	Support vector machine
	Fuzzy neural network
Hybrid methods	**Coulomb counting and OCV combination**
	Coulomb counting and Extended Kalman Filter combination
	Per-unit system and Extended Kalman Filter (EKF) combination

Table 2.1: List of the methods for SoC estimation [6]

Coulomb Counting method

The estimation of *SoC* of a Li-Ion battery in this method is based on monitoring both the voltage V_{bat} and the current I_{bat}. The operation mode of the battery is recognized by the direction of current through the battery system.

The information needed to carry monitoring are the measurement of the battery voltage, the current flowing through it and the operating temperature. Coulomb counter Δ*Q* is used to track the *SoC* when the battery is charged or discharged.

The amount of charges Δ*Q* in an operating period τ is obtained by a temporal integration of the measured charging/discharging current I_{bat} like expressed in the following equation:

$$\Delta Q = -\int_{t}^{t+\tau} I_{bat} dt \qquad (2\text{-}3)$$

The variance ΔQ that will be used in the following equations is negative if the battery is in discharge, positive if in charge.

Charging Mode

In this mode, the coulomb counter is presented by Q_{gained} as expressed in the following equation which represents the quantity of charges accumulated during τ.

$$Q_{gained}(t+\tau) = Q_{gained}(t) + \Delta Q \qquad (2\text{-}4)$$

So the variation of the state of charge gained in this same operating period is obtained by the relation:

$$\Delta SoC\ (t+\tau) = \frac{Q_{gained}(t+\tau)}{Q_{rated}} \times 100\% \qquad (2\text{-}5)$$

By cumulating the previous state of charge indication and the obtained one can have the instantaneous value of SoC:

$$SoC\ (t+\tau) = SoC\ (t) + \Delta SoC\ (t+\tau) \qquad (2\text{-}6)$$

Discharging Mode

In the discharging mode the coulomb counter is presented by Q_{lost}, which represents the amount of charge lost in the operating period τ by the equation:

$$Q_{lost}(t+\tau) = Q_{lost}(t) + \Delta Q \tag{2-7}$$

Self-Discharging Mode

At the battery storage periods, the percentage of monthly self-discharging is converted to amount discharged per hour; this amount is designed by the constant $q_{per/hour}$. Considering a 5% rate of self-discharge per month $q_{per/hour}$ is approximated to 0.0016 Ah. So the quantity of charges dissipated in the open circuit period Q_{oc} is calculated by the equation representing the accumulative losses during the storage hours.

$$Q_{OC}(h+1) = Q_{OC}(h) + q_{per/hour} \tag{2-8}$$

Temperature effect

The temperature variation affects considerably the state of charge of the Li-Ion battery. In order to improve accuracy of the battery state of charge, a temperature coefficient α is introduced and multiplied by the value of the calculated *SoC*. This coefficient varies depending on the operating temperature according to the intervals variations [7]:

$$\alpha = \begin{cases} 0.5, & T < -20° \\ 1, & -20° \leq T \leq 40° \\ 0.8, & T > 40° \end{cases} \tag{2-9}$$

Nevertheless, the Coulomb counting method is an open-loop algorithm and could result in significant inaccuracies due to uncertain disturbances and variables such as noise, temperature, current, etc. Also, there are difficulties in determining the initial value of SoC which causes a cumulative effect. In addition, the estimation accuracy depends highly on the current sensors used which may be affected by measurement errors, which also result in cumulative effect. Furthermore, the method needs complete discharging of the cell and periodic capacity calibration to obtain maximum capacity which shorten the battery lifespan [1].

Coulomb Counting and Kalman Filter combination

The object of hybrid models is to benefit from the advantages of each method and obtain a globally optimal estimation performance. Since the information contained in the individual estimating method is limited, a hybrid method can maximize the available information, integrate individual model information and make the best use of the advantages of multiple estimating methods thus improving the estimation accuracy [6].

2.2 Inputs

- Historical profile of measured battery currents

  - $[I_{batt}]$ = A
- Historical profile of measured battery and battery cells voltages
  - $[U_{batt}]$ = V
- Historical profile of SoC from battery pack
  - [SoC] = %

Resolution: 1/min

I_{batt}, U_{batt} should be average for the time period.

2.3 Outputs

- Current state of charge estimate in energy
  - $[SoC_E]$ = kWh

2.4 Internal parameters

- Current state of charge estimate in percentage
  - $[SoC_{\%}]$ = %
- see "Submodule for identification of battery parameters" internal parameters

2.5 Frequency of submodule calls

The submodules is executed every minute.

2.6 Algorithm

Coulomb-counting converted into energy:

$$E_{batt}(k+1) = E_{batt}(k) - I_{batt}(k+1) \cdot t_{sampling} \cdot \sum U_{batt}(k+1) \tag{2-10}$$

in case of I_{batt} charge (-) or discharge (+).

A standard Extended Kalman filter approach is implemented for the state of charge estimation. The basic principle of Kalman filtering is to construct a model of the real system which uses the internal variables which one wishes to estimate as inputs, while the outputs are some measurable variables. In this case the internal variable is the state of charge, while the measured variables are the voltage and current variables. The model of the battery contains the OCV(SoC) function and the internal resistance as parameters, which are determined during the parameter estimation phase.

The battery model provides a prediction for the battery voltage and the state of charge for the next time step based on the previous state of charge and the current values. In the correction step the actual measured voltages are used for the correction of the SoC internal model variable.

The stored energy is corrected based on the corrected SoC value.

3 Submodule for prediction of the photovoltaic array energy production (M.PE.3)

3.1 Submodule inputs

Table 3.1: Required inputs for the photovoltaic array production prediction submodule.

Variable name	Variable annotation	Variable description
Photovoltaic array power meters data	E_{pv}	Photovoltaic array energy production measured directly on the array power meter
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I_{\mathrm{diff}}^{h}, I_{\mathrm{dir}}^{\mathrm{n}}$ Remaining pilot sites: $T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).
Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

3.1.1. Solar irradiance data

Depending on the availability of solar irradiance measurements on different pilot sites throughout the project, two separate sets of weather measurements inputs are used.

On the UNIZG-FER pilot site, where direct normal and diffuse horizontal irradiance measurements are available, they are used as submodule inputs and paired with the same forecasted variables during submodule operation.

Due to high costs of direct and diffuse irradiance sensors other pilot sites provide measurements of global horizontal and tilted global irradiations which are then used as submodule inputs. Since measured and forecasted irradiances are now different, during submodule operation, forecasted direct and diffuse irradiance, solar angles (obtained through the use of Pysolar python library), geographical pilot site data and current datetime, are used for calculation of global horizontal and tilted global irradiances thus matching the measured and forecasted irradiance variables.

3.2. Submodule outputs

Table 3.2: Outputs of the photovoltaic array production prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the submodule)	θ_{PV}	Needed for on-line operation of the submodule
Predicted profile of the photovoltaic array energy production (for on-line operation of the submodule)	$(E_{\mathrm{pv}})_{\mathrm{N}}$	Needed for the MPC module on the microgrid level

Since the photovoltaic array production model is based on artificial neural networks (as presented in the following section 3.3) the predictions sometimes tend to reach impossible values, e.g. negative values when fluctuating around 0 or slightly positive values during night time with no solar irradiation. The structure of neural networks prohibits the model to incorporate exact boundaries on the model outputs, therefor all generated predictions are post-processed in order to avoid such unexpected values. Therefore, all negative predicted values are set to 0, as well as all values between dusk and dawn. Dusk and dawn times for a specific pilot site location are calculated using Astral python library.

3.3. Methodology

Based on a detailed description of artificial neural networks (ANN) given in [8], in the following sections a condensed description of ANNs structures and learning algorithms is given, together with a description of prediction module structure and operation schemes.

3.3.1. Artificial neural networks

Understanding of the human brain functioning and its learning and adaptation abilities made researchers try imitating its structure in order to imitate its capabilities in the computer systems. The basic element of the brain is a neural cell or neuron. Human brain contains 10^{11} neurons interconnected in the network with more than 10^{15} links. Although the neuron structure is rather simple, because of the immense number of links among them, a brain can perform the most complex operations. Schematic representation of a biological neuron is shown in Figure 3.1.

Neuron is composed of the cell body (soma), axon and a number of dendrites. Front end of an axon is connected to the cell body and its back end is split in a large number of branches. These branches are terminated by telodendria with their terminal buttons that touch dendrites of the other neurons. The terminal buttons contain numerous small bags with transmitters. A small distance between a telodendron of one neuron and a dendrite of another is called a synapse. Axon of one neuron forms synaptic interactions with many other neurons. Impulses generated in the cell body travel through an axon to a synapse.

Depending on the efficiency of each synaptic transfer, action potentials of different intensity come over dendrites to the cell body where they are then collected and processed. If their cumulative value is greater than the neuron sensitivity threshold, a cell body generates an action potential which is spread over the axon to the other neurons, and if it is lower, the neuron remains inactive and does not generate an action potential. From the signal processing perspective, neuron operation can be divided in *synaptic operation* which gives a certain relevance (weight) to each input signal and *somatic operation* which collects all the "weighted" input signals, and due to their cumulative values, generates or does not generate a signal which is transferred towards other neurons.

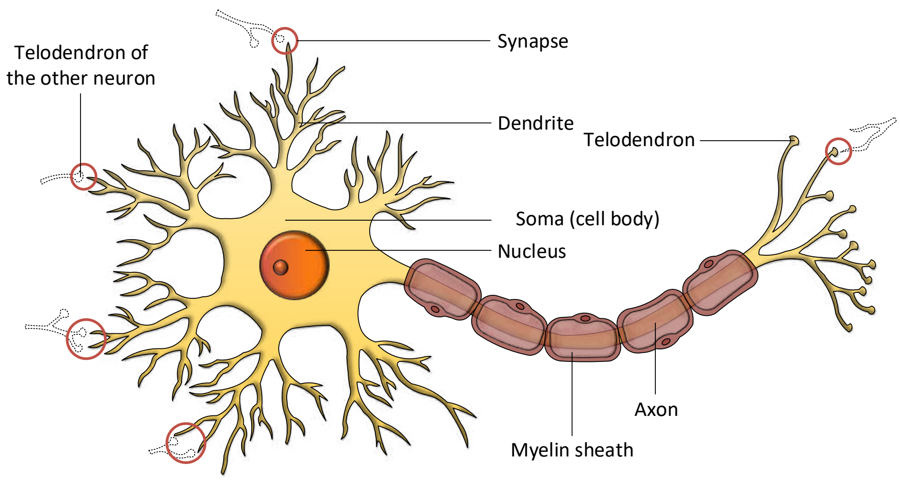


Figure 3.1: Schematic representation of a biological neuron.

3.3.1.1. Artificial neuron model

Early research in the field of artificial neurons was published by McCulloh and Pitts in 1943 and 1947 [9], [10]. Their model was based on a simple implementation of synaptic and somatic operations and was called a perceptron. Schematic representation of a perceptron is shown in Figure 3.2.

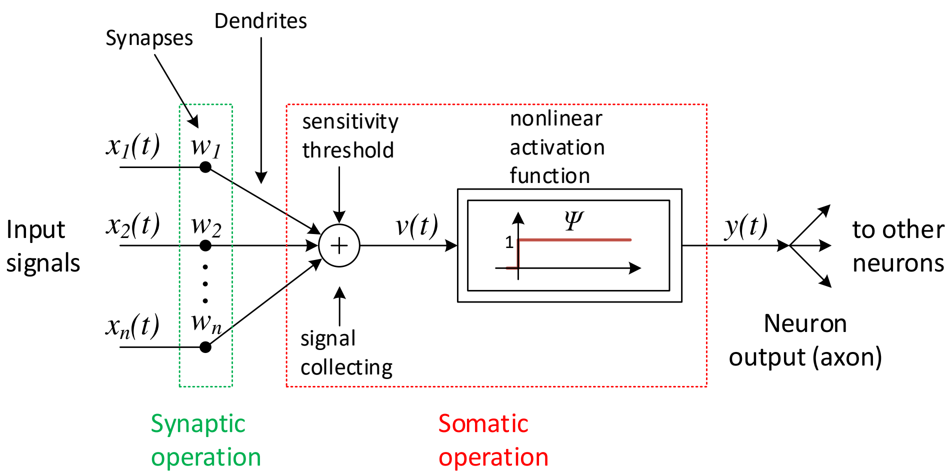


Figure 3.2: Schematic representation of a perceptron.

Synaptic operation is performed by multiplying input signals x_i with their weight coefficients w_i. Sum of all weighted signals is compared to a neuron sensitivity threshold w_{n+1}. If this sum is greater than a sensitivity threshold, nonlinear activation function ψ generates an output signal y equal to 1, and if it is less, neuron output is zero.

Mathematically, a perceptron can be described using these relations:

$$v(t) = \sum_{t=1}^{n} w_i(t)\, x_i(t) - w_{n+1}, \tag{3-1}$$

$$y(t) = \psi(v), \tag{3-2}$$

where:

$\boldsymbol{x_u} = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is a vector of neuron input signals;

$\boldsymbol{w_s} = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ is a vector of neuron input signals;

$\boldsymbol{w_{n+1}}$ is a neuron sensitivity threshold;

$\boldsymbol{v(t)}$ is a similarity measure between input signals and synaptic weight coefficients (result of the confluence operation);

$\boldsymbol{\psi(t)}$ is a nonlinear activation function;

$\boldsymbol{y(t)}$ is a neuron output.

However, because of the too simple model of a neuron, especially because of the discontinuity in nonlinear activation function, perceptron is not able to solve some simple operations. These constraints of the perceptron can be overcome by applying a continuous differentiable activation function. Sigmoid functions are commonly used as activation functions because it was proved that the ANNs composed of at least three layers of neurons with sigmoid functions can represent any continuous function. One of the most commonly used activation functions is *tansig* defined by the following expression:

$$\psi(v) = \frac{2}{1 + e^{-2g_0 v}} - 1, \tag{3-3}$$

where g_o is an activation gain and it is usually set to 1. Because of an extension of the initial model, in literature neurons with sigmoid activation functions are also referred to as perceptrons.

Neuron models can be divided in two groups: static and dynamic models. Static neuron models, as opposed to dynamic ones, do not contain dynamic elements and their output depends exclusively on current values of input signals and weight coefficients. In this deliverable only ANNs with static neuron models are analysed.

3.3.1.2. Multilayer perceptron

Static neural networks are most commonly used ANNs, especially in identification and control applications. A basic element of the static ANN is a static neuron. In static ANNs neurons are organised in a feedforward way, i.e.: each neuron can be connected to the network inputs and/or to other neurons, but in the way that no feedback connections are formed. Therefore, static ANNs do not contain any dynamic elements and that makes them statically stable which is their most important advantage in relation to dynamic ANNs. However, in order to model a dynamic system, delayed input and output signals have to be explicitly included in the vector of input signals of the static ANN. The most commonly used static ANNs are multilayer perceptrons (MLP) whose structure is presented in Figure 3.3. MLPs consist of perceptrons organized in serially connected layers. Layers are often labelled with numbers $0, 1, 2, \cdots, L$, while for the number of nodes in the l-th layer we use label $n(l)$. The zeroth layer only transfers the input vector to an input of the first layer, L-th layer is an output layer, while layers between them are called hidden layers. Every neuron in a hidden layer is connected to all the neurons in two neighbouring layers with unidirectional feedforward connections. Connections between neurons of the neighbouring layers are represented by synaptic weight coefficients which act as signal gains on the corresponding connections. Values of the synaptic weight coefficients determine the network behaviour, i.e.: its ability of approximating a nonlinear function.

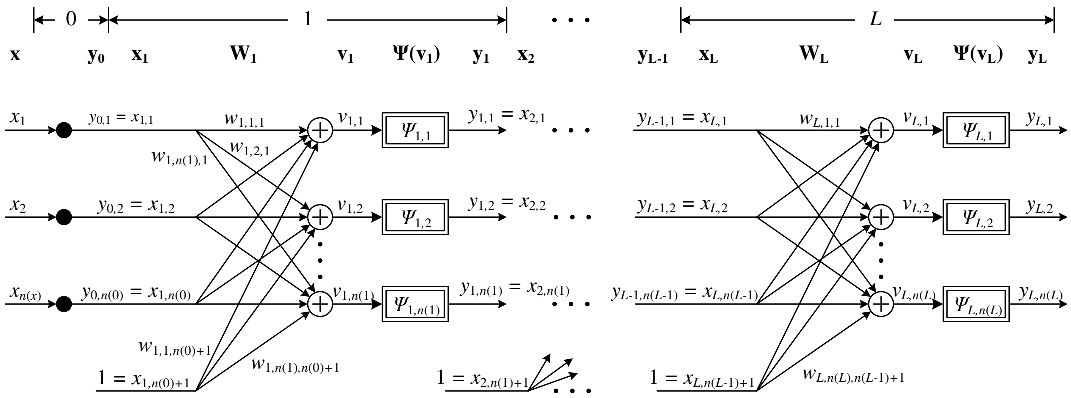


Figure 3.3: Schematic representation of a multilayer perceptron.

Mathematically, MLPs can be described by the following relations:

$$y_0 = x, \tag{3-4}$$
$$x_l = \left[y_{l-1}^T, 1\right]^T, \qquad 1 \le l \le L, \tag{3-5}$$
$$v_l = W_l \cdot x_l, \qquad 1 \le l \le L, \tag{3-6}$$
$$y_l = \psi(v_l), \qquad 1 \le l \le L, \tag{3-7}$$

where:

$\boldsymbol{x} = \left[x_1, x_2, \cdots, x_{n(x)}\right]^T$ is a vector of the network input od dimension $n(x)$;

$\boldsymbol{y_0} = \left[y_{0,1}, y_{0,2}, \cdots, y_{0,n(0)}\right]^T$ is an output vector of the 0-th layer of dimension $n(0)$;

$\boldsymbol{x_l} = \left[x_{l,1}, x_{l,2}, \cdots, x_{l,n(l-1)}, x_{l,n(l-1)+1}\right]^T$ is an input vector to the l-th layer (input $x_{l,n(l-1)+1} = 1$ multiplied by corresponding weight coefficient gives a scalar bias to neurons of the l-th layer);

$\boldsymbol{v_l} = \left[v_{l,1}, v_{l,2}, \cdots, v_{l,n(l)}\right]^T$ is an output vector of the confluence operation of the l-th layer;

$\boldsymbol{y_l} = \left[y_{l,1}, y_{l,2}, \cdots, y_{l,n(l)}\right]^T$ is an output vector of the l-th layer;

$$\boldsymbol{W_l} = \begin{bmatrix} \boldsymbol{w_{l,1,1}} & \cdots & \boldsymbol{w_{l,1,j}} & \cdots & \boldsymbol{w_{l,1,n(l-1)}} & \boldsymbol{w_{l,1,n(l-1)+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{w_{l,i,1}} & \cdots & \boldsymbol{w_{l,i,j}} & \cdots & \boldsymbol{w_{l,i,n(l-1)}} & \boldsymbol{w_{l,i,n(l-1)+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{w_{l,n(l),1}} & \cdots & \boldsymbol{w_{l,n(l),j}} & \cdots & \boldsymbol{w_{l,n(l),n(l-1)}} & \boldsymbol{w_{l,n(l),n(l-1)+1}} \end{bmatrix}$$ is a weight coefficient matrix of the synaptic connections of the l-th layer, dimension of which is $n(l) \times (n(l-1)+1)$;

$\boldsymbol{\Psi_l}(\boldsymbol{v_l}) = \left[\Psi_{l,1}(v_{l,1}), \Psi_{l,2}(v_{l,2}), \cdots, \Psi_{l,n(l)}(v_{l,n(l)})\right]^T$ is an activation function vector of the l-th layer (usually $\Psi_{l,1} = \Psi_{l,2} = \cdots \Psi_{l,n(l)}$).

The most commonly used activation function in the hidden layer is *tansig*, while in the output layer linear activation function is used. The activation gain is usually set to one.

The most important properties of the ANNs are universal approximation, learning and adaptation. ANN property of approximating any continuous function to an arbitrary accuracy is its most important property from the perspective of modelling, identification and control of nonlinear processes. Learning and adaptation properties enable that an adequately calibrated ANN has the generalization ability when the data that was not present in the calibrating data set comes to its input.

3.3.1.3. Neural network learning algorithms

Learning algorithm tunes network parameters in order to achieve its desired behaviour. In identification and control of nonlinear dynamic systems desired behaviour of a neural network is usually known, so error-based algorithms are used for the learning/calibrating procedure. Schematic representation of the error-based algorithm for neural network learning is shown in Figure 3.4.

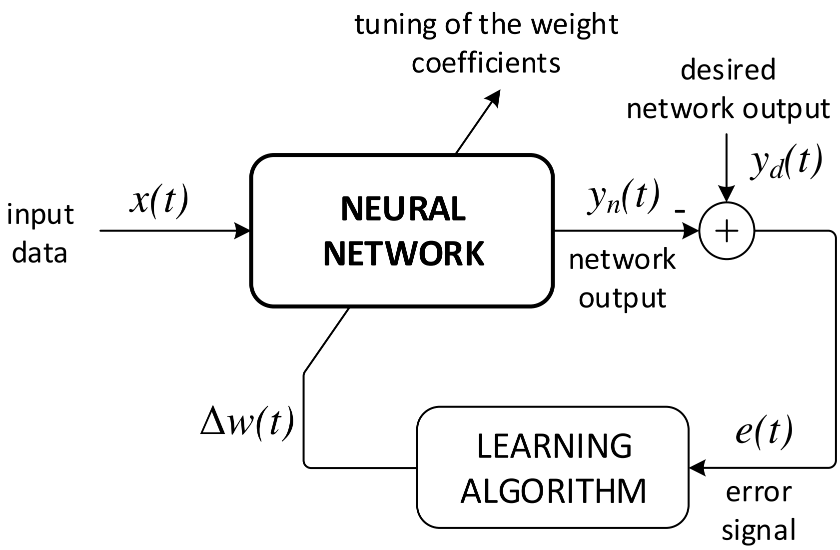


Figure 3.4: Schematic representation of the error-based algorithm for neural network learning.

Resulting neural network response y_n to the input data is compared to the external reference signal y_d, which represents desired network behaviour, generating error signal e based on which the learning algorithm changes synaptic weight coefficients of the network in order to improve its behaviour, i.e.: to decrease the error. As an error measure a criterion function $\Im(\Theta)$ is used and it can be any positive scalar function dependent on ANN parameters $\mathbf{\Theta}$. The most commonly used criterion function is defined as:

$$\Im(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{v=1}^{N} e(v,\boldsymbol{\Theta}) \cdot e^T(v,\boldsymbol{\Theta}) = \frac{1}{2}\sum_{v=1}^{N}\sum_{i=1}^{n(L)} e_i^2(v,\boldsymbol{\Theta}) = \frac{1}{2} e^{*T}(\boldsymbol{\Theta}) \cdot e^*(\boldsymbol{\Theta}), \tag{3-8}$$

where ν is a number of the measured sample, N is an overall number of measured samples, $e^*(\Theta)$ is the error vector of the whole measured data set, which is of dimension $N_e = N \cdot n(L)$.

There are two basic approaches in minimizing the criterion function $\Im(\Theta)$: non-recursive and recursive. According to the non-recursive approach, function $\Im(\Theta)$is minimized such that network parameter changes are determined based on the complete set of N measured samples. According to the recursive approach, function $\Im(\Theta)$ is minimized based on a local criterion function $\Im_\nu(\Theta)$, i.e. network parameters are changed after each measured sample.

Learning algorithm tunes network parameters until the criterion function reaches its minimum. Minimum of the criterion function $\Im(\Theta)$ can be formally defined by its Taylor series expansion in vicinity of the parameter vector Θ^0 for which the minimum is obtained, and by ignoring its third and higher order terms:

$$\Im(\boldsymbol{\Theta}) \cong \Im(\boldsymbol{\Theta^0}) = \nabla\Im^T(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta^0}} \cdot \Delta\boldsymbol{\Theta} + \frac{\mathbf{1}}{\mathbf{2}}\Delta\boldsymbol{\Theta^T} \cdot \boldsymbol{H}(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta^0}} \cdot \Delta\boldsymbol{\Theta}, \tag{3-9}$$

where:

$\Delta\boldsymbol{\Theta} = \boldsymbol{\Theta} - \boldsymbol{\Theta^0}$;

$\nabla\Im(\boldsymbol{\Theta})$ is a gradient vector of the criterion function:

$$\nabla\Im(\boldsymbol{\Theta}) = \left[\frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}}, \frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}}, \cdots, \frac{\partial\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}}\right]; \tag{3-10}$$

$\boldsymbol{H}(\boldsymbol{\Theta}) = \nabla^2\Im(\boldsymbol{\Theta})$ is a Hessian matrix of the criterion function:

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \begin{bmatrix} \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_1^2} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{1}}\partial\theta_{\boldsymbol{n}(\theta)}} \\ \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}\partial\theta_{\mathbf{1}}} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_2^2} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\mathbf{2}}\partial\theta_{\boldsymbol{n}(\theta)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}\partial\theta_{\mathbf{1}}} & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}\partial\theta_{\mathbf{2}}} & \cdots & \frac{\partial^2\Im(\boldsymbol{\Theta})}{\partial\theta_{\boldsymbol{n}(\theta)}^2} \end{bmatrix}. \tag{3-11}$$

For the criterion defined by (3-8), gradient vector and Hessian matrix become:

$$\nabla\Im(\boldsymbol{\Theta}) = \boldsymbol{J}^T(\boldsymbol{\Theta}) \cdot e^*(\boldsymbol{\Theta}), \tag{3-12}$$

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \nabla^2\Im(\boldsymbol{\Theta}) = \boldsymbol{J}^T(\boldsymbol{\Theta}) \cdot \boldsymbol{J}(\boldsymbol{\Theta}) + \sum_{\boldsymbol{i=1}}^{\boldsymbol{N_e}} e_i^*(\boldsymbol{\Theta})\nabla^2 e_i^*(\boldsymbol{\Theta}), \tag{3-13}$$

where $\boldsymbol{J}(\boldsymbol{\Theta})$ is a Jacobian matrix:

$$\boldsymbol{J}(\boldsymbol{\Theta}) = \begin{bmatrix} \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{1}}} & \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{2}}} & \cdots & \frac{\partial e_1^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{n}(\theta)}} \\ \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{1}}} & \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{2}}} & \cdots & \frac{\partial e_2^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{n}(\theta)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{1}}} & \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{2}}} & \cdots & \frac{\partial e_{Ne}^*(\boldsymbol{\Theta})}{\partial \theta_{\boldsymbol{n}(\theta)}} \end{bmatrix}. \tag{3-14}$$

Parameter vector $\Theta = \Theta^*$ will be the minimum argument of the function $\Im(\Theta)$ if the following conditions are fulfilled:

$$\nabla\Im(\Theta^*) = 0, \tag{3-15}$$
$$\Delta\Theta^{\mathrm{T}} \cdot \mathrm{H}(\Theta^*) \cdot \Delta\Theta > 0. \tag{3-16}$$

Therefore, tuning of the ANN parameters Θ is in fact a *nonlinear optimisation* problem where the criterion function $\Im(\Theta)$ is the objective function of the optimisation problem. *Gradient methods* are most commonly used nonlinear optimisation techniques. The main problem in applying gradient methods in ANN learning procedure is calculating a gradient vector of the criterion function over the network parameters. This problem has slowed research and application of ANNs for a while, but was successfully solved using the backpropagation algorithm. More details can be found in [8].

Tuning of the ANN parameter vector Θ is based on an iterative procedure:

$$\Theta(k+1) = \Theta(k) + \Delta\Theta(k) = \Theta(k) + \alpha(k)s_d(k), \tag{3-17}$$

where:

$s_d(k)$ is the minimum searching direction in the k-th iteration of the optimisation procedure (it is based on an information on a function $\Im(\Theta)$);

$\alpha(k)$ is the learning coefficient in the k-th iteration of the optimisation procedure (it determines the step size in the searching direction).

Depending on the procedure of determining the minimum searching direction $s_d(k)$, gradient methods can be divided into four groups:

- *Steepest descent methods*: $s_d(k) := -\nabla\Im\big(\Theta(k)\big)$;
- *Conjugate gradient methods:* $s_d(k) := -\nabla\Im\big(\Theta(k)\big) + \beta(k) \cdot s_d(k-1)$, where $\beta(k)$ is a scalar parameter which ensures conjugacy;
- *Newton methods:* $s_d(k) := -\left[\nabla^{2\Im}\big(\Theta(k)\big)\right]^{-1}\nabla\Im\big(\Theta(k)\big)$;
- *Quasi-Newton methods* [11], [12]*:* $s_d(k) := -S(k)\nabla\Im\big(\Theta(k)\big)$ where $S(k) \cong \left[\nabla^{2\Im}\big(\Theta(k)\big)\right]^{-1}$.

ANN learning algorithms are named based on the corresponding nonlinear optimisation methods which are used: steepest descent algorithms, conjugate gradient algorithms etc.

3.3.2. Applying neural networks to system modelling

In the last 20 years neural network applications for predicting variables in ecological and technical systems have become a well-known procedure in a research community [13]. In the early phases of their applications, ANNs were considered as a novel approach in system modelling and the majority of published papers in that period were related to applying ANNs in different systems and exploring their advantages in relation to the well-known statistic approaches [14]. Many review papers in this research area did not only affirm a potential of using the ANNs in prediction systems, but they also noted an importance of developing a standard methodology in the model development procedure using ANNs. Clearly defined methodology is an important procedure for all modelling methods, but especially in ANN modelling because models are developed based on the available data and they are not explicitly based on the physical system that is modelled, therefore, a possibility of developing a model which is not very meaningful is increased.

Main steps in developing the prediction model using ANNs are shown in Figure 3.5. Flow of data and outcomes for each step are also shown. First step in model development process is a choice of appropriate model outputs (variables which are going to be predicted) and potential inputs. A choice of potential inputs is based on *a priori* knowledge on the modelled process and on data availability. Selected data have to be processed (scaled, filtered, lagged) for being in an appropriate form for the next model development steps.

A general ANN prediction model can be expressed in the following form:

$$\mathrm{Y} = \mathrm{f}(\mathrm{X}, \mathrm{W}) + \mathrm{e}, \tag{3-18}$$

where Y is a model output vector, X is a model input vector, W is a model parameter vector (weight coefficients), f is a function which defines input-output relationship and e is a model error vector. Therefore, in model development process we need to define model inputs X, a functional relationship f defined by the ANN structure and ANN parameter vector W. Model inputs are determined using the so called Input Variable Selection (IVS) procedures which are described in subsection 3.3.3. Result of this step are model development data which are then divided in calibration and validation data sets. Calibration data are used in ANN learning algorithms for determining the optimal model parameters, while validation data are used for validating the calibrated model on the independent data set. If implicit regularization is used as a stopping criterion of the learning algorithm, calibration data are divided in training and testing data sets.

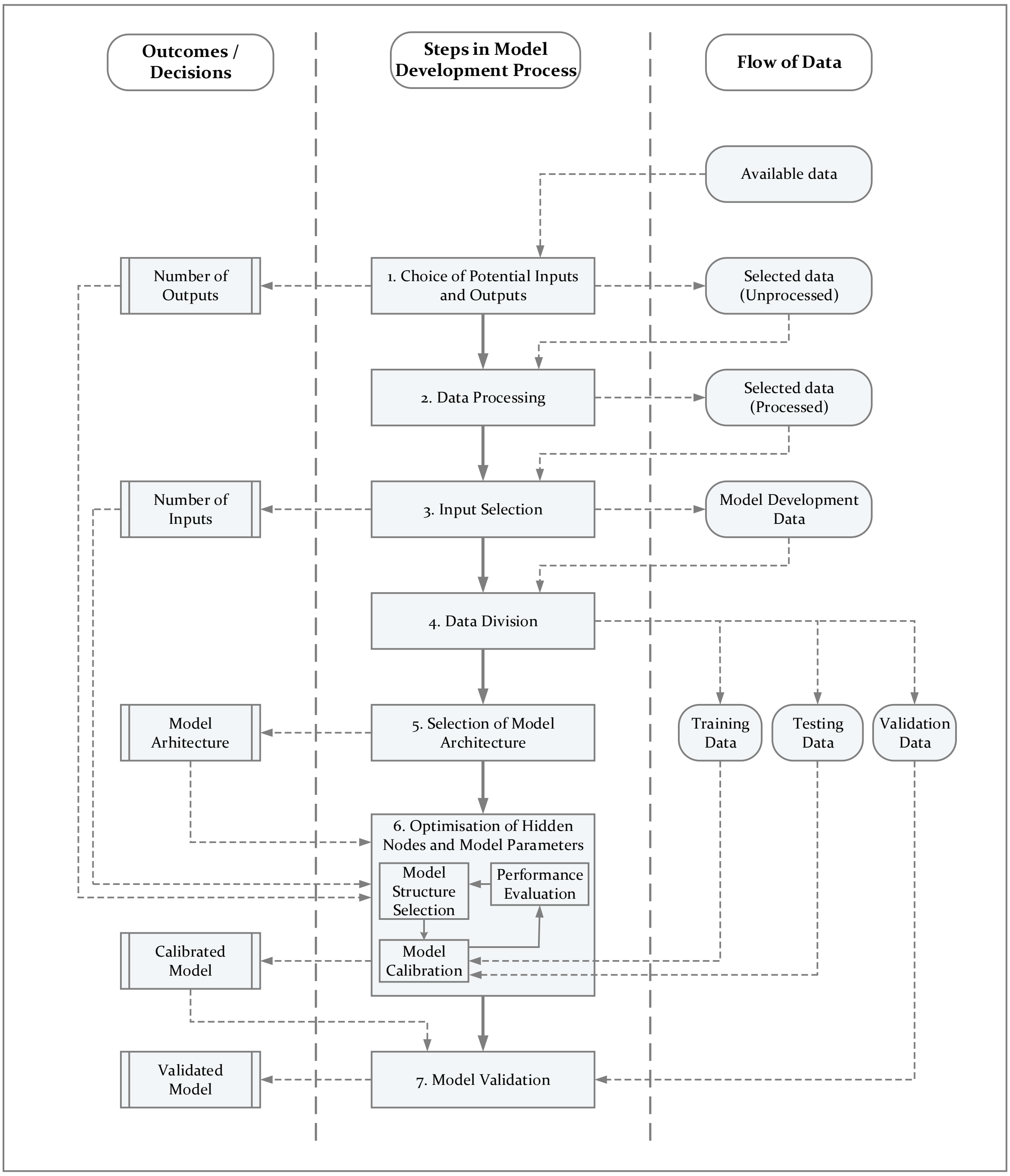


Figure 3.5: Main steps in the model development process using artificial neural networks [14].

The main objective of the ANN learning process is to find the global minimum of the criterion function $\Im(\Theta)$. However, in modelling of dynamic systems which inherently contain noise, the global minimum of the criterion function is not the optimal solution because the obtained model does not assure the best generalization properties. In the first phase of the ANN learning process a decrease of the criterion function $\Im(\Theta)$ on the training data leads to a decrease of the criterion function $\Im^{t}(\Theta)$ on the testing data. However, after certain number of iterations, value of the criterion function $\Im^{t}(\Theta)$ starts increasing although $\Im(\Theta)$ is still decreasing and, therefore, further adjusting of the ANN parameters leads to a deterioration of its generalization properties. This problem can be solved by early stopping of the learning process when a criterion function value on the testing data starts increasing. This procedure is called an *implicit regularization*.

Next step implies choosing a number of hidden layers and a number of neurons in each layer. The optimal structure of ANN is usually determined iteratively [14]. For a fixed structure, optimal parameters of the ANN are determined using learning procedure and they depend on the choice of learning algorithm and on initial ANN parameters. In general case criterion function is nonconvex and applying gradient methods can trap model parameter vector in a local minimum of the criterion function which is not the optimal solution. Therefore, a calibration process implies a number of calibration instances for different initial values of model parameters. ANN, defined by its structure and parameters, which has the minimal criterion function value on the calibration data is then validated on the validation data set. To ensure that a model development process results in the best possible model, it is required that training, testing and validation data sets have the same statistical properties [15].

3.3.3. Input variable selection procedure

One of the most important steps in modelling of complex systems is selection of the appropriate input variables. However, this step is usually not concerned to be of an extreme importance and most of the input variables are determined heuristically or based on *a priori* knowledge of the system which can result in including too many or too little input variables [16].

As a consequence of omitting one or more relevant input variables, model will not be able to describe the whole dynamics and phenomena of the system. Possibility of omitting relevant input variables is much greater for time series in which input candidates are not only different variables, but also their lagged values (unless dynamic ANNs are used) which significantly increases the number of potential input variables. Including too many input variables can be caused by poorly assessed relevance of an input variable or by existence of a redundancy among them, where some of the chosen variables contain some useful information, but are interdependent, so they contain a redundant information. This case leads to an increase in a number of local minima in the criterion function [14] and makes it harder to determine the optimal model parameters if a gradient method is used for ANN learning. On the other hand, with an increase of input variables, a number of model parameters is also increased which, as a consequence, leads to decreased speed and quality of the learning procedure. Furthermore, existence of an input variable which does not affect the output variable can lead to a deterioration of ANN generalization properties, i.e. the model will perform poorly on data that were not used during model calibration procedure.

These considerations indicate that the optimal ANN input variable set consist of the minimal set of variables which can describe the system behaviour well enough. A number of IVS algorithms were developed and they can be classified in *wrapper* and *filter* algorithms [17].

3.3.3.1. Wrapper algorithms

IVS using wrapper algorithms is based on developing a number of ANNs with different input vectors and the choice of an appropriate input set is determined based on performance of the corresponding ANN. The main drawback of this approach is that such a procedure can last very long because it is required to develop a large number of ANNs whereas the development of each implies an appropriate choice of the ANN structure and the learning algorithm. Additionally, appropriateness of the input variables chosen for a certain ANN architecture is not guaranteed for another architecture, so the application of the obtained input set is rather limited [16].

For d potential input variables, a number of possible input subsets is $2^d - 1$. Therefore, because of the large computational and time requirements, all possible input variable combinations are almost never tested. The most commonly used wrapper algorithms are forward selection, backward elimination and genetic algorithms [17].

Forward selection is an incremental procedure for forming the optimal input variable set in which a number of variables is incrementally increased. In the beginning, one out of d variables, for which an ANN with the best performance is obtained, is chosen. Then, the input set is enlarged by the next one out of $d - 1$ remained variables for which an ANN performance is most improved. A procedure is repeated until adding a new variable to the input set does not lead to a significant improvement of the ANN performance.

Backward elimination is a procedure inverse to a forward selection, i.e. the input variable set is incrementally reduced. The procedure starts with an input set which contains all the potential input variables and the least relevant variables are progressively eliminated from the input set. This procedure is computationally more intensive than the forward selection because a large number of inputs requires learning an ANN with much larger number of parameters.

Genetic algorithms introduce stochastic elements in the procedure of selecting the optimal input variable set, increasing a possibility of finding the optimal set. Genetic algorithms show their advantages in relation to forward selection and backward elimination when the candidate set contains variables which only combined with other variables show their relevance to an output variable, while taken separately, do not have an excessive importance.

3.3.3.2. Filter algorithms

Unlike wrapper, filter algorithms use statistical measure of dependence between an output variable and potential inputs as a criterion for input selection. Uncoupling IVS procedure and model calibration does not only increase the modelling efficiency, but also extends possible applications of the obtained input set. However, efficiency of a filter algorithm is highly dependent on the statistical measure employed [16].

The most commonly used statistical measure of dependence is a linear correlation coefficient whose main drawback is that it only determines the linear dependence between variables which is particularly problematic in the model development using ANNs because they are used as an alternative to linear regression when a dependence between model inputs and output is nonlinear. Therefore, it is more meaningful to use an appropriate nonlinear statistical measure of dependence, like mutual information [14]. Unlike linear correlation coefficient, mutual information is also sensitive to dependences which are reflected in higher input-output correlation moments – mutual information is equal to zero if and only if two variables are strictly independent [18].

Apart from inputs relevance, IVS procedures should also consider redundancy of the input variables. In order to do so, a suitable algorithm based on partial mutual information (PMI) was developed and it is described in the next subsection.

3.3.3.3. Input variable selection algorithm based on partial mutual information

For a given continuous random variable X with a codomain $C(X)$, Shannon entropy is defined as:

$$\mathrm{H(X)} = -\int_{\mathrm{C(X)}} \mathrm{f(x)} \ln \mathrm{f(x)}\ \mathrm{dx}, \tag{3-19}$$

where x is an outcome of random variable X and $f(x)$ is its probability density function (pdf). Entropy is a term well-known in the information theory and it represents an informational description of random events and defines a measure of the information content, i.e. random variable uncertainty. Mutual information of two random variables, X and Y, is defined as:

$$\mathrm{I(X;Y)} = \int_{\mathrm{C(Y)}} \int_{\mathrm{C(X)}} \mathrm{f(x,y)} \ln\left(\frac{\mathrm{f(x,y)}}{\mathrm{f(x)f(y)}}\right) \mathrm{dxdy}, \tag{3-20}$$

where $f(x)$ and $f(y)$ are pdfs of the variables X and Y, respectively, and $f(x,y)$ is a joint pdf of the random vector (X,Y). Mutual information can be expressed using entropies as:

$$\mathrm{I(X;Y)} = \mathrm{H(X)} + \mathrm{H(Y)} - \mathrm{H(X,Y)}, \tag{3-21}$$

where $H(X)$ and $H(Y)$ are entropies of the random variables X and Y, respectively, and $H(X,Y)$ is a joint entropy of the random vector (X,Y). Mutual information represents a reduction in uncertainty of the random variable Y knowing the random variable X and *vice versa*. Figure 3.6 depicts the dependency among mutual information and entropies of the random variables X and Y.

Here, $H(Y|X)$ is conditional entropy of Y given X, that is, the amount of uncertainty in the random variable Y when the value of X is known, and it is formally defined as:

$$\mathrm{H(Y|X)} = \int_{\mathrm{C(Y)}} \int_{\mathrm{C(X)}} \mathrm{f(x,y)} \ln\left(\frac{\mathrm{f(x)}}{\mathrm{f(x,y)}}\right) \mathrm{dxdy}. \tag{3-22}$$

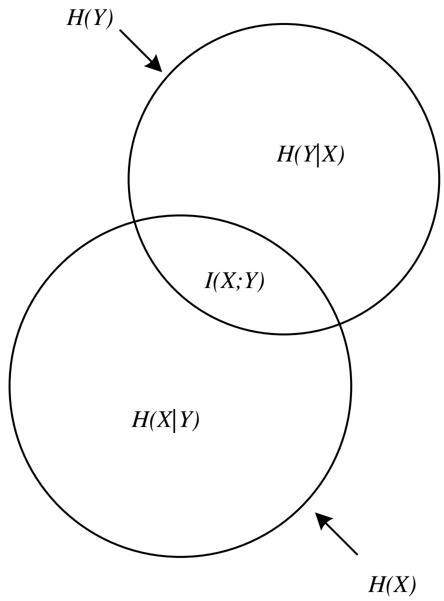


Figure 3.6: Venn diagram showing a relationship among mutual information and entropies of random variables X and Y.

Let us now consider the third random variable, Z. A part of a mutual information $I(Z;Y)$ which is not contained in X, $I(Z;Y|X)$, is called a partial mutual information and it is determined using the following expression:

$$\mathrm{I(Z;Y|X) = H(X,Z) + H(X,Y) - H(X) - H(X,Y,Z).} \tag{3-23}$$

Given X and the already reduced uncertainty $H(Y|X)$ shown in Figure 3.6, the PMI $I(Z;Y|X)$ is defined as the further reduction in uncertainty of the random variable Y that is gained by the additional mutual observation of the random variable Z.

Figure 3.7 depicts the dependence among PMI, individual and joint entropies of the random variables X, Y and Z. PMI is invariant under strictly monotonic transformations which makes it robust against possibly nonlinear distortions among random variables [19] and this is one of its most important advantages in relation to the linear correlation. However, a problem in determining a mutual information is that pdfs of the random variables have to be known. In practice, the real pdfs are not known and it is needed to estimate them. This topic is covered in the next subsection.

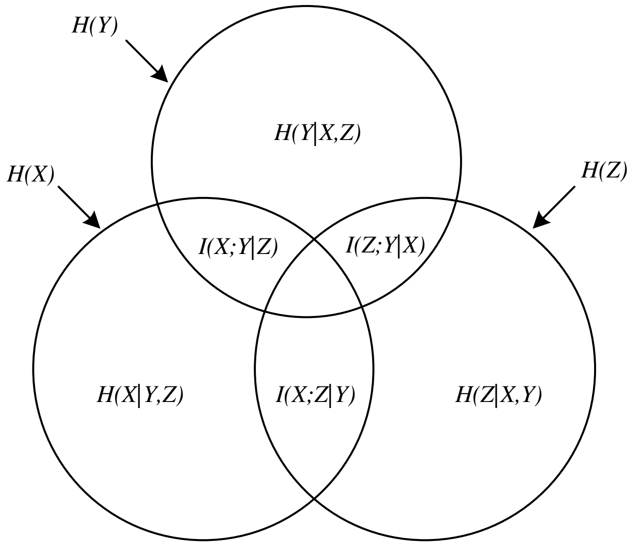


Figure 3.7: Venn diagram showing a relationship among partial mutual information and entropies of the random variables X, Y and Z.

PMI-based IVS algorithm is presented in [20]. Details of the algorithm are presented here:

Algorithm 1: Partial mutual information-based input variable selection

Input: output variable Y, potential input variables C
Result: chosen input variables X

Initialise $X \leftarrow \emptyset$
while $C \neq \emptyset$ **do**
  for each $c \in C$
    Estimate $I(c, Y|X)$
  Determine $c_s \in C$ that maximises $I(c, Y|X)$
  if *algorithm termination criterion is satisfied* **then**
    Stop running the algorithm
  Move c_s to X

In [16] a number of algorithm termination criteria are analysed. In this work a predefined number of the most relevant input variables was used as a termination criterion.

3.3.3.4. Estimating partial mutual information

Considering the expression (3-19) it can be seen that for estimating an entropy of the random variable, it is first required to determine its pdf which is estimated from the available historical data, i.e. from the considered random variable outcomes. There are two main approaches in estimating a pdf: *parametric* and *non-parameteric*.

The parametric approach assumes that data are drawn from a known parametric family of distributions, for example the normal distribution with mean μ and variance σ^2. Estimating the pdf then becomes a problem of estimating the parameters μ and σ^2. The non-parametric approach does not assume a form of the pdf, so non-parametric methods are usually much more robust and accurate than the parametric ones. A review of the most commonly used non-parametric estimation methods can be found in [21].

One of the most commonly used non-parametric pdf estimation methods is *kernel density estimation* and this method is proposed in [20] in the original version of Algorithm 1. However, this approach has some drawbacks -- apart from the fact that it is computationally very intensive and that it requires relatively large number of data samples for an accurate estimation, its behaviour is dependent on the kernel function parameters. This problem becomes even harder when a dimension of the random variable is increased [22]. Much more accurate and computationally less intensive pdf estimation method is *k-th nearest neighbour method*. The method in which an entropy of the random variable is directly determined is presented in [19] and it is described here.

Let us consider three continuous time series, $\{x_t\}$, $\{y_t\}$ and $\{z_t\}$, which represent the outcomes of random processes $\{X_t\}$, $\{Y_t\}$ and $\{Z_t\}$, respectively. For each vector $v_t \equiv \{x_t, y_t, z_t\}$, $t = 1,2,\cdots,N$ and a fixed integer k, $1 \leq k \ll N$, a distance $\varepsilon_k(t)$ to its k-th neighbour is defined. It means that a set $\{v_{t^*}\}$, where $t^* = 1,2,\cdots,N$, $t^* \neq t$, contains $k-1$ vectors with distances from v_t less than $\varepsilon_k(t)$ and $N-k-1$ vectors with the distance greater than $\varepsilon_k(t)$.

Therefore, for each t distance of v_t to each element of $\{v_{t^*}\}$ is determined:

$$\varepsilon(\mathrm{t}) = \{||\mathrm{v}_{\mathrm{t}^*} - \mathrm{v}_\mathrm{t}||\}. \tag{3-24}$$

This set is then sorted and distance $\varepsilon_k(t)$ is determined by selecting the k-th element of the sorted set. The distance is determined using max norm, i.e. $||\cdot|| = \max\{||\cdot||_x, ||\cdot||_y, ||\cdot||_z\}$, where $||\cdot||_x$, $||\cdot||_y$ and $||\cdot||_z$ can be any norm, but this algorithm suggests using max norm as well. Let us now define a vector $w_t \equiv \{x_t, z_t\}$, $t = 1,2,\cdots,N$.

For each t a number of vectors in $\{w_{t^*}\}$ with distances strictly less than $\varepsilon_k(t)$ is determined:

$$\mathrm{N}_{\mathrm{xz}}(\mathrm{t}) = \#\{\mathrm{t}^* \neq \mathrm{t};\ ||\mathrm{w}_{\mathrm{t}^*} - \mathrm{w}_\mathrm{t}|| < \varepsilon_\mathrm{k}(\mathrm{t})\}. \tag{3-25}$$

where # denotes a number of elements in the set. In a similar way $N_{xy}(t)$ and $N_x(t)$ are defined, for which w_t is defined using vectors $\{x_t, y_t\}$ and $\{x_t\}$, respectively. PMI is estimated using the following expression:

$$\hat{I}(Z;Y|X) = \frac{1}{N}\sum_{t=1}^{N}\left[h_{N_{xz}(t)} + h_{N_{xy}(t)} - h_{N_x(t)}\right] - h_{k-1}, \tag{3-26}$$

where h_n is the n-th negative harmonic number defined as $h_n = -\sum_{i=1}^{n} i^{-1}$ [19].

The k-th nearest neighbour method is computationally much faster than kernel methods are and, regardless of a number of considered variables dimension, it requires defining only one scalar parameter, k.

Here, we analyse the properties of the PMI estimator in case of the normal distribution for which PMI can be determined analytically, as shown in [19]. Multivariate normal distribution of the random vector $X \in R^n$ with mean $a \in R^n$ and covariance matrix $R \in R^{n\times n}$ is defined by its pdf:

$$\mathrm{f}(X) = \frac{1}{(2\pi)^{n/2}\sqrt{R}} exp\left(\frac{1}{2}(x-a)^T R^{-1}(x-a)\right), \tag{3-27}$$

and it is denoted as $X \sim \mathcal{N}_n(a, R)$ where $|R|$ denotes a determinant of the covariance matrix R. For n-dimensional normal distribution $\mathcal{N}_n(a, R)$ entropy is determined using the following expression:

$$H(X) = \frac{n}{2}(1 + ln2\pi) + \frac{1}{2} ln|R|. \tag{3-28}$$

3.3.4. Structure of the prediction model

This deliverable analyses an identification procedure for prediction models with time horizon of 12-36 hours. One of the main issues in developing such a multiple-output system is how to assess its performance, i.e.: how to define a criterion which will tell us if one model is better than the other. The response is trivial if each output of one model outperforms the corresponding output of the other model, but generally it is not the case. The simplest approach is to define a local criterion function for each output and a global criterion function could be e.g.: a sum of the local criterion functions. The first drawback of this approach is that we are usually more concerned about sooner prediction hours than about hours at the end of a prediction horizon, so we do not want to give the same weight to each local criterion function. An alternative is to use weighted sum of the local criterion functions as a global criterion, but a question of how to choose these weights remains open. The second drawback is that such a model has the same input vector which is used for describing input-output relationship for each output, which generally does not have to be the optimal choice. Certainly, developing a separate model for each output can at least perform as well as one model with multiple outputs. The first advantage of this approach is that defining a criterion function is trivial because for single-output models the local criterion corresponds to the global criterion. The second advantage is that such an approach does not necessarily imply a unique input vector for each model. The main drawback of this approach is that the whole developing process, including IVS, defining the optimal model structure and model calibration has to be carried out multiple times which can be computationally very intensive for a large prediction horizon. The concept of this approach is depicted in Figure 3.8.

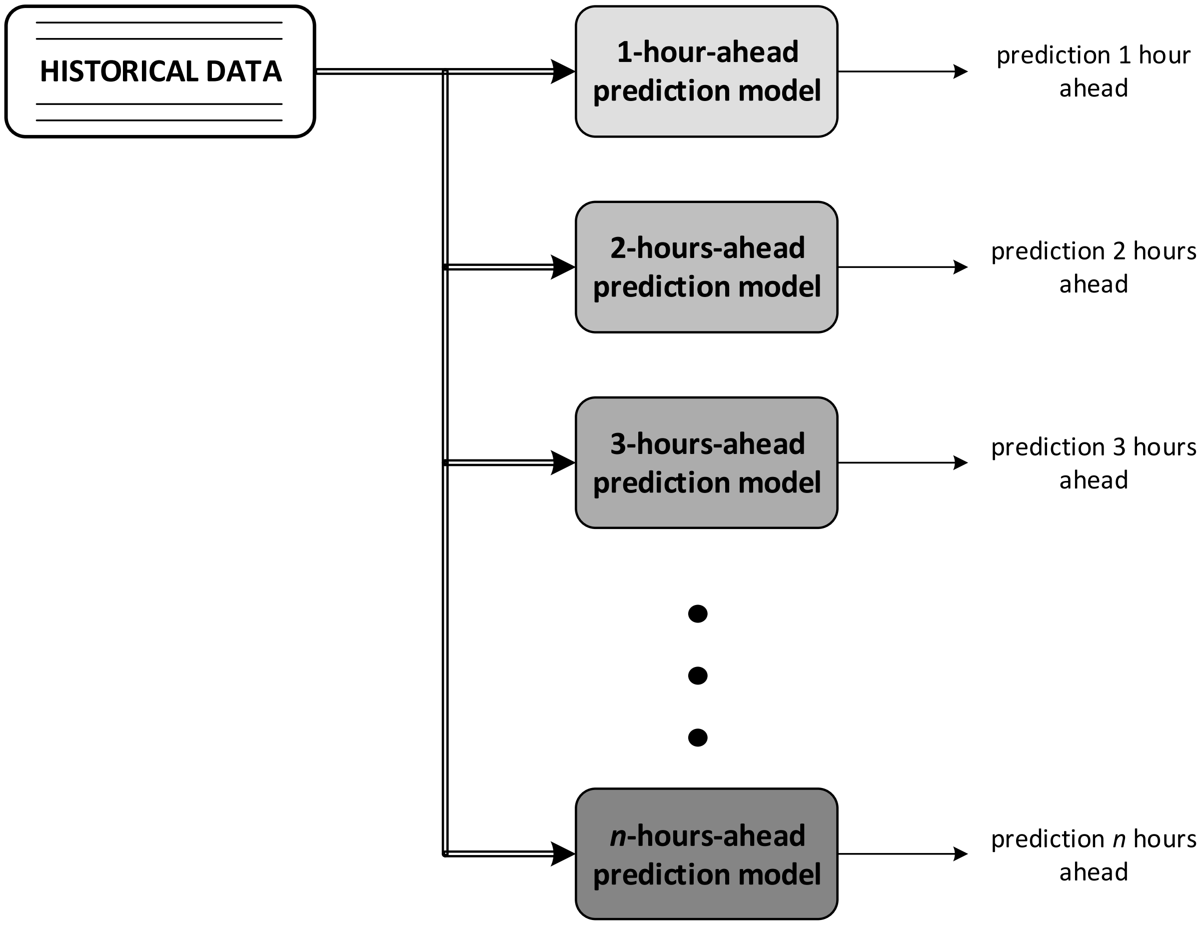


Figure 3.8: A static approach of the prediction system which uses a separate model for each system output.

Unlike the above-mentioned *static* approaches, the third approach uses the fact that the prediction system is considered as dynamic, i.e.: its output depends on past outputs. This *dynamic* approach is depicted in Figure 3.9. The main idea behind this approach is that the model does not have to use all the actual data, but also the provisional data, e.g.: output of the 1-hour-ahead model is a prediction for one hour ahead and this value can be used by the same model for predicting for two hours ahead. Analogously, this procedure can be repeated for obtaining the prediction for k hours ahead. It is expected that this approach will be less accurate than the one shown in Figure 3.8. because in this case a prediction error of the model is accumulated over the whole prediction horizon. However, if the performance of such an approach is not much worse than the one of the static approach, from the computational point of view, applying dynamic approach is much more efficient and contains significantly less parameters. Additionally, in some applications a larger prediction horizon may be required. Extension of the existing prediction model to a larger prediction horizon using dynamic approach is trivial; for the static approach this is not the case. Therefore, a dynamic approach is chosen for the prediction system.

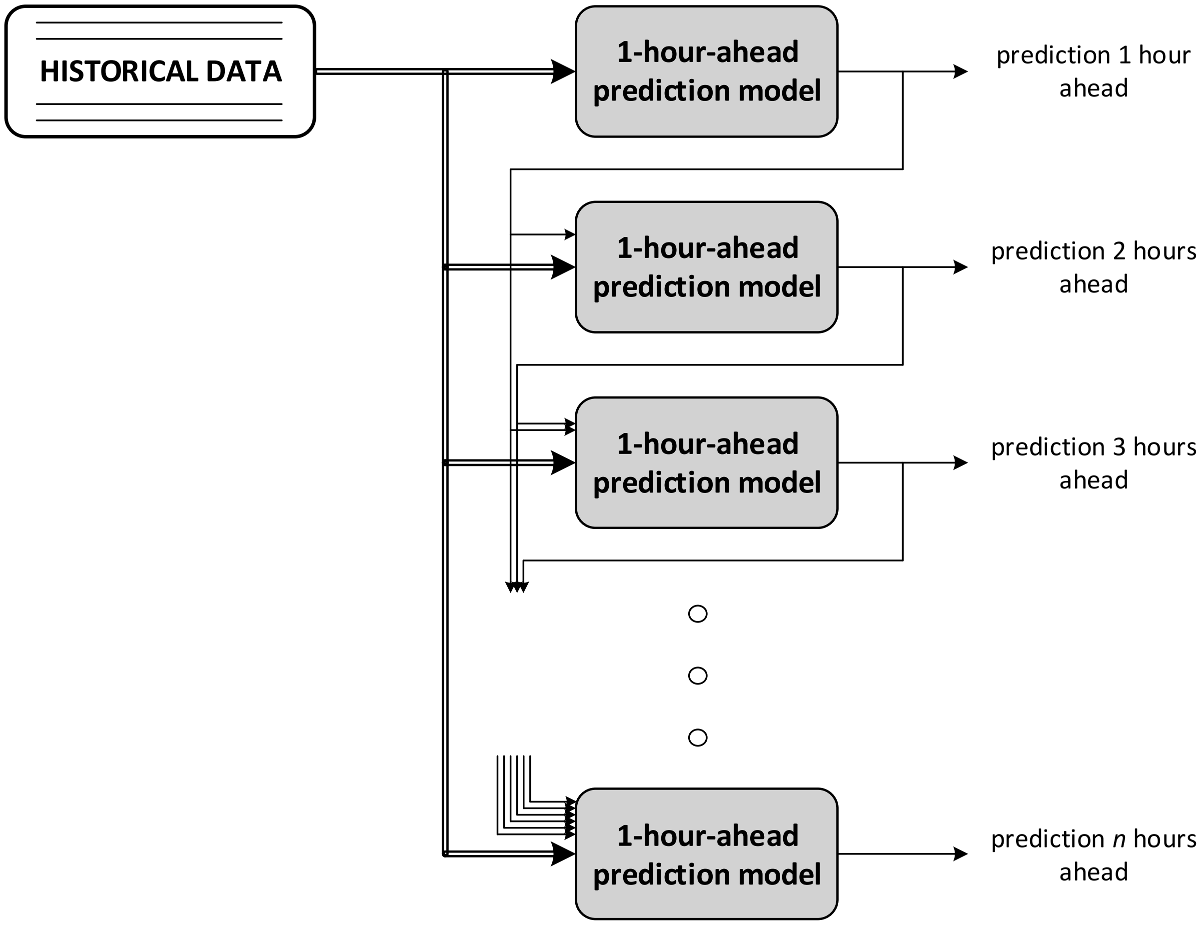


Figure 3.9: A dynamic approach of the prediction system which uses a single model for estimating system outputs for the whole prediction horizon.

3.3.4.1. Adaptive structure of the prediction system

It is often the case that historical data used for calibrating the prediction model do not cover the complete set of possible input-output vectors or that predicted variable values that occurred in past differs from values for the coming period due to factors which were not considered or did not have a significant impact on the variable during model calibration process. Occurrence of these factors can lead to poor predicting abilities of the existing prediction model. Therefore, for robust operation of the prediction system the model should be able to adapt to possible changes in the system.

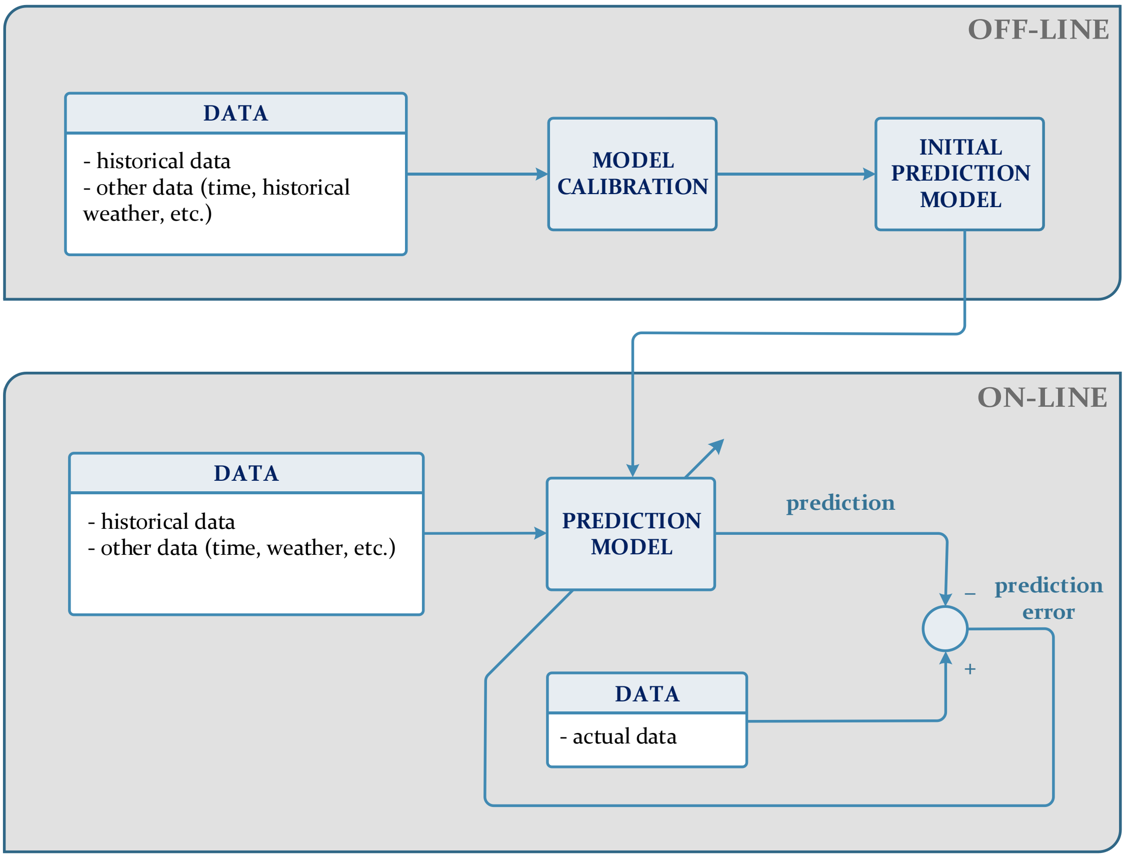


Figure 3.10: Adaptive module structure / a principle overview.

Modified structure of the prediction module is shown in Figure 3.10. The system is composed of two parts: *off-line* and *on-line*. In the off-line part historical data are used for obtaining the initial prediction model and this procedure is described in subsection 3.3.2. The on-line part of the module uses the initial model developed in the off-line part in order to generate predictions. When the data are available, they are compared to the corresponding predictions which results in the prediction error for the certain time instant. Model parameters are then tuned such that the prediction error is decreased. The presented procedure of using the feedback information on prediction accuracy for model parameters tuning introduces an adaptation ability to module.

3.3.4.2. Possible approaches to the on-line tuning of model parameters

Most real systems are time-variant. In order to track changes in the system, its model parameters should be continuously estimated. The on-line part of the prediction system, mentioned in the previous section, is the tool for continuous tuning of the model parameters such that the model tracks the actual predicted variable evolution as accurately as possible.

Artificial neural network (ANN) is a flexible model structure that can be easily and systematically calibrated and adapted. There is a large number of methods suggested in literature for the so called *recursive* neural network learning. Some of them are based on the recursive approximation of typical gradient methods [8], [23]. On the other hand, some recursive methods are based on the methodology for dynamic system state estimation [24]-[27]. These methods are based on the state-space representation of the ANN model [28]:

$$\mathrm{w_{k+1} = w_k + r_k,} \tag{3-29}$$
$$\mathrm{d_k = G(x_k, w_k) + e_k,} \tag{3-30}$$

where G is a function which defines the input-output mapping and is determined by the ANN structure, x_k is an input vector, w_k is a vector of ANN parameters and e_k is an error vector. In (3-29) a vector of parameters w_k corresponds to a stationary process with identity state matrix, driven by process noise r_k. ANN model written in this form enables using extended Kalman filter (EKF) or unscented Kalman filter (UKF) for the ANN parameter estimation. However, the ANN models with relatively large number of inputs and nodes in the hidden layer result in a large number of parameters, and applying EKF or UKF becomes intractable due to numerical stability issues [29]. On the other hand, recursive gradient methods for ANN learning are quite robust and their application is not limited to ANNs with a small number of parameters. Therefore, this approach in recursive ANN learning is analysed hereinafter.

3.3.4.3. Applying the on-line tuning procedure in normal operation

We use the prediction model developed within subsection 3.3.2 as an initial prediction model for the on-line part of prediction system (see Figure 3.10). Gradient descent method with momentum term is used for the recursive ANN learning. ANN parameters Θ are updated based on the following relation:

$$\mathrm{\Delta\Theta(k) = -\alpha\nabla\mathfrak{I}_v\big(\Theta(k)\big) + \gamma_m\Delta\Theta(k-1),} \tag{3-31}$$

where $\Delta\Theta(k) = \Theta(k+1) - \Theta(k)$, α is the learning coefficient, $\nabla\mathfrak{I}_v\big(\Theta(k)\big)$ is the gradient of local criterion function on the corresponding data set and γ_m is the nonnegative momentum term which speeds up the learning convergence while attenuating the parasitic oscillations [8]. If the parameter vector Θ is to be updated using more than one data sample, we consider two different learning styles: (i) *incremental learning* in which the model parameters are updated consecutively after each data sample is presented to the model; and (ii) *batch learning* in which the parameters are updated once after all the data samples are presented. The recursive ANN learning is performed using MATLAB® Neural Network Toolbox [30].

The on-line tuning parameters, learning coefficient α and momentum term γ_m can be determined based on the initial set of data that were used for obtaining the initial model. However, those data might not contain an evident variation in predicted variable, thus no significant difference in the performance of off-line and on-line model would be observed. Therefore, on-line tuning parameters can be determined based on the performance of on-line prediction model on the modified testing data – e.g. a linear trend is added to the original data such that predicted variable mean increases by 50% of the initial mean per month.

3.3.4.4. Concept of conditional adaptation (outliers handling)

In addition to the normal operation, another possible scenarios which affect the prediction system can occur. In the normal operation scenario we assumed that data do not contain potentially irregular or corrupted data samples (referred to as *outliers*). However, it is often the case that data on actual data are corrupted -- using these data samples within the on-line tuning procedure could cause an undesirable model behaviour. Instantaneous change in mean may be a result of many different external factors that influence the predicted variable, but it may also be caused by a meter problem – in the latter case data are characterised as corrupted.

The basic idea in avoiding the on-line tuning procedure using corrupted data is by marking those data, i.e. if a data sample is suspected to be an outlier, it is marked and that data sample will not be used in the on-line tuning procedure. In order to recognize an outlier occurrence, min/max values of the model inputs are used as boundaries for filtering the outliers.

4. Submodule for non-controllable consumption prediction (M.PE.4)

Submodule for prediction of the total non-controllable energy consumption on the microgrid level.

4.1. Submodule inputs

Table 4.1: Required inputs for non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Historical profile of the non-controllable energy consumption on the microgrid level	$E_{e,\mathrm{nc}}$ $E_{t,\mathrm{nc}}$	Non-controllable energy consumption on the microgrid level, electrical energy for all pilot sites except Idrija pilot sites where both electrical and thermal energy on the microgrid level are considered
Weather measurements	UNIZG-FER pilot site: $T_{\mathrm{env}}, I_{\mathrm{diff}}^{h}, I_{\mathrm{dir}}^{\mathrm{n}}$ Remaining pilot sites: $T_{\mathrm{env}}, I_{\mathrm{glo}}^{h}, I_{\mathrm{glo}}^{\mathrm{t}}$	Measured weather variables: temperature, diffuse horizontal and direct normal irradiance (UNIZG-FER site), global horizontal and tilted global irradiance (remaining sites).
Weather predictions	$(T_{\mathrm{env}})_{\mathrm{N}}, (I_{\mathrm{dir}}^{\mathrm{n}})_{\mathrm{N}}, (I_{\mathrm{diff}}^{\mathrm{h}})_{\mathrm{N}}$	Forecasted weather variables (temperature, direct normal and diffuse horizontal irradiance).

Time indicators	τ	Variables representing time of the day, time of the week and day of the year. Calculated from current and historical datetimes.

4.1.1. Non-controllable thermal energy consumption

Non-controllable thermal energy consumption on the microgrid level represents the microgrid level electrical (and thermal in case of Idirija pilots) energy consumption that is not controlled by the microgrid MPC modules. It is measured/calculated differently on each considered pilot site depending on the configuration of the pilot site microgrid level and available measurement equipment present on the site. Determination of the non-controllable electrical and thermal energy consumption for different pilot sites is presented in the following Table 4.2.

Table 4.2 Non-controllable electrical and thermal energy consumption determination.

Pilot site	Electrical energy	Thermal energy
UNIZG-FER	Overall consumption (energy exchange) of the building - electricity consumption on fan coils - electricity consumption of the chiller - electricity consumption of the circulation pump of the heat exchanger - energy exchange with the battery system + production of the photovoltaic system	
HEP		
Idrija (school building)	Overall electrical energy consumption of the building	Heat exported from DHW tank to building
Idrija (sports centre building)	Overall electrical energy consumption of the building	Heat exported from DHW tank to building
EON	Overall electrical energy consumption of the building - electrical energy consumption of heat pumps (4 water chillers) - electrical energy consumption of electric heaters + electrical energy production of the PV system	

	- el. energy consumption of fan coils (4 heating/cooling circuits)	
STREM (school building)	Consumption on the central electric meter - single fan coil consumption (z.pe.1 output)	
STREM (retirement care centre)	Consumption on the central meter - load of the cooling machine (HVAC.MPC.2 cooling) - battery power	
EPHZHB	Overall electrical energy consumption of the building - electrical energy consumption of fan-coils - electrical energy consumption of heat pump - electrical energy consumption of electric boiler	

4.1.2 Solar irradiance data

Depending on the availability of solar irradiance measurements on different pilot sites throughout the project, two separate sets of weather measurements inputs are used.

On the UNIZG-FER pilot site, where direct normal and diffuse horizontal irradiance measurements are available, they are used as submodule inputs and paired with the same forecasted variables during submodule operation.

Due to high costs of direct and diffuse irradiance sensors other pilot sites provide measurements of global horizontal and tilted global irradiations which are then used as submodule inputs. Since measured and forecasted irradiances are now different, during submodule operation, forecasted direct and diffuse irradiance, solar angles (obtained through the use of Pysolar python library), geographical pilot site data and current datetime, are used for calculation of global horizontal and tilted global irradiances thus matching the measured and forecasted irradiance variables.

4.2 Submodule outputs

Table 4.2: Outputs of the non-controllable consumption prediction submodule.

Variable name	Variable annotation	Variable description
Prediction model parameters (for off-line operation of the	θ_{ug}	Needed for on-line operation of the submodule.

submodule)		
Predicted profile of the non-controllable electricity consumption, predicted profile of the non-controllable thermal load (for on-line operation of the submodule)	$(E_{e,nc})_{\mathrm{N}}$ $(E_{t,nc})_{\mathrm{N}}$	Needed for the MPC module on the microgrid level

Since the non-controllable consumption model is based on artificial neural networks (as presented in the following section 4.3) the predictions sometimes tend to reach impossible values, e.g. slightly negative electrical energy values during the night when there is no PV panels production. The structure of neural networks prohibits the model to incorporate exact boundaries on the model outputs, therefor all generated predictions are post-processed in order to avoid such unexpected values.

4.3 Methodology

Non-controllable electrical and thermal energy consumption on the microgrid level are modeled using artificial neural networks with the same procedure as described for the M.PE.3 module, in Section 3.3.

5 Submodule for identification of controllable load parameters (M.PE.6)

5.1 Theory

Controllable loads include the following units:

- hot water boilers,
- water chillers.

Both units can be described as a simple heat storage system with one heat tank. For all cases the medium storage can be considered of a fixed volume. The energy content of the storage is indicated by temperature of the medium.

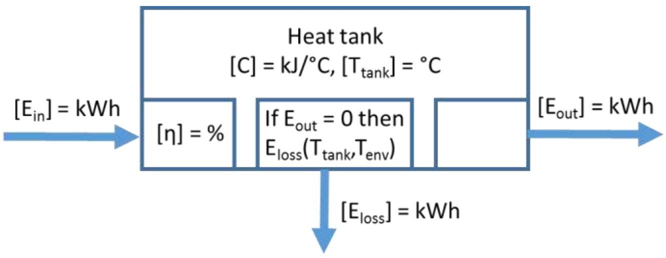


Figure 5.1: Simple physical model of controllable loads

Water boiler: η ~ 100% Water boiler: $E_{out} \neq 0$

Water chiller: η > 100% (COP) Water chiller: $E_{out} \neq 0$

5.2 Inputs

- Historical controllable load measured state, x_l
  - $[T_{heatstorage}]$ = °C
- Historical controllable load energy input (electricity), E_{in}
  - $[E_{in}]$ = kWh
- Historical controllable load energy output, E_{out}
  - $[E_{out}]$ = kWh
- Room temperature measurement, T_{env}
  - $[T_{env}]$ = °C
- Parameter of the COP model of the chiller (if relevant), COP

Resolution: 1/min

The temperature of the heat tank is:

- buffer tank temperature measured at middle height of the tank in case of the water chiller
- equivalent temperature calculated based on three measurement (top, middle, bottom) in case of water boiler.

5.3 Outputs

- Simplified model of the controllable load, θ_l
  - Specific heat for the heat tank
    - $[C_{ht}]$ = kWh / °C
  - Energy loss per second in the function of temperature difference of the heat tank and its environment
    - $[P_{loss}(\Delta T)]$ = W/°C

5.4 Internal parameters

Internal parameters:

- stepping - dimensionless
  - Some controllable loads have more than one input power levels, e.g. one room with four individually controlled heaters. Stepping means the number of the possible power levels for the controllable loads. (supposing equidistant power levels).

- $[P_{stepping}]$ = kW
  - The power value for one equidistant power level.
- $[T_{plus\_physical}]$ = °C – absolute
  - Absolute maximum temperature level for the unit (physical limit or operation requirement).
- $[T_{minus\_physical}]$ = °C – absolute
  - Absolute minimum temperature level for the unit (physical limit or operation requirement).
- type = {chiller, boiler}
  - Type of the controllable load.

5.5 Frequency of submodule calls

The submodule is planned to be called once per year.

5.6 Algorithm

The operation of this module has the purpose of periodically recalibrating several parameters of these systems. After a longer period (e.g. yearly) P_{loss} has to be recalculated based on historical data.

5.6.1 Initial parameter and output setup

All initial values must be entered manually or their determination requires a manually controlled identification process.

Parameter	Water chiller	Water heater
$T_{plus\_physical}$	comfort limit based on the unit's documentation	based on the unit's documentation
$T_{minus\_physical}$	freeze limit based on the unit documentation	based on the unit's documentation
stepping	based on the unit documentation	based on the unit documentation
$P_{stepping}$	based on the unit documentation	based on the unit documentation
C_{ht}	volume of the tank (m^3) * density of water (1000 kg/m^3) * specific heat of water (4.2 kJ/kg°C)	volume of the tank (m^3) * density of water (1000 kg/m^3) * specific heat of water (4.2 kJ/kg°C)
$P_{in,min}$	based on the unit's documentation	based on the unit's documentation
$P_{in,max}$	based on the unit's documentation	based on the unit's documentation
P_{loss}	identification process	identification process

5.6.2 Identification process

Identification procedure for P_{loss}:

1. search the historical data for idle periods (no input and no output)
2. check the idle periods for one hour long intervals with different ΔTs
3. during the idle period:
    a. calculate the average temperature difference between the heat tank and the environment –> ΔT
    b. calculate the temperature change during that idle period –> $\Delta E = C_{ht}(T_2\text{-}T_1)$
    c. calculate the time interval for the idle period –> t_{idle}
4. $P_{loss}(\Delta T)= \Delta E/ t_{idle}$
5. based on the losses at different ΔTs one can create a polynomial function to more precisely describe the losses

6 Submodule for domestic hot water energy consumption prediction (M.PE.5)

6.1 Introduction

Domestic hot water tank is controllable load. Water in tank is heated from two sources: immersed tubular heat exchanger and electric water heater. This way water can be heated from two different energy sources. Heat to exchanger is supplied from microgrid boiler room. Energy transferred is measured with heat meter. Heat transfer can be switched on or off with switching the circulation pump.

Electric heater which can heat water with electric energy. Electric heater has three power stages which are switched independently from each other.

- Stage 1: 15 kW
- Stage 2: 15 kW
- Stage 3: 5 kW
- Total: 35 kW

Consumed energy is measured with electric meter.

Each heat source is described with mathematical model. State of the energy accumulated in tank is described with *equivalent temperature*.

The physical model, algorithm and identification procedure is explained and defined in chapter: Submodule for identification of controllable load parameters (M.PE.6).

6.2 Inputs

- Efficiency of electric water heater is: η_{el} ~ 100%.
- Efficiency of heat exchanger was measured and is: η_{ex} ~ 90%.
- Historical energy input (electricity), $[E_{in\_el}]$ = kWh
- Historical thermal energy input, $[E_{in\_th}]$ = kWh
- Historical energy output – hot water consumption, $[E_{out}]$ = kWh
- Room temperature measurement, T_{env} is constant: ~ 20 °C
- Temperature at height 1: $[T_{DHW\_1}]$ = °C
- Temperature at height 2: $[T_{DHW\_2}]$ = °C
- Temperature at height 3: $[T_{DHW\_3}]$ = °C
- Domestic hot water state: $[DHW_{state}]$ = dimensionless

Energy consumption is calculated from measured water flow

Resolution: 15 min

6.3 Outputs

- The state variable of DHW tank - historical equivalent temperature, $[T_{DHW}]$ = °C
- Simplified model of the DHW tank, θ_{DHW}
- Constraints:
    a. Minimum and maximum allowed temperature values
    b. Maximum energy exchanged using the heat exchanger: P_{max_th} = kWh
    c. Maximum energy exchanged using the heaters (combined): P_{max_el} = kWh
    d. Antilegionella schedule

Legionella protection will be done as a temporary rise of DHW setpoint temperature. This shall be done weekly according to provided schedule.

Energy Balance

The well-mixed assumption implies that all water in the tank is at the same temperature. To calculate the water temperature, the model analytically solves the differential equation governing the energy balance of the water tank:

$$mc_p\frac{dT}{dt} = q_{net}$$

c_p = specific heat of water

T = temperature of the tank water

t = time

q_{net} = net heat transfer rate to the tank water

The net heat transfer rate is the sum of gains and losses due to multiple heat transfer pathways. The imported and exported energy is measured with calorimeter. The losses linear depended of equivalent temperature of the tank and environment temperature.

$$Q_{in} - Q_{out} = m*c*_{d}T$$

$$Q_{in} = Q_{exch} + Q_{heater} - (Q_{loss} + Q_{load}) = m*c*_{d}T$$

Domestic hot water consumed is measured with three flow meters. The sum of energy is calculated from measured flow and temperature of water at the top of tank:

$$Q_{load} = q*c*(T_{DHW\_3} - T_{fresh})$$

T_{fresh} – Fresh tap water at constant temeperature of 15 °C

Model

DHW tank simplified model is linear and incorporates heat losses of the tank.

Energy loss: $[P_{loss}(\Delta T)] = W/°C$

Energy loss per second in the function of temperature difference of the heat tank and its environment:

$$P_{loss}(\Delta T)= \Delta E/ t_{idle}$$

The model is stored in MySQL database in JSON format:

{‘A’:’[0.0025]’; ‘B’:’[1,1,-1]’}

The heat exchanger

Having maximum power for the current conditions (supply temperature, water temperature in the tank...) is not enough. The algorithm needs the formula and coefficients, so maximum heat exchange can be calculated for any conditions on the prdiction horizon.

$$Q_{exch} = k*q*(T_{supply} - T_{DHW})$$

Bibliography

[1] M.A. Hannan, M.S.H. Lipu, A. Hussain, A. Mohamed (2017), A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. Renewable and Sustainable Energy Reviews. 78. 834-854.

[2] Shugang Jiang (2011), A Parameter Identification Method for a Battery Equivalent Circuit Model

[3] Thomas R. B. Grandjean, Andrew McGordon, Paul A. Jennings (2017), Structural Identifiability of Equivalent Circuit Models for Li-Ion Batteries. Energies, vol. 10

[4] Dr. Gregory L. Plett (2015), Lecture notes and recordings for ECE5720: Battery Management and Control

[5] Joakim Andersson (2017), Lifetime estimation of lithium-ion batteries for stationary energy storage systems

[6] Wen-Yeau Chang (2013), The State of Charge Estimating Methods for Battery: A Review. ISRN Applied Mathematics, vol. 2013

[7] Ines Baccouche, Asma Mlayah, Sabeur Jemmali, Bilal Manai, Najaoua Essoukri Ben Amara (2015), Implementation of a Coulomb Counting Algorithm for SoC estimation of Li-Ion Battery for Multimedia Applications. CoRR, vol. abs/1803.10654

[8] I. Petrović, M. Baotić, and N. Perić, Inteligentni sustavi upravljanja: Neuronske mreže, evolucijski i genetički algoritmi. University of Zagreb Faculty of Electrical Engineering and Computing, 2011.

[9] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133, dec 1943.

[10] W. Pitts and W. S. McCulloch, "How we know universals: The perception of auditory and visual forms," Bulletin of Mathematical Biophysics, vol. 9, pp. 127–147, 1947.

[11] K. Levenberg, "A method for the solution of certain problems in least squares," Quart. Applied Math., vol. 2, pp. 164–168, 1944.

[12] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," SIAM Journal on Applied Mathematics, vol. 11, no. 2, pp. 431–441, 1963.

[13] H. R. Maier and G. C. Dandy, "Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications," Environmental Modelling & Software, vol. 15, no. 1, pp. 101 – 124, 2000.

[14] H. R. Maier, A. Jain, G. C. Dandy, and K. Sudheer, “Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions,” Environmental Modelling & Software, vol. 25, no. 8, pp. 891 – 909, 2010.

[15] R. May, H. Maier, and G. Dandy, “Data splitting for artificial neural networks using SOM based stratified sampling,” Neural Networks, vol. 23, no. 2, pp. 283 – 294, 2010.

[16] R. J. May, H. R. Maier, G. C. Dandy, and T. G. Fernando, “Non-linear variable selection for artificial neural networks using partial mutual information,” Environmental Modelling & Software, vol. 23, no. 10–11, pp. 1312 – 1326, 2008.

[17] K. Suzuki, Artificial Neural Networks - Methodological Advances and Biomedical Applications. InTech, 2011.

[18] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” Phys. Rev. E, vol. 69, p. 066138, Jun 2004.

[19] S. Frenzel and B. Pompe, “Partial mutual information for coupling analysis of multivariate time series,” Phys. Rev. Lett., vol. 99, p. 204101, Nov 2007.

[20] A. Sharma, “Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — A strategy for system predictor identification,” Journal of Hydrology, vol. 239, no. 1–4, pp. 232 – 239, 2000.

[21] B.W. Silverman and P. J. Green, Density Estimation for Statistics and Data Analysis. London: Chapman and Hall, 1986.

[22] T. Duong and M. L. Hazelton, “Cross-validation bandwidth matrices for multivariate kernel density estimation,” Scandinavian Journal of Statistics, vol. 32, no. 3, pp. 485–506, 2005.

[23] L. Ngia and J. Sjoberg, “Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm,” IEEE Transactions on Signal Processing, vol. 48, no. 7, pp. 1915–1927, 2000.

[24] I. Rivals and L. Personnaz, “A recursive algorithm based on the extended Kalman filter for the training of feedforward neural models,” Neurocomputing, vol. 20, no. 1–3, pp. 279–294, 1998.

[25] X.Wu and Y.Wang, “Extended and Unscented Kalman filtering based feedforward neural networks for time series prediction,” Applied Mathematical Modelling, vol. 36, no. 3, pp. 1123–1131, 2012.

[26] R. Zhan and J.Wan, “Neural network-aided adaptive unscented Kalman filter for nonlinear state estimation,” IEEE Signal Processing Letters, vol. 13, no. 7, pp. 445–448, 2006.

[27] R. Van der Merwe and E. Wan, “The square-root unscented Kalman filter for state and parameter-estimation,” in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 6, pp. 3461–3464, 2001.

[28] S. Haykin, Kalman Filtering and Neural Networks. Adaptive and Learning Systems for Signal Processing, Communications and Control Series, Wiley, 2004.

[29] L. Padilla and C. Rowley, “An adaptive-covariance-rank algorithm for the unscented Kalman filter,” in Proceedings of the IEEE Conference on Decision and Control, pp. 1324–1329, 2010.

[30] MATLAB, version 8.1.0 (R2013a). Natick, Massachusetts: The MathWorks Inc., 2013.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Model predictive control module for microgrid management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER)
Authors (Partners)	Danko Marušić, Mario Vašak (UNIZGFER)
Contact person	Danko Marušić (UNIZGFER)
Abstract (for dissemination)	The deliverable outlines the model predictive control module on the building microgrid level for hierarchical management of building subsystems. It is focussed on the input-output interfaces of the module. The internal logic of the module is presented in more detail in the annexed document.
Keyword List	Controllable Storage, Controllable Source, Controllable Load, Model Predictive Control, Energy Exchange Command, Grid Interaction, Demand Response, Flexibility Contracting

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.2.1	Mario Vašak (UNIZGFER)
v0.2	15 January 2019	Updated version	Mario Vašak (UNIZGFER)
v0.3	17 June 2019	Updated version	Mario Vašak (UNIZGFER)
v0.4	26 June 2019	Updated version	Mario Vašak, Danko Marušić (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of modules.

The modules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on microgrid level model predictive control.

The microgrid level model predictive control module is presented via corresponding interfacing tables that explain what data are used by it as inputs and what are its output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following model predictive control module is designed, commissioned and tested on the microgrid level:

M.MPC.1 – module for model predictive control that in off-line operation interacts with other MPC modules and decides on building optimal daily operation and the contracted amount of building flexibility, and in on-line operation decides on the controllable generation, storage or loading entities actuation on the microgrid level as well as on controllable electricity consumption pricing on the lower levels.

In the following chapter the module is presented with its interface tables showing which data it uses as inputs and which data it provide as outputs to be at the disposal to other submodules. Detailed explanations of algorithms behind it are provided in Annex 1. It inherits the developments presented in the previously delivered 3Smart document D4.2.1 (related to model predictive control) and additionally shows its further improvements based on feedback from the pilot buildings.

Source and sink for the data used by the module is a properly structured 3Smart database. Its structure in the part concerned by the module is provided in Annex 2.

2 M.MPC.1 modules

M.MPC.1 module is used for model predictive control that in off-line operation interacts with other MPC modules and decides on the optimal building daily operation and the contracted amount of building flexibility, and in on-line operation decides on the controllable generation, storage or loading entities actuation on the microgrid level again in interaction with lower-level MPC modules and in interaction with the short-term grid-level modules.

M.MPC.1 module is tested in all pilot buildings of 3Smart where it may be constituted considerably differently based on available control points and based on the fact whether only controllable electricity consumption from HVAC or both controllable electricity and heat affect the microgrid level operation.

In all cases M.MPC.1 interacts in off-line operation with the central HVAC system off-line MPC (HVAC.MPC.1 or HVAC.MPC.2) and with the long-term grid-side modules (see D5.4.3), while in on-line operation it interacts with the central HVAC system on-line MPC and with the short-term grid-side modules.

On UNIZGFER building M.MPC.1 controls the battery storage system, the same holds for HEP, Strem retirement and care centre and EPHZHB buildings. In Idrija M.MPC.1 controls the CHP unit and the heating in domestic hot water tank by heat and electricity. In Idrija M.MPC.1 receives predicted controllable heat from the central HVAC level, unlike all other sites where the central HVAC level provides the predicted controllable electricity consumption to M.MPC.1. In Strem school M.MPC.1 does not control any device, just interconnects the prices and demand response conditions from the grid modules towards the central HVAC MPC. In EON building M.MPC.1 controls power production from the photovoltaic system and also the electric heaters in several rooms.

The module interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Model predictive control module for microgrid energy flows control

Variable name	Variable annotation	Variable description
Module inputs		
Cumulative predicted controllable energy consumption that needs to be served by the microgrid; electricity and heat	E_L, E_H	Energy inputs optimized within the HVAC level MPC
Grid price conditions	c_{DA}	Prices and conditions obtained from the distribution grid/grids
Prediction of the overall non-controllable electricity/heat/gas consumption profile	$E_{L,nc}$, $E_{H,nc}$, $E_{gas,nc}$	
Prediction of the overall controllable generation profile of electricity/heat	$E_{L,c}$, $E_{H,c}$	
Parameters of the battery/batteries model, if exist	θ_{BAT}	
Battery/batteries state of charge, if exists	SoC	
Internal states of controllable loads, if exist (e.g., representative temperature in the domestic hot water boiler or in electric heaters controllable room)	x	
Controllable loads state limits, if exist (e.g., upper and lower limit of the domestic hot water boiler or in electric heaters controllable room)	x_{min}, x_{max}	
Controllable loads energy conversion model, if exist (e.g. power to heat model in the domestic hot water boiler or refrigerator)	θ_{cont}	
State of the heat storage, if exists (e.g., representative temperature of the heat tank)	T_{hs}	
Parameters of the heat storage model	θ_{hs}	
Parameters of the energy conversion units model, if exist (e.g., CHP)	θ_{ec}	
Prediction of usage conditions of controllable loads, if applicable (e.g. temperature of the cold water supply in the domestic hot water tank and profile of hot water usage)	d_{cont}	
Prediction of maximum energy inputs from the photovoltaic system	$P_{PV,max}$	Obligatory needed when power production from the photovoltaic system is

		controllable, otherwise can be merged with non-controllable electricity consumption
Maximum power price	c_{Pmax}	
Day-ahead prices	c_{DA}	
Intra-day prices	$c_{ID,\mathrm{penal}}, c_{ID,\mathrm{incent}}$	If not provided along the entire prediction horizon, considered to be equal to 1.2c_DA
Storage degradation price	c_{Batt}	
Table 2.2 – Module outputs – should be separate table		
Module outputs		
Profile of the energy exchange with battery/batteries storage	E_{BAT}	The value valid for the first sampling period is to be transmitted to the interface module
Profile of the energy command for controllable load actuation	$E_{L,c}^{*}$	For controllable loads continuous energy commands are generated which are then transferred to on-off on the interface submodule
Profile of the energy conversion units actuation	E_{ecu}	E.g., CHP command for heat generation with electricity as the side-product or vice versa
Local characterization of the value function of the optimization around the planned profile of controllable loads E^{EL}, E^{H}	$J^{*}(E_L, E_H)$	This local characterization is transmitted back to the zone and central HVAC level (constitutes of price coefficients and polytopic localization)
Energy exchange profiles with the grid and other data for the grid: electrical energy, heat energy, gas energy, respectively	E_G, E_G^H, E_G^{gas} other conditions	Conditions obtained through optimization of the interaction with distribution grid/grids

Bibliography

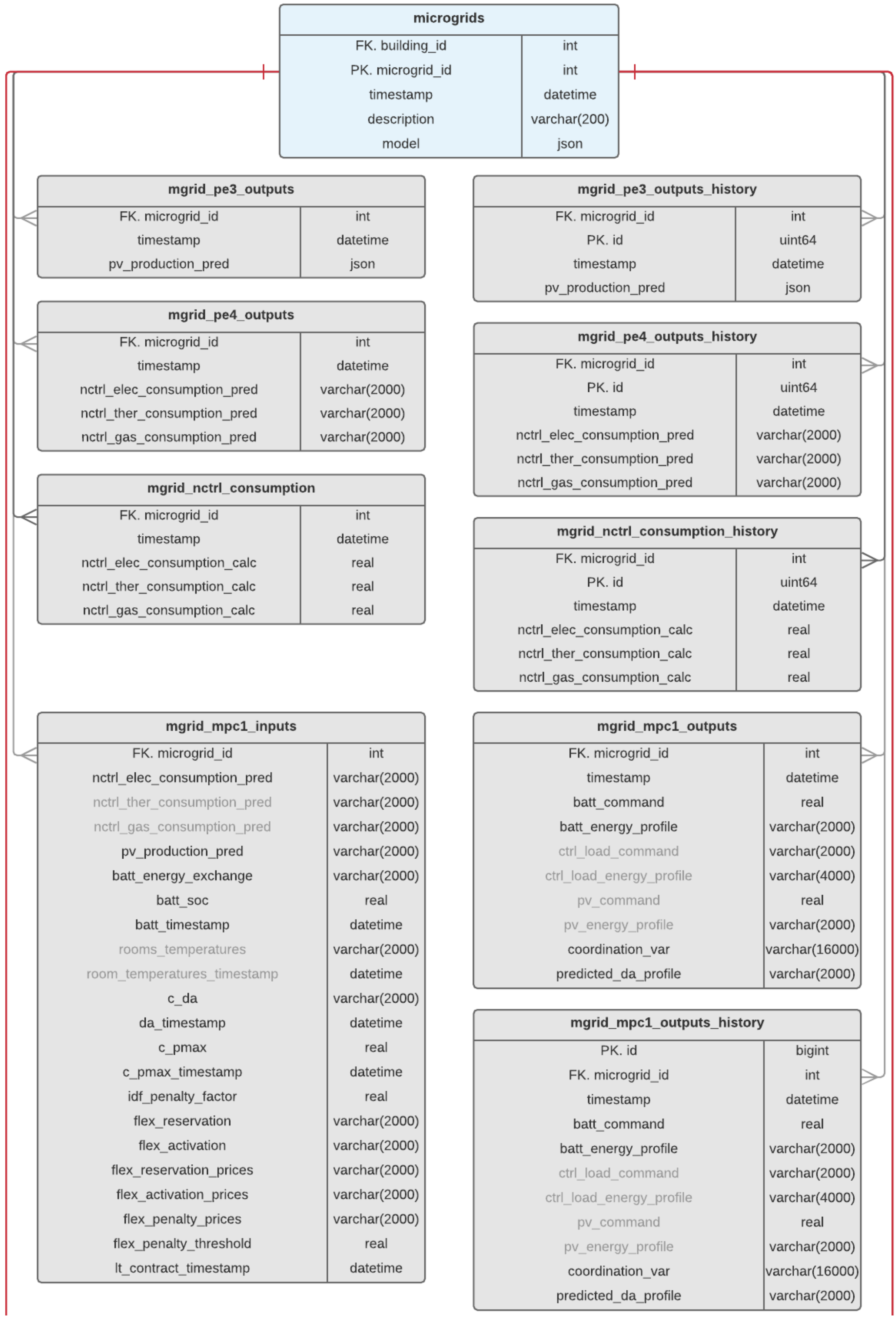
[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

[2] 3Smart D5.4.3. Final grid-side energy management software module. June 2019.

Annex 1 -- Open software module for microgrid level consumption management – Model predictive control module

Provided as a separate document.

Annex 2 -- 3Smart database organization for open software module for the microgrid level management – Model predictive control module


microgrids
FK. building_id int
PK. microgrid_id int
timestamp datetime
description varchar(200)
model json
mgrid_pe3_outputs
FK. microgrid_id int
timestamp datetime
pv_production_pred json
mgrid_pe3_outputs_history
FK. microgrid_id int
PK. id uint64
timestamp datetime
pv_production_pred json
mgrid_pe4_outputs
FK. microgrid_id int
timestamp datetime
nctrl_elec_consumption_pred varchar(2000)
nctrl_ther_consumption_pred varchar(2000)
nctrl_gas_consumption_pred varchar(2000)
mgrid_pe4_outputs_history
FK. microgrid_id int
PK. id uint64
timestamp datetime
nctrl_elec_consumption_pred varchar(2000)
nctrl_ther_consumption_pred varchar(2000)
nctrl_gas_consumption_pred varchar(2000)
mgrid_nctrl_consumption
FK. microgrid_id int
timestamp datetime
nctrl_elec_consumption_calc real
nctrl_ther_consumption_calc real
nctrl_gas_consumption_calc real
mgrid_nctrl_consumption_history
FK. microgrid_id int
PK. id uint64
timestamp datetime
nctrl_elec_consumption_calc real
nctrl_ther_consumption_calc real
nctrl_gas_consumption_calc real
mgrid_mpc1_inputs
FK. microgrid_id int
nctrl_elec_consumption_pred varchar(2000)
nctrl_ther_consumption_pred varchar(2000)
nctrl_gas_consumption_pred varchar(2000)
pv_production_pred varchar(2000)
batt_energy_exchange varchar(2000)
batt_soc real
batt_timestamp datetime
rooms_temperatures varchar(2000)
room_temperatures_timestamp datetime
c_da varchar(2000)
da_timestamp datetime
c_pmax real
c_pmax_timestamp datetime
idf_penalty_factor real
flex_reservation varchar(2000)
flex_activation varchar(2000)
flex_reservation_prices varchar(2000)
flex_activation_prices varchar(2000)
flex_penalty_prices varchar(2000)
flex_penalty_threshold real
lt_contract_timestamp datetime
mgrid_mpc1_outputs
FK. microgrid_id int
timestamp datetime
batt_command real
batt_energy_profile varchar(2000)
ctrl_load_command varchar(2000)
ctrl_load_energy_profile varchar(4000)
pv_command real
pv_energy_profile varchar(2000)
coordination_var varchar(16000)
predicted_da_profile varchar(2000)
mgrid_mpc1_outputs_history
PK. id bigint
FK. microgrid_id int
timestamp datetime
batt_command real
batt_energy_profile varchar(2000)
ctrl_load_command varchar(2000)
ctrl_load_energy_profile varchar(4000)
pv_command real
pv_energy_profile varchar(2000)
coordination_var varchar(16000)
predicted_da_profile varchar(2000)

mgrid_mpc1_inf_da_inputs	
FK. microgrid_id	int
nctrl_elec_consumption_pred	varchar(2000)
nctrl_ther_consumption_pred	varchar(2000)
nctrl_gas_consumption_pred	varchar(2000)
pv_production_pred	varchar(2000)
batt_energy_exchange	varchar(2000)
batt_soc	real
batt_timestamp	datetime
rooms_temperatures	varchar(2000)
room_temperatures_timestamp	datetime
c_da	varchar(2000)
da_timestamp	datetime
c_pmax	real
c_pmax_timestamp	datetime
idf_penalty_factor	real
flex_reservation	varchar(2000)
flex_activation	varchar(2000)
flex_reservation_prices	varchar(2000)
flex_activation_prices	varchar(2000)
flex_penalty_prices	varchar(2000)
flex_penalty_threshold	real
lt_contract_timestamp	datetime

mgrid_mpc1_inf_da_outputs	
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
informative_da_profile	varchar(2000)

mgrid_mpc1_inf_da_outputs_history	
PK. id	bigint
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
informative_da_profile	varchar(2000)

mgrid_mpc1_dec_da_inputs	
FK. microgrid_id	int
nctrl_elec_consumption_pred	varchar(2000)
nctrl_ther_consumption_pred	varchar(2000)
nctrl_gas_consumption_pred	varchar(2000)
pv_production_pred	varchar(2000)
batt_energy_exchange	varchar(2000)
batt_soc	real
batt_timestamp	datetime
rooms_temperatures	varchar(2000)
room_temperatures_timestamp	datetime
c_da	varchar(2000)
da_timestamp	datetime
c_pmax	real
c_pmax_timestamp	datetime
idf_penalty_factor	real
flex_reservation	varchar(2000)
flex_activation	varchar(2000)
flex_reservation_prices	varchar(2000)
flex_activation_prices	varchar(2000)
flex_penalty_prices	varchar(2000)
flex_penalty_threshold	real
lt_contract_timestamp	datetime

mgrid_mpc1_dec_da_outputs	
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
declared_da_profile	varchar(2000)

mgrid_mpc1_dec_da_outputs_history	
PK. id	bigint
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
declared_da_profile	varchar(2000)

mgrid_mpc1_lt_inputs	
FK. microgrid_id	int
nctrl_elec_consumption_pred	varchar(2000)
nctrl_ther_consumption_pred	varchar(2000)
nctrl_gas_consumption_pred	varchar(2000)
pv_production_pred	varchar(2000)
c_da	varchar(2000)
da_timestamp	datetime
c_pmax	real
c_pmax_timestamp	datetime
idf_penalty_factor	real
flex_reservation	varchar(2000)
flex_activation	varchar(2000)
flex_reservation_prices	varchar(2000)
flex_activation_prices	varchar(2000)
flex_penalty_prices	varchar(2000)
flex_penalty_threshold	real
lt_contract_timestamp	datetime

mgrid_mpc1_lt_outputs	
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
predicted_da_profile	varchar(2000)
flexibility_offer	varchar(8000)

mgrid_mpc1_lt_outputs_history	
PK. id	bigint
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
predicted_da_profile	varchar(2000)
flexibility_offer	varchar(8000)

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX I TO D4.5.3 MICROGRID LEVEL MODEL PREDICTIVE CONTROL SUBMODULE

Open software module for energy flows management in building's microgrid – Model predictive control module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	UNIZGFER
Authors (Partners)	Vinko Lešić, Danko Marušić, Mario Vašak (UNIZGFER)
Contact person	Danko Marušić (UNIZGFER)
Abstract (for dissemination)	This deliverable discusses the MPC design procedure for microgrid-level of energy management in buildings. It interacts with a lower-level MPC module/submodule and also with grid-side modules. The annex provides the internal logic of module operation, both in off-line mode when used for optimization of building daily operation and flexibility bidding and in on-line operation when it decides on behaviour of controllable microgrid elements and interacts with grid-side and other MPC modules.
Keyword List	MPC, microgrid, batteries, building operation flexibility, day-ahead consumption, intra-day consumption

Revision history

Revision	Date	Description	Author (Organization)
0.1	15 December 2017	First incomplete draft	Mario Vašak, Vinko Lešić, Danko Marušić (UNIZGFER)
0.2	22 December 2017	Final draft	Mario Vašak, Vinko Lešić, Danko Marušić (UNIZGFER)
1.0	31 December 2017	Final	Mario Vašak, Vinko Lešić, Danko Marušić (UNIZGFER)
1.1	15 September 2018	Initial version taken from deliverable D4.2.1. Mode predictive control module, for further updating based on feedback from pilots	Mario Vašak (UNIZGFER)
1.2	15 January 2019	Updated document	Mario Vašak (UNIZGFER)
1.3	18 June 2019	Updated document	Mario Vašak, Danko Marušić, Vinko Lešić (UNIZGFER)
2.0	30 June 2019	Final draft	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

This D4.5.3 annex describes the optimal control module of the microgrid level. The module is positioned as the highest in the hierarchy and as a connection to the electricity distribution grid and energy markets. The core of the module is a model predictive control algorithm based on a linear program. It minimises the cost of building operation in various time-variable market conditions. It also manages energy storage operation while respecting the system limitations and their degradation in time. Renewable energy sources production and non-controllable building consumption is also taken into account based on weather forecast and historical data. Besides the control outputs for controllable microgrid elements, the outputs are also consumption pricing characterization sent towards lower hierarchy levels as a basis for overall building consumption adjustment to current market conditions and the predicted energy exchange with the electricity grid.

The presented optimal control module represents the core concept for enabling active participation of buildings in energy markets, for providing ancillary services to the grid and in general for acting as a flexible prosumer in future smart cities and smart grids.

1. List of symbols

In the remainder of this annex is the following notation employed. Variables that are indexed with respect to the time instant on the prediction horizon are denoted with subscript k (e.g. x_k). Variables that are stacked over the prediction horizon are denoted in bold notation (e.g. $\mathbf{x}$).

Variables

C	storage capacity
η	storage efficiency
x	state vector
E	energy
A	model state matrix
B	model input matrix
u	control vector
d	disturbance vector
J_{MP}	maximum power cost
J_{DA}	day-ahead cost
J_{IDf}	intra-day: day-ahead following cost
J_{BD}	battery degradation cost
ε	slack (mathematical, substitute) variable
N	prediction horizon
c_{DA}	day-ahead prices
T_d	sampling time
F	constraints matrix (related to states)
G	constraints matrix (related to inputs)
w	constraints vector

Subscripts

0	initial value
$\min$	minimum value
$\max$	maximum value
dch	discharging variable
ch	charging variable
L	load variable
G	grid variable
$HVAC$	variable
BAT	battery variable
B	boiler variable
H	heater variable
CHP	combined heating and power variable
PV	photovoltaic variable
l	low level variable
m	medium level variable
h	high level variable
BD	battery degradation

Abbreviations

CFTOC	constrained finite time optimal control
MPC	Model Predictive Control
CR	critical Region
HVAC	Heating, ventilation, and air conditioning
DA	Day-ahead
EMS	Energy management system

Superscripts

(i)	Hierarchical control iteration number
$*$	optimal variable – exempt is E_G^* that refers to energy reference (day-ahead)

2. Introduction

One of the main focuses of the 3Smart project is to derive a modular energy management tool for buildings that can be easily adapted to different configurations of the building and that adds upon the existing building automation system. This "3Smart" energy management concept consists of three levels put in the hierarchical organization: zones comfort level, central HVAC level and microgrid level, as shown in Fig. 2.1. Further on, each of the modules incorporates three different modules: prediction and estimation, optimal control and interfaces to the equipment (Fig. 2.2). Each of the considered pilot locations has specific configurations where all modules or levels are not necessarily present. High level of flexibility is therefore targeted to achieve easy modifications and adaptation to particular buildings.

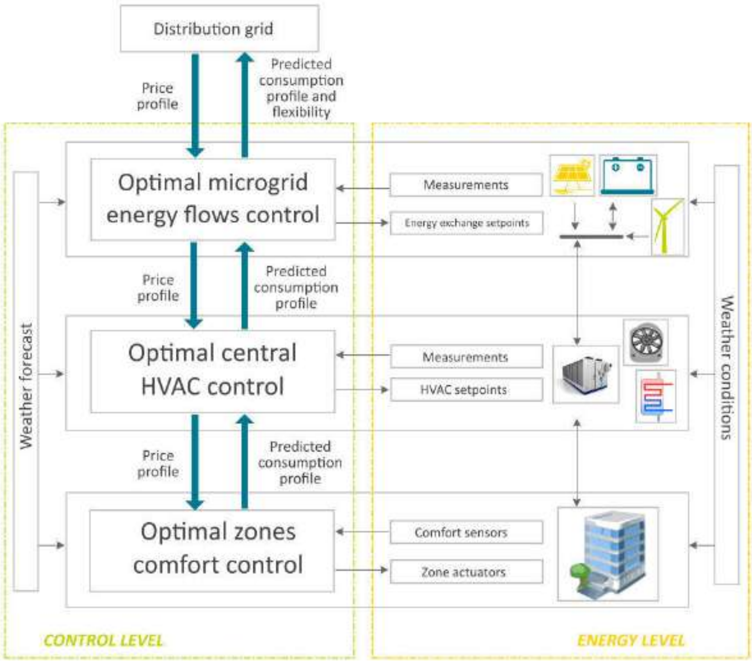


Figure 2.1: Functional diagram of the 3Smart EMS hierarchy on the building side.

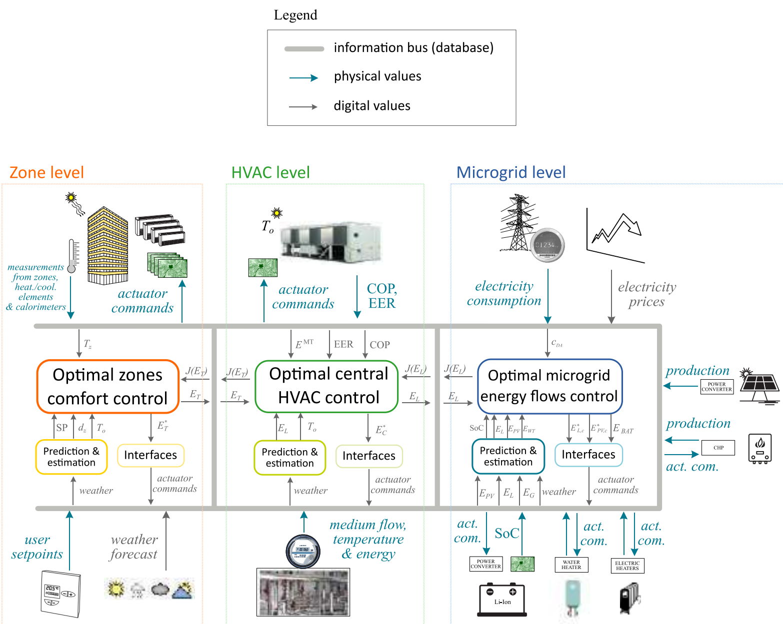


Figure 2.2: Modules schematics of the 3Smart EMS concept.

The deliverable presents the optimisation algorithm that takes into account renewable energy sources production profiles, total building electrical energy consumption, variable prices, electricity market and distribution grid conditions as well as different characteristics and requirements. This is all performed while considering electrical and thermal storages dynamics included through mathematical models of storages and physical system limitations.

Finally, the building microgrid level is the highest level in the building-side EMS hierarchy. Microgrid level introduces a possibility to manage energy storages, controllable production and controllable loads on the building level to induce minimum energy costs with respect to the planned energy consumption and production profile while it makes the building an active entity on the smart grid or on the district-level smart energy distribution system, i.e. enables further modular build-up of the concept beyond the building and towards the smart city.

The microgrid level optimally balances the electrical and possibly thermal energy flows from corresponding production and conversion units (photovoltaic arrays, small wind turbines, CHPs), to controllable or non-controllable loads while economically optimally engaging flows from/to storage units and from/to the utility grid in accordance with technical constraints on the flows that need to be respected. Sometimes also production units and non-HVAC consumption units may be controllable directly by the microgrid level which gives an additional flexibility (e.g., boilers for domestic hot water may be an example).

Figure 2.7 presents the microgrid level optimisation process and connection with the real physical equipment. Information about energy requirements are provided from lower levels and electricity exchange terms are received from the aggregator. The MPC for microgrid energy flows takes into account the following information: non-controllable consumption profile prediction E_{L}, local generation profile prediction from photovoltaics E_{PV} and wind turbine E_{WT}, storages state of charge (e.g. for batteries and hydrogen-based storage with fuel cells available, $\mathrm{SoC}_{\mathrm{BAT}}$ and $\mathrm{SoC}_{\mathrm{FC}}$), controllable loads state and electricity exchange terms. While taking into account all the physical constraints of the available equipment, optimisation algorithm computes the required charge/discharge energy requirements for electrical storages, (e.g. batteries E_{BAT}) and the energies for controllable loads, production and conversion units, which are further passed to information database and via available interface submodules finally delivered as reference values for targeted power converters or on-off switch commands for the loads. In standalone operation, microgrid simply uses available storages for buying at low price and selling at high price while satisfying load requirements. In hierarchically interconnected operation, microgrid level transforms electricity prices towards lower hierarchy levels by smart storages, loads and production units actuation.

The balancing of available power production with the requirement of building electrical energy consumption in variable electricity market scenarios and with available energy storages is performed by model predictive controller for energy flows management.

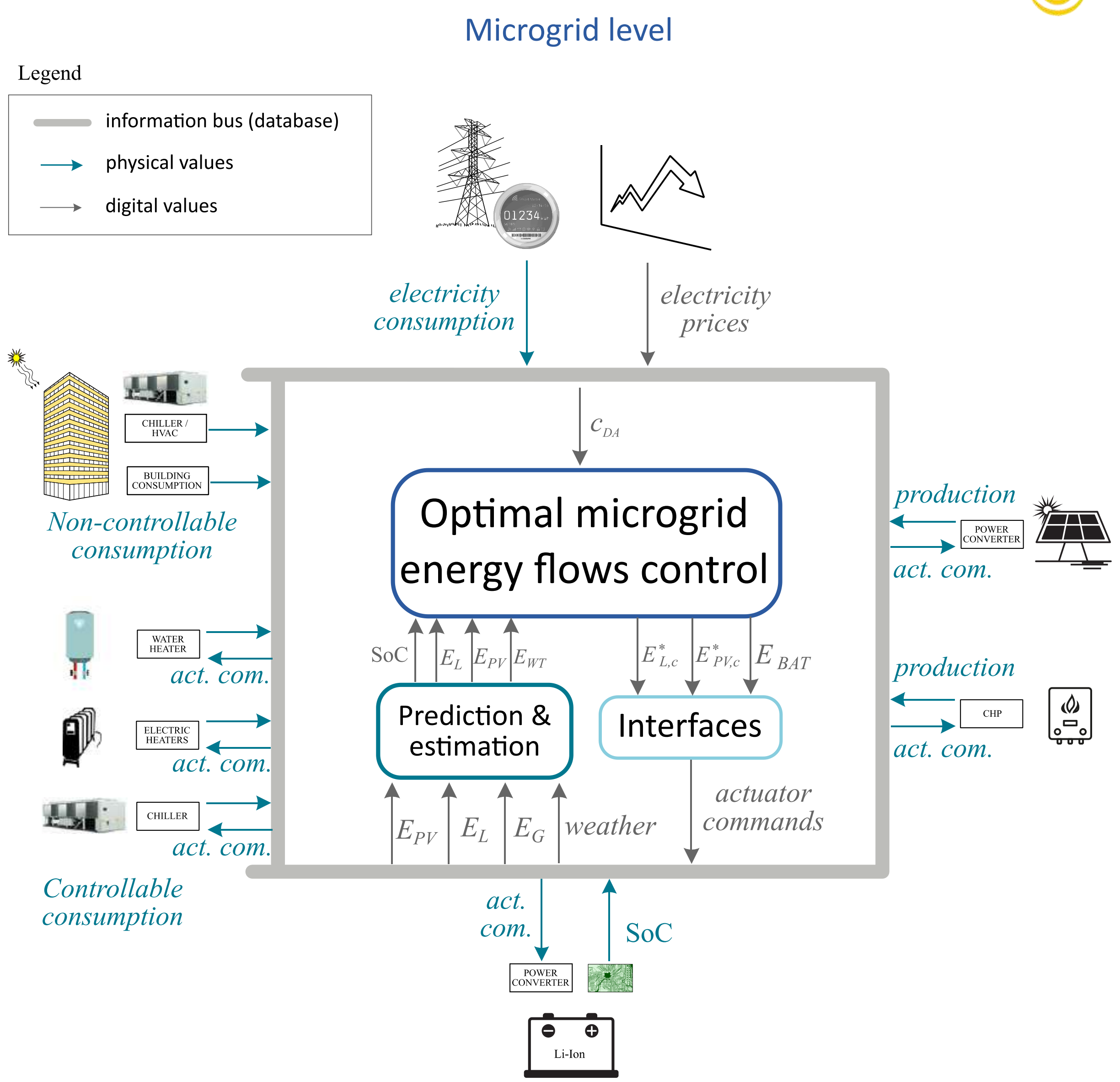


Figure 2.7: Microgrid level optimal control concept.

Table 2.1: Model predictive control module for microgrid energy flows control

Variable name	Variable annotation	Variable description
Module inputs		
Cumulative predicted controllable energy consumption that needs to be served by the microgrid; electricity and heat	E_L, E_H	Energy inputs optimized within the HVAC level MPC
Grid price conditions	c_{DA}	Prices and conditions obtained from the distribution grid/grids
Prediction of the overall non-controllable electricity/heat/gas consumption profile	$E_{L,nc}$, $E_{H,nc}$, $E_{gas,nc}$	
Prediction of the overall controllable generation profile of electricity/heat	$E_{L,c}$, $E_{H,c}$	
Parameters of the battery/batteries model, if exist	θ_{BAT}	

Battery/batteries state of charge, if exists	SoC	
Internal states of controllable loads, if exist (e.g., representative temperature in the domestic hot water boiler or in electric heaters controllable room)	x	
Controllable loads state limits, if exist (e.g., upper and lower limit of the domestic hot water boiler or in electric heaters controllable room)	x_{min}, x_{max}	
Controllable loads energy conversion model, if exist (e.g. power to heat model in the domestic hot water boiler or refrigerator)	θ_{cont}	
State of the heat storage, if exists (e.g., representative temperature of the heat tank)	T_{hs}	
Parameters of the heat storage model	θ_{hs}	
Parameters of the energy conversion units model, if exist (e.g., CHP)	θ_{ec}	
Prediction of usage conditions of controllable loads, if applicable (e.g. temperature of the cold water supply in the domestic hot water tank and profile of hot water usage)	d_{cont}	
Prediction of maximum energy inputs from the photovoltaic system	$P_{PV,max}$	Obligatory needed when power production from the photovoltaic system is controllable, otherwise can be merged with non-controllable electricity consumption
Maximum power price	c_{Pmax}	
Day-ahead prices	c_{DA}	
Intra-day prices	$c_{ID,\mathrm{penal}}, c_{ID,\mathrm{incent}}$	If not provided along the entire prediction horizon, considered to be equal to 1.2c_DA
Storage degradation price	c_{Batt}	
Table 2.2 – Module outputs – should be separate table		
Module outputs		
Profile of the energy exchange with battery/batteries storage	E_{BAT}	The value valid for the first sampling period is to be transmitted to the interface module
Profile of the energy command for controllable load actuation	$E^{*}_{L,c}$	For controllable loads continuous energy commands are generated which are then transferred to on-off on the

		interface submodule
Profile of the energy conversion units actuation	E_{ecu}	E.g., CHP command for heat generation with electricity as the side-product or vice versa
Local characterization of the value function of the optimization around the planned profile of controllable loads E^{EL}, E^{H}	$J^*(E_L, E_H)$	This local characterization is transmitted back to the zone and central HVAC level (constitutes of price coefficients and polytopic localization)
Energy exchange profiles with the grid and other data for the grid: electrical energy, heat energy, gas energy, respectively	E_G, E_G^H, E_G^{gas} other conditions	Conditions obtained through optimization of the interaction with distribution grid/grids

3. Mathematical model of energy storages

When observing microgrid with a sampling time large enough to disregard the transients on its energy links, it is sufficient to observe the energy balancing condition and storages dynamics. Dynamics of a microgrid storage unit are modeled via its state-of-charge (SoC):

$$SOC_{k+1} = SOC_k - \frac{1}{C}\eta\, E_{\mathrm{BAT},k} \quad (3\text{-}1)$$

which describes how the SoC changes with applied charging or discharging energy (E_{batt}) for the next sample time instant $k+1$. Variables C and η denote capacity and efficiency of the system. As shown in (3-2), energies and efficiencies are split to discharging and charging components ('dch' and 'ch' subscripts) such that $E_{\mathrm{dch}} \geq 0$ and $E_{\mathrm{ch}} \leq 0$. This is done to avoid high calculation complexity of a mixed-integer problem formulation for optimization of microgrid energy flows [1]. For the charging part, efficiency of $\eta_{\mathrm{ch}} < 1$ implies dissipated energy of the process as it is intuitively clear. For the discharging part, observed from the perspective of electrical storage, $\frac{1}{\eta_{\mathrm{dch}}} \geq 1$ means that more energy is sent from the electrical storage than it reaches the microgrid. The storage dynamics equation (1) are valid for all types of microgrid storages, assuming that the transients are far less (one order of magnitude or more) than the sample time interval. Therefore, it is valid for batteries and fuel-cells but also applicable for various thermal storages.

$$SOC_{k+1} = SOC_k - \frac{1}{C}\left(\frac{1}{\eta_{\mathrm{dch}}}E_{\mathrm{dch},k} + \eta_{\mathrm{ch}}E_{\mathrm{ch},k}\right), \quad (3\text{-}2)$$

For the MPC implementation, (3-2) is formulated as:

$$x_{k+1} = Ax_k + Bu_k, \quad (3\text{-}3)$$

where x_k is a storages SoC vector, u_k is a vector of energies exchanged between the microgrid and the storage systems between time instants k and $k+1$, $u = [E_{\mathrm{dch}}, E_{\mathrm{ch}}]^T$, and A and B are corresponding model matrices.

In particular for batteries, both η_{ch} and η_{dch} include battery and power converter efficiency and can be considered constant. The charging and discharging energy, i.e. power within the sampling time interval, is a subject to physical limitations dictated by power converter limitations and battery conditions (SoC-charging current characteristics). Mathematically, this is included in the optimisation problem simply as:

$$u_{\mathrm{min}} \leq u_k \leq u_{\mathrm{max}}, \tag{3-4}$$

while storage SoC operation between capacity limits (e.g. 10% and 100%) as:

$$x_{\mathrm{min}} \leq x_k \leq x_{\mathrm{max}}. \tag{3-5}$$

In general, the microgrid energy balance condition (energy conservation law on the microgrid link) implies that sum of production, consumption and storage energies equal to zero at all times, i.e. all the produced energy is either consumed, saved to storages or sold back to the utility grid. Mathematically, this is described as the condition satisfied at every discrete time step k:

$$E_{G,k} = E_{L,k} - \mathbf{1}_d^T d_k - \mathbf{1}_u^T u_k, \tag{3-6}$$

where $E_{G,k}$ is the energy exchanged with the utility grid, $E_{L,k}$ is the energy supplied to the load (building consumption), d_k is a vector of energy productions of different generation units in the microgrid and $\mathbf{1}_d$ and $\mathbf{1}_u$ are appropriately sized vectors of ones introduced to mathematically represent the summation of all production contributors (e.g. photovoltaic arrays) and controllable loads (e.g. HVAC system).

4. Interaction of building-side and grid-side EMS

Electrical energy market and distribution grid are independent systems entities where each of them can provide the building certain market conditions related to exchange of energy between the building and the grid. The hierarchical control system on the building side takes into account the announced market conditions and decides how to control the building climate and internal building energy flows through all the levels (zones, central heating/cooling medium preparation, microgrid) in the optimal way such that minimum-or-no discomfort are established at minimum overall price for the building.

A common time base of operation (i.e. sampling time) is defined as the time interval within which the cost does not change. Intra-day requests for the regulation are possible to deliver within the next sampling time, which is chosen in respect to available computational resources (data acquisition, server computer power, solver license etc.).

The building-side EMS optimizes the comfort and the overall economic cost of the energy exchange with the utility grids. Comfort is transformed into energy and finally into cost through suitable weighing factors and model-based transformation.

In this section it is assessed what are the different parts of the cost for energy exchange profile between the building and the grid, considering the existence of market and distribution grid entities. These parts are all put individually to the model predictive control form. But firstly, a common time base of operation needs to be assessed in the sense of time intervals within which the cost does not change. We refer to it as the polling interval and its duration is named the common sampling time. Instantaneous response to the grid requesting for a change in building power consumption at any moment is left out of scope. The slowest possible reaction time to the emergency event for the configuration considered here amounts one common sampling time, which is chosen in respect to available computational resources (data acquisition, server computer power, solver license etc.) for all pilot sites.

The economic parameters for defining the cost for energy exchange with the grid that are to be taken into account in the building-side EMS are given in the sequel. These parts are all integrated in the MPC form.

4.1 Cost of maximum power

This is usually calculated on a monthly basis. In certain day of the month, only the increase of the maximum power compared to maximum registered power from the previous days in the month is penalized, at the beginning of the month the initial maximum power is set to zero. This cost is only charged for power taken from the distribution network, not for power sold.

The optimisation objective is therefore to keep the consumed energy as close as possible to 0. This would only be possible of course for the real case of zero-energy building and in reality certain deviation is necessary for normal operation of the building. Mathematically, this is represented as:

$$J_{MP} = c_{\mathrm{Pmax}}\varepsilon_1, \qquad (4\text{-}1)$$
$$\text{s.t.} \begin{cases} \varepsilon_1 \geq 0 \\ \varepsilon_1 \geq E_G - \varepsilon_2 \\ \varepsilon_2 = \max E_{G1,\dots,N} \end{cases},$$

where ε_1 , ε_2 are so-called slack variables (substitute variables) required for mathematical representation. Variable ε_2 is the maximum of total energy consumption required by the building, over the prediction horizon (worst case), obtained from the previous consumption for current month. Relation (4-1) is graphically illustrated in Fig. 4.1., which shows that up to ε_2, the cost ε_1, is maintained the same as this is already the price that has to be paid this month as that peak consumption already occurred in the past. If additional consumption is required to maintain the building operation, the cost ε_1 increases linearly with coefficient of c_{Pmax} and the overall optimisation cost J_{MP} increases. The MPC problem (minimisation of the cost function while respecting the constraints) tends to keep ε_1 as low as possible. Economically, this means the lowest cost of operation with regard to the maximum power criterion.

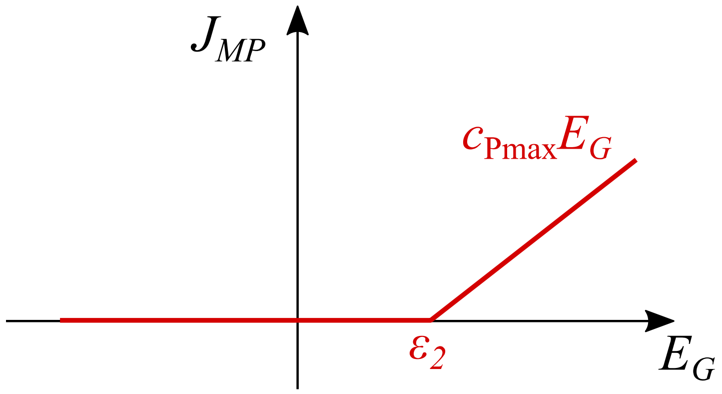


Figure 4.1: Maximum power cost function sketch.

4.2 Cost of energy, day-ahead

The aggregator is a market entity and it participates in the market by bidding demand profile (e.g., 24-hour kWh/h blocks) for a certain day-ahead (DA) price (here we assume all bids are successful). DA prices are based on power exchange data (HUPX, SIPEX, EEX). They will be used by building-side EMS for the optimization process, i.e. for cost minimization which results in optimal demand profiles for given prices. The values for day-ahead prices and the resulting demand profile need to be known 12 hours prior to start of the observed day, meaning from 12 to 36 hours before the dispatch between the two midnights. It should be mentioned that the building-side EMS receives day-ahead energy prices (24 prices, for each hour). The building will (e.g. at 9:00 each day) send to the aggregator the energy consumption profile it would apply between the following two midnights in case of a fixed price determined as the average price for the previous day, such that the aggregator can better bid on the market and finally send prices to the building/buildings.

Day-ahead building energy consumption is the energy required for internal building processes and for balancing between production, consumption and storages. Mathematically, it is put to a form of minimising the energy bought from the utility grid:

$$J_{DA} = c_{DA}^T E_G, \qquad \text{s.t.} \begin{cases} \mathbf{x} = \boldsymbol{A}x_0 + \boldsymbol{B}\mathbf{u} \\ \mathbf{x}_{\min} \le \mathbf{x} \le \mathbf{x}_{\max} \\ \mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max} \end{cases}, \tag{4-2}$$

where E_G can also have negative sign, referring to the process of selling the excess energy back to the grid. The expression (4-2) also incorporates storage dynamics and physical limitations (storage capacity and allowed charging/discharging characteristics). If the market price c_{DA} is a constant vector, (4-2) yields control variables for energy-optimal building operation. The building-side EMS needs to obey the DA profile it declared. In case it is not able to due to some unforeseen events later in the day, it needs to participate in the intra-day market to compensate for the incorrect DA forecast.

4.3 Intra-day pricing

At the intra-day market, the building-side EMS needs to pay for energy which is missing/surplus from the profile announced day-ahead. There are two options for valorising the intra-day prices related to this in the project: *i)* simpler one: values will be taken as day-ahead energy prices multiplied by a factor of 1.2 (20% higher) and thus the cost for the hour when the deviation occurs will be expressed as: (energy in the hour * DA hour price)+(absolute value of energy deviation from DA * 1.2 DA hour price); *ii)* instead of DA*1.2 existing intra-day prices are used (considered as forecasts and used as historical values from markets with available intra-day historical price data).

The mathematical representation for the case of 20% higher price is:

$$J_{IDf} = 1.2c_{DA}^{T}\|E_G - E_G^*\|_1, \tag{4-3}$$

which puts a 20% higher cost on following the reference trajectory E_G^* of the day-ahead production. Since the price in (4-3) is higher than in (4-2), the optimisation will give priority to intra-day following than to reduction of consumption unless the possible reduction itself gives higher savings than those 20%. An exemplary consumption profile is illustrated in Fig. 4.2.

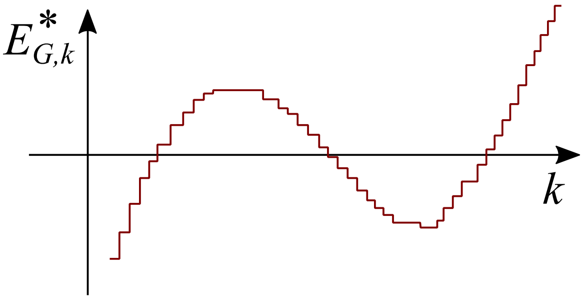


Figure 4.2: Exemplary day-ahead building energy consumption profile.

4.4 Flexibility provision towards the grid

The flexibility provision towards the grid follows the logic from D5.4.3 (grid-side modules description).

4.5 Costs of battery degradation

As always defined by manufacturers, batteries have limited duration and life span, which is expressed as number of charging/discharging cycles. This can be put to a form of the exact price for battery degradation per charging/discharging, or in terms of MPC framework put as part of the cost function:

$$J_{BD} = c_{BD}u_k, \tag{4-5}$$

where price c_{BD} depend on battery type and manufacturer. For more details about the approach, see [6], from which the prices for $LiFePO_4$ batteries are chosen as $c_{BD} = 0.0784\frac{€}{kWh}$.

5. Optimisation problem formulation

Microgrid optimisation submodule and the corresponding MPC algorithm for energy flows management outputs a decision when to buy from or sell energy to the utility grid and in which amounts, i.e., when to charge and discharge storages. The overall optimisation problem is a constrained finite time-optimal control problem based on a linear program [3]. As described before, it is a complex function of different contradictory requirements. The complete objective function of the microgrid MPC submodule incorporates parts from sections 4.1-4.5:

$$J = J_{MP} + J_{DA} + J_{IDf} + J_{IDd} + J_{BD}. \tag{5-1}$$

The microgrid MPC optimization problem is posed as follows:

$$J = \sum_{k=0}^{N-1} \left(\varepsilon_{1,k} + c_{DA,k} E_{G,k} + 1.2 c_{DA,k} \left\| E_{G,k} - E^*_{G,k} \right\|_1 - 5 c_{DA,k} \left\| \Delta E^*_{G,k} \right\|_1 + 100 c_{DA,k} \left\| E_{G,k} - \varepsilon_{3,k} \right\|_1 + c_{BD,k} u_k \right)$$

$$\text{s.t.} \begin{cases} \boldsymbol{\varepsilon}_1 \geq 0 \\ \boldsymbol{\varepsilon}_1 \leq \mathbf{c}_{\mathrm{Pmax}} (\mathbf{E}_G - \boldsymbol{\varepsilon}_2) \\ \boldsymbol{\varepsilon}_2 = \max \mathbf{E}_{G1,\dots,N} \\ \mathbf{x} = \boldsymbol{A} x_0 + \boldsymbol{B} \mathbf{u} \\ \mathbf{E}_{G,k} = \mathbf{E}_L - \mathbf{1}_d^T \mathbf{d} - \mathbf{1}_u^T \mathbf{u}, \\ \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \\ \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \\ \boldsymbol{\varepsilon}_3 = \mathbf{E}^*_G + \Delta \mathbf{E}^*_G \end{cases}$$

where bold notation represents vectors and matrices stacked over the prediction horizon. In particular, for e.g. microgrid configuration consisted of HVAC, electric heaters, boiler, CHP, battery storage and various non-controllable loads, the energy balance equation is:

$$E_{HVAC} + E_H + E_B + E_{CHP} + E_L = E_{PV} + E_{BAT} + E_G, \tag{5-2}$$

from which the E_G is derived, or (5-2) is simply put as a constraint to (5-1). Vector u_k is comprised of energies from (5-2) related to controllable systems and d of ones related to non-controllable systems.

6. Hierarchical coordination of the modules

All of the 3Smart modules can operate independently and in case of absent modules. However, when put to coordinated operation, the savings are significantly increased as proven in [2] (additional savings of 15%) for the case of zone and microgrid modules. This is achieved only by further mathematical (software) adjustments, without additionally required hardware or installations.

A significant feature in building applications is observed in a proximity of energy and economical optimum. The coordination method used in 3Smart exploits the proximity premise for dividing a problem into hierarchy levels suitable for fast convergence from low level optimization criterion to the higher one. With parametric formulation, both criteria are expected in the same or adjacent critical region (CR) while shifting between them is trivial in complexity and time requirements. Lower hierarchy level control variables are treated as a parametric disturbance of the higher hierarchy level problem and further transformed towards global optimization criterion through a parametric problem value function. A critical region is a subset of parameters that yield the same set of active constraints, i.e., constraints that are satisfied with equality sign in the optimal solution. The hierarchical decomposition keeps the modules and corresponding technologies apart and independent. The implementation is therefore eased with minimal on-site modifications and different technologies are interconnected only by means of provided price and consumption signals.

A common approach in joining the MPC problems is by concatenating the matrices and the cost functions of each problem into one larger control problem formulation subject to augmented model and joint constraints. In 3Smart EMS concept, level separation is retained and the coordination is performed by exchanging information about optimized energy consumption price and consumption profiles, which are respected in both level operation. Parametric coordination of hierarchy levels exploits the multiparametric MPC and CRs with simple explicit control law. The original algorithm for multiparametric MPC was proposed in [2] and its segments are utilized for hierarchical coordination. The distinction is that only a single CR is determined at one iteration and no additional partitioning of the parameter space is performed. The 3Smart EMS modules are generally represented as in the sequel.

- Zone comfort MPC as low-Level (LL) problem:

$$\begin{aligned} J_l^* = \min_{u_l} J_m^*(u_l) + f_l(u_l, x_{l0}) \\ G_l u_l \le F_l x_{l0} + E_l \end{aligned} \qquad (6\text{-}1)$$

- HVAC level MPC as mid-level (ML) problem:

$$\begin{aligned} J_m^* = \min_{u_m} J_h^*(u_m) + f_m(u_l, u_m, x_{m0}) \\ G_m u_m \le F_m x_{m0} + G_{ml} u_l + E_m \end{aligned} \qquad (6\text{-}2)$$

- Microgrid MPC as high-level (HL) problem

$$\begin{aligned} J_h^* = \min_{u_h} f_h(u_h, u_m, x_{h0}) \\ G_h u_h \le F_h x_{h0} + G_{hl} u_m + E_h \end{aligned} \qquad (6\text{-}3)$$

Initialization step sets the following: $J_m^{*(-1)} = \|\cdot\|_1, J_h^{*(-1)} = \|\cdot\|_1, CR_m^{*(-1)} = \Re^{n_{ul}}, CR_h^{*(-1)} = \Re^{n_{um}}$.

The algorithm executes problems (6-1), (6-2) and (6-3) iteratively. Each iteration ($i = 0,1, \dots, m$) consists of the following 4 steps:

1. LL with $J_m^{*(i-1)}$ and $u_l \in CR_m^{(i-1)} \rightarrow u_l^{*(i)}$
2. ML with $J_h^{*(i-1)}$, $u_l = u_l^{*(i)}$and $u_m \in CR_h^{(i-1)} \rightarrow u_m^{*(i)}$
3. HL with $u_m = u_m^{*(i)} \rightarrow CR_h^{*(i)}\left(u_m^{*(i)}\right), J_h^{*(i)}, u_h^{*(i)}\left(u_m^{*(i)}\right)$

4. ML with $J_h^{*(i)}, u_l = u_l^{*(i)}$ and $u_m \in CR_h^{(i)} \rightarrow CR_m^{*(i)}\left(u_l^{*(i)}\right), J_m^{*(i)}, u_m^{*(i)}\left(u_l^{*(i)}\right)$

Graphically, iterations through critical regions are presented in Fig. 6.1. The initial solution u_l^* is shifted along the decreasing value of J_m^* from (6-1) such that all physical and comfort constraints are satisfied, which is expressed by triggering the constraints from (6-2). Previous energy-optimal control inputs u_l^* are now transformed to price-optimal ones $u_l^{*(i)}$, within the critical region $CR_m^{(i-1)}$. Finally, the price optimal higher level control signals $u_m^{*(i)}$ within the critical region are obtained from (6-2). The case of LL-ML iteration is graphically illustrated in Fig. 6.1.a. The same applies for ML-HL, and finally also for the whole path of $\mathrm{LL} \rightarrow \mathrm{ML} \rightarrow \mathrm{HL} \rightarrow \mathrm{ML}$ as in steps 1-4 above.

The approach is based on the premises that the non-coordinated solution on the lower level is near the optimum achieved through coordination. In practice, the premise of solution proximity proved to be justified for the case of building connected to a variable price energy market. When such assumptions are compromised, the CR boundaries are hit before the zone level constraints and the solution is to be found outside the current CR. The procedure is then iteratively executed until the solution is found in the active CR where zone level constraints are triggered (Fig. 6.1.b) or microgrid constraints are activated (adjacent CR is non-existent).

The number of levels (modules) can be arbitrary and iterations remain the same: start from the lowest, climb up to the highest and determine current optimizers with computed respective costs and critical regions from the previous iteration. On the highest level compute the new critical region and cost for the level below and go back through the levels by repeating that, up to the second lowest level. Then everything is set for the next iteration.

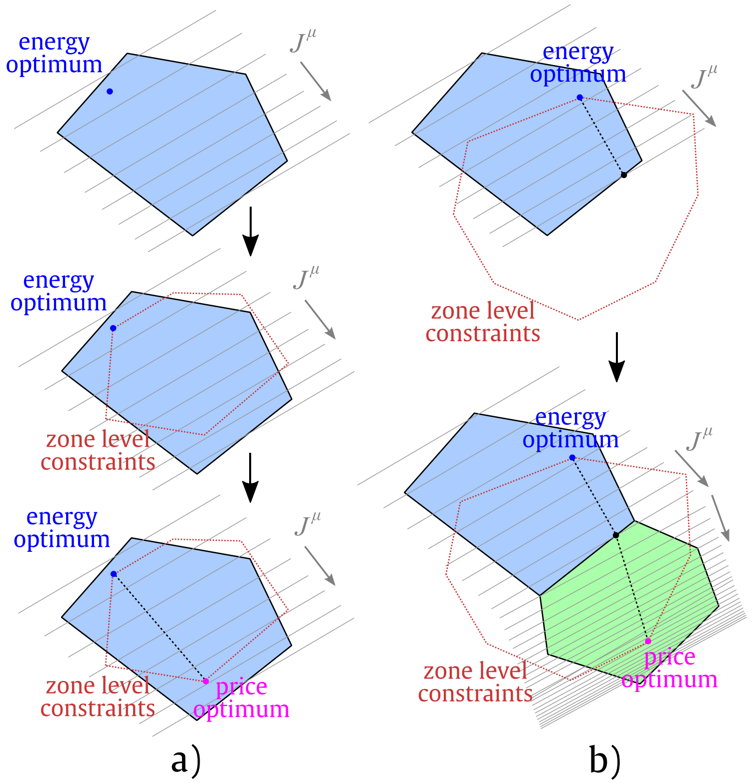


Figure 6.1: Geometric representation of parametric coordination in the space of control vector: a) solution is inside the starting CR, b) iteration to a nearby CR. Parallel lines represent the cost function with higher density reflecting a smaller value.

7. Conclusion

This annex describes a design procedure of a microgrid level MPC module of 3Smart building energy management system. The controller is designed to operate as a highest level in the hierarchically organized building energy management system. The MPC optimizes total energy and microgrid operation costs in presence of time-varying building consumption, renewables production, energy prices and demand response. Outputs of the MPC are storage reference energy profiles with chosen sampling time.

The report also sets the outline for a coordinated hierarchical control of all levels of 3Smart building energy management system based on iterative approach with parametric MPC.

Bibliography

[1] M. Gulin, M. Vašak, M. Baotić, "Analysis of Microgrid Power Flow Optimization with Consideration of Residual Storages State", *Proceedings of the European Control Conference 2015*, pp. 3131-3136, 2015.

[2] V. Lešić, A. Martinčević, M. Vašak, "Modular energy cost optimization for buildings with integrated microgrid", *Applied Energy*, vol. 197, pp. 14-28, 2017.

[3] J. B. Rawlings, D. Q. Mayne, "Model Predictive Control: Theory and Design", Nob Hill Publishing, Madison, Wisconsin, 2015 (download URL).

[4] A. Martinčević and M. Vašak, D4.4.1 – Open software module for zones consumption management, University of Zagreb, Faculty of Electrical Engineering and Computing, 2017.

[5] Borrelli F, Bemporad A, Morari M. Geometric algorithm for multiparametric linear programming. *Journal of Optimization Theory and Applications,* vol. 118, no. 3, pp. 515-540, 2003.

[6] A. Parisio, E. Rikos, L. Glielmo, “A Model Predictive Control Approach to Microgrid Operation Optimization”, *IEEE Transactions on Control Systems Technology*, vol. 22, pp. 1813-1827, 2014.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D4.5.3

Final building-side energy management software module – Interfacing submodules for microgrid management

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Debrecen (UNIDEBTTK), Marko Baša (E3)
Authors (Partners)	Arpad Racz (UNIDEBTTK), Marko Baša (E3), Mario Vašak (UNIZGFER)
Contact person	Mario Vašak (UNIZGFER)
Abstract (for dissemination)	The deliverable gives an overview of modules that interface energy exchange commands from the microgrid level model predictive control to building actuators – here the input/output data of the modules are provided and a detailed logic regarding modules operation is provided in the annexed document.
Keyword List	Interfacing Controllable Load, Interfacing Controllable Storage, Interfacing Controllable Generation, Battery System, Domestic Hot Water Tank, Photovoltaic System, CHP

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on D4.2.1	Mario Vašak (UNIZGFER)
v0.2	15 January 2019	Updated version	Mario Vašak (UNIZGFER)
V0.3	19 June 2019	Updated version	Arpad Racz (UNIDEBTTK), Marko Baša (E3), Mario Vašak (UNIZGFER)
V1.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

Integrated energy management of buildings and grids installed with the 3Smart project is on the side of buildings divided into three vertical levels – zone level, central HVAC system level and microgrid level. In each of these levels the energy management algorithms are classified into three parts – (i) prediction and estimation, (ii) model predictive control, and (iii) equipment interfacing -- and the algorithms are implemented via a sequence of submodules.

The submodules are designed, commissioned and tested on different pilot buildings in the Danube region.

Within this deliverable the focus is put on microgrid level interfacing submodules.

Each submodule is presented via an interfacing table that explains what data are used by the submodules as inputs and what are the final output data. The algorithms behind are in more detail explained in the annexed document.

1 Introduction

Within the 3Smart project the following interfacing submodules are designed, commissioned and tested on the microgrid level:

M.I.1 – submodule for interfacing commands for energy exchange with the battery system (tested in UNIZGFER, HEP, STREM retirement and care centre and EPHZHB pilot buildings within 3Smart);

M.I.2 -- submodule for interfacing commands for energy exchange with the domestic hot water tank (tested in IDRIJA school and sports centre pilot buildings within 3Smart);

M.I.4 -- submodule for interfacing commands of heating energy to electric room heaters (tested in EON pilot building within 3Smart);

M.I.5 – submodule for interfacing commands for energy injection from the photovoltaic system to the building (tested in EON pilot building within 3Smart);

M.I.6 – submodule for interfacing commands for energy injection from CHP (tested in IDRIJA school and sports centre pilot buildings within 3Smart).

In the following chapters the mentioned submodules are presented with their interface tables showing which data they use as inputs and which data they provide as outputs to be at the disposal to other submodules or to be used for building actuators in the microgrid. Detailed explanations of algorithms behind each of the submodules are provided in the previously delivered 3Smart document D4.2.1 (related to interfacing on the microgrid level) -- it is here updated with included feedback from operation on different pilots and provided as Annex 1.

Source and sink for the data used by the submodules is a properly structured 3Smart database. Its structure in the part concerned by the microgrid level interfacing submodules is provided in Annex 2.

2 M.I.1 submodule

M.I.1 submodule is used for interfacing commands for energy exchange with the battery system. Within 3Smart it is tested in UNIZGFER, HEP, STREM retirement and care centre and EPHZHB pilot buildings.

The submodule interface is defined in Table 2.1 and Table 2.2.

Table 2.1: Inputs of the M.I.1 submodule

Current SoC	SoC_E SoC	Historical profiles of SoC calculated by the battery pack controller and M.I.1 module
Commanded energy exchange AC side	E_{bat}	Energy command from M.MPC.1
Exchanged energy	E_{ex}	Energy exchange of the last

		minute
MPC sampling time	T_{MPC}	System-wide parameter

Table 2.2: Outputs of the M.I.1 submodule

Variable name	Variable annotation	Variable description
Reference power for power converter (AC side)	P_{ref}	Written into the building energy management's database

3 M.I.2 submodule

M.I.2 submodule is used for interfacing commands for energy exchange with the domestic hot water tank. Within 3Smart it is tested in IDRIJA school and sports centre pilot buildings.

The submodule interface is defined in Table 3.1 and Table 3.2.

Table 3.1: Required inputs for the M.I.2 submodule

Variable name	Variable annotation	Variable description
Commanded heat energy consumption	E_cmd_ht	Energy that should be injected to tank in 15min interval.
Commanded electric energy consumption	E_cmd_el	Energy that should be injected to tank in 15min interval.
Electric energy meter measurements data	DHW_emeter	Electric energy consumption data
Heat meter measurements	DHW_cal	Heat energy consumption data
Electric heaters technical data	φ_{DHW_el}	pilot specific data depending on the installation of the electric heaters.
Heat exchanger technical data	φ_{DHW_ht}	pilot specific data depending on the installation of the heat exchanger.

Table 3.2: Outputs of the M.I.2 submodule

Variable name	Variable annotation	Variable description
Switch heater 1 ON	DHW_heater_1	Switch ON the heater
Switch heater 2 ON	DHW_ heater_2	Switch ON the heater
Switch heater 3 ON	DHW_ heater_3	Switch ON the heater

Switch pump ON	DHW_pump	Switch ON the pump

4 M.I.4 submodule

M.I.4 submodule is used for interfacing commands of heating energy to heat storage systems, like boilers or rooms with electric room heaters (tested in EON pilot building within 3Smart).

The submodule interface is defined in **Table 4.1** and **Table 4.2**.

Table 4.1: Inputs of the M.I.4 submodule

Variable name	Variable annotation	Variable description
Controllable load energy command from the microgrid MPC module	E_l	Energy command from M.MPC.1
Exchanged energy	E_{ex}	Energy exchange of the last minute
MPC sampling time	T_{MPC}	System-wide parameter

Table 4.2: Outputs of the M.I.4 submodule

Variable name	Variable annotation	Variable description
Array of ON/OFF signals for electric heaters	u_l	Written into the building energy management's database

5 M.I.5 submodule

M.I.5 submodule is used for interfacing commands for energy injection from the photovoltaic system to the building. Within 3Smart it is tested in EON pilot building.

The following tables provide the input-output interface of the submodule (Table 5.1 and Table 5.2).

Table 5.1: Inputs of the M.I.5 submodule

Variable name	Variable annotation	Variable description
Commanded energy exchange	E_{PV}	Energy command from M.MPC.1
Exchanged energy	E_{ex}	Energy exchange of the last minute
MPC sampling time	T_{MPC}	System-wide parameter

Table 5.2: Outputs of the M.I.5 submodule

Variable name	Variable annotation	Variable description
Commanded power out for solar inverters	$p_{PV,out}$	Written into the building energy management's database

6 M.I.6 submodule

M.I.6 is a submodule used for interfacing commands for energy injection from CHP. Within 3Smart it is tested in IDRIJA school and sports centre pilot buildings.

The following tables provide the input-output interface of the submodule (Table 6.1 and Table 6.2).

Table 6.1: Required inputs for the M.I.6 submodule

Variable name	Variable annotation	Variable description
Commanded heat energy produced	Et_CHP	heat energy to be produced in time interval (MPC sampling period)
Heat meter measurements	Eexp_CHP	Heat energy produced - heat energy measured on calorimeter in time interval

Table 6.2: Outputs of the M.I.6 submodule

Variable name	Variable annotation	Variable description
Desired power	P_{E_CHP}	Reference electrical power for CHP
Start CHP signal	CHP_StartCmd	CHP_StartCmd

Bibliography

[1] 3Smart D4.1.1. Building-side EMS concept and information exchange interfaces definition. June 2017.

Annex 1 – Open software module for microgrid level consumption management – Interfacing submodules

Provided in a separate document.

Annex 2 – 3Smart database organization for open software module for the microgrid level management – Interfacing submodules

mgrid_mpc1_lt_inputs	
FK. microgrid_id	int
nctrl_elec_consumption_pred	varchar(2000)
nctrl_ther_consumption_pred	varchar(2000)
nctrl_gas_consumption_pred	varchar(2000)
pv_production_pred	varchar(2000)
c_da	varchar(2000)
da_timestamp	datetime
c_pmax	real
c_pmax_timestamp	datetime
idf_penalty_factor	real
flex_reservation	varchar(2000)
flex_activation	varchar(2000)
flex_reservation_prices	varchar(2000)
flex_activation_prices	varchar(2000)
flex_penalty_prices	varchar(2000)
flex_penalty_threshold	real
lt_contract_timestamp	datetime

mgrid_mpc1_lt_outputs	
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
predicted_da_profile	varchar(2000)
flexibility_offer	varchar(8000)

mgrid_mpc1_lt_outputs_history	
PK. id	bigint
FK. microgrid_id	int
timestamp	datetime
batt_energy_profile	varchar(2000)
ctrl_load_energy_profile	varchar(4000)
pv_energy_profile	varchar(2000)
coordination_var	varchar(16000)
predicted_da_profile	varchar(2000)
flexibility_offer	varchar(8000)

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

ANNEX I TO D4.5.3 -- MICROGRID LEVEL INTERFACING MODULES

Open software module for energy flows management in building's microgrid – Interfacing submodules

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	30 months
Work Package	4
Task	4.5
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 2.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	UNIDEBTTK, E3
Authors (Partners)	Árpád Rácz, Sándor Misák, István Szabó, Gergő Borbély, Réka Nagy-Szentesi, András Mucsi (UNIDEBTTK), Marko Baša (E3)
Contact person	Arpad Racz (UNIDEBTTK)
Abstract (for dissemination)	A comprehensive report on 3Smart for open software module for energy flows management in building's microgrid is given, for interface submodules towards the equipment in the field. This is an annex to D4.5.3 in which the logic of each of the submodules is described.
Keyword List	3Smart, open software module, building microgrid, battery system, renewable energy, controllable load

Revision history

Revision	Date	Description	Author (Organization)
0.1	14 July 2017	First draft	Árpád Rácz, Sándor Misák, István Szabó, Réka Nagy-Szentesi, András Mucsi (UNIDEBTTK) , Marko Baša (E3)
0.2	15 November 2017	Alpha version	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK) , Marko Baša (E3)
0.3	29 December 2017	Final draft	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK) , Marko Baša (E3)
1.0	30 December 2017	Corrections	Árpád Rácz, Sándor Misák, István Szabó, András Mucsi (UNIDEBTTK) , Marko Baša (E3)
1.1	15 September 2018	Taken over from D4.2.1 for further upgrade based on feedback from pilots	Mario Vašak (UNIZGFER)
1.2	15 January 2019	Updated document	Mario Vašak (UNIZGFER)
1.3	21 June 2019	Updated document	Mario Vašak (UNIZGFER)
1.4	26 June 2019	Final draft	Arpad Racz (UNIDEBTTK), Marko Baša (E3), Mario Vašak (UNIZGFER)
2.0	30 June 2019	Final quality-checked version	Mato Baotić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of contents

Executive summary

The main objective of the project Smart Building – Smart Grid – Smart City (3Smart) funded within the Interreg Danube Transnational Programme is to provide technology and legislative setup for cross-spanning energy management of buildings and utility grids, foremost electricity distribution grids.

One of the main pillars in reaching that objective is to derive a modular energy management tool for buildings, which can be easily adapted to different configurations of the building and adds upon the existing building automation system. This “3Smart” energy management concept consists of three modules put in the hierarchical organization: zones comfort module, central HVAC module and microgrid energy flows module. Further on, each of the modules incorporate three different submodules: prediction and estimation, optimal control and interfaces to the equipment. Each of the considered pilot locations has a specific configuration where all modules are not necessarily present. High level of flexibility is therefore targeted to achieve easy modifications for particular pilots.

This annex describes the interfacing submodules on the microgrid level. These modules are essential for transmitting MPC command towards the building equipment in the field.

1 Introduction

One of the main focuses of 3Smart project is to derive a modular energy management tool for buildings that can be easily adapted to different configurations of the building and adds upon the existing building automation system. This "3Smart" energy management concept consists of three modules put in the hierarchical organization: zones comfort module, central HVAC module and microgrid energy flows module, as shown in Fig. 1. Further on, each of the modules incorporate three different submodules: prediction and estimation, optimal control and interfaces to the equipment (Fig. 2). Each of the considered pilot locations has specific configuration where all modules are not necessarily present. High level of flexibility is therefore targeted to achieve easy modifications for particular pilots.

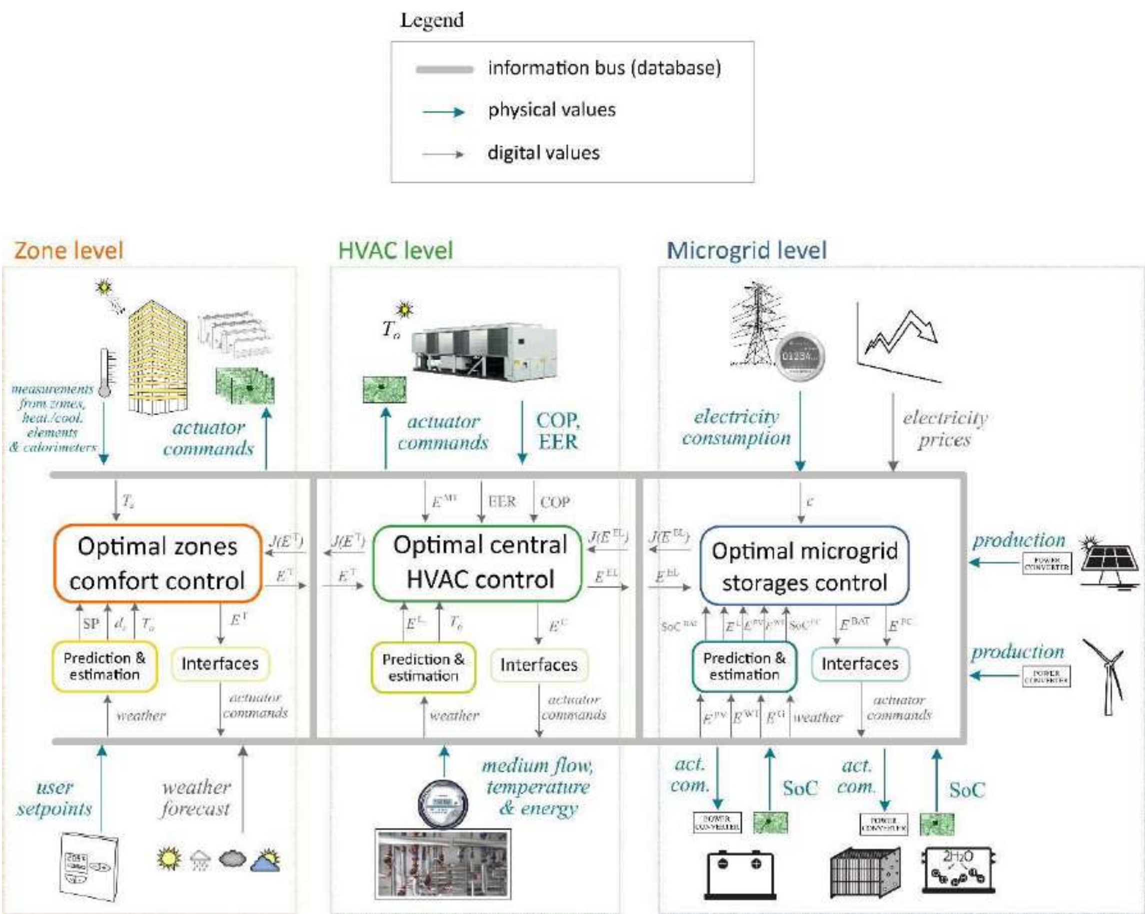


Figure 1: Functional diagram of the 3Smart EMS hierarchy on the building side.

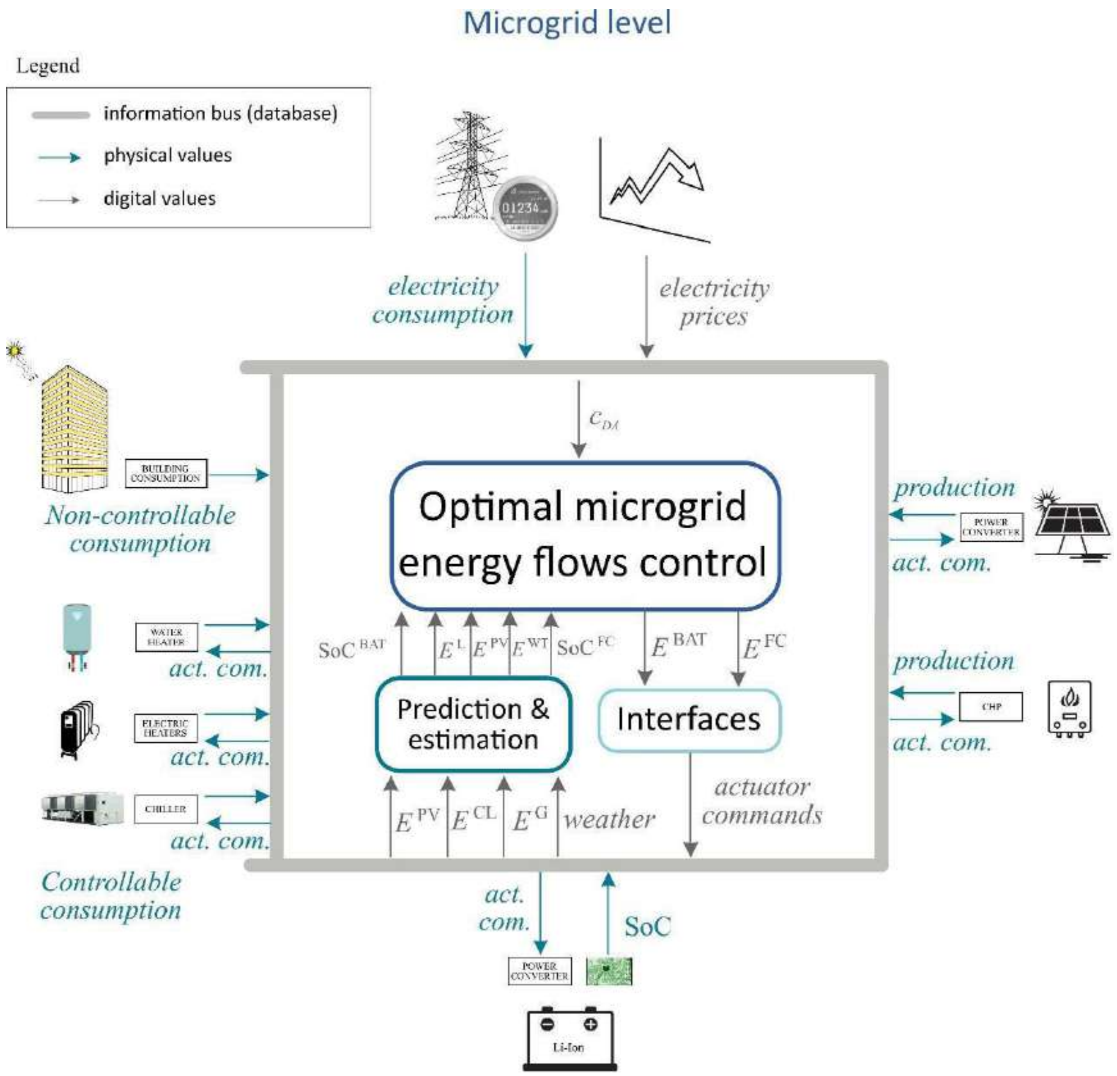


Figure 2: Modules schematics of the 3Smart EMS concept.

This document contains the descriptions of the following submodules:

- Submodule for issuing commands towards the storage power converter based on the commanded energy exchange signals
- Submodule for issuing commands towards the heaters in domestic hot water tank
- Submodule for issuing commands towards the controllable loads based on the commanded energy exchange signals
- Submodule for issuing commands towards the controllable PV inverters based on the commanded energy exchange signals
- Submodule for issuing commands towards the CHP system

2 M.I.1 -- Module for issuing commands towards the storage power converter based on the commanded energy exchange signals

2.1 Theory

Tasks of this module

Based on the energy command from microgrid MPC module the module M.I.1 provides the required energy flow from/to batteries. This control is realized via a **current reference signal** (DC side) to the power converter or via the power reference on the AC side of the power converter. During this process the module needs to respect the following limitations:

- state-of-charge of the batteries
- allowed maximum current for batteries
- the currently required charging method (CC vs. CV)
- battery status information (temperature, error signals)
- longevity of the battery pack

Challenges of $LiFePO_4$ batteries [1]

- Long voltage relaxation time to reach its open circuit voltage (OCV) after a current pulse
- Time-, temperature-, and SOC-dependent hysteresis
- Very flat OCV-SOC curve for most of the SOC range

Charging of $LiFePO_4$ batteries [2]

During the conventional lithium ion charging process, a conventional Li-ion Battery containing lithium iron phosphate ($LiFePO_4$) needs two steps to be fully charged:

1. uses **constant current (CC)** to reach about **60% State of Charge (SOC)**;
2. takes place when charge voltage reaches **3.65V** per cell, which is the upper limit of effective charging voltage. Turning from constant current (CC) to **constant voltage (CV)** means that the charge current is limited by what the battery will accept at that voltage, so the charging current tapers down asymptotically.

To put a clock to the process, step 1 (60% SOC) needs about one hour and the step 2 (40% SOC) needs another two hours.

2.2 Inputs

- Current SoC
  - [SoC_E] = kWh
  - [SoC] = %
- Commanded energy exchange

   - $[E_{batt}]$ = kWh
- Battery system energy exchange of the last minute, E_{ex}
   - $[E_{ex}]$ = kWh
- MPC sampling time
   - $[T_{MPC}]$ = s

Resolution: 1/min for SoC, $1/T_{MPC}$ for E_{batt}

2.3 Outputs

- Reference power for power converter AC side
   - $[P_{ref}]$ = W

2.4 Frequency of submodule calls:

1/min

2.5 Algorithm

For every 15 min cycle:

- $E_{sum} = 0$
- For every 1 min cycle (i):

At the end of the 1 min cycle:

$$E_{sum} = E_{sum} + E_{ex,i}$$

3 M.I.2 -- Module for issuing heating commands towards the domestic hot water tank

The output of the MPC is the energy that the heater and exchanger should consume in the time frame. These two parameters are stored as records in database. Each command is one record with time when command should be applied. Time granularity is 15 minute interval in prediction horizon. In case of a real device, the on-off state could be modulated in order to obtain the right amount of energy.

A model for a single element electric water heater is presented. Same alghoritem Power step is defined with max power of each heating element of electric heater.

3.1 Inputs

- Energy command from the microgrid MPC module: [E_{MPC_e}] = kWh
- Energy command from the microgrid MPC module: [E_{MPC_ex}] = kWh
- Heat energy imported to excager in the last minute: [E_{out_ex}] = kWh
- Electric energy consumed on el. heaters in the last minute: [E_{out_e}] = kWh
- Power of stage 1: [P_{el_1}] = kW
- Power of stage 1+2: [P_{el_2}] = kW
- MPC sampling time [T_{MPC}] = min
- „valid from“ timestamp

3.2 Outputs

- ON/OFF signal for every power stage of electric heater:
  - Cmd_{el_1}
  - Cmd_{el_2}
  - Cmd_{el_3}
- ON/OFF signal for circulation pump for heat exchanger Cmd_{ex}

3.3 Internal parameters

- Controllable load heat energy exchange of the interval: [E_{sum_ex}] = kWh
- Controllable load electric energy exchange of the last interval: [E_{sum_el}] = kWh
- Max Energy produced on heater stages in one MPC interval: [E_{el_1}], [E_{el_2}] = kWh
- Energy to produce till the end of current MPC interval: [E_{remain_el}] = kWh
- Energy to produce till the end of current MPC interval: [E_{remain_ex}] = kWh
- Current time

3.4 Frequency of submodule calls:

1/min

3.5 Algorithm

$$E_{el\_1} = P_{el\_1} / 60 * T_{MPC}$$

$$E_{el\_2} = P_{el\_2} / 60 * T_{MPC}$$

$$E_{ex} = P_{ex} / 60 * T_{MPC}$$

For every 1 minute cycle:

$$E_{sum\_el} = E_{out\_e} + E_{sum\_el}$$

$$E_{remain\_el} = E_{MPC\_el} - E_{sum\_el}$$

Cmd_{el_1}, Cmd_{el_2}, Cmd_{el_3} = OFF

IF ($E_{remain\_el} > 0$):

IF ($E_{remain\_el} < E_{\_el\_1}$): Cmd_{el_1} = ON

ELSE IF ($E_{remain\_el} < E_{\_el\_2}$): Cmd_{el_1}, Cmd_{el_2} = ON

ELSE: Cmd_{el_1}, Cmd_{el_2}, Cmd_{el_3} = ON

$E_{sum\_ex} = E_{out\_ex} + E_{sum\_ex}$

$E_{remain\_ex} = E_{MPC\_ex} - E_{sum\_ex}$

Cmd_{el_3} = OFF

IF ($E_{remain\_ex} > 0$): Cmd_{ex} = ON

For every 15 min cycle:

$E_{sum\_el} = 0$

$E_{sum\_ex} = 0$

4 M.I.4 -- Module for issuing commands towards the controllable loads based on the commanded energy exchange signals

4.1 Inputs

- Controllable load energy command from the microgrid MPC module, E_l
  - $[E_l]$ = kWh
  - Controllable load energy exchange of the last minute, E_{ex}
  - $[E_{ex}]$ = kWh
- MPC sampling time
  - $[T_{MPC}]$ = min

The temperature of the heat buffer is:

- the room temperature in case of room heating,
- buffer tank temperature measured at middle height of the tank in case of a water chiller

- equivalent temperature calculated based on three measurement (top, middle, bottom) in case of a water boiler.

Resolution: 1/min, E_l is changed with sampling frequency of the microgrid-level MPC module

$T_{heatstorage}$ should be the current value at the time stamp

4.2 Outputs

- For hot water boilers, water chillers and room heaters, u_l
  - Array of ON/OFF signals

4.3 Internal parameters

- Simplified model of the controllable load, θ_l
  - $N_{stepping}$ - dimensionless
  - $[P_{in,min}]$ = kW
  - $[P_{in,max}]$ = kW

4.4 Frequency of submodule calls:

1/min

4.5 Algorithm

$$E_{stepping} = \frac{P_{in,max}}{N_{stepping}} \cdot 60s$$

For every 15 min cycle:

- $E_{sum} = 0$
- For every 1 min cycle (i):

$N_{stepping}$ = 1) {

If (E_{sum} < E_l)

then **the unit ON**

else **the unit is OFF**

} Else {

$$remaning\_steps = \frac{E_l - E_{sum}}{E_{stepping}}$$

$$op\_level = round\_closest\_integer\left(\frac{remaning_{steps}}{T_{MPC} - i \cdot 1min}\right)$$

Operate at *op_level* for 1 minute.

}

At the end of the 1 min cycle:

$$E_{sum} = E_{sum} + E_{ex,i}$$

5 M.I.5 -- Module for issuing commands towards the controllable PV inverters based on the commanded energy exchange signals

5.1 Inputs

- Commanded energy exchange, E_{PV}
    - $[E_{PV}]$ = kWh

  PV system energy exchange of the last minute, E_{ex}
    - $[E_{ex}]$ = kWh
- MPC sampling time
    - $[T_{MPC}]$ = s

Resolution: 1 / 15 min

5.2 Outputs

- Commanded power output for solar inverters, $p_{PV,out}$
    - $[p_{PV,out}]$ = % (of the nominal power of the inverter)

5.3 Frequency of submodule calls:

1/min

5.4 Algorithm

For every 15 min cycle:

- E_{sum}=0
- For every 1 min cycle (i):
    - E_{sum}=E_{sum}+$E_{ex,i}$
    - If E_{sum}< $E_{PV,ex}$ then inverter to maximum output
        - else inverter output to zero

This algorithm gives the safest approach to energy commands. It gives the highest possibility to fulfilling the energy commands. Also it is not very grid-friendly.

6 M.I.6 -- Module for CHP plant interfacing

The output of the MPC is the heat energy that the CHP should produce in the time frame. These values are stored as records in database. Each command is one record with time when command should be applied. Time granularity is 15 minute interval in prediction horizon. Output of the module is a power demand value, sent to CHP controller, in order to obtain the right amount of energy. In real life, CHP unit cannot follow desired power setpoint changes very fast. To manage this issue, desired power is recalculated every minute, based on energy remaining to be produced in time interval.

6.1 Inputs

- Energy command from the microgrid MPC module: [E_{MPC_el}] = kWh
- Electric energy produced in the last minute: [E_{out_E}] = kWh
- CHP max power: P_{el_max} = kW
- MPC sampling time [T_{MPC}] = min
- Power gain factor: k
- „valid from“ timestamp

6.2 Outputs

- Power command: [P_{el}] = kW

6.3 Internal parameters

- Controllable load electric energy exchange of the last interval: [E_{sum_el}] = kWh
- Energy to produce till the end of current MPC interval: [E_{remain_el}] = kWh
- The current minute count in current MPC interval: [T]
- Current time

6.4 Frequency of submodule calls:

1/min

6.5 Algorithm

For every 1 minute cycle:

$E_{sum\_el} = E_{out\_e} + E_{sum\_el}$

$E_{remain\_el} = E_{MPC\_el} - E_{sum\_el}$

$P_{el} = 0$

IF ($E_{remain\_el} > 0$):

$P_{el} = E_{remain\_el} / (T_{MPC} - T) * k$

$T = T + 1$

For every 15 min cycle:

$E_{sum\_el} = 0$

$T = 0$

Bibliography

[1] Simplified Extended Kalman Filter Model for SOC Estimation of Commercial Power-Oriented LFP Lithium Battery Cells - https://uk.mathworks.com/company/newsletters/articles/simplified-extended-kalman-filter-model-for-soc-estimation-of-commercial-power-oriented-lfp-lithium-battery-cells.html - Website viewed on 14th of July, 2017.

[2] How to charge Lithium Iron Phosphate Rechargeable Lithium Ion Batteries - https://www.powerstream.com/LLLF.htm - Website viewed on 14th of July, 2017.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D5.4.3.

Final grid-side energy management software module – Short-Term Day-Ahead Module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	5
Task	4.3
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Mostar Faculty of Mechanical Engineering, Computing and Electrical Engineering
Authors (Partners)	Tomislav Capuder (UNIZGFER), Mirna Gržanić (UNIZGFER), Martin Bolfek (HEP ODS) , Paula Perović (UNIZGFER)
Contact person	Paula Perović (UNIZGFER)
Abstract (for dissemination)	The document explains all variables and input data that are used for Short-term Day-ahead module inputs; it describes the final outputs of the module
Keyword List	Energy management, Demand response, SCADA, Flexibility provider

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on 5.2.1	Tomislav Capuder, Paula Perović (UNIZGFER)
v0.2	12 December 2018	Draft version	Tomislav Capuder, Paula Perović (UNIZGFER)
v0.3	10 January 2019	Draft version	Paula Perović (UNIZGFER)
v0.4	11 January 2019	Draft version	Mirna Gržanić, Tomislav Capuder (UNIZGFER)
V0.5	15 June 2019	Draft version	Paula Perović (UNIZGFER), Mirna Gržanić (UNIZGFER)
V0.6	24 June 2019	Final version	Tomislav Capuder (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Hrvoje Pandžić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

One of the objectives of the project Smart Building – Smart Grid – Smart City (3Smart), is creation of an integrated and modular energy management tool for the DSO to use buildings as their assets and to utilize their flexibility in order to more efficiently plan investment into the distribution grid. The tool is organized in two main submodules, Long-term and Short-term module. The Long-term module is used by DSO for offline planning and it allows DSO to plan their investments based on reserving buildings flexibility services and substituting CAPEX with higher OPEX paid to the building for providing services. Long-term Annual module calculates the DSO needed flexibility and Multi-Annual module calculates the flexibility service fee (activation, reservation and penalty prices). On the other hand, the short-term module is used by the DSO to optimize usage of flexibility services and schedule. Day-ahead module is used for determining time windows for utilizing daily flexibility coming from reservation windows in the long-term contract and the intra-day module is used for improving daily schedule.

The focus of deliverable 5.4.3 is on energy management tools interfaces. Deliverable 5.4.3. has 4 outputs, one for each submodule. The deliverable explains every variable that is used as submodule inputs and describes the final output of the modules. Database outlook is described in Annex 1 while submodules algorithms and logic are provided in Annex 2. This document presents Short-term Day-ahead module interface tables.

1 Introduction

Short-term day-ahead module is constructed to utilize the contracted flexibility reserve in the long-term module. The module is used for day-to-day operations for optimizing the usage of building flexibility potential as the distribution network/system operator asset. It is cast as AC OPF algorithm to define how much of maximal reserved capacity from the long-term contract will be activated in the next day (and for how long). Day-ahead flexibility is based on load predictions, up to 36 hours before realization of utilization. Database scheme is provided in Annex I, and detailed logic and algorithms are described in Annex II.

1.1. Day-ahead module interface tables

Day-ahead interface tables are described in Table 1 and Table 2. Interface tables are compatible with DSO database tables “info_grid”, “ac_opf_module_load_input”, “building_flexibility_table” ,“ac_opf_module_results” and “dso_to_building_da_flexibility_activation_profiles”.

Table 1: Day-ahead input interface table

Variable name	Variable annotation	Variable description	Source
Active power matrix	p_mat	Active power matrix contains calculated daily active power profiles for all nodes in the the grid. Matrix dimensions are number of rows x number of time intervals (96)	Neplan, SCADA, DSO replacement curves or PowerFactory (actual measurements from AMR)
Reactive power matrix	q_mat	Reactive power matrix contains calculated daily reactive power profiles for all nodes in the the grid. Matrix dimensions are number of rows x number of time intervals (96)	Neplan, SCADA, DSO replacement curves or PowerFactory (actual measurements from AMR)
Grid description table	grid_description_table	Grid description table contains infromations where every row describes one lines with Line Name, Node From, Node To, Lenght, Resistance in Ohm/km, Reactance in Ohm/km, Maximum Rated Current in Ampers	DSO data, simulation tools such as Neplan, PowerFactory or GREDOS (in case excel tables with data)
Building flexibility table	Building_flexibility_table	Building_flexibility_table defines flexibility reservation (minimum and maximum reserved capacity, when and how long)	Long-term Contract
Building predicted	Building_predicted_da_profile	Building predicted profile	Building MPC

profile based on Day Ahead market prices		based on Day Ahead market prices is proved by building and it presents building load prediction for next day, calculatd base on day-ahead market price	

Table 2: Day Ahead output interface table

Variable name	Variable annotation	Variable description
Voltage matrix	u_mat	Voltage matrix contains calculated daily voltage profiles for all nodes in the grid. Matrix dimensions are number of rows times number of time intervals (96)
Current matrix	i_mat	Current matrix contains calculated daily voltage profiles for all nodes in the the grid. Matrix dimensions are number of rows times number of time intervals (96)
DA flexibility activation profile	da_flexibilty_activation_profile	DA flexibility activation profile calculated by AC OPF.

Conclusion

Short-term Day-ahead module input interface includes grid topology, load profiles predictions from distribution management tool (or it can be created from replacement curves), flexibility reservations from the Long-term contract and the predicted building behaviour based on day-ahead market prices. The outputs are AC OPF results for voltage and current network behaviour and DA requested flexibility. Figure 1. depicts energy management submodules interconnection. Each module is described with table, the left column contains input data and the right column contains output data.

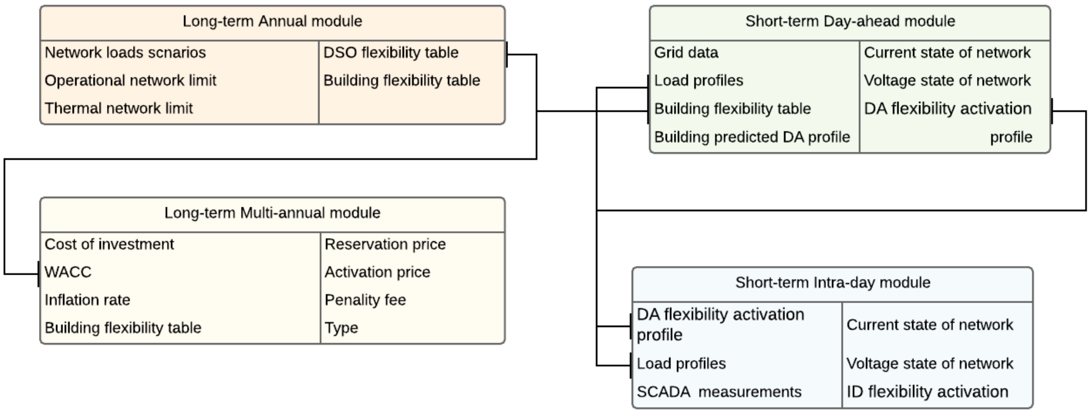


Figure 1. Energy managent submodules inteconnection

Annex I: Database outlook

DA module database structure and its relations are describe with Figure II.

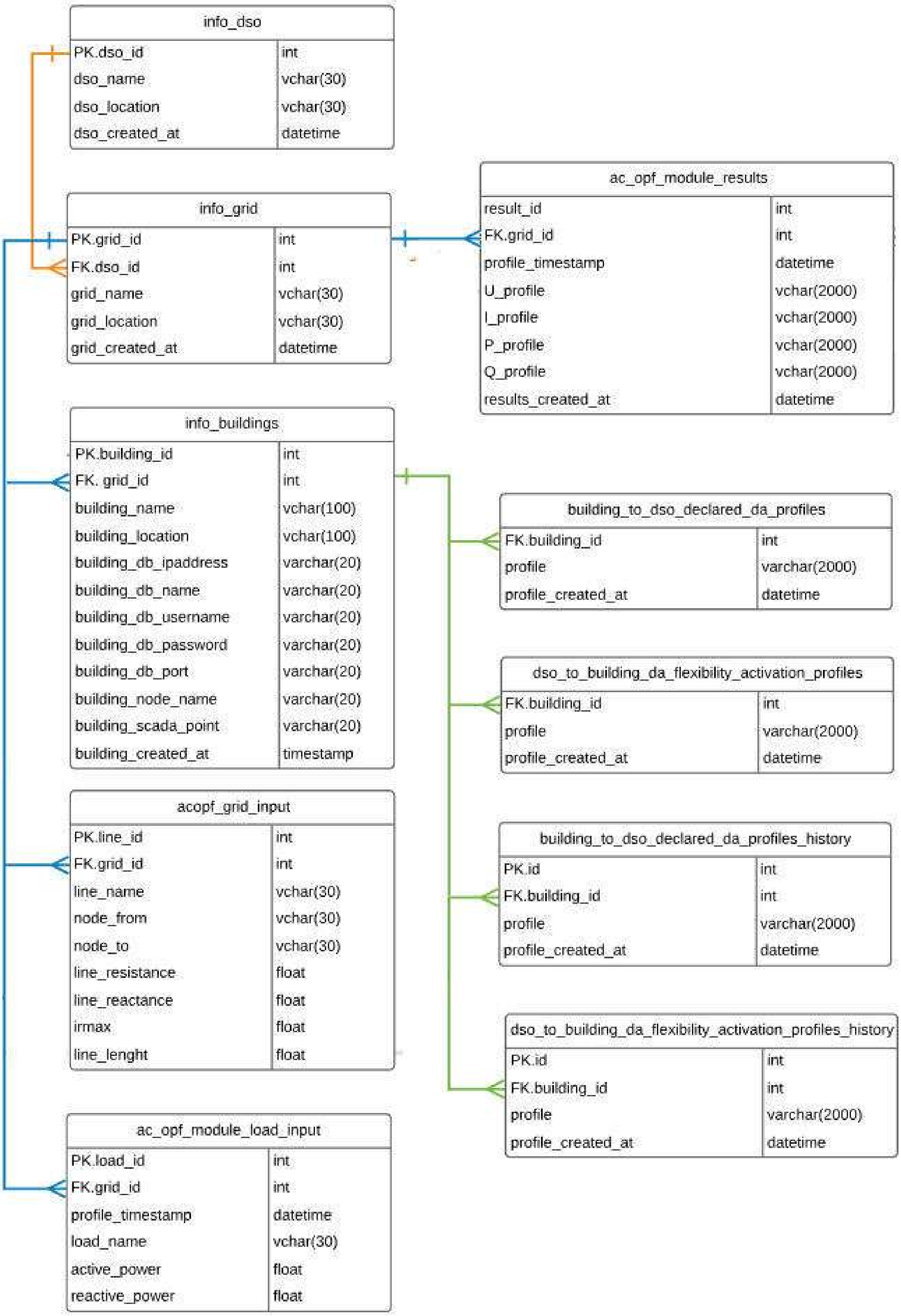


Figure I. Day-Ahead module database tables and relations

Database communication model is depicted with Figure III. The database communicates with other databases via TCP/IP protocol. The general idea of communication protocol is allowing entity (DSO, buildings and retailer) approach to database tables of other entities with python scripts, which presents one of the layers of regular programs (such as mpc, ac opf, day ahead module, intraday module). Entity can directly connect to other databases. Data privacy is guaranteed with entities access level to database. Every entity will have new account with read-only access to database. building will have read only access to the DSO and Retailer databases and vice-versa. For example, if the building has to deliver passive profiles to the DSO, the building just has to put the profiles into a certain table available to the DSO. The DSO will read this table at a certain time of day. After data is read, data timestamp will be checked to determine if the data is fresh. Even higher level of privacy can be guaranteed if the access is provided only for tables with needed data for specific entity.

The Python layer will connect to the database, read the data and write it to its own tables. The data structure of exchanged data will be formatted as JSON object. JSON data format is plain text which is easily readable by built-in Python JSON parsing libraries and easily convertible to other data types. Here is an example of a structure consisting of a 2x2 matrix A, 2x1 vector b and a description string:

Model = '{"A": [[0.027734449084562574, 0.2975643267384046], [0.6371333688566387, 0.38837159365408835]], "b": [[0.3884257461956241], [0.6990043851887695]], "description": "state space model matrices"}'

Data will be stored on principles of dictionary (key:value), and items are separated by “,”.

JSON data structure will be stored in database as varchar(2000) data type.

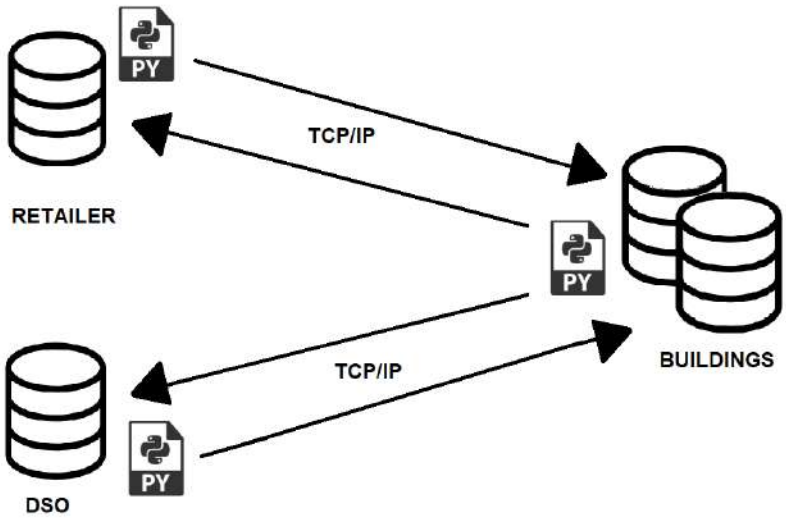


Figure II Communication model between building and DSO, and communication model between building and retailer. Entities will have read-only access to other database output tables. Python program will automatically connect and read data and write to its own database

Table I describes all communication events (DBD stands for day before delivery). Communication will be implemented with python scripts running on entity server defined with column ”Reads data” in table Scripts will read entity data from entities described in table’s column “Puts data at disposal” at a certain time described with column “Time”. General data exchange structure contains objects: the first is json object with 15 min resolution energy profiles and time when first time horizon occurs, and second object is timestamp when data was created. In case there are no fresh data in the table, scripts will repeat the check 10 minutes later. Table II contains json definitions and data examples for non-scalar data. Measurement units may be changed if necessary.

Table I Chronological communication events for DA module with detailed exchange structure

Time (UTC)	Data exchange	D.5.2.1 Nomenclature	Reads data	Puts data at disposal
DBD, 10:00	Informative DA profiles (json), profile created at (timestamp)	Informative DA schedule of the building	Retailer	Building
DBD, 13:15	DA prices (json), Profile created at (timestamp)	DA prices	Building	Retailer
DBD, 14:00	Declared DA profile (json), profile created at (timestamp)	DA schedule of the building	Retailer	Building
DBD, 14:00	Declared DA profile (json), profile created at (timestamp)	DA schedule of the building	DSO(DA)	Building
DBD, 14:15	DA flexibility activation profile (json), profile created at (timestamp)	DA flexibility profile by the DSO	Building	DS0(DA)

Table II Json definitions for dat exchanged in DA module

Communication event ID	JSON DATA
General definition	{“profile name”: [vector – float – 96 values], “Measuring unit”: “mu”, “Valid from”:“(timestamp yyyy-mm-dd hh:mm:ss)”}
1	{"Informative_DA_profile": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,- 4.75474499685196,-7.05352031484420,- 5.04552454338058,1.98883041147715,15.5199743752825,30.2243591989718,4 2.3750160963821,48.6996128188282,44.0724614014147,33.4154602577768,19 .8069030463219,6.79379492247743,0.346017609948462,-2.30051566857788,- 2.10296945666405,-0.70158591015846,-1.68728914210829,- 3.48502195975458,-5.93454746246068,-8.55606118169379,- 9.97439866415701, -10.9480980304388,-11.5617755279102,- 11.9988086060426,-12.5542121763336,-13.3104639796222,- 14.3500838192498,-15.6410980791624,-17.3420595721479,- 18.6963549900087,-19.2700190856583,-18.7612213674154,- 16.5841650293630,-13.7369740391108,-10.8515696391588,- 8.53751494017917,-7.82298188488800,-7.89979480781464,- 8.35233441647422,-8.69405392448411,-7.96185434358636,- 6.81912958501434,-5.52032972642133,-4.37530285097900,- 4.09029359681548,-4.08694862794924,-4.14444794247938,- 4.02488936856972,-3.20115730971145,-2.17170873726020,- 1.15928853152464,-0.396016163235099,-0.323319434636732,- 0.573300018627128,-0.967378458249253,-1.29913927110666,-

	1.18648307394541,-0.864926954692977,-0.427156949382357,0,0.167044947284598,0.196985558515131,0.122075657712569,0,-0.0384652719847160,-0.0485771390292825,-0.0316919637025193,0,0.00672663117077245,0.00880493437779292,0.00592319087065342,0,0,0,0], "Measuring_unit": "kW", "Valid_from": "2018-09-17 2:00:00"}
2	{"DA_prices": [0.05207, 0.05207, 0.05207, 0.05207, 0.05203, 0.05203, 0.05203, 0.05203, 0.05001, 0.05001, 0.05001, 0.05001, 0.05201, 0.05201, 0.05201, 0.05201, 0.05201, 0.05201, 0.05201, 0.05201, 0.05202, 0.05202, 0.05202, 0.05202, 0.07604, 0.07604, 0.07604, 0.07604, 0.07892, 0.07892, 0.07892, 0.07892, 0.07891, 0.07891, 0.07891, 0.07891, 0.07646, 0.07646, 0.07646, 0.07646, 0.07842, 0.07842, 0.07842, 0.07842, 0.07897, 0.07897, 0.07897, 0.07897, 0.07791, 0.07791, 0.07791, 0.07791, 0.0774, 0.0774, 0.0774, 0.0774, 0.07849, 0.07849, 0.07849, 0.07849, 0.08001, 0.08001, 0.08001, 0.08001, 0.0801, 0.0801, 0.0801, 0.0801, 0.09, 0.09, 0.09, 0.09, 0.08915, 0.08915, 0.08915, 0.08915, 0.09572, 0.09572, 0.09572, 0.09572, 0.08786, 0.08786, 0.08786, 0.08786, 0.07715, 0.07715, 0.07715, 0.07715, 0.07499, 0.07499, 0.07499, 0.07499, 0.04648, 0.04648, 0.04648, 0.04648], "Measuring_unit": "EUR/kWh", "Valid_from": "2018-10-22 22:00:00"}
4, 5	{"Declared_DA_profile": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-4.40881457120924,-6.55389304153656,-4.71567490719656,1.80858823194224,14.2292798223350,28.0499605547677,39.9898919078737,47.0781254403744,44.9627637978742,37.1946817818429,26.0879184797042,14.5324136930805,7.16204232894512,2.80565510662295,1.34244968217499,1.93025454387579,2.59541455847064,3.23638385255257,3.16861489241684,2.01281679068190,-0.504769000907987,-3.67899371588490,-6.86867892635720,-9.44771190741795,-10.2758631035784,-10.3660853060221,-10.2345200069028,-10.4410650451395,-12.1245077838722,-14.3034068037717,-16.5210240279394,-18.2619054356731,-18.5806791482442,-18.0898428911675,-17.0505569358873,-15.8125466825114,-15.0399250698480,-14.4384015745078,-13.9447774650335,-13.4279366704656,-12.5725771324292,-11.5245083989525,-10.2992369435445,-8.94509047241126,-7.61639367907759,-6.25332636192884,-4.89183184662384,-3.58120121353285,-2.29945807622209,-1.25126521897513,-0.538192266877596,-0.216249124347829,-0.387578120586701,-0.799564497800280,-1.27212876817409,-1.62955619555073,-1.61616034188853,-1.37332137608418,-0.977899964219016,-0.540682239998261,-0.260419166637827,-0.0815193778964 04,0,0,0,0,0,0,0,0,0,0,0,0], "Measuring_unit": " kW", "Valid_from": "2018-10-22 22:00:00"}

6	{"DA_flexibility_activation_profile": [0,0.606574912975372,1.02600697833224,0.901944678545856,0,- 2.27583404120944,-4.78750458145718,-6.86169525355939,- 7.89602137707201,-6.88349111270873,-4.82005198123357,- 2.30099744070119,0,0.963681997196315,1.12643135531563,0.6924897481065 69,0,-0.246250131913775,-0.311559428811153,- 0.203594181209020,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.0392156399009166,0.05 82947845934666,0.0445349730406702,0,-0.221776913727023,- 0.339384397697267,- 0.268242412118061,0,0.754335110221086,1.22703148009311,1.049747187681 53,0,-2.51646497247400,-5.26534283912386,-7.50956736614412,- 8.60120481504233,-7.41812971373539,-5.12880048916783,- 2.41089986671777,0,0.790747739909002,0.805732681406043,0.393906517906 737,0,0.591157743760178,1.46158826336528,2.35136785391709,2.953206429 19753,2.61767974068478,1.85926744565602,0,0,-0.360428592314349,- 0.421298798687064,- 0.258999447772524,0,0.0921004944173757,0.116526952537633,0.0761466587 007446,0,-0.0185005333725646,-0.0247864383567575,- 0.0171095030911426,0,0.0181509673970601,0.0277763588380361,0.02195385 98271621,0,-0.0617373186504593,-0.100424376983395,- 0.0859148351311970,0,0.240854438581268,0.482810606928905,0.6542404483 57125,0.703951486877629,0.409762564606634,0.117735784028214,- 0.04139617671033],"Measuring_unit": " kW", "Valid_from": "2018-10-22 22:00:00"}

Annex II: Day-ahead module logic and algorithm

The first step in utilizing the needed flexibility is through day-ahead load forecast, which must be specific to grid zones specified by the EMS. The load forecast is used to determine the need for demand response services together with the estimated period of service. The grid side EMS module will compare the predicted load, which is based on predefined data (namely the load profiles from the customers, network parameters either automated from NEPLAN of inputted and .csv parameters, etc.), against the network limits set in a predefined table of the respective grid scenario. The outcome is an estimated flexibility need for the next day. In general, the sequence of events defining the need for next day flexibility for the DSO, utilized from the flexibility provider (in this case the building), is defined as follows (and presented in Figure III.):

- **Passive profiles (or those not coming from optimization based on DA market prices) are used by the retailer to bid in the power exchange and by the DSO to plan the next day operation.** This is shown in Figure III. By arrows "Informative DA schedule of the building" going from the building to both the retailer and the grid side EMS. In the project we do not focus on the retailer side of market bidding on a DA ahead market. For the DSO (grid-side EMS) this aspect is covered by historical recorded consumption data of the building. Additionally, the building can send assumed profiles which are not based on optimization driven by historical DA market prices.
- **The retailer sends the DA price profile to the flexible user – the building**.
- **The building computes its optimal schedule based on the prices from the day-ahead market, its current conditions and predictions as well as the long-term contract for flexibility provision**. In Figure III. these two steps are presented with the arrows of DA prices being provided to the building by the retailer (or an aggregator). The project does not focus on how these prices are defined by the retailer, however it assumes they reflect the DA prices on the power exchange.

  This schedule/profile will, at least at the beginning when the DSO does not have sufficient information to forecast behaviour of flexible buildings, be different than that predicted by the DSO who considers buildings to be passive consumers (note: the DSO does not know or has information on how the EMS of the building works nor how it will schedule its consumption based on market prices. In case the DSO has adequate tools and enough historical data on building flexible scheduling it can adjust its forecasting tool).
- **The building communicates the DA price driven optimal schedule to the DSO (grid side EMS)**. This information is a valuable source of information for the DA grid side EMS module in defining needed flexibility for the next day. In Figure III. This is referred to as "DA schedule with flexibility". Unlike the Informative DA schedule profile, which is based on historical data and not on knowledge of buildings capability to flexibly schedule its actuators based on DA prices, this is the profile that defines electricity exchange between the building and power system (distribution network).
- **The same DA schedule is also sent to the retailer to assist him plan its market position** (arrow indicating "DA schedule of the building) such as procurement of balancing energy etc (again presented by arrows on the retailer side). Potentially, the need for self-balancing of the retailer/aggregator portfolio could again be procured from the building (flexibility provider). This aspect is not currently in the focus of the 3Smart project, however in Figure

III. it is shown as arrows “Request for flexibility” and “Flexibility availability with prices” in grey colour.

- **DSO runs the AC Optimal Power Flow calculation (AC OPF) based on the latest data from the network users (this is DSO grid-side EMS tool).** In addition to the information received from the network users (such as flexible buildings) and own field operators (such as scheduled maintenance or network upgrades/reconnections) the grid-side EMS relies on the flexibility contracts from the long-term module which constrain both the time windows and the power which the grid side EMS can demand from the flexibility provider.
- **Calculated flexibility needs are communicated with the building**. The calculated profile determines set points for the building to provide DSO flexibility based on long-term contract defined values (in terms of time windows and power). At this point the operational points for the next day of the building are defined based on knowledge and information available at that point (day before the delivery). These needs are defined as 15-minutes time steps of energy exchange profile (meaning the DSO activates long term contracts as time windows for providing the flexibility and power for each time step). This is shown in Figure III. with the arrow “DA flexibility profile by the DSO”. The prices for the building to provide the service are defined in the long-term contract.
- **Modified DA schedule with DSO flexibility needs are sent to the retailer**. This is done in order to provide the information to the retailer in order to be able to assign deviations/imbalances from those defined at the closure of the DA in power exchange. In Figure III. Arrow “DA profile with DSO flex.req.” presents this aspect. Information about requested/procured flexibilities and deviations from the DA market schedule should also be sent to the Transmission System Operator.

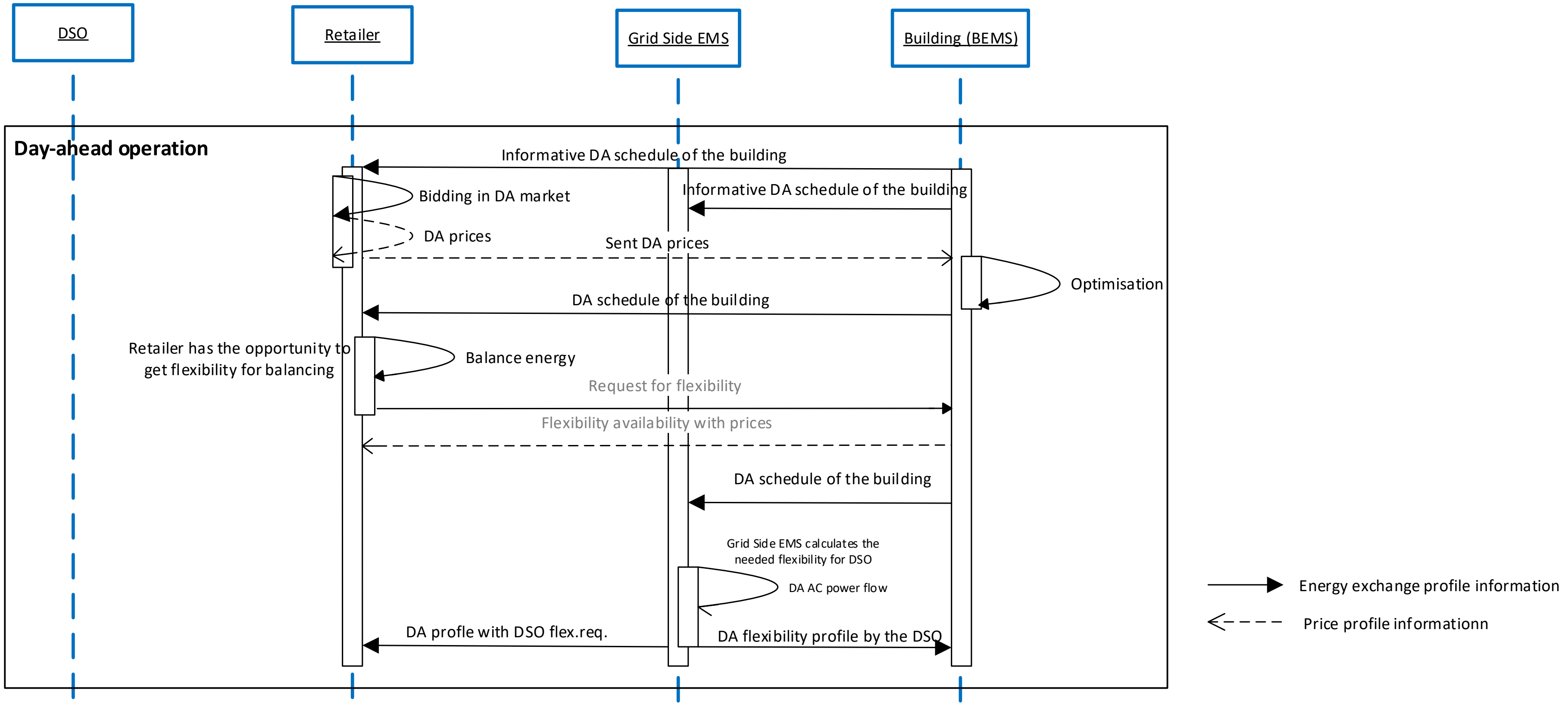


Figure III: Day-ahead operation diagram

Annex II: AC Optimal Power Flow

To model any power systems network, one must understand physical properties and mathematical representations of network elements. When analysing load flows (magnitude and direction of power flowing through the network) key elements that need to be modelled are power lines (both cables and overhead lines) and transformers. Most common way of representing line elements is a PI scheme, presented in Figure IV:

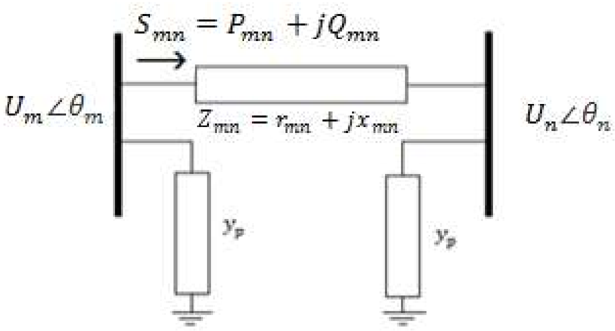


Figure IV PI section of line

Apparent power S_{mn} is a complex variable expressed with real part (active power P_{mn}) and imaginary part (reactive power Q_{mn}) where j satisfies $j^2 = -1$. Voltages $U_m\angle\theta_m$ and $U_n\angle\theta_n$ are presented as complex variables with amplitudes U_m, U_n and phasors θ_m, θ_n. The polar form of voltage at busbar m is $U_m cos\theta_m + jU_m sin\theta_m$. If voltage (or any other variable/constant) is expressed just as $\boldsymbol{U_m}$, it represents a complex variable with neglected phasor due to the simplicity. $\boldsymbol{Z_{mn}}$ is a complex constant representing impedance of the line between busbar m and n, composed of real part resistance r_{mn}, and reactance x_{mn} as an imaginary part. Admittance is express with (1):

$$\boldsymbol{Y_{mn}} = \frac{1}{\boldsymbol{Z_{mn}}} = \frac{1}{r_{mn} + jx_{mn}} = \frac{r_{mn} - jx_{mn}}{r_{mn}^2 + x_{mn}^2} = g_{mn} - jb_{mn} \#(1)$$

Real part of admittance is a conductance g_{mn} :

$$g_{mn} = \frac{r_{mn}}{r_{mn}^2 + x_{mn}^2} \#(2)$$

while imaginary part is a susceptance b_{mn} (3):

$$b_{mn} = \frac{x_{mn}}{r_{mn}^2 + x_{mn}^2} \#(3)$$

y_p is a shunt admittance of the line.

Current I_{mn} is defined as (4):

$$\boldsymbol{I_{mn}} = (\boldsymbol{U_m} - \boldsymbol{U_n}) \cdot \boldsymbol{Y_{mn}} \#(4)$$

Apparent power S_{mn} is equel to (5), where * marks complex conjugated variable (or a number):

$$\boldsymbol{S_{mn}} = \boldsymbol{U_m} \cdot \boldsymbol{I_{mn}}^* = \boldsymbol{U_m} \cdot (\boldsymbol{U_m} - \boldsymbol{U_n})^* \cdot \boldsymbol{Y_{mn}}^* \#(5)$$

Incorporating expresions (1-4) in (5), apparent power is presented with (6):

$$\boldsymbol{S_{mn}} = (g_{mn} + jb_{mn}) \cdot (U_m^2 - U_m U_n \cos(\theta_m - \theta_n) - jU_m U_n \sin(\theta_m - \theta_n))\#(6)$$

AC power flow relaxation in distribution networks

Methods used for converting non-convex AC power flow model to convex relaxations are:

- *Ben-Tal, Network* and *Copper plate* which are linear relaxation methods that significantly extend the scope of the solution and do not guarantee global optimal solution;
- *Second Order Cone Programming* relaxation (*SOCP*) first time presented in [6];
- *Semidefinite Programming* (*SDP*) relaxation explained in [7-8].

[9-11] show the relaxation accuracy of the relaxations for radial network.

DistFlow model is used for *Alternating Current Optimal Power Flow* (AC OPF) calculations in this Deliverable. Grid model is presented in Figure 4.3:

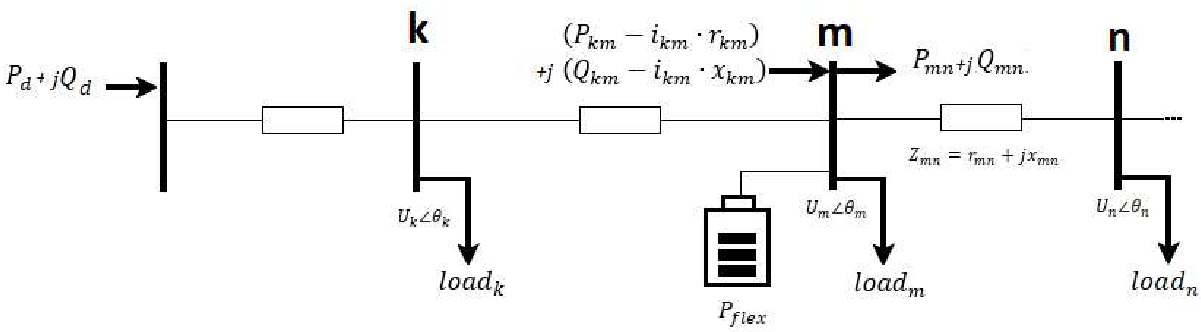


Figure III Grid model

The model is based on the quadratic Kirchhoff Voltage Law (10-11) and the current on the line mn is calculated as (12):

$$\boldsymbol{U_{n,t}^2} = |\boldsymbol{U_{n,t}}|^2 = |\boldsymbol{U_{m,t}} - \boldsymbol{I_{mn,t}}\boldsymbol{Z_{mn}}|^2 \#(10)$$

$$|\boldsymbol{U_{n,t}}|^2 = |\boldsymbol{U_{m,t}}|^2 - 2(r_{mn}P_{mn,t} + x_{mn}Q_{mn,t}) + \\ + |\boldsymbol{I_{mn,t}}|^2 (r_{mn}{}^2 + x_{mn}{}^2)\#(11)$$

$$|\boldsymbol{I_{mn,t}}|^2 = \frac{|\boldsymbol{S_{mn,t}}|^2}{|\boldsymbol{U_{m,t}}|^2} \#(12)$$

Where $\boldsymbol{U_{m,t}}$ and $\boldsymbol{U_{n,t}}$present voltage of the bus m and n, , $\boldsymbol{I_{mn,t}}$ current on the line mn flowing from bus m to n, $\boldsymbol{Z_{mn}}$ impedance of the line mn, r_{mn} resistance, x_{mn} reactance, $P_{mn,t}$ active power and $Q_{mn,t}$ reactive power flowing from bus m to n.

Equations listed above are non-linear and non-convex and thus cannot be solved using commercial solvers. *Second Order Cone Programming* (*SOCP*) relaxation is a convex relaxation of DistFlow model and can be solved with commercial solvers. *SOCP* relaxations of the above problem are presented with (13-14):

$$u_{n,t} = u_{m,t} - 2\left(r_{mn}P_{mn,t} + x_{mn}Q_{mn,t}\right) + \\ +i_{mn,t}\left({r_{mn}}^2 + {x_{mn}}^2\right)\#(13)$$

$$P_{mn,t}^2 + Q_{mn,t}^2 = i_{mn,t}u_{m,t}\#\#(14)$$

Where absolute value of quadratic variables of voltage $\left|\boldsymbol{U_{m,t}}\right|^2$ and $\left|\boldsymbol{U_{n,t}}\right|^2$, as well as current $\left|\boldsymbol{I_{mn,t}}\right|^2$ are replaced with linear variables (15-17):

$$\left|\boldsymbol{U_{m,t}}\right|^2 = u_{m,t}\#(15)$$

$$\left|\boldsymbol{U_{n,t}}\right|^2 = u_{n,t}\#(16)$$

$$\left|\boldsymbol{I_{mn,t}}\right|^2 = i_{mn,t}\#(17)$$

Because of non-convexity, (14) is relaxed as (19):

$$P_{mn,t}^2 + Q_{mn,t}^2 \leq \ i_{mn,t}u_{m,t}\#\#(19)$$

The voltage at each node n and current on the line mn are limited with (20-21):

$$0.81u^{nominal} \leq u_{n,t} \leq 1.21u^{nominal}\#(20)$$

$$i_{mn,t} \leq I_{MAX}^2\#(21)$$

Upper and lower voltage bounds are determined in grid codes and may differ among the countries (here ±10% of nominal value is used).

The active and reactive power balance of load buses are shown in (22) and (23).

$$load_{m,t}^{active} + P_{flex} = \sum_{k\in K}\left(P_{km,t} - i_{km,t}\cdot r_{km}\right) - \sum_{n\in N}\left(P_{mn,t}\right)\ \#(22)$$
$$load_{m,t}^{reactive} = \sum_{k\in K}\left(Q_{km,t} - i_{km,t}\cdot x_{km}\right) - \sum_{n\in N}\left(Q_{mn,t}\right)\ \#(23)$$

Where $load_{m,t}^{active}$ and $load_{m,t}^{reactive}$ present inflexible or must-served load of the consumer. P_{flex} is required flexibility from the DSO.

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D5.4.3

Final grid-side energy management software module – Short-Term Intra-Day Module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	5
Task	4.3
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	University of Zagreb Faculty of Electrical Engineering and Computing (UNIZGFER), University of Mostar Faculty of Mechanical Engineering, Computing and Electrical Engineering
Authors (Partners)	Tomislav Capuder (UNIZGFER), Mirna Gržanić (UNIZGFER), Martin Bolfek (HEP ODS) , Paula Perović(UNIZGFER)
Contact person	Paula Perović(UNIZGFER)
Abstract (for dissemination)	The document explains all variables and input data that are used for Short-term Day-ahead module inputs; it describes the final outputs of the module
Keyword List	Energy management, Demand response, SCADA, Flexibility provider

Revision history

Revision	Date	Description	Author (Organization)
v0.1	1 September 2018	Initial version based on 5.2.1	Tomislav Capuder, Paula Perović (UNIZGFER)
v0.2	12 December 2018	Draft version	Tomislav Capuder, Paula Perović (UNIZGFER)
v0.3	10 January 2019	Draft version	Paula Perović (UNIZGFER)
v0.4	12 January 2019	Draft version	Mirna Gržanić, Tomislav Capuder (UNIZGFER)
V0.5	14 June 2019	Draft version	Paula Perović (UNIZGFER), Mirna Gržanić (UNIZGFER)
V0.6	24 June 2019	Final version	Tomislav Capuder (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Hrvoje Pandžić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

One of the objectives of the project Smart Building – Smart Grid – Smart City (3Smart), is creation of an integrated and modular energy management tool for the DSO to use buildings as their assets and to utilize their flexibility in order to more efficiently plan investment into the distribution grid. The tool is organized in two main submodules, Long-term and Short-term module. The Long-term module is used by DSO for offline planning and it allows DSO to plan their investments based on reserving buildings flexibility services and substituting CAPEX with higher OPEX paid to the building for providing services. Long-term Annual module calculates the DSO needed flexibility and Multi-Annual module calculates the flexibility service fee (activation, reservation and penalty prices). On the other hand, the short-term module is used by the DSO to optimize usage of flexibility services and schedule. Day-ahead module is used for determining time windows for utilizing daily flexibility coming from reservation windows in the long-term contract and the intra-day module is used for improving daily schedule.

The focus of deliverable 5.4.3 is on energy management tools interfaces. Deliverable 5.4.3. has 4 outputs, one for each submodule. The deliverable explains every variable that is used as submodule inputs and describes the final output of the modules. Database outlook is described in Annex 1 while submodules algorithms and logic are provided in Annex 2. This document presents Short-term Day-ahead module interface tables.

1 Introduction

Short-term AC OPF module calculation to determining building flexibility profile for next day is based on predicted load profiles and building profile based on forecast. Since this advanced decision can deviate from actual events, the actual need for flexibility can be compensate (more precisely estimated and activated) with Short-term Intra-day module. ST ID Module presents intra-day operations that allow DSO to improve day-ahead schedule with real time measurements and prices.

Database scheme is provided in Annex I, and detailed logic and algorithms are described in Annex II.

1.1. Intra-Day module interface tables

Intra-day interface tables are described in Table 1 and Table 2. Interface tables are compatible with DSO database table "id_triggering_module".

Table 1: Intra-day input interface table

Variable name	Variable annotation	Variable description
Day-ahead flexibility profile activation	Da_flexibility_activation_profile	Building flexibility activation profile calculated in day-ahead module.
Predicted feeder consumption profiles	Predicted_feeder_consumtion	Substation power consumption profile based on power profiles that are used as input for AC OPF calculation
SCADA measured feeder consumption	Scada_measured_consumtion	SCADA measured feeder consumption is an average power measured on feeder for time interval of 15 minutes
Timestamp of SCADA measurements	End_of_measurement_period	Timestamp that describes time horizon of measured data.

Table 2: Intra-day output interface table

Variable name	Variable annotation	Variable description
ID flexibility activation	id_flexibility_activation_request	ID flexibility value calculated for next time horizon
Timestamp of flexibility activation	id_flexibility_activation_request _timestamp	Timestamp defining when building should activate ID flexibility

Conclusion

ST ID module calculates total predicted consumption from the close to real-time SCADA (or smart meter) measurements and DA flexibility activation profiles. From DSO SCADA, ST ID receives real-time measurements that are averaged to 15 minute interval. The final output is ID flexibility activation (Figure I)

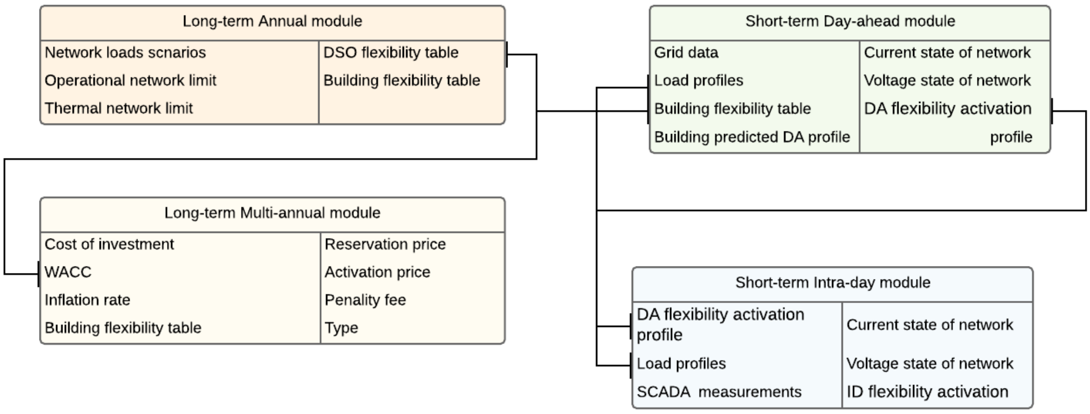


Figure I. Energy managent submodules inteconnection

Annex I

ID module database structure and it's relations are described with Figure II.

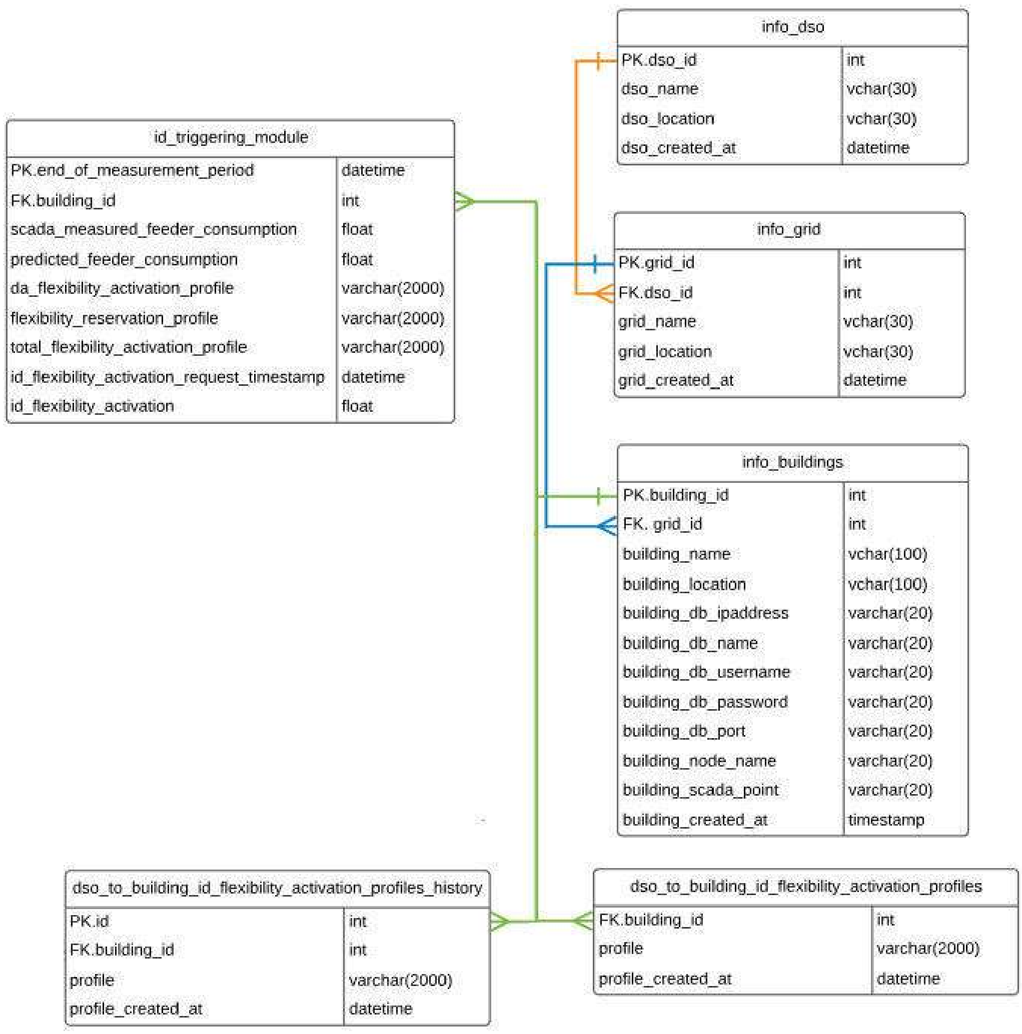


Figure II: Intra-Day module database tables and relations

Database communication model is depicted with Figure III. The database communicates with other databases via TCP/IP protocol. The general idea of communication protocol is allowing entity (DSO, buildings and retailer) approach to database tables of other entities with python scripts, which presents one of the layers of regular programs (such as mpc, ac opf, day ahead module, intraday module). Entity can directly connect to other databases. Data privacy is guaranteed with entities access level to database. Every entity will have new account with read-only access to database. building will have read only access to the DSO and Retailer databases and vice-versa. For example, if the building has to deliver passive profiles to the DSO, the building just has to put the profiles into a certain table available to the DSO. The DSO will read this table at a certain time of day. After data is read, data timestamp will be checked to determine if the data is fresh. Even higher level of privacy can be guaranteed if the access is provided only for tables with needed data for specific entity.

The Python layer will connect to the database, read the data and write it to its own tables. The data structure of exchanged data will be formatted as JSON object. JSON data format is plain text which is easily readable by built-in Python JSON parsing libraries and easily convertible to other data types. Here is an example of a structure consisting of a 2x2 matrix A, 2x1 vector b and a description string:

Model = '{"A": [[0.027734449084562574, 0.2975643267384046], [0.6371333688566387, 0.38837159365408835]], "b": [[0.3884257461956241], [0.6990043851887695]], "description": "state space model matrices"}'

Data will be stored on principles of dictionary (key:value), and items are separated by “,”.

JSON data structure will be stored in database as varchar(2000) data type.

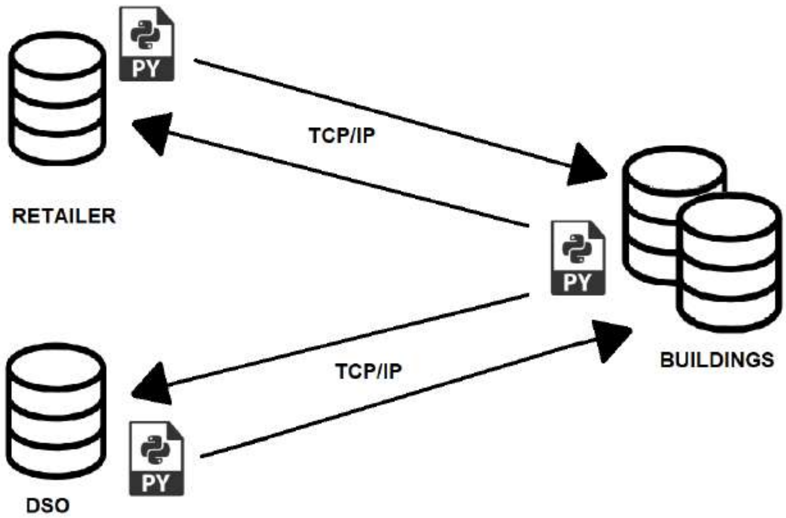


Figure III Communication model between building and DSO, and communication model between building and retailer. Entities will have read-only access to other database output tables. Python program will automatically connect and read data and write to its own database

Table I describes all communication events (DOD stands for day of delivery). Communication will be implemented with python scripts running on entity server defined with column ”Reads data” in Table I. Scripts will read entity data from entities described in table’s column “Puts data at disposal” at a certain time described with column “Time”. General data exchange structure contains objects: the first is json object with 15 min resolution energy profiles and time when first time horizon occurs, the second object is timestamp when data was created. In case there is no fresh data in the table, scripts will repeat the check 10 minutes later. Events will occur when values measured are higher than triggering values and only if it happens during reserved time windows contracted in advance (during long term contracting) or if the event happens before day-ahead defined time step for activation of flexibility (meaning the AC OPF predicted time t, but the event was on an ID triggered by measurements on 15-minute sample time or more and within the flexibility time window). Table II contains json definitions and data examples for non-scalar data. Measurement units may be changed if necessary.

Table I Chronological communication events for ID module with detailed exchange structure

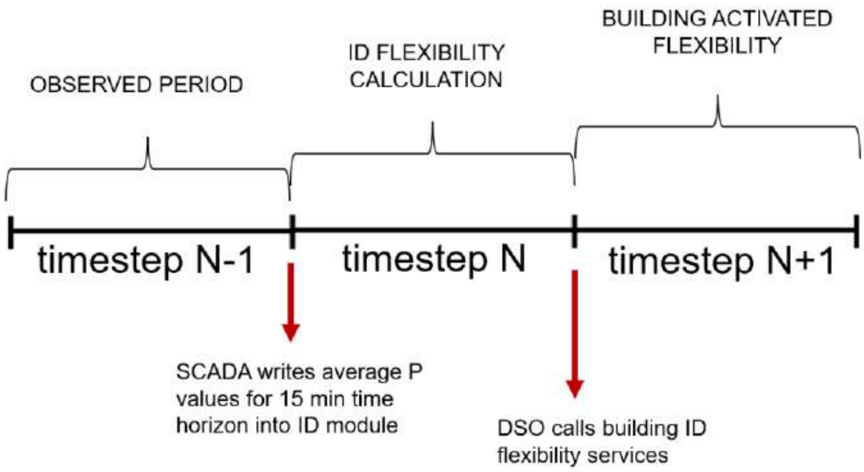
Time (UTC)	Data exchange	D.5.2.1 Nomenclature	Reads data	Puts data at disposal
All the time	SCADA measurements (float)	Scada measurements	DSO(ID)	DSO (SCADA)
DOD, every 15 minutes	ID flexibility activation (float)	Activation of needed flexibility	Building	DSO(ID)

Table II Json definitions for dat exchanged in ID module

Communication event ID	JSON DATA
General definition	{“profile name”: [vector – float – 96 values], “Measuring unit”: “mu”, “Valid from”:“(timestamp yyyy-mm-dd hh:mm:ss)”}
1	{"ID_flexibility_activation": [0.0392156399009166], "Measuring_unit": " kW", "Valid_from": "2018-10-22 22:00:00"}

Annex II

Intra-day operation is described with Figure IV. At the beginning of N-time interval, ID module gets SCADA measurements from the grid substation as an average value of N-1-time horizon. During N-time intervals, ID module compares measured and predicted consumption. If the measured value is greater than predicted consumption, than ID module checks if the building can provide flexibility service. If building flexibility service is reserved for the next time period in long term contract and it is not called by DA module, then ID module calculates ID flexibility activation (algorithm is described in Table III). Algorithm is based on extrapolation of real-time measurements; the extrapolated data is compared with DA predicted feeder consumption in order to check if the triggering events occurs earlier than forecasted in DA module. If the peek occurs earlier, id module cancels flexibility call by Day-ahead module.




```
#profiles
scada_measurements                                                                              =
predicted_feeder_consumption
da_flexibility profiles

from last 20 measurements:
        extrapolate  for next 3 periods

#for t in range(now, now+3 periods)
compare extrapolilate data with predicted_feeder_consumption
        if extrapolilate data < predicted_feeder_consumption:
                if da_flexibility_profiles [t+2] =! 0:
                        id_flexibility [t+2] = 0
                else:
                        id_flexibility [t+2] =   da_flexibility_profiles [t+2]
```

Figure IV: Intra-day module operations

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D5.4.3.

Final grid-side energy management software module-Long-term Annual module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	5
Task	5.4
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	EON
Authors (Partners)	Gábor Péter Mihály, Kata Sánta (EON), Tomislav Capuder, Paula Perović, Mario Vašak (UNIZGFER)
Contact person	Tibor Béni (E.ON)
Abstract (for dissemination)	The document contains final technical solution description for the long term grid-side energy management module of the 3Smart tool.
Keyword List	Grid-side flexibility tools, Energy Management system

Revision history

Revision	Date	Description	Author (Organization)
v0.1	20 December 2018	First draft	Katalin Décseiné Giczi (EON)
v0.2	20 June 2019	Draft	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON)
V0.3	24 June 2019	Draft revision	Paula Perović, Tomislav Capuder (UNIZGFER)
v0.4	27 June 2019	Draft version	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON)
v0.5	28 June 2019	Draft version	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON)
v0.6	29 June 2019	Final Draft version	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON), Paula Perović, Tomislav Capuder (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Hrvoje Pandžić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

One of the objectives of the project Smart Building – Smart Grid – Smart City (3Smart) is creation of an integrated and modular energy management tool for the DSO to use buildings as their assets and to utilize their flexibility in order to efficiently plan investment into the distribution grid. The tool is organized in two main submodules, Long-term and Short-term module. The Long-term module is used by the DSO for offline planning and it allows the DSO to plan their investments based on reserving building flexibility services by substituting CAPEX with higher OPEX paid to the building for providing services. Long-term Annual module calculates the DSO needed flexibility and Multi-Annual module calculates the flexibility service fees (activation, reservation and penalty prices). On the other hand, short-term module is used by the DSO to optimize usage of flexibility services and schedule. Day-ahead module is used for determining time windows for utilizing daily flexibility coming from reservation windows in the long-term contract and the intra-day module is used for improving daily schedule.

The focus of deliverable 5.4.3 is on energy management tools interfaces. Deliverable 5.4.3. has 4 outputs, one for each submodule. The deliverable explains every variable that is used as submodule inputs and describes the final output of the modules. Database outlook is described in Annex 1 while submodules algorithms and logic are provided in Annex 2. This document presents Long-term Annual module interface tables.

1 Introduction

An important part of the deliverable is the description of the long-term module functionalities from the DSO point of view, as well as that of a distributed demand response provider, in the market environment. Currently, neither the DSO or demand response providers participate in the market. The market, in this deliverable, is seen through participation of multiple stakeholders exchanging money and services in a transparent way either through tenders or, more preferably, at power exchange. This means that the DSO does not have any information of the accepted demand profiles from its users and operates the network based on vast experience and available historical data. On the other hand, not enabling market access to the distributed energy sources results in overbuilding and underutilization of the DSO assets.

Integrated tool for Long term energy management of building for DSO, installed with the 3Smart project, is divided in two modules: Annual and Multiannual.

Each submodule is presented via an interfacing table that explains which data is used by the submodules as inputs and what is the final output data. The algorithms behind are explained in more details in the annexed document.

1. Calendar submodule
2. Calculation input submodule
3. Flexibility calculation submodule
4. DSO flexibility table (Output)

The modules are designed, commissioned and tested on different pilot location.

In this deliverable the focus is put on Annual modules and its variables and algorithms

1.1. Long term Annual module interface tables

An important feature of the LT module is that inputs and outputs of the module are directly written from/in database of other modules; namely the DSO flexibility table serves as an input for the Multiannual module via database. The following Input/output table describes not only the database entities but the LT excel inputs and outputs as well.

Table 1. Long-term Annual input interface table

Variable name	Variable annotation	Variable description	Source
Calendar	Calendar	The Calendar contains the following columns: Date, Scenario Name-> here the DSO staff has to enter all days of the given year and inbuilt function seeks the type of day: WEEKDAY, SATURDAY, SUNDAY. These type of days relevant for scenario calculation in utility. Special days - manual entry-> the inbuilt function can not	DSO Staff manually enters the input data for Calendar.

		recognise the special national holidays therefore DSO Staff has to enter them and function will copies them into Scenario Name column. Scenario Name, Count(): Here the inbuilt function calculates the number of different type of days for DSO flexibility table where the number of different type of days will be relevant.	
Calculation input	**Calculation input**	Contains of: Thermal limit of cable/ line and Operational limit for all months, Year, Time- Month – Type of Day load flow calculation result matrix. These inputs are used for DSO flexibility table where the needed flexibility for DSO is calculated.	DSO Staff manually enter the necessary data. The load flow calculation results are extracted from NEPLAN.

Table 2. Long-term Annual oputput interface table

Variable name	Variable annotation	Variable description
Flexibility calculation	Flexibility calculation	In Calculation input table the user can push the button: Show calculation. After this action the Flexibility calculation table displays the results for the given month for sake of checking the results by DSO Staff.
DSO Flexibility table	**dso_flexibility_table**	This table contains the needed flexibility by DSO for the given contractual period which is a crucial input for Microgrid module (the Microgrid module sends the answer of the Building in terms of available flexibility service by Building). Content: Month, Type of day, Flexibility requirement [kW], Time interval (Start), Time interval (Length), Flexibility requirement [kWh], Pcs of type of days

		This Variable appears in the database as a table, but contains additional inputs for LT workflow: PK.id; FK.contract and year/month. They refers to the Building name, Contract ID, the contractual periode.

Conclusion

Web based workflow tool is developed for both the Annual and Multiannual modules. The central point of LT workflow is an excel ("3Smart_LT module_v4.xlsm") which requires manual inputs from DSO staff. DSO staff has to be aware of this manual interventions, nevertheless the LT workflow gives a logical guide for it.

The interconnection of the long-term and short-term modules of the grid-side EMS is crucial for running daily load flow calculations. The daily AC load flow calculation will use the distribution network and load parameters from the standardized database/ or table prepared by the DSO and the basic logic is similar to the long-term scenario-based calculations. This AC OPF (optimal power flow) module completes the optimization calculation described in the short-term module description. An important output of the LT module is the DSO and Building flexibility table which is a key input to the AC OPF to precisely calculate the needed flexibility utilization from offered amount and time interval.

Furthermore, another key interconnection is between Long Term module and Building side Microgrid module: based on DSO Flexibility table results (submodule) the Microgrid will answer with a so called Building Flexibility table which contains the capability of the Building to provide requested flexibility service.

Important changes appearing in D5.4.3 when compared to module described in 5.3.1 Open software module for long-term level of grid-side energy management- Annual module description and in its basic elements:

- The former version of module contained a more fragmented LT excel where the following submodules existed:
    - Pre-Input_Legend,
    - Pre-Input_Dates,
    - Calendar which

These are now organised into one submodule: Calendar. This submodule contains all relevant information and makes the usage of submodule easier.

- Pre-Input_Scenarios,
- Processed input

These are organised into one submodule, into Calculation input. The submodule contains all relevant information as the former submodule, however makes the usage of submodule easier.

Annex I

1. Database architecture

The structure is envisioned in a way that the output tables of one entities database are defined in the same way as the input tables of another entities database, which makes understanding of the database structure a crucial part of the development process. The Long Term database is not separated into Annual and Multiannual ones, the building_flexibility_table and flexibility_unit_prices_and_penalty belong to Multiannual, the other part of database belong to Annual part.

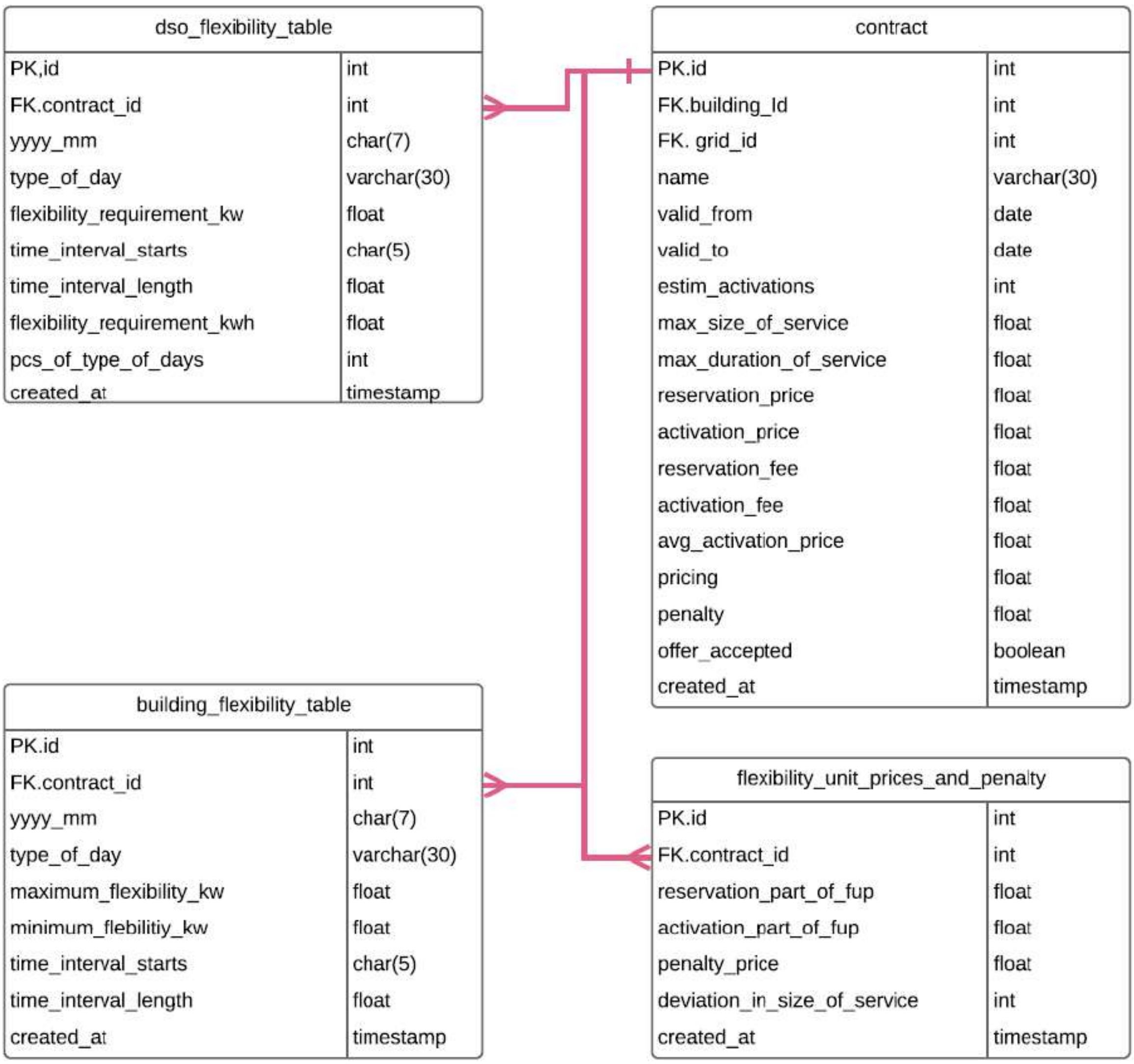


Figure 1 – Long-term database tables and relations

2. Communication events between the modules related to long-term grid-side operation

The below table describes the communication mechanism of the LT workflow which contains steps regarding both Annual and Multiannual modules. The main characteristic of the description is to focus on Grid-Building interaction and related database manipulations.

Table 3. Chronological communication events with detailed exchange structure

ID	Time (UTC)	Data exchange/ activity	D.5.3.1 (Annual and Multiannual) Nomenclature	module	Reads data	Puts data at disposal	Tri-gger
1	till December, before contract agreement	Calculation of flexibility needs, prices, penalty and quality of service by using " 3Smart_LT module_v3.xlsm"	Result: DSO Flexibility table; Flexibility unit prices,penalty; Output for long term contract sheets	LT module	DSO (staff)	DSO (staff)	0
2	till December, before contract agreement	Importing results of " 3Smart_LT module_v3.xlsm"	Result: DSO Flexibility table; Flexibility unit prices,penalty; Output for long term contract data base tables	LT module	DSO (LT)(script 1)	DSO (staff)	0
3	After step 2	Building EMS Microgrid module is fetching data from LT database		Microgrid	Building	DSO (LT)	0
4	After step 3	Building calculate flexibility offer	Result: Building Flexibility database table, (Microgrid database)	Microgrid		Building	0
5	After step 4	DSO (LT) module is fetching data from Microgrid database		LT	DSO (LT) (script2)	Building	0
6	After step 5	Generating file from Building Flexibility table	Result: Building Flexibility table in Excel	LT	DSO (staff)	DSO (LT) (script3)	0
7	After step 6	Contract preparation by DSO, inserting Building Flexibility table into " 3Smart_LT module_v3.xlsm"	Result: Output for long term contract sheet	LT		DSO (staff)	
8	After step 7	Acceptance/Rejection of Building offer	Result: Offer acceptance sheet (Yes/No)	LT		DSO (staff)	
9	After step 8	Importing Output for long term contract sheet of " 3Smart_LT module_v3.xlsm"	Result: The details of contract in Database	LT	Building	DSO (LT) (script4)	

Annex II

Logic of Open software module for long-term level of grid-side energy management - Annual module

1. Functional description of the long-term operation

This chapter contains functional and everyday operational description of the long-term module, based on and relating to the D5.1.1. document. It is important to note that day-to-day operation of the long-term and the short-term modules are interconnected:

- the two modules use similar logic,
- the long-term module is a data source for the short-term module and Building side EMS as well.

1.1 Inputs, outputs and functions of the long-term module

The envisioned operation of the long-term module is described by the following sequence diagram. The inputs and functions of the various participants are listed below the diagram (an output of a party corresponds to an input of another participant).

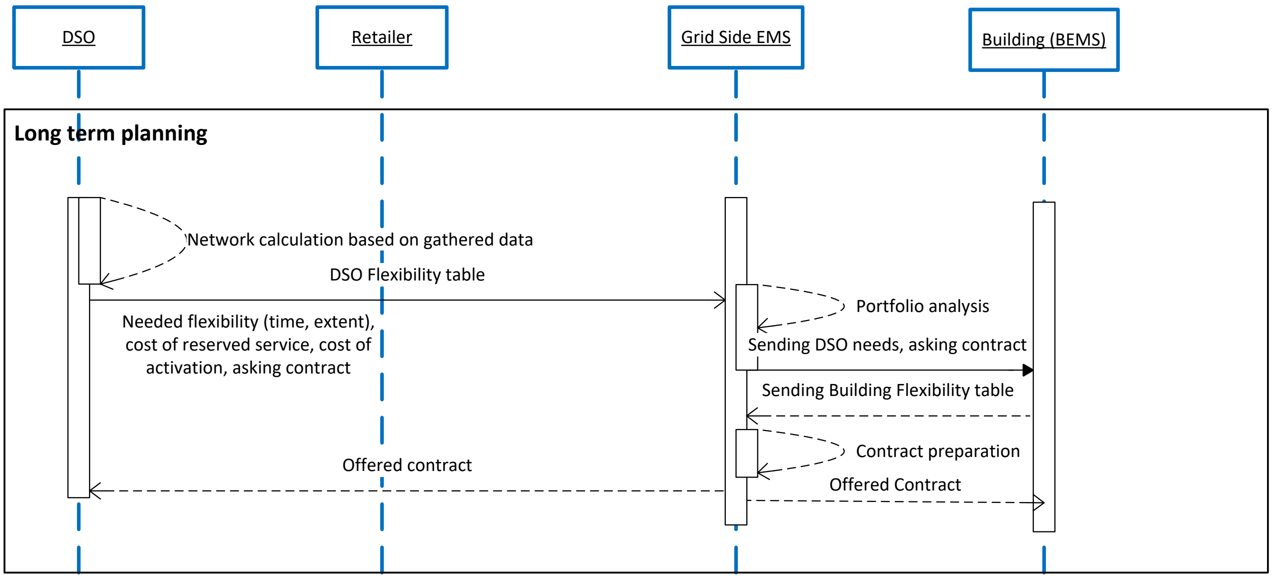


Figure 2: Long term planning sequence diagram

The following tables describes the above sequence diagram steps from input, output and activities point of view, this is basically follows a business logic and not a database manipulation.

Input list for Grid-side EMS:

Table 2 – Inputs for Grid-side EMS

ID	Name of Input	Description	Quality requirement
Ltl_1	Network parameters and load curves for daily LF calculation Direction: DSO → Grid-side EMS (Database/table with standardized inputs)	The D5.4.1. documents serve as inputs. The necessary data to build grid models for simulation: network parameters, load profiles at each node or load profiles of the given feeder and the connected customers. The load profiles either result from remote measurement based on AMR, or are derivates from the Synthetic Load Profiles (SLP) used in DSO which can differ by customer types.	Both the network data and the load data should be detailed enough for the short-term daily LF submodule to be able to run grid simulation daily, i.e. network parameters of the lines, nodes, transformers, load data at nodes, voltage data at selected busbars (if available).
Ltl_2	Operational limit table of the DSO	In order to compare the load curve with the network limits the DSO has to deliver a so-called limit table which describes the operational limit of the network (e.g. 70% of the thermal limit of the cable, in kVA, etc.)	The table has to describe obviously the limits, in this way the load curve or voltage band (results of LF calculation) can be compared with the operational limits. After the comparison, the result will be the needed flexibility (time, amount).
Ltl_3	Long-term planning results Direction: DSO → Grid-side EMS	The DSO must carry out the investigation in its own interest about the network part which shows any constraints that can be mitigated either by shiftable load or any renewable adjustment related to the given network part. DSO runs several scenarios which consider first the passive load profile, after that with DSO estimated flexible profiles, then with the delivered flexible load profiles by Prosumers: these calculations should result in flexibility needs in kW and activation time in Date, hour based on comparison of load flow calculation and the operational limit of the given network part. Operational limits also constitute a separate table which has to be sent to the Grid-side EMS. (Further details on limits and flexibility calculation will be given in chapter	DSO has to answer the following points: 1. needed flexibility amount for specific days of the year 2. activation time - flexibility needs pairs for specific days of the year Proposed price for the given flexibility: in case of reservation in EUR/kW, in case of activation in EUR/kWh. Reservation should contain: flexibility needs - activation time and duration pairs, frequency, price (in EUR/kW). Activation should contain: flexibility needs - activation time and duration pairs,

		5.) In addition, DSO has to calculate the avoidable cost of the traditional network intervention, i.e. the investment and the foreseeable operation cost of a network which would be built/upgraded instead of the constrained network part. This will be the lower limit of the total price of the flexibility service, which equals to reservation and indicative activation prices multiplied by the requested frequency of the services.	frequency of services, indicative price per activation (e.g. EUR/kWh).
LtI_4	Building Flexibility Direction: Building → Grid-side EMS	The given Building considers the unit prices (both for reservation and activation) calculated by DSO and based on it the Building could calculate the flexibility service (kW, time interval).	The Building returns an answer for the Grid-side EMS request: the amount of flexibility in kW and time intervals.

Input list for Building:

Table 3 – Inputs for Building

ID	Name of Input	Description	Quality requirement
LtI_5	Offer for a contract Direction: Grid-side EMS → Building	The Grid-side EMS collects all needs from the DSO regarding the part of the network with potential providers of flexibility service (point of view for DSO). Based on Building answer in terms of flexibility contract and unit prices the Grid side EMS will create a long term contract offer.	Needed flexibility amount, time and duration for specific days of the year with price.

Input list for DSO:

Table 4 – Inputs for DSO

ID	Name of Input	Description	Quality requirement
LtI_6	Offered contract with prices Direction: Grid-side EMS → DSO	Based on Building flexibility service and already calculated unit price the Grid side EMS creates a long term contract which will be received both by DSO and Building.	Offered flexibility amount, time and duration for specific days of the year with prices.

Functions of Grid-side EMS:

Table 5 – Grid-side EMS functions

ID	Name of Function	Description	Result/Output
GSEMS_F1	Reception of network parameters and load curves for daily LF	The Grid-side EMS receives the network parameters and load curves, examines the	Prepared data in standardized database (table) with distribution

	calculation preparation	completeness of the data and with help of LF calculator it calculates the needed flexibility for DSO. The network parameters in LF calculator will be inserted into short term database as well for DA AC OPF.	network and load parameters for daily LF calculations, the LF can run based on the received data and give reliable results. Long term network data can be used in short term database.
GSEMS_F2	Reception of Long-term planning results and creation of contract offer for Buildings based on them	The Grid-side EMS receives the Long-term planning results, examines their completeness, inserts them into the existing portfolio, calculates an own price based on portfolio optimization (i.e. simulates the behaviour of the Aggregator with some assumed margin), gives contract offer for Building.	A structured offer for the Building.
GSEMS_F3	Delivery of long-term contract for DSO	The Grid-side EMS receives the Validated contract with prices from the Buildings, compares them with the DSO needs, makes a portfolio optimum, gives long-term contract for the DSO for the given network part, completed with the necessary information (offered flexibility amounts, activation times and durations, offered prices, penalty consideration, etc.)	Validated LT contract both for DSO and Building.

Function (Activity) of Building:

Table 6 – Building functions

ID	Name of Activity	Description	Result/ Output
B_F1	Techno-economic calculation based on long-term contract offer	The building considers the DSO flexibility needs and the given unit prices and create a Building flexibility service answer (Building flexibility table) with amount of flexibility in kW and corresponding time intervals.	Validated contract with prices or/and modification proposals for Grid-side EMS.

Function (Activity) of DSO:

Table 7 – DSO functions

ID	Name of Activity	Description	Result/ Output
DSO_F1	Provision of data for daily LF calculation	Provision of the input data for the daily LF calculation	D5.4.1 document
DSO-F2	Provision of draft long-term contact	Provision of the input data for long-term contact	Long-term contract data

1.2 Connection with the short-term module

The interconnection of the long-term and short-term modules of the grid-side EMS is crucial for running daily load flow calculations. The daily AC load flow calculation will use the distribution network and load parameters from the standardized database/ or table prepared by the DSO. The assumed logical connection of the long-term calculation method and AC OPF is described in the flow chart diagram in Figure 3. What is important is the DSO and Building flexibility table from LT excel, because AC OPF also will use the needed and offered flexibility both in term of amount and time interval.

The envisioned operation of the short-term module is described by the sequence diagram below:

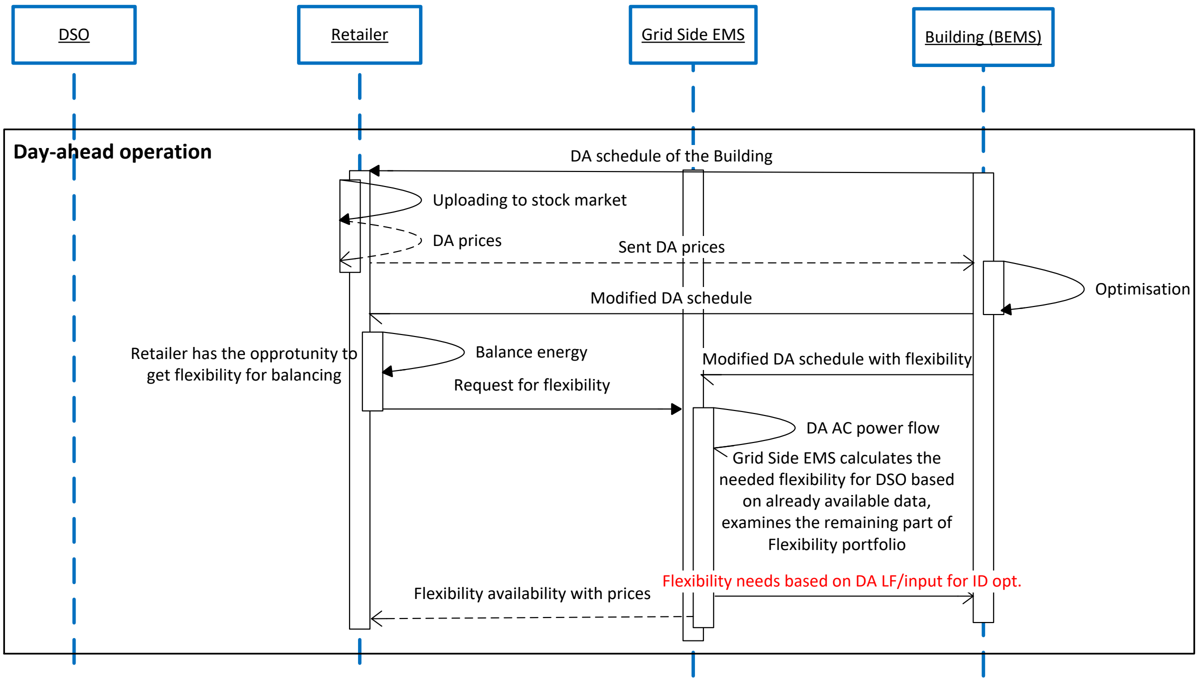


Figure 3: Day-ahead operation sequence diagram

The following two figures are extracted from Microsoft Excel. They show an example of the flexibility requirement calculation.

Date	Time	Estimated load by AC power flow [kW]	Estimated load by AC power flow which exceeds the Operational limit [kW]	Operational limit Pop[kW]	Network limits Pmax [kW]	Predefined Operational limit (% of Pmax)	Activated Flexibility needs (=Estimated maximum load- Operational limit(%)*Pmax)[kW]	Calculated Flexibility needs (=Estimated maximum load- Operational limit(%)*Pmax)[kW]	Activation Price/kWh	Duration	start	end
2017.05.22	15:30	60	0	70	100	70,00%	0	0	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	15:45	65	0	70	100	70,00%	0	0	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	16:00	75	75	70	100	70,00%	23	5	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	16:15	75	75	70	100	70,00%	23	5	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	16:30	80	80	70	100	70,00%	23	10	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	16:45	80	80	70	100	70,00%	23	10	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	17:00	85	85	70	100	70,00%	23	15	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	17:15	87	87	70	100	70,00%	23	17	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	17:30	90	90	70	100	70,00%	23	20	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	17:45	92	92	70	100	70,00%	23	22	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO
2017.05.22	18:00	93	93	70	100	70,00%	23	23	25 Eurcent	3 hours	Signal from the DSO	3 ours from "on"-signal, or by earlier signal from the DSO

Figure 4: The logic of DSO flexibility needs calculation

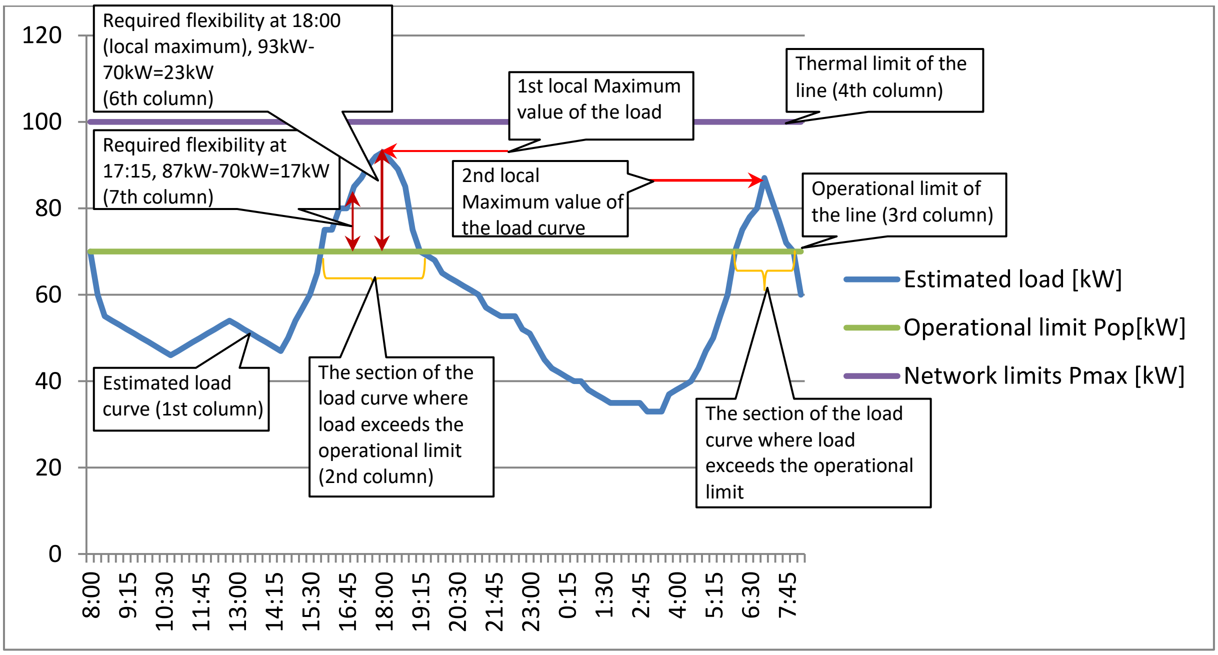


Figure 5: Visual example of logic of DSO flexibility needs calculation

The figures above demonstrate how the Grid-side EMS should calculate the needed flexibility based on daily LF calculation and predefined parameters (operational parameters of the network) in the short-term module. The daily LF calculation will be the basis of the activation in the intraday period:

1. 1st column: daily LF estimates the load curve of the selected network element.
2. 2nd column: Grid-side EMS shows the load which exceeds the operational limit.
3. 7th column: theoretical value of the needed flexibility. The Grid-side EMS calculates the difference between load curve and the operational limit of the network. E.g. if 100 kW would be the thermal limit of the line and 70% would be the operational limit, and the load curve

shows 87 kW, then the needed flexibility is 17 kW. If the value is positive, then the Grid-side EMS should send an activation signal to the Building.

In other words: the system finds the extreme values of the examined variable based on daily LF calculation; this procedure is executed on a daily basis. The Grid-side EMS examines the load curve and calculates the differences between the pre-set limit and the estimated curve. The start of the activation time window is defined as the period when the curve exceeds the operational limit for the first time. The deactivation time corresponds to the time when the curve falls below the operational limit.

2. Running scenario simulations for long-term planning

According to Deliverable D5.1.1. Grid-side EMS concept and information exchange interfaces definition, the long-term module can be divided into 4 main steps:

1. Initial distribution network state calculations
2. Calculating DSOs flexibility needs
3. Calculating costs of flexibility depending on the source
4. Contracting flexibility – planning future distribution networks

This chapter covers step 1 of the above, the need for scenario simulations and the method of running the initial scenario calculations are described.

2.1 Network planning and the flexibility option

Traditionally, DSOs ensure the security and reliability of operation, and the required quality of service by reinforcing the network. If wisely exploited, demand response creates a chance to postpone the costly conventional investments that should be completed in a few years.

Demand response can remedy several issues, including:

- keeping voltage within the allowed operational range
- congestion management, thermal limit of cables and overhead lines
- operational limit of lines (e.g. due to a potential n-1 state, weather conditions)
- capacity of transformers
- peak shaving, loss reduction
- asymmetry.

First and foremost, one needs to run load flow simulations according to its conventional network planning approach. In the next few paragraphs, a brief background of calculation methods is provided.

Distribution network planning at E.ON:

The NEPLAN software is used for network modelling. First, the network topology and equipment data are exported from the GIS software. For high and medium voltage network planning, measurement data from winter and summer national measurement days and the SCADA system, and long-term load forecast based on time series forecasting are used. Currently, the planning of the MV network is based on the scenario with no dispersed generation infeed. The measured distributed generation is added to the measurements at HV/MV substations to determine the maximum load of each MV feeder. The improvement of this planning method is under progress. Currently we do not run time series load flow for the MV calculations but we plan to do so, or at least examine more static states in the future.

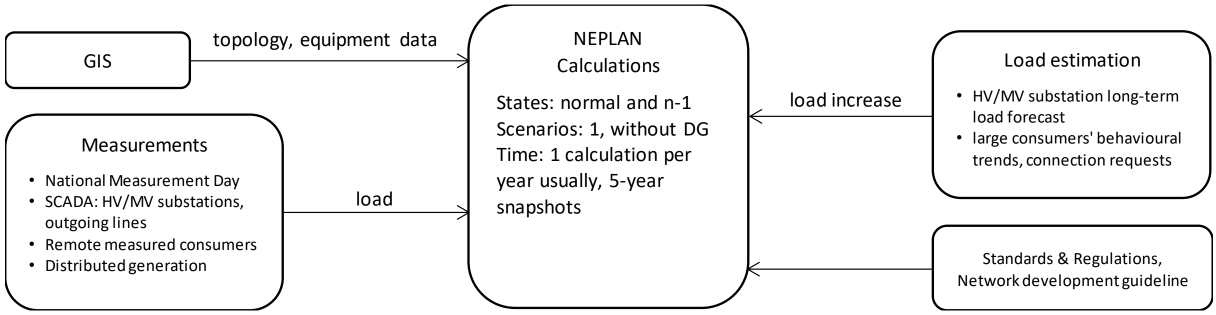


Figure 5: Network planning logic

The weak points of the grid are identified by analyzing line cross section, current, voltage drop (normal and n-1), loading of elements (normal and n-1), connectable power (normal and n-1) and network loss. (Connectable power: based on the allowed range of voltage and loading limits, the amount of extra load (in 0,5 MW steps) that could be connected to each node is determined.)

The low voltage distribution network planning is based on time series LF calculation. Regarding distributed generation, both scenarios are considered, i.e. with and without distributed generation infeed. On LV network both measurements and substitution profiles (the yearly consumption of the consumers and profile types) are used, and a series of 8760 LFs is run, so a calculation for each hour of each day in a year. The calculation results are compared to actual measurements if available – the comparisons have proved this combined method is promising, and we aim to adapt our MV calculations according to it.

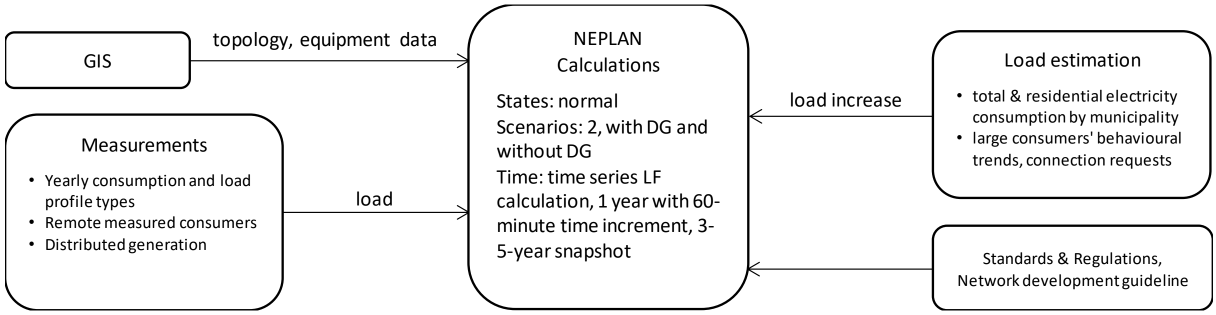


Figure 6: Network planning logic with consideration of DG

The weak points of the grid are identified by analyzing line cross section, current, voltage drop, loading of elements and network loss. Balanced 3-phase systems are modelled, and the impact of asymmetry is evaluated afterwards when elaborating the results of the LF calculation.

2.2 Scenarios and calculation procedure

Running time series load flow analysis is a reasonable way to estimate the annual or seasonal flexibility requirements: the number of activations, the time, the duration and the size of the desired flexibility service. The D5.1.1. document suggests running 3 conceptually different AC load flow calculations for the designated days, weeks, seasons. All LFs would be time series calculations, therefore load (and generation) profiles for entire days would be used. The proposed calculations are:

1. AC LFs with conventional approach to loads, thus the traditional LF calculation without flexible loads. This involves calculation with load forecast (demand and generation, increase or decrease) as well, according to the DSO's own network planning methods.
2. Repetition of the AC LFs with the introduction of flexible loads. New profiles should be created for the estimated new load patterns of prosumers, the "flexible load substitution demand curves". Estimating the flexible consumers' behaviour is a key step for two reasons. Although we can derive the needed flexibility for the whole network part in question, we need to know the available flexibility of the prosumers, too. Theoretically, it is possible to satisfy the total amount of the needed flexibility (if the Aggregator gathers all possible shiftable loads alongside a MV line), but in our case with only one building offering flexibility service, perhaps only a part of the DSO's demand can be realised by this sole shiftable load. The second reason is that in other period(s) of the day (or preceding/following day) the building eventually will consume the energy it „lost" to supply flexibility. A simple example is in the next figure, where the building compensates the energy deficit in the morning and in the evening. The DSO should keep that in mind and make sure that the modified demand curves do not compromise the network security.

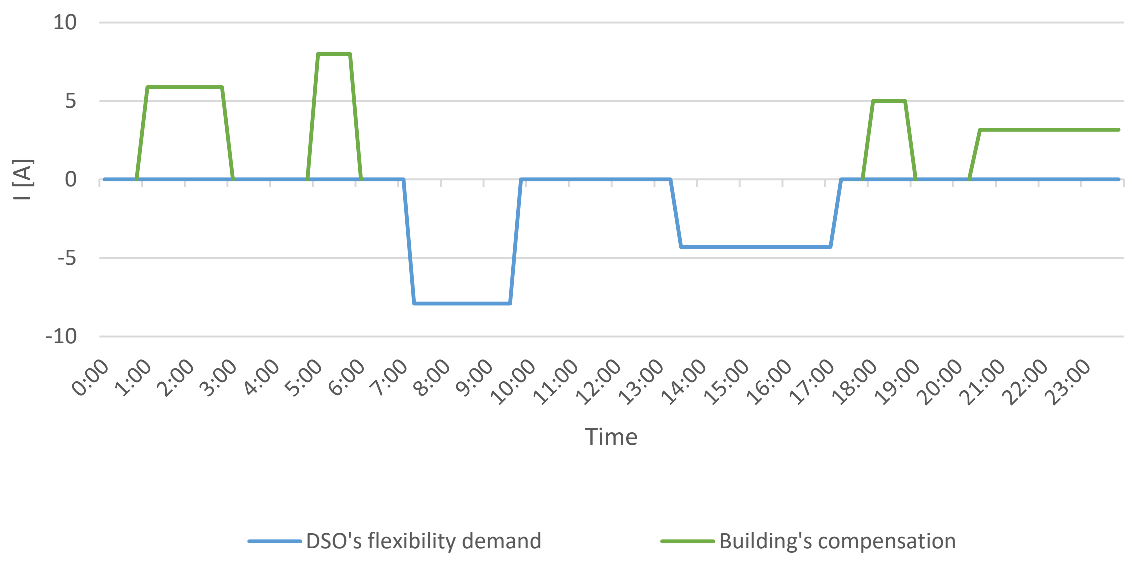


Figure 7 – Flexibility demand and compensation

3. Demand patterns are provided by the flexible prosumers themselves. These profiles are then run through the AC LF.

Usually, as the load pattern varies throughout the year, DSOs examine multiple network states while planning network upgrades and investments. Presumably the planning methods already cover several of the following scenarios, whether LF series with daily profiles or single LFs are run for the designated times:

- time of year: seasons, months,
- day types: weekdays, Saturdays, Sundays (holidays),
- time of day: time of peak and base load during the day in case of “static” LF, or time series LF with 24-hour profiles (with quarter hourly or hourly resolution),
- meteorological conditions,
- calculation with and without renewable or distributed energy sources,
- calculation with different network configurations (e.g. n-1 state).

For 3Smart, 36 scenarios are derived from E.ON’s low voltage calculation method. Load values derived from actual measurements are used where available (mostly larger customers with AMR devices) and several substitution profiles are used for the rest. These profiles yield the 36 scenarios:

- for each profile type, we distinguish between the 12 months,
- in each month, there are 3 different daily load curves: weekday, Saturday and Sunday.

Each scenario means a calculation with the load curves for one day. E.ON usually calculates with hourly resolution (24 values for a day), but if other DSOs have the option to calculate with 15-minute resolution (96 values for a day), that is great. If 12 profiles for the months (for example just for each season) or different profiles for Saturday and Sundays (just weekday and weekend) are not used, that can be sufficient and means less than 36 scenarios.

The step-by-step calculation procedure is described in the flow chart below. Sections 4.3, 5.1 and 5.2 are marked with the capitals in the figure. The linkage between the AC Optimal Power Flow calculation (for the short-term module functionalities) and the traditional load flow calculation is virtual, because the AC OPF will be run on similar basic data (GIS data) but from time perspective, the calculation is separated. The long-term calculation precedes the short-term, and the long-term calculation frequency is much smaller.

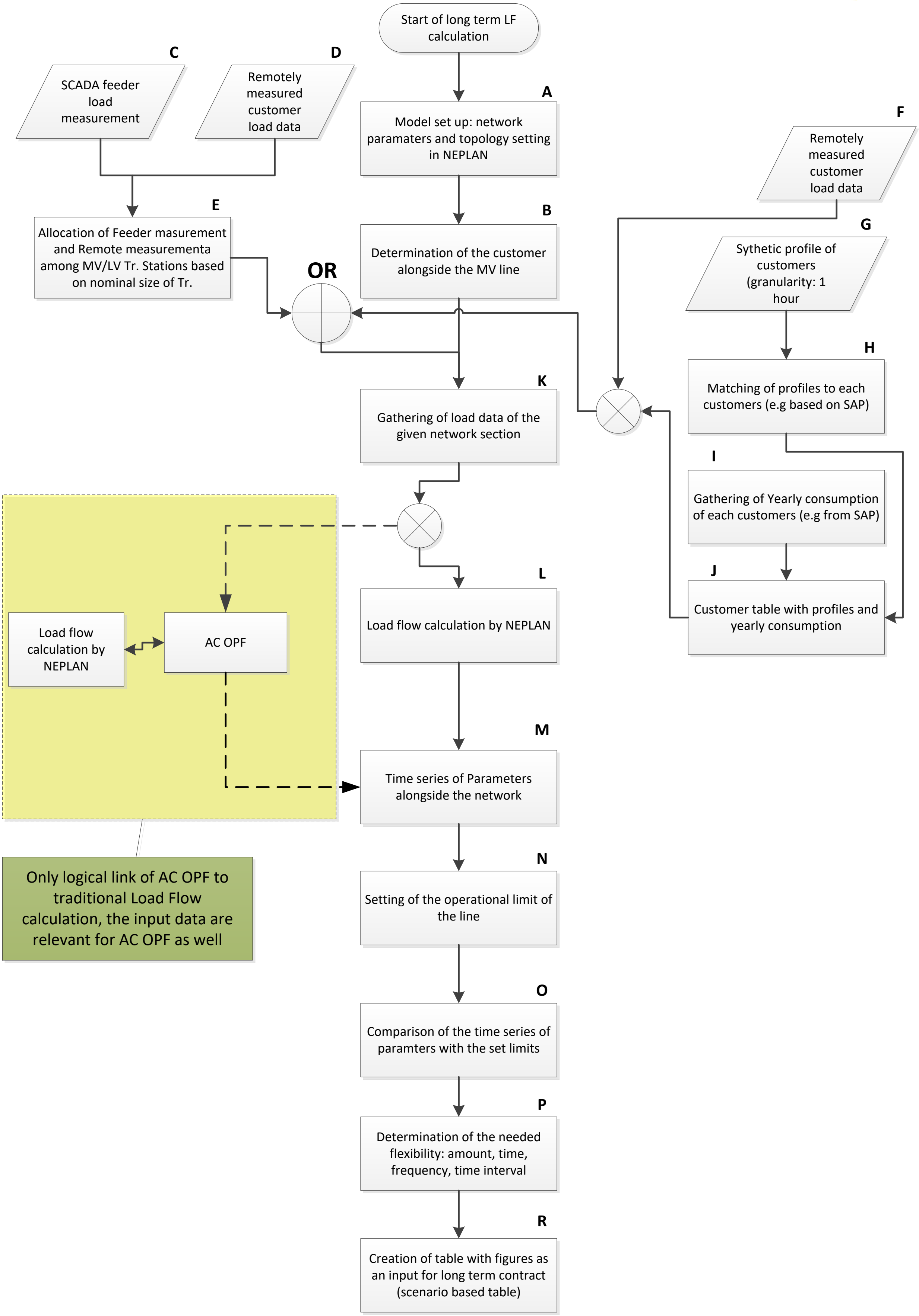

Start of long term LF calculation
A
Model set up: network paramaters and topology setting in NEPLAN
B
Determination of the customer alongside the MV line
C
SCADA feeder load measurement
D
Remotely measured customer load data
E
Allocation of Feeder masurement and Remote measurementa among MV/LV Tr. Stations based on nominal size of Tr.
OR
F
Remotely measured customer load data
G
Sythetic profile of customers (granularity: 1 hour
H
Matching of profiles to each customers (e.g based on SAP)
I
Gathering of Yearly consumption of each customers (e.g from SAP)
J
Customer table with profiles and yearly consumption
K
Gathering of load data of the given network section
L
Load flow calculation by NEPLAN
Load flow calculation by NEPLAN
AC OPF
Only logical link of AC OPF to traditional Load Flow calculation, the input data are relevant for AC OPF as well
M
Time series of Parameters alongside the network
N
Setting of the operational limit of the line
O
Comparison of the time series of paramters with the set limits
P
Determination of the needed flexibility: amount, time, frequency, time interval
R
Creation of table with figures as an input for long term contract (scenario based table)

Figure 9: Logical process flow chart of DSO flexibility needs calculation

2.3 Input data of scenario simulations

A Grid model

The first step is to determine and model the part of the network that will be investigated. Presumably this grid model was already built for deliverable D5.4.1. "Grid models for grid simulation in professional simulation tools". The technical parameters provided in the D5.4.1. document can be regarded as the database with the necessary network parameters. The figure below represents E.ON's NEPLAN model for 3Smart: a MV feeder with 4 MV/LV substations.

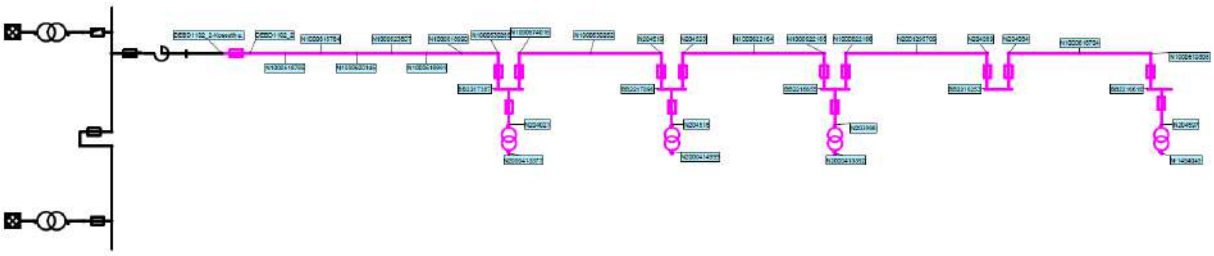


Figure 10: Example for a Grid topology in one pilot site (Debrecen)

B, K: Customers and load data

The consumers and generators that are connected to this part of the grid should be identified and modelled, or more conveniently, sum measurements (e.g. total load of a transformer) can be used if available. Gathering and allocating the load data can be a difficult process, as there are several options. You may have continuous measurements, estimated data with load profiles, or the combination of the two.

C, D, E: Measurements

If synthetic load profiles cannot be used, the sole data source for creating load profiles are continuous measurements. If some or all customers are (remotely) measured and their consumption is available, individual profiles (load curves) can be created for the load elements in the model that represent these consumers.

If individual measurement is not available for all customers, a collective profile can be allocated to the rest based on a general network measurement. For example, there are SCADA measurements in the HV/MV substations for the medium voltage feeders. In the grid model the measured sum value is assigned to the first element, this is the total load of the MV line. The remotely measured consumers' values are assigned to their individual load elements. The rest of the load is allocated to the remaining load elements (one for each MV/LV transformer station) in relation to the nominal power of the MV/LV transformers but with the same profile (so their load curves normalized are all the same). Regarding the load allocation, other methods can be used (for example, E.ON plans to consider the number and yearly consumption of the customers soon).

G, H, I, J (with F): Load profiles (with measurements)

E.ON uses load profiles along with the yearly consumption and the profile type of each user. The loading curves were derived from the profiles used for forecasting the consumption of profile-settled users. There are different curves for workdays, Saturdays and Sundays in each month. There are two categories for domestic consumers (normal and direct load control), 5 for business consumers and categories for continuous consumption, e-mobility, renewable heating/cooling and solar panels as well. The direct load control profile was altered, and a seasonal profile was introduced for holiday resort areas. Figure 11 represents the daily profile curves of the more common profile types for the 3 scenarios in January (weekday, Saturday and Sunday).

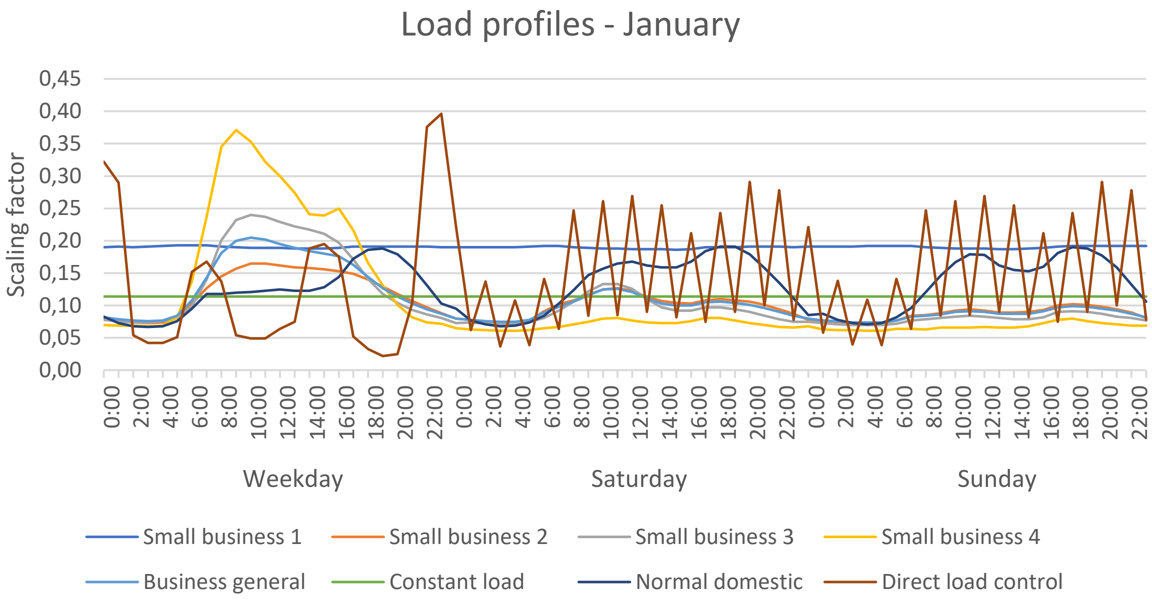


Figure 11: Example for Synthetic Load Profiles (SLP)

To demonstrate the seasonal variations as well, the following figures represent the same profile types for 36 scenarios. The daily profiles are sorted after one by one in this order: January weekday, January Saturday, January Sunday, February weekday, February Saturday, February Sunday, etc.

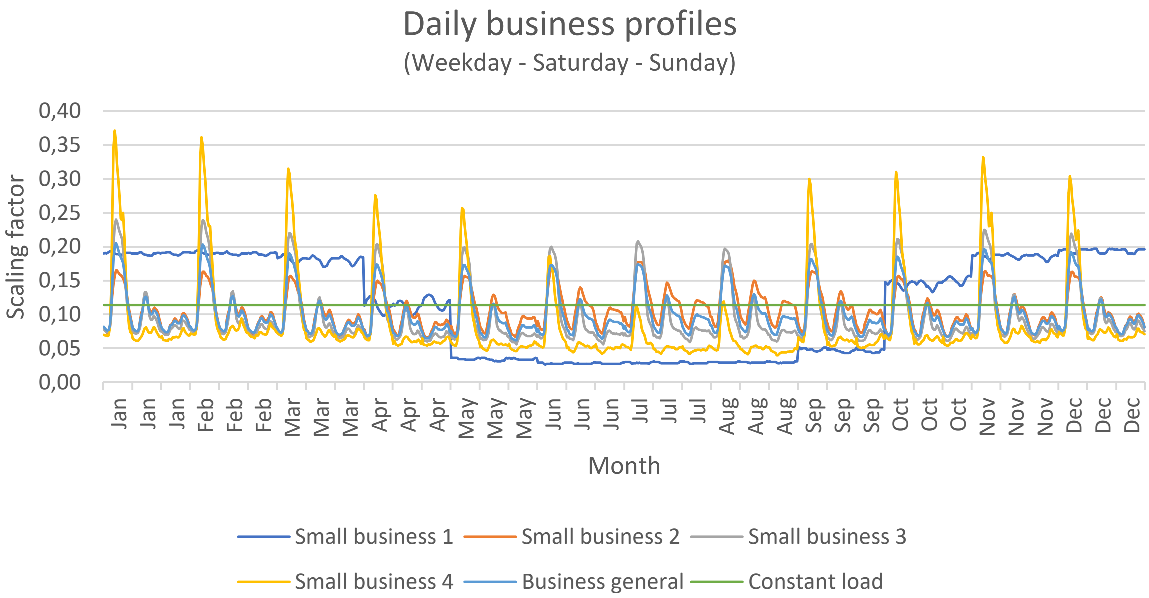


Figure 12: SLP in terms of different type of days

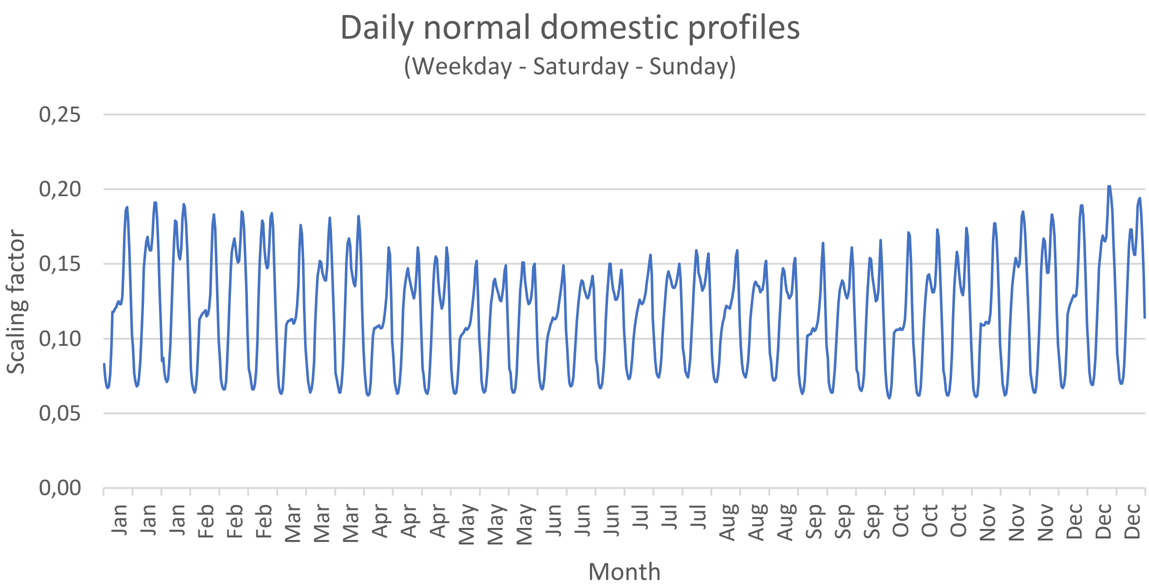


Figure 13: Domestic SLP

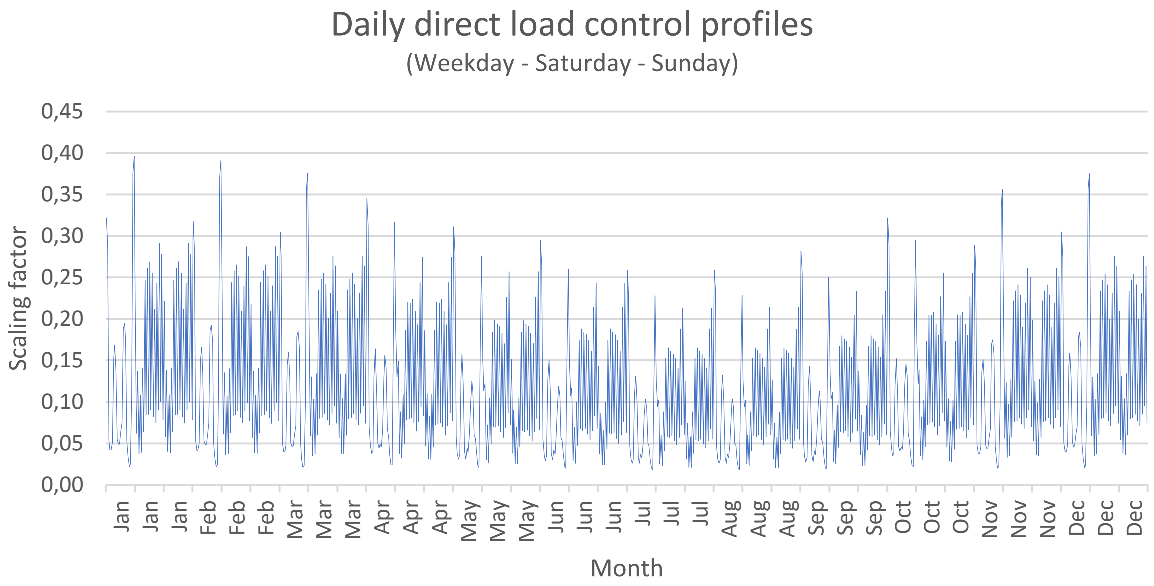


Figure 14: SLP of customer with Direct Load control (e.g control of boilers via radiowave)

Household size solar power plants are modelled as common loads with a negative load profile. Yearly generation is calculated from their built-in capacity (kVA), it is assumed that they would operate at full power for 1100 hours per year (a utilization hour specific for Hungary). The consumption of consumers with solar power plants is modified after reviewing their past measured yearly consumptions. For every month, the first 7 days are calculated without PV generation (representing cloudy weather) and the remainder of the month with PV generation (clear weather). Figure 15 demonstrates the daily PV profiles, one for each month.

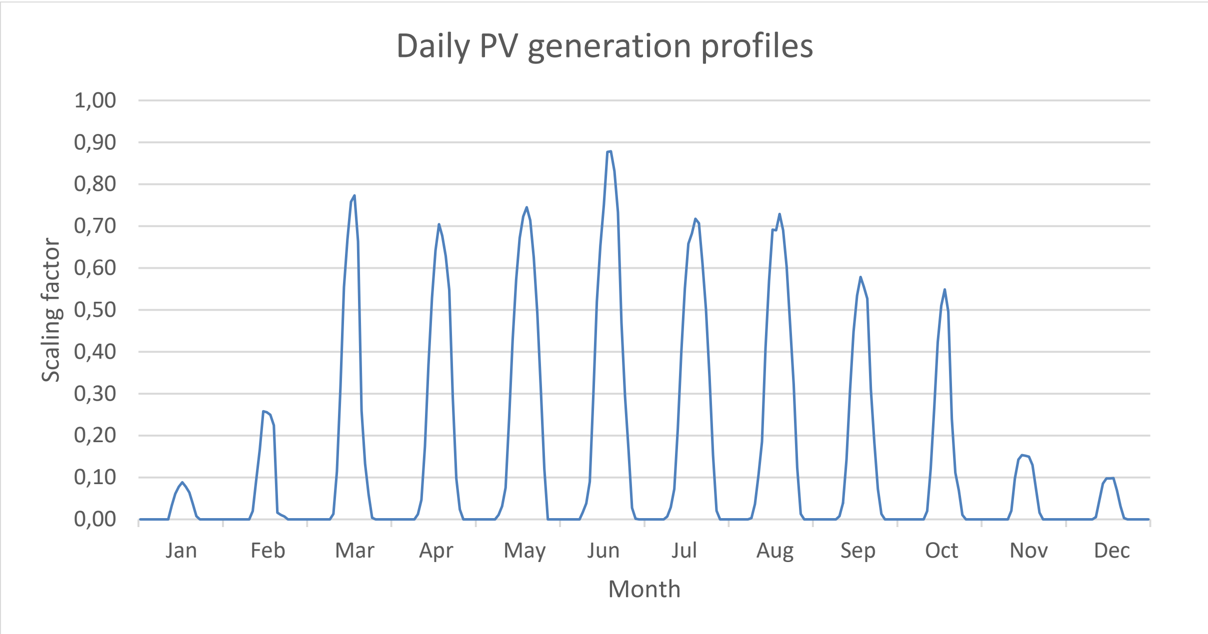


Figure 15: SLP of PV generation

The yearly consumption and the profile type of the customers are obtained from SAP and recorded in the adequate NEPLAN variables. To prepare the individual load curve of these customers, the scaling factor of the respective profile is multiplied by the yearly consumption for each LF calculation.

If there are remotely measured customers in addition to the profile-settled users, individual profiles (load curves) can be created for the load elements in the model that represent these consumers. Like in the previous case with measurements, steps D and F are basically the same. These measurements are usually rearranged so that for each hour the highest values are sorted to the first week (the days without PV generation) to get the worst-case scenario.

L: Time series load flow calculation

Run the LF simulation for each scenario. For example, one January weekday means 24 (hourly) or 96 (quarter hourly) calculations.

M: Calculation results

Save the results of those network elements that are crucial, interesting, where there is a chance that technical network constraints are violated or where specific technical parameters need to be improved. In the next chapter the total load of the MV line is examined, so the results of the first element after the feeder node (marked with a red rectangle in the figure) are needed.

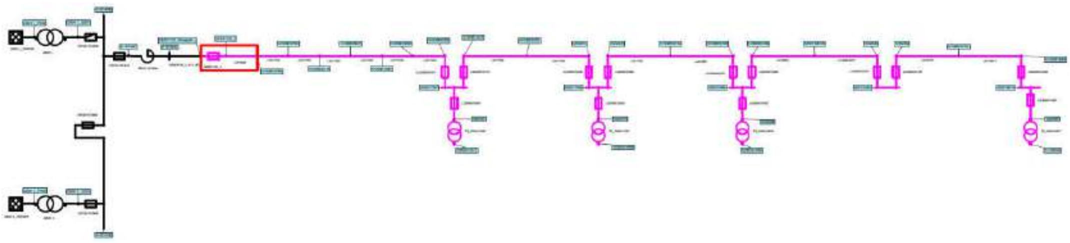


Figure 16: Total load element on the grid topology

3. Definition and calculation of the limit values and flexibility tables

This chapter covers the second main step of the long-term module, calculating the DSO's flexibility needs. The method of determining the flexibility need is described and illustrated with figures.

3.1 Results analysis

N: Setting the limit values for critical network elements

Analyze the available data – the results of scenario simulations and/or measurements. Identify the weak points of the network. Define the limits. It can be the thermal limit of the cables, the operational limit of lines (e.g. 70%) or transformers (e.g. 60-80% in summer), minimum and maximum allowed voltage, etc.

In this example, the total load of the medium voltage feeder is observed. Both simulation results and past measurements are available. Figure 17 demonstrates the measured hourly average, maximum and minimum current in January 2018. Figure 18 represents the quarter hourly averages in 2016. The figures demonstrate that the load is the highest in the winter months and that it is remarkably variable in a brief period (the city's transport company operates a transformer station on this MV line). However, it is far below the cables' thermal limit. The grid-side EMS can be tested with peak shaving in the autumn and winter months based on the average values. In this example, the 95% percentile of the whole yearly data is picked, which is about 40 A. The same can be done for just workdays, or different values can be set for the summer period for example.

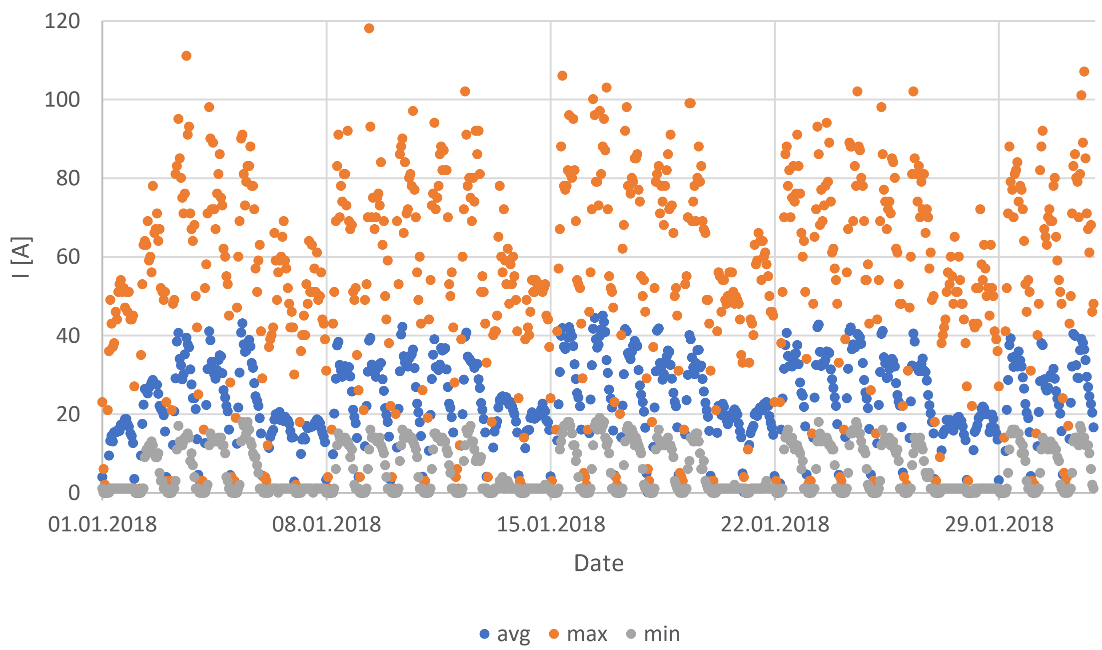


Figure 17: Measured current (1-hour values)

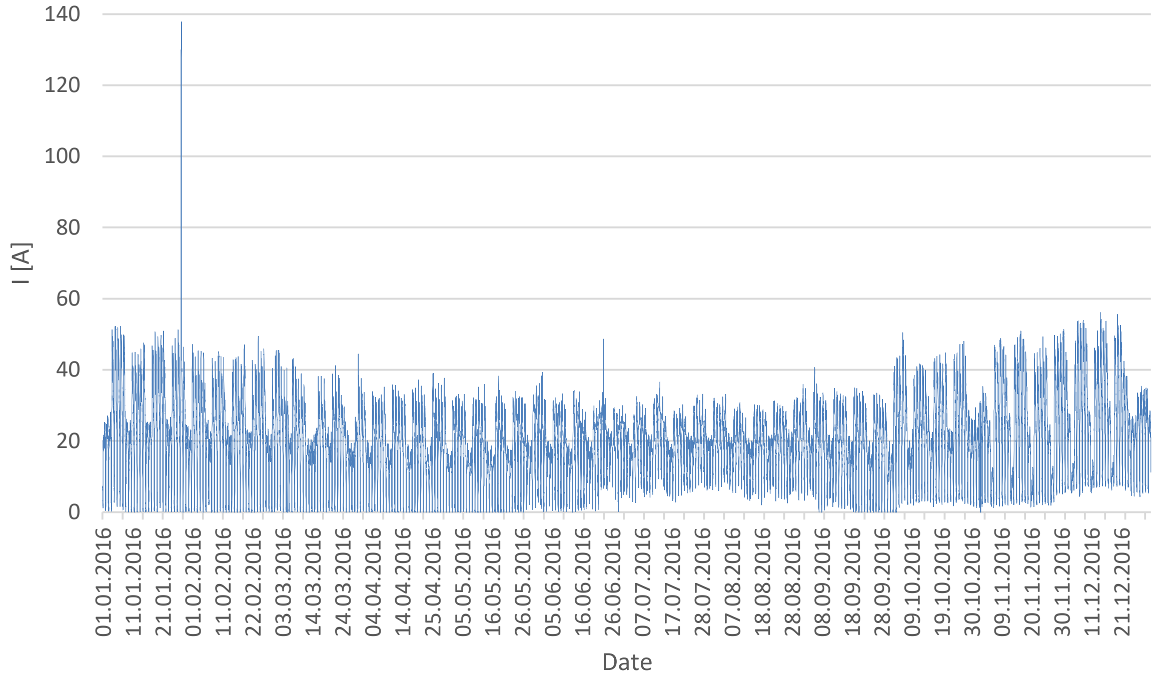


Figure 18: Measured current (15-minute values)

O: Comparison

Examine the scenarios one-by-one to estimate whether the set limits are violated. If only synthetic profiles are used, that is already a normalized load profile. In this example, January weekdays are examined. In the simulation both synthetic load curves and measurements were used. Several days were simulated due to the measurements, so a normalized curve was created. In the next figure, the thin lines represent the individual days and the thick red line represents the normalized scenario.

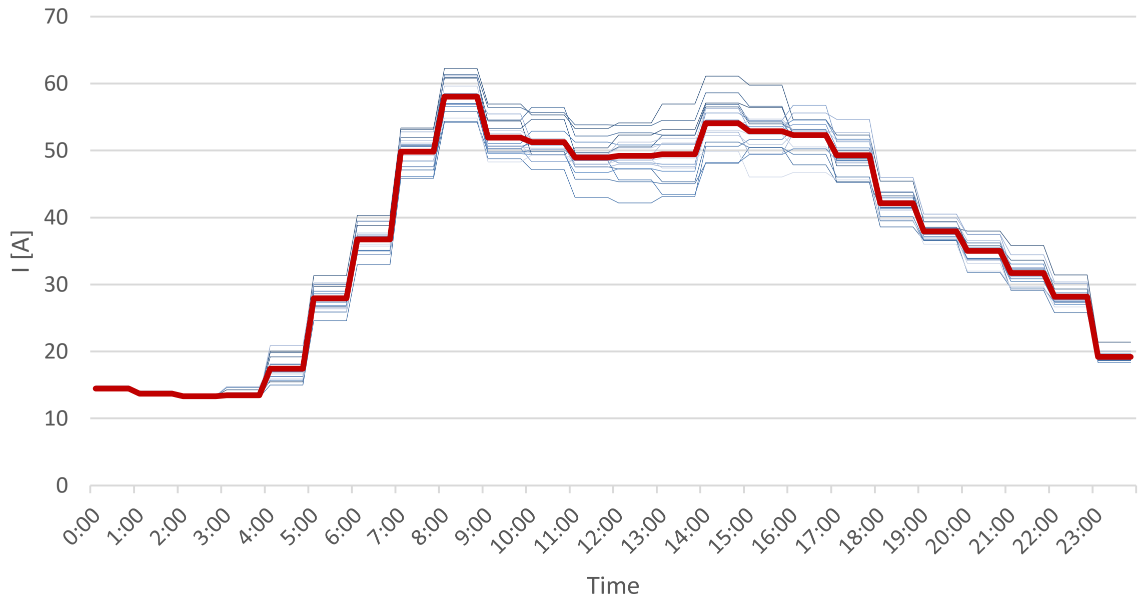


Figure 19: January weekdays simulation

Measurements are available for the appointed network element and can be analysed as well. The measurements for weekdays in January are examined. This was in 2016, hence the lower values.

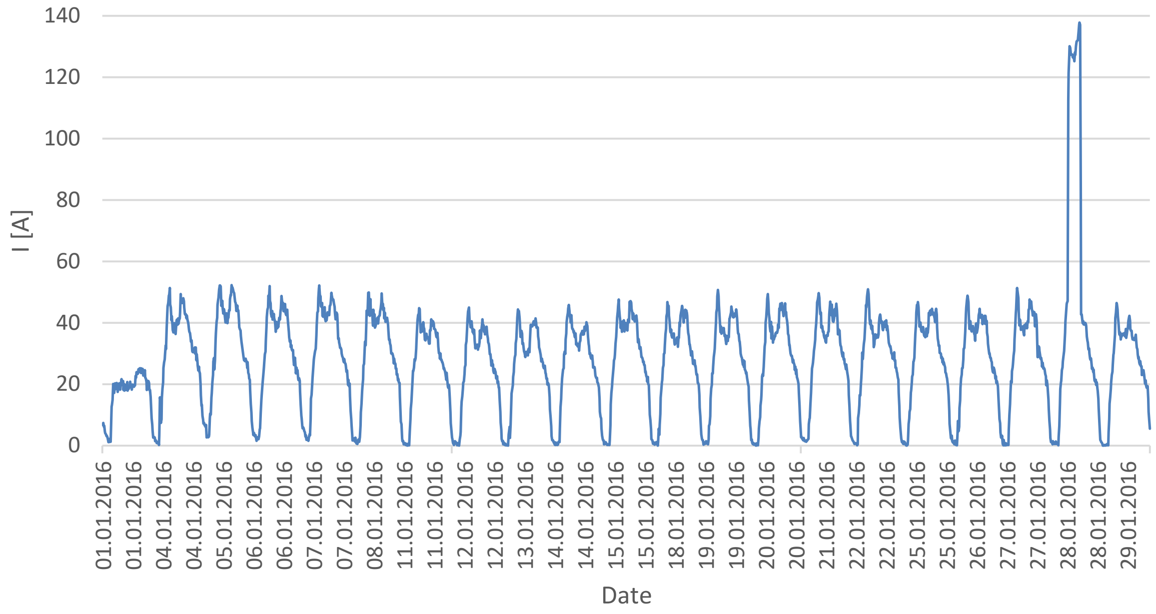


Figure 20: Measurements from January weekdays

All extremities should be excluded, use credible data only. January 1 was a holiday and on January 28 there was a load increase due to temporary network reconfiguration. In figure 21, the thin lines represent the selected days and the thick red line represents the normalized profile created from them by calculating the average value for every quarter hour.

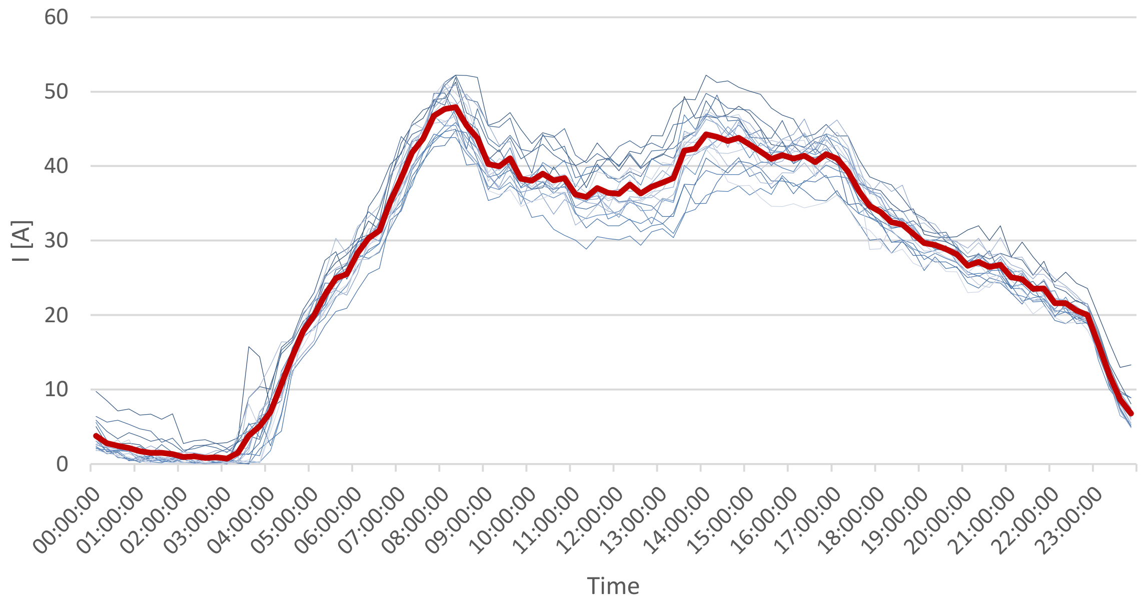


Figure 21: Measurements from selected January weekdays

The normalized scenario should be compared to the set limit. In this case the 40 A limit is exceeded in the morning and in the afternoon hours on a typical January weekday.

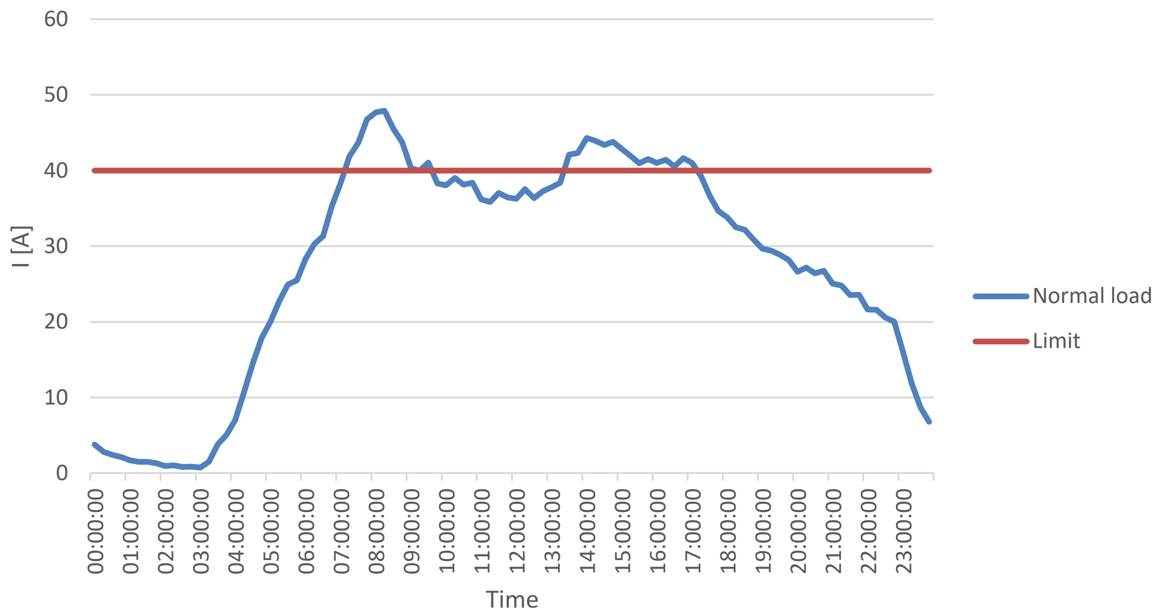


Figure 22: Normalized January weekday

3.2 Calculation of flexibility

P: Determination of the needed flexibility

The next step is to translate this comparison into flexibility needs in terms of power/energy, duration of service and time slots. Figure 23 describes the load curve the DSO would like to achieve and the corresponding flexibility requirement.

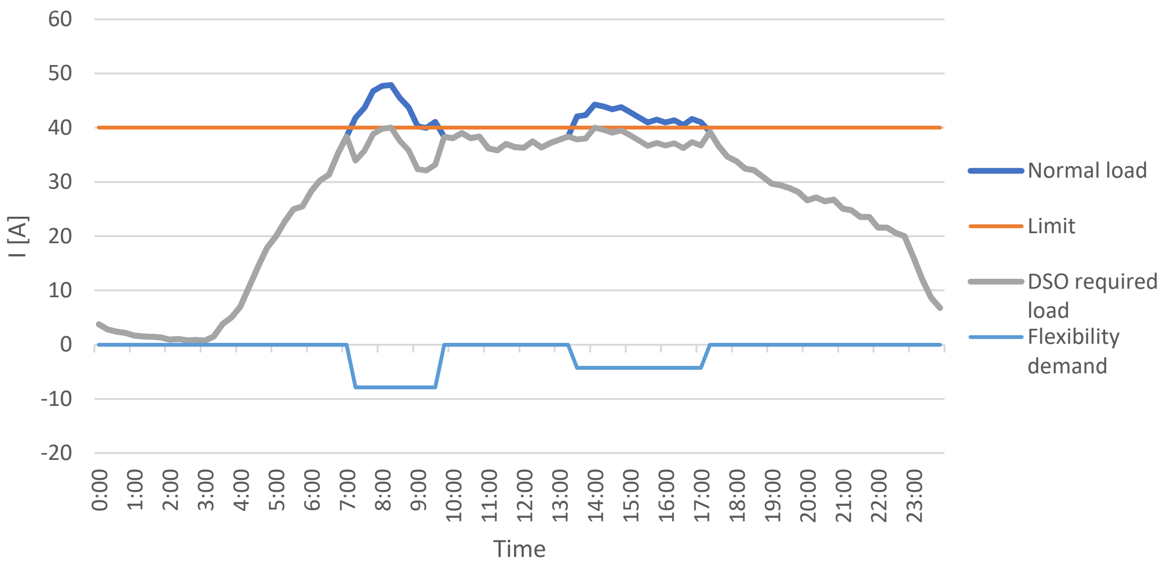


Figure 23: January weekday with flexibility demand

For this first, January weekday scenario, there are two periods when the limit is exceeded.

- The first time slot is between 7:15 and 9:45, for a duration of 2,5 hours. The maximum difference is 7,9 A, the DSO requests a load decrease of 150 kW.
- The second time slot is between 13:30 and 17:15, for a duration of 3,75 hours. The maximum difference is 4,3 A, the DSO requests a load decrease of 80 kW.

The above two are the DSO's flexibility needs for January workdays – in 2018 these would have been valid for January 2-5, 8-12, 15-19, 22-26 and 29-31.

For the 3Smart pilots of course, one building may be able to deliver those flexibility needs just partially, and the building will consume that shifted energy in other hours. Theoretically, security control would be the next step. This involves:

- running calculations with worst case scenarios to make sure the grid stays safe throughout the day,
- checking for a few years ahead to see if defined flexibility requirements remain feasible solution when including load forecast,
- suggesting restrictions if necessary, e.g. regarding the period between the two flexibility events.

The next figure illustrates an example when the DSO's total flexibility request is delivered and the prosumers compensate the load shift between 1:00-3:00, 5:00-6:00, 18:00-19:00 and 20:30-0:00.

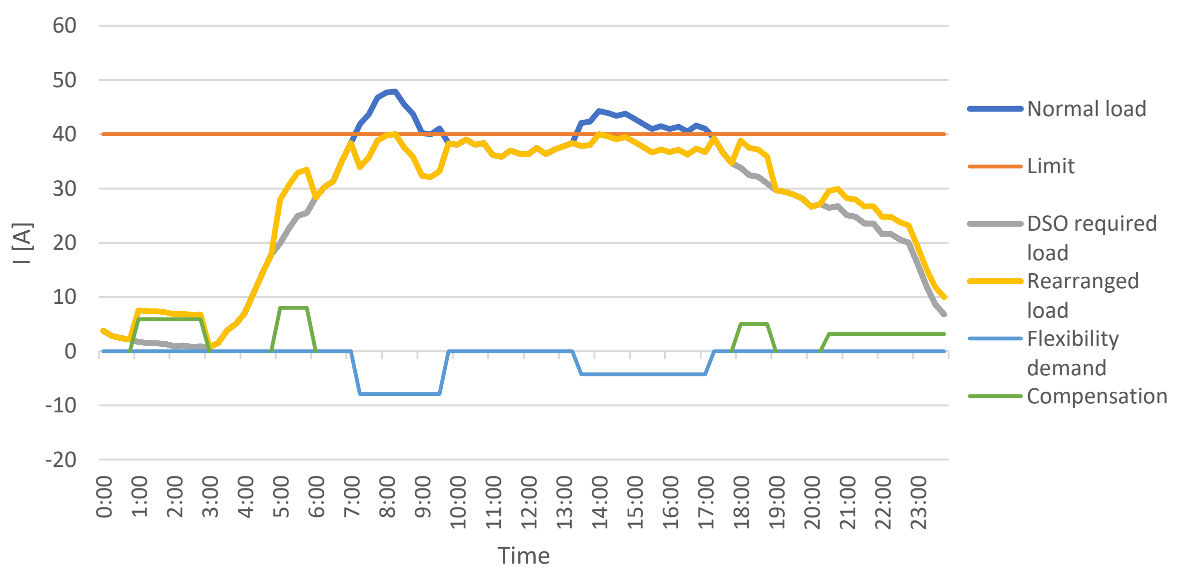


Figure 24: January weekday with demand response

R: Flexibility tables

The DSO examines all scenarios in the way described above with the Annual software module and lists the flexibility requirement(s) and time slot(s) for all cases when the defined limit is violated. Then these results are translated into the inputs below (LtI_3) and a table containing all flexibility requests during the contract period is sent to the Grid Side EMS database to make a long-term contract proposal for the flexible prosumer (this is the part of the Multiannual module operation already).

Table 8 – DSO flexibility requirements

Month	**Type of day**	**DSO flexibility requirement [kW]**	**Time interval**	**Time interval**	**Pcs of type of day**
1…12	***workday / Saturday / Sunday***	***direction, volume***	***start***	***length***	
2018-01	WEEKDAYS	-150	07:15	2.50	22
2018-01	WEEKDAYS	-80	13:30	3.75	22

4. Annual software module

DSO has to deliver the above-mentioned flexibility table which informs about the needed flexibility for the specific network part. The basis of the calculation, i.e. the input is the result of the Load Flow calculation for each month, for each selected, specific type of day (weekdays, Saturday, Sunday). This means 36 scenarios with 96 lines, since the calculation is made in quarter-hourly resolution.

The LT module excel file consists of Annual and Multiannual parts. The Annual part can be split into so called sub modules:

1. Calendar submodule
2. Calculation input submodule
3. Flexibility calculation submodule
4. DSO flexibility table (Output)

The interrelationship of these modules is described below:

Calendar

The Calendar sheet contains the type of days in each month of the given year which is used in DSO Flexibility table in Output.

The holidays have to be handled manually.

Calculation input

The Calculation input_Scenarios contains the essence, the result of network examination, this is a time series of network load assigned to months and types of days.

The network limits can be edited here.

The Calculation starts the calculation of the needed flexibility. Beside of this – for sake of testing - the user can choose given month-type of day setting and display the calculation on the Flexibility calculation page, in this way the tester can check the calculation of

Flexibility calculation

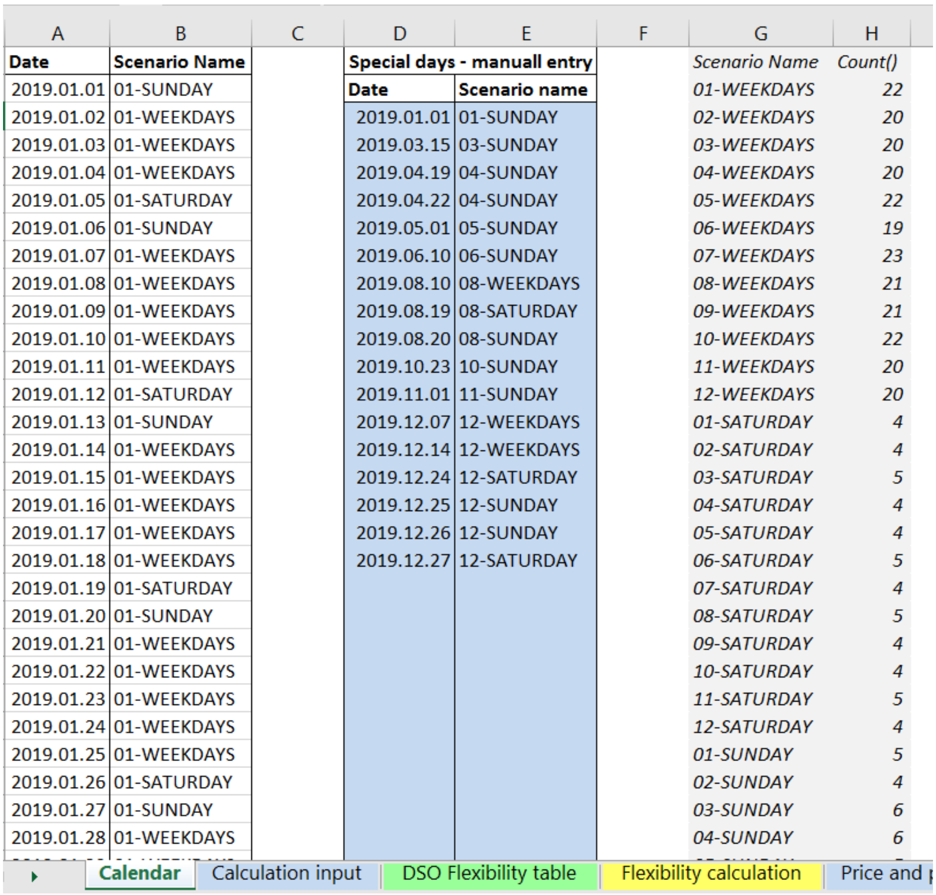
This is the core of the module, here the software compares the operational limits of the network with the load curves served by scenarios, and results in flexibility needs in kW and corresponding time intervals.

DSO flexibility table (Output)

The result of the flexibility calculation is the DSO Flexibility table, which contains line by line the needed flexibility in kW and time interval for specific month and type-of-days. Beside of the „raw“ flexibility data the table contains the number of the specific type of days per month which later can help in the Multiannual module to count all the requested flexibility in kWh in the year and it will be used for unit price calculation of the Activation part of the request.

Calendar

The below screen shot contains the Date sheet which is an assistant table. Here the user defines the type of day for the given date, and the special days (national holidays, Christmas, etc.) has to be handled. This table also counts the type of days within each month which is used in DSO Flexibility table in Output.



A	B	C	D	E	F	G	H
Date	Scenario Name		Special days - manuall entry			Scenario Name	Count()
2019.01.01	01-SUNDAY		Date	Scenario name		01-WEEKDAYS	22
2019.01.02	01-WEEKDAYS		2019.01.01	01-SUNDAY		02-WEEKDAYS	20
2019.01.03	01-WEEKDAYS		2019.03.15	03-SUNDAY		03-WEEKDAYS	20
2019.01.04	01-WEEKDAYS		2019.04.19	04-SUNDAY		04-WEEKDAYS	20
2019.01.05	01-SATURDAY		2019.04.22	04-SUNDAY		05-WEEKDAYS	22
2019.01.06	01-SUNDAY		2019.05.01	05-SUNDAY		06-WEEKDAYS	19
2019.01.07	01-WEEKDAYS		2019.06.10	06-SUNDAY		07-WEEKDAYS	23
2019.01.08	01-WEEKDAYS		2019.08.10	08-WEEKDAYS		08-WEEKDAYS	21
2019.01.09	01-WEEKDAYS		2019.08.19	08-SATURDAY		09-WEEKDAYS	21
2019.01.10	01-WEEKDAYS		2019.08.20	08-SUNDAY		10-WEEKDAYS	22
2019.01.11	01-WEEKDAYS		2019.10.23	10-SUNDAY		11-WEEKDAYS	20
2019.01.12	01-SATURDAY		2019.11.01	11-SUNDAY		12-WEEKDAYS	20
2019.01.13	01-SUNDAY		2019.12.07	12-WEEKDAYS		01-SATURDAY	4
2019.01.14	01-WEEKDAYS		2019.12.14	12-WEEKDAYS		02-SATURDAY	4
2019.01.15	01-WEEKDAYS		2019.12.24	12-SATURDAY		03-SATURDAY	5
2019.01.16	01-WEEKDAYS		2019.12.25	12-SUNDAY		04-SATURDAY	4
2019.01.17	01-WEEKDAYS		2019.12.26	12-SUNDAY		05-SATURDAY	4
2019.01.18	01-WEEKDAYS		2019.12.27	12-SATURDAY		06-SATURDAY	5
2019.01.19	01-SATURDAY					07-SATURDAY	4
2019.01.20	01-SUNDAY					08-SATURDAY	5
2019.01.21	01-WEEKDAYS					09-SATURDAY	4
2019.01.22	01-WEEKDAYS					10-SATURDAY	4
2019.01.23	01-WEEKDAYS					11-SATURDAY	5
2019.01.24	01-WEEKDAYS					12-SATURDAY	4
2019.01.25	01-WEEKDAYS					01-SUNDAY	5
2019.01.26	01-SATURDAY					02-SUNDAY	4
2019.01.27	01-SUNDAY					03-SUNDAY	6
2019.01.28	01-WEEKDAYS					04-SUNDAY	6

Figure 25: Calendar

Column E: type of the special days in the appropriate format, e.g. "01-WEEKDAYS", e.g if 01.01 is new year it means SUNDAY because of the load pattern, but (in some case in HU there are some Saturdays when employees have to work instead of another day, therefore it will be WEEKDAYS from load pattern point of view: e.g 08.20 is a public holiday, Tuesday-> match SUNDAY to it, 08.19 is Monday but based on Governmental decision it is nor workday, the employees have to work on 08.10 Staurday instead of 08.19 Monday, therefore 08.10 will be WEEKDAYS, the 08.19 will be SATURDAY. The matching depends on the National holiday plan.

Calculation input

At the beginning of calculation user can set the operational limit of the network. The Limit table is used to record the load limits (operational limits) of the concerned network part (either LV, MV or MV/LV Tr.). The possibility of selection among different limits month by month helps the testing of the system. If we used only one limit for whole year then perhaps the flexibility service would be necessary only in January and December, but in this way, we can test the operation of the system in any time.

A	B	C
Thermal limit of cable/ line	5000	kW
Operational limit (January)	970	kW
Operational limit (February)	970	kW
Operational limit (March)	730	kW
Operational limit (April)	730	kW
Operational limit (May)	730	kW
Operational limit (June)	690	kW
Operational limit (July)	690	kW
Operational limit (August)	690	kW
Operational limit (September)	830	kW
Operational limit (October)	830	kW
Operational limit (November)	830	kW
Operational limit (December)	970	kW

Figure 26: Limits

The below picture of shows the Scenarios which contain the DSO load flow calculation or SCADA measurement. What is important is that the granularity of time series is 15 minutes, the cells contain kW, and the columns serve to distinguish the specific months, type of days from each other. For each month, for each type of day there are time series of network load results, in this case 36 scenarios were calculated but it varies DSO by DSO, if there is no possibility to gather the information in this way then it is also acceptable if e.g. there will be winter-summer seasonal differentiation and there will be weekday-non-weekday differentiation.

E	F	G	H	I	J	K	L	M	N	O	P
Time	January - Weekdays	February - Weekdays	March - Weekdays	April - Weekdays	May - Weekdays	June - Weekdays	July - Weekdays	August - Weekdays	September - Weekdays	October - Weekdays	November - Weekdays
0:00	273,68	264,08	265,07	265,55	261,87	302,74	331,17	323,32	311,98	279,74	265,01
0:15	273,68	264,08	265,07	265,55	261,87	302,74	331,17	323,32	311,98	279,74	265,01
0:30	273,68	264,08	265,07	265,55	261,87	302,74	331,17	323,32	311,98	279,74	265,01
0:45	273,68	264,08	265,07	265,55	261,87	302,74	331,17	323,32	311,98	279,74	265,01
1:00	264,95	256,18	253,75	236,20	230,06	275,24	298,13	293,16	270,06	247,30	255,62
1:15	264,95	256,18	253,75	236,20	230,06	275,24	298,13	293,16	270,06	247,30	255,62
1:30	264,95	256,18	253,75	236,20	230,06	275,24	298,13	293,16	270,06	247,30	255,62
1:45	264,95	256,18	253,75	236,20	230,06	275,24	298,13	293,16	270,06	247,30	255,62
2:00	259,24	252,78	248,85	233,08	224,53	268,00	292,47	284,72	259,22	241,20	251,65
2:15	259,24	252,78	248,85	233,08	224,53	268,00	292,47	284,72	259,22	241,20	251,65
2:30	259,24	252,78	248,85	233,08	224,53	268,00	292,47	284,72	259,22	241,20	251,65
2:45	259,24	252,78	248,85	233,08	224,53	268,00	292,47	284,72	259,22	241,20	251,65
3:00	287,86	261,78	256,57	231,15	223,31	264,45	285,43	281,25	256,36	239,44	257,27
3:15	287,86	261,78	256,57	231,15	223,31	264,45	285,43	281,25	256,36	239,44	257,27
3:30	287,86	261,78	256,57	231,15	223,31	264,45	285,43	281,25	256,36	239,44	257,27
3:45	287,86	261,78	256,57	231,15	223,31	264,45	285,43	281,25	256,36	239,44	257,27
4:00	420,45	387,60	350,32	236,88	224,98	267,68	286,66	281,30	260,04	250,94	395,11
4:15	420,45	387,60	350,32	236,88	224,98	267,68	286,66	281,30	260,04	250,94	395,11
4:30	420,45	387,60	350,32	236,88	224,98	267,68	286,66	281,30	260,04	250,94	395,11
4:45	420,45	387,60	350,32	236,88	224,98	267,68	286,66	281,30	260,04	250,94	395,11
5:00	582,24	543,01	480,14	334,12	309,03	338,35	353,62	348,47	326,96	363,19	529,73
5:15	582,24	543,01	480,14	334,12	309,03	338,35	353,62	348,47	326,96	363,19	529,73
5:30	582,24	543,01	480,14	334,12	309,03	338,35	353,62	348,47	326,96	363,19	529,73
5:45	582,24	543,01	480,14	334,12	309,03	338,35	353,62	348,47	326,96	363,19	529,73
6:00	745,08	703,39	602,45	479,01	429,07	432,84	421,62	419,58	431,52	495,03	695,03
6:15	745,08	703,39	602,45	479,01	429,07	432,84	421,62	419,58	431,52	495,03	695,03
6:30	745,08	703,39	602,45	479,01	429,07	432,84	421,62	419,58	431,52	495,03	695,03

Figure 27: Scenarios

As it was already mentioned the basis is the 36 scenarios - of course it depends on the DSO policy, practice how many scenarios will be calculated - but beside of the scenarios the module user has to give as an input the operational limits of the network. (A thermal limit of the network part can be recorded for the user's own information.)

After that the user has to run the calculation. Based on the background calculation the output table is being filled in, which will be the so-called Flexibility table for the specific network part.

A	B	C	D	E	F	G	H
Operational limit (June)	690	kW		1:00	264,95	256,18	253,75
Operational limit (July)	690	kW		1:15	264,95	256,18	253,75
Operational limit (August)	690	kW		1:30	264,95	256,18	253,75
Operational limit (September)	830	kW		1:45	264,95	256,18	253,75
Operational limit (October)	830	kW		2:00	259,24	252,78	248,85
Operational limit (November)	830	kW		2:15	259,24	252,78	248,85
Operational limit (December)	970	kW		2:30	259,24	252,78	248,85
Calculate				2:45	259,24	252,78	248,85
				3:00	287,86	261,78	256,57
				3:15	287,86	261,78	256,57
Choose year:				3:30	287,86	261,78	256,57
2019				3:45	287,86	261,78	256,57
				4:00	420,45	387,60	350,32
Choose column for calculation check:				4:15	420,45	387,60	350,32
				4:30	420,45	387,60	350,32
Show calculation				4:45	420,45	387,60	350,32
				5:00	582,24	543,01	480,14
				5:15	582,24	543,01	480,14
The following input is needed for calculations:				5:30	582,24	543,01	480,14
Cells *B2-B13* : operational limit for the given month				5:45	582,24	543,01	480,14
Cell *A18* : chosen year (updates data in *Calendar* sheet as well!)				6:00	745,08	703,39	602,45
Cells *F3-AO98* : 15-minute load data input				6:15	745,08	703,39	602,45
If *Calendar* sheet is valid, click on the *Calculate* button to run!				6:30	745,08	703,39	602,45
				6:45	745,08	703,39	602,45
The following is required for the calculation check:				7:00	745,08	968,00	786,49
Cell *A21* : have to choose the month and type of month for checking				7:15	1041,94	968,00	786,49
				7:30	1041,94	968,00	786,49
				7:45	1041,94	968,00	786,49
				8:00	1038,75	943,45	796,13

Figure 28: Calculation command

Flexibility calculation

The calculation part of the module can be checked on the Flexibility calculation sheet:

Scenario	Time	Estimated load by AC power flow [kW]	Estimated load by AC power flow which exceeds the operational limit [kW]	Operational limit P_{op} [kW]	Network limit P_{max} [kW]	Predefined operational limit (% of P_{max})
December - Sunday	0:00	285	0	970	5000	19%
December - Sunday	0:15	285	0	970	5000	19%
December - Sunday	0:30	285	0	970	5000	19%
December - Sunday	0:45	285	0	970	5000	19%
December - Sunday	1:00	277	0	970	5000	19%
December - Sunday	1:15	277	0	970	5000	19%
December - Sunday	1:30	277	0	970	5000	19%
December - Sunday	1:45	277	0	970	5000	19%
December - Sunday	2:00	274	0	970	5000	19%
December - Sunday	2:15	274	0	970	5000	19%
December - Sunday	2:30	274	0	970	5000	19%
December - Sunday	2:45	274	0	970	5000	19%
December - Sunday	3:00	273	0	970	5000	19%
December - Sunday	3:15	273	0	970	5000	19%
December - Sunday	3:30	273	0	970	5000	19%
December - Sunday	3:45	273	0	970	5000	19%
December - Sunday	4:00	307	0	970	5000	19%
December - Sunday	4:15	307	0	970	5000	19%
December - Sunday	4:30	307	0	970	5000	19%
December - Sunday	4:45	307	0	970	5000	19%
December - Sunday	5:00	323	0	970	5000	19%
December - Sunday	5:15	323	0	970	5000	19%
December - Sunday	5:30	323	0	970	5000	19%
December - Sunday	5:45	323	0	970	5000	19%
December - Sunday	6:00	340	0	970	5000	19%

Calendar | Calculation input | DSO Flexibility table | Flexibility calculation | Price and penalty | Flexibilit

Figure 29: Flexibility calculation

The calculation compares the operational limit with the scenarios (Load flow calculation results) and determines the intersection of the load curve and the operational limit. After that the module seeks the local maximum(s) of the curve and determines the needed flexibility which is the local maximum of the curve. Based on this calculation the module fills in the output table which is the concrete Flexibility table, see below:

DSO Flexibility table (Output)

The output contains the number of specific type of days within the specific month (Pcs of type of days), which is calculated on the Calendar sheet.

Month	Type of day	Flexibility requirement [kW]	Time interval (Start)	Time interval (Length)	Flexibility requirement [kWh]	Pcs of type of days
2019-01	WEEKDAYS	-71,94	7:15	1,75	-125,90	22
2019-01	WEEKDAYS	-39,02	13:00	3,00	-117,06	22
2019-03	WEEKDAYS	-66,13	7:00	3,00	-198,39	20
2019-03	WEEKDAYS	-20,51	13:00	2,00	-41,03	20
2019-04	WEEKDAYS	-62,86	8:00	2,00	-125,72	20
2019-06	WEEKDAYS	-16,01	9:00	1,00	-16,01	19
2019-06	WEEKDAYS	-51,02	12:00	5,00	-255,08	19
2019-07	WEEKDAYS	-1,93	14:00	2,00	-3,86	23
2019-10	WEEKDAYS	-59,98	8:00	2,00	-119,96	22
2019-11	WEEKDAYS	-97,73	7:00	2,00	-195,46	20
2019-11	WEEKDAYS	-47,66	13:00	4,00	-190,63	20
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	
				0,00	0,00	

Calendar | Calculation input | DSO Flexibility table | Flexibility calculation | Price and penalty | Flexibility unit prices,

Figure 30: Output table

Attention: the negative sign refers to the downward flexibility, i.e. the DSO needs a load reduction. If the sign would be positive then it would mean an upward flexibility, i.e DSO would need a load increase (e.g due to voltage increase because of renewables). In our case we will use only the negative sign, and the calculation is prepared for load reduction. Later on during the unit price calculation we use absolute value function to avoid negative prices.

The Flexibility table will give the numerical description of the below picture, attention, the below picture is only a visual example/aid, the numbers do not fit to above tables:

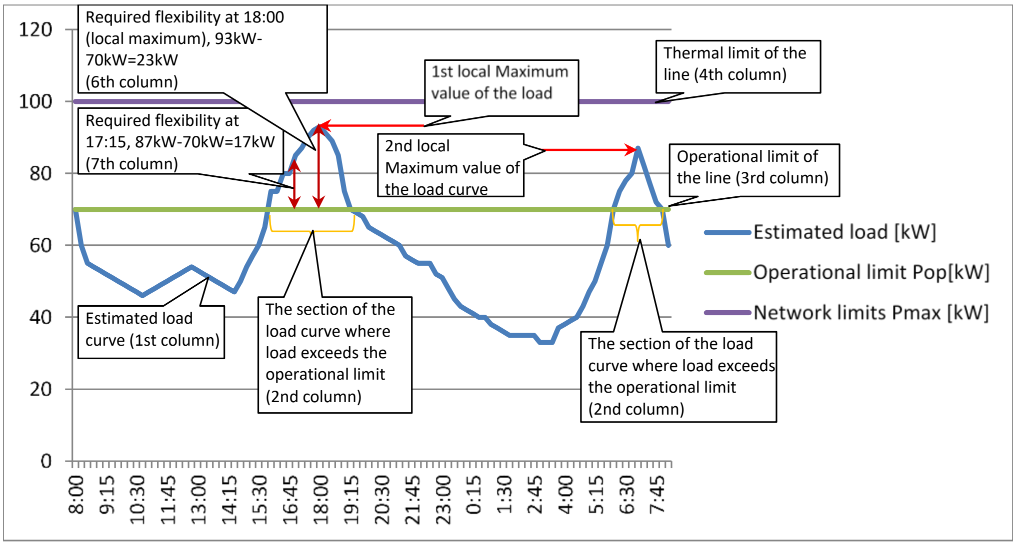


Figure 31: Visualized flexibility calculation

That is, it gives the needed flexibility for the network with the needed time interval. Furthermore, it calculates the flexibility service in kWh as well, because the Flexibility table will be the input for the Multiannual software module, where we have to use the kWh dimension as well for the unit price calculation.

The programmed SW module of LT module can be found below as embedded file:

3Smart_LT module_v4.xlsm

Description of Visual Basic Macros (VBM) used in Annual SW module

This part of the document presents the VBA macro logic of the flexibility calculation in LT_module.*xlsm'*.

The user can run the calculation in the sheet *'Calculation input'* by clicking on button *'Calculate'*. The input data should be given in the sheet *'Calculation input'*. Only the blue marked cells can be edited, otherwise the macro will not run correctly.

The following input should be checked in any case before calculation:

- The scenario values: If possible, it is recommended to fill each column with data. Otherwise, blank columns will not be processed into the output table.
- The thermal limit have to be given in the yellow cell in *'Calculation input'*.
- The operational limit has to be given for each month! If any cell is left blank, the calculation will return with some hashtag values.

After calculation, the distinct month-daytype values can be checked. There is a combobox in cell 'A21' in *'Calculation input'*. After clicking the button below, the calculation will be shown for the chosen daytype in *'Flexibility calculation'*.

The control buttons in this Excel are linked with VBA scripts running in the background. The monthly operational limits are stored in a 12-element array. The month, daytype, local maxima values, time interval start and time interval end values are stored in separate 200-element arrays. For these procedures, various indexes used while counting and looping. At the end, the output rows are generated from the array values (defined as maximum 200). The 200 value limits the rows of flexibility table, i.e. maximum 200 raws can be editied in flexibility table (in this way more local maximums of daily load curve can be handled.

```
Sub Calculate(opLimit(), networkLimit)

Dim currentLoad, i, j, monthIndex, localMaximum
Dim monthArray(200), daytypeArray(200) 'month and date types are written out using arrays
Dim maximumArray(200), startArray(200), endArray(200), maximumIndex, intervalIndex 'local maxima and date intervals are written out using arrays
i = 2 'starts with row "i + 1" on the sheet "Calculation input", row "i" on "Flexibility calculation"
j = 6 'starts with row "j" on the sheet "Calculation input", runs until an empty cell is found
maximumIndex = 1
intervalIndex = 1

Do Until ActiveWorkbook.Sheets("Calculation input").Cells(2, j).Value = ""

    'Check the month
    monthIndex = ValidateMonth(j)
    ReadWriteLimits opLimit(), networkLimit, monthIndex

    ActiveWorkbook.Worksheets("Flexibility calculation").Range("C2:C97").ClearContents

    For rowNum = 2 To 97
        currentLoad = ActiveWorkbook.Sheets("Calculation input").Cells(rowNum + 1, j).Value
        ActiveWorkbook.Sheets("Flexibility calculation").Cells(rowNum, 3).Value = currentLoad
        ActiveWorkbook.Sheets("Flexibility calculation").Cells(rowNum, 1).Value = ActiveWorkbook.Sheets("Calculation input").Cells(2, j).Value
    Next

    ActiveWorkbook.Sheets("Flexibility calculation").Calculate

    'writes time intervals into array:
    TimeIntervalsIntoArray startArray(), endArray(), intervalIndex

    'writes local maxima (based on calculation) into array:
    LocalMaximaIntoArray monthArray(), daytypeArray(), maximumArray(), maximumIndex

    j = j + 1
Loop

ActiveWorkbook.Sheets("DSO Flexibility table").Range("A3:D200").ClearContents
ActiveWorkbook.Sheets("DSO Flexibility table").Range("I3:I200").ClearContents

'WRITES OUTPUT VALUES TO SHEET 'DSO FLEXIBILITY TABLE':
Dim rowOutput
rowOutput = 3
```

Figure 32: VBA script

If the flexibility requirement of a daytype is not greater than 0, then it is filtered out from the result table.

There are 2 VBA macro script modules. One for the overall calculation, and another for checking the calculated values for a selected daytype.

The sub procedures of the calculating VBA macro script are the following:

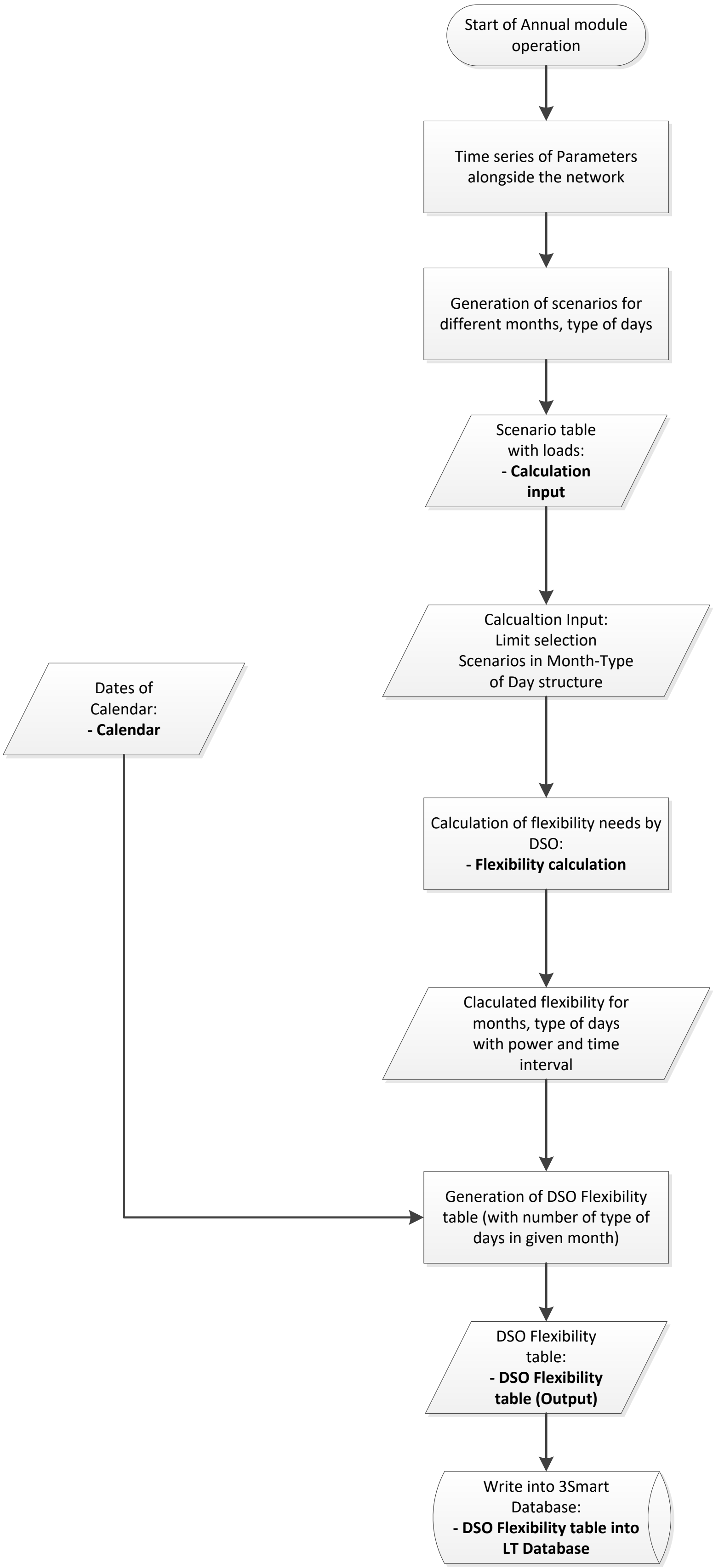
- **Szamolas_Click**: The main sub which runs at first by clicking on the *'Calculate'* button. Writes the operational limits into an array, and calls **Calculate** procedure
- **Calculate**: Goes through each input column, and each cell within those, copies the load values into the calculating sheet. Uses monthly index by calling **ValidateMonth**. Calls the **TimeIntervalsIntoArray** and **LocalMaximaIntoArray** procedures per column. At the end, writes the array values into the output rows.

- **TimeIntervalsIntoArray**: Saves the time interval start and end per daytype to separate arrays, then uses 3 hidden helper columns to choose local maxima per daytype.
- **LocalMaximaIntoArray**: Saves the chosen local maxima values into an array.
- **ValidateMonth**: Assigns an index value to each month for calculation and looping.

The sub procedure of the checking VBA macro script is the following:

- **Gomb8_Click**: Runs by clicking on the *'Show calculation'* button. Checks the combobox value, then writes the related values to the sheet *'Flexibility calculation'*.

Flow chart for Annual software module



It can be deduced from the above flow chart where a Database interaction is necessary.

Bibliography

[1] 3Smart deliverable D5.1.1. Grid-side EMS concept and information exchange interfaces definition. December 2017.

[2] 3Smart deliverables D5.4.1. Grid models for grid simulation in professional simulation tools. December 2017.

[3] iPower: Development of a DSO-market on flexibility services.
http://orbit.dtu.dk/en/publications/development-of-a-dsomarket-on-flexibility-services(2c6422eb-5190-4421-a79c-2692db0567ca).html
March 2013.

[4] AccountingExplained.com. NPV and Inflation.
https://accountingexplained.com/managerial/capital-budgeting/npv-and-inflation
March 2018.

[5] “Energie Koplopers: Flexibility from residential power consumption: A new market opportunities”

Project Deliverable Report

Smart Building – Smart Grid – Smart City

http://www.interreg-danube.eu/3smart

DELIVERABLE D5.4.3.

Final grid-side energy management software module – Long -term Multiannual module

Project Acronym	3Smart
Grant Agreement No.	DTP1-502-3.2-3Smart
Funding Scheme	Interreg Danube Transnational Programme
Project Start Date	1 January 2017
Project Duration	36 months
Work Package	5
Task	5.4
Date of delivery	**Contractual:** 30 June 2019 **Actual:** 30 June 2019
Code name	**Version:** 1.0 Final ☒ Final draft ☐ Draft ☐
Type of deliverable	Report
Security	Public
Deliverable participants	EON, UNIZGFER
Authors (Partners)	Gábor Péter Mihály (EON), Kata Sánta, Gábor Hornyák (EON), Tomislav Capuder, Paula Perović, Mario Vašak (UNIZGFER)
Contact person	Tibor Béni (EON)
Abstract (for dissemination)	The deliverable contains the final technical solution description for long term grid-side energy management module of the 3Smart tool.
Keyword List	Grid-side flexibility tools, Energy Management system

Revision history

Revision	Date	Description	Author (Organization)
v0.1	20 December 2018	First draft	Katalin Décseiné Giczi (EON)
v0.2	20 June 2019	Draft	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON)
v0.3	28 June 2019	Draft	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON)
v0.4	29 June 2019	Final draft	Tibor Béni, Kata Sánta, Gábor Hornyák, Gábor Mihály Péter (E.ON), Paula Perović, Tomislav Capuder (UNIZGFER)
v1.0	30 June 2019	Final quality-checked version	Hrvoje Pandžić, 3Smart quality assessment manager, Mario Vašak (UNIZGFER)

Table of Contents

Executive summary

One of the objectives of the project Smart Building – Smart Grid – Smart City (3Smart) is creation of an integrated and modular energy management tool for the DSO to use buildings as their assets and to utilize their flexibility in order to efficiently plan investment into the distribution grid. The tool is organized in two main submodules, Long-term and Short-term module. The Long-term module is used by the DSO for offline planning and it allows the DSO to plan their investments based on reserving building flexibility services by substituting CAPEX with higher OPEX paid to the building for providing services. Long-term Annual module calculates the DSO needed flexibility and Multi-Annual module calculates the flexibility service fees (activation, reservation and penalty prices). On the other hand, short-term module is used by the DSO to optimize usage of flexibility services and schedule. Day-ahead module is used for determining time windows for utilizing daily flexibility coming from reservation windows in the long-term contract and the intra-day module is used for improving daily schedule.

The focus of deliverable 5.4.3 is on energy management tools interfaces. Deliverable 5.4.3. has 4 outputs, one for each submodule. The deliverable explains every variable that is used as submodule inputs and describes the final output of the modules. Database outlook is described in Annex 1 while submodules algorithms and logic are provided in Annex 2. This document presents Long-term Multiannual module interface tables.

1 Introduction

An important part of the deliverable is the description of the long-term module functionalities from the DSO point of view, as well as that of a distributed demand response provider, in the market environment. Currently, neither the DSO or demand response providers participate in the market. The market, in this deliverable, is seen through participation of multiple stakeholders exchanging money and services in a transparent way either through tenders or, more preferably, at power exchange. This means that the DSO does not have any information of the accepted demand profiles from its users and operates the network based on vast experience and available historical data. On the other hand, not enabling market access to the distributed energy sources results in overbuilding and underutilization of the DSO assets.

Integrated tool for Long term energy management of DSO, installed with the 3Smart project, is divided in two modules: Annual and Multiannual. Both modules are run on a yearly basis. The Multiannual module is divided into the next submodules:

1. Price and penalty
2. Flexibility unit prices, penalty
3. Building Flexibility table
4. Output for long term contract

Each part of the module consists of three parts: i) input for specific calculation function, ii) the calculation function itself and iii) the output of the specific calculation function.

The modules are designed, commissioned and tested on different pilot location.

In this deliverable the focus is put on Multiannual module, its logic, algorithm and variables.

Each submodule is presented via an interfacing table that explains what data is used by the submodule as inputs and what is the final output data. The algorithms behind the module are explained in more details in the annexed document.

1.1. Long term Multiannual (MA) module interface tables

An important feature of the MA LT module is that all inputs and outputs of the module are directly written from/in database of other modules (e.g. Annual, DA). In case of the Multiannual module there is only one table which does not appear as database table in other modules, namely Price and penalty, while other excel tables appear in other module databases either as input (Building Flexibility table, Flexibility unit prices, penalty) or as output (Output for long term contract). In that sense, the following Input/output table describes not only the database entities but the LT excel inputs as well.

Table 1. Long-term Multiannual input interface table

Variable name	Variable annotation	Variable description	Source
Price and penalty sheet	Price and penalty	The calculator sheet needs the following inputs by DSO Staff: WACC, Inflation, The cost	DSO decides on input data for Price and penalty

		of investment, Ratio of used flexibility price, Year (considered life time). Reservation ratio, Penalty price multiplicator, Quality threshold (max. deviation in size of service without penalty). The main results: Reservation part of Flexibility unit price, Activation part of Flexibility unit price, Penalty which are used other input table, namely: Flexibility unit prices, penalty.	
Flexibility unit prices, penalty sheet	**flexibility_unit_prices_and_penalty**	Contains of: Contract valid from, Contract valid until, Reservation part of Flexibility unit price, Activation part of Flexibility unit price, Penalty price (per kWh non-delivered below the threshold), Deviation in size of service (Quality threshold): Max. These values serve as an input for the Building Microgrid module, beside of DSO Flexibility table which is part of Annual module.	Price and penalty sheet
Building Flexibility table sheet	**building_flexibility_table**	This table contains the provided (planned) flexibility by Building for the given contractual period which is a crucial input for DSO for creation of Long-term contract (in database: contract)	Building Microgrid module

Table 2. Long-term Multiannual output interface table

Variable name	Variable annotation	Variable description
Output for long term contract	**contract**	The following are either the results of the contract table or manual entries (me) by DSO Staff (after completion, the LT workflow send it to Building for acceptance): Contract valid

		from, Contract valid until, Est. no. of activations during period, Maximum Size of service in power (kW), Max. duration of service per activation (h), On – Trigger (me), Off – Trigger (me), Maximum allowed activation time (me), Quality of Service (me), Unit price of Reservation (EUR/kW/(15min)), Unit price of Activation (EUR/kWh), Reservation fee for the contractual period, Activation fee for the whole contractual period, Average activation price/activation, Pricing , Penalty if failed supply (only termination criteria has to be manual entry), Building offer accepted (manual entry by Building)

Conclusion

Web based workflow tool is developed for both the Annual and Multiannual modules. The central point of LT workflow is an excel ("3Smart_LT module_v4.xlsm") which requires manual inputs from DSO staff. DSO staff has to be aware of these manual interventions, nevertheless the LT workflow gives a logical guide for it.

The interconnection of the long-term and short-term modules of the grid-side EMS is crucial for running daily load flow calculations. The daily AC load flow calculation will use the distribution network and load parameters from the standardized database/ or table prepared by the DSO and the basic logic is similar to the long-term scenario-based calculations. This AC OPF (optimal power flow) module completes the optimization calculation described in the short-term module description. An important output of the LT module is the DSO and Building flexibility table which is a key input to the AC OPF to precisely calculate the needed flexibility utilization from offered amount and time interval.

Furthermore, another key interconnection is between Long Term module and Building side Microgrid module: based on DSO Flexibility table results (submodule) the Microgrid will answer with a so-called Building Flexibility table which contains the capability of the Building to provide requested flexibility service.

Important changes appearing in D5.4.3 when compared to module described in 5.3.1 Open software module for long-term level of grid-side energy management- Multiannual module description:

- the former version of the module contained a more fragmented LT excel. Although the following submodules were created, they have been slightly changed/adjusted
  - *Inputs for flexibility cost,*
  - *Calc. of available resource,*
  - *Output of flexibility cost,*
  - *Input for unit price calc._1,*
  - *Unit price calculation,*
  - *Input for longtermcontract_1*

  The above submodules are now organized into one submodule: Price and penalty. This submodule contains all relevant information and makes the usage of submodule easier.
  The *Input for unit price calc._2* table served as an input for unit price calculation, but in the unified LT excel it is part of the DSO Flexibility table.
- *Output of unit price calc., Input for longtermcontract_1* are organized into one submodule, into *Flexibility unit prices, penalty*.
- *Input for longtermcontract_2* table is renamed to Building Flexibility table.
- *Long term contr. preparation* table was merged into *Output for long term contract* table.
- Beside of the restructuring of LT excel, the Reservation unit price calculation was replaced by a new, more logical calculation which is described in Annex. The numerical example also followed this change and is shown in Annex.

Annex I

1. Database architecture

The structure is envisioned in a way that the output tables of one entities database are defined in the same way as the input tables of another entities database, which makes understanding of the database structure a crucial part of the development process. The Long Term database is not separated into two databases (Annual and Multi Annual); the building_flexibility_table and flexibility_unit_prices_and_penalty belong to Multiannual, the other part of database belong to Annual part.

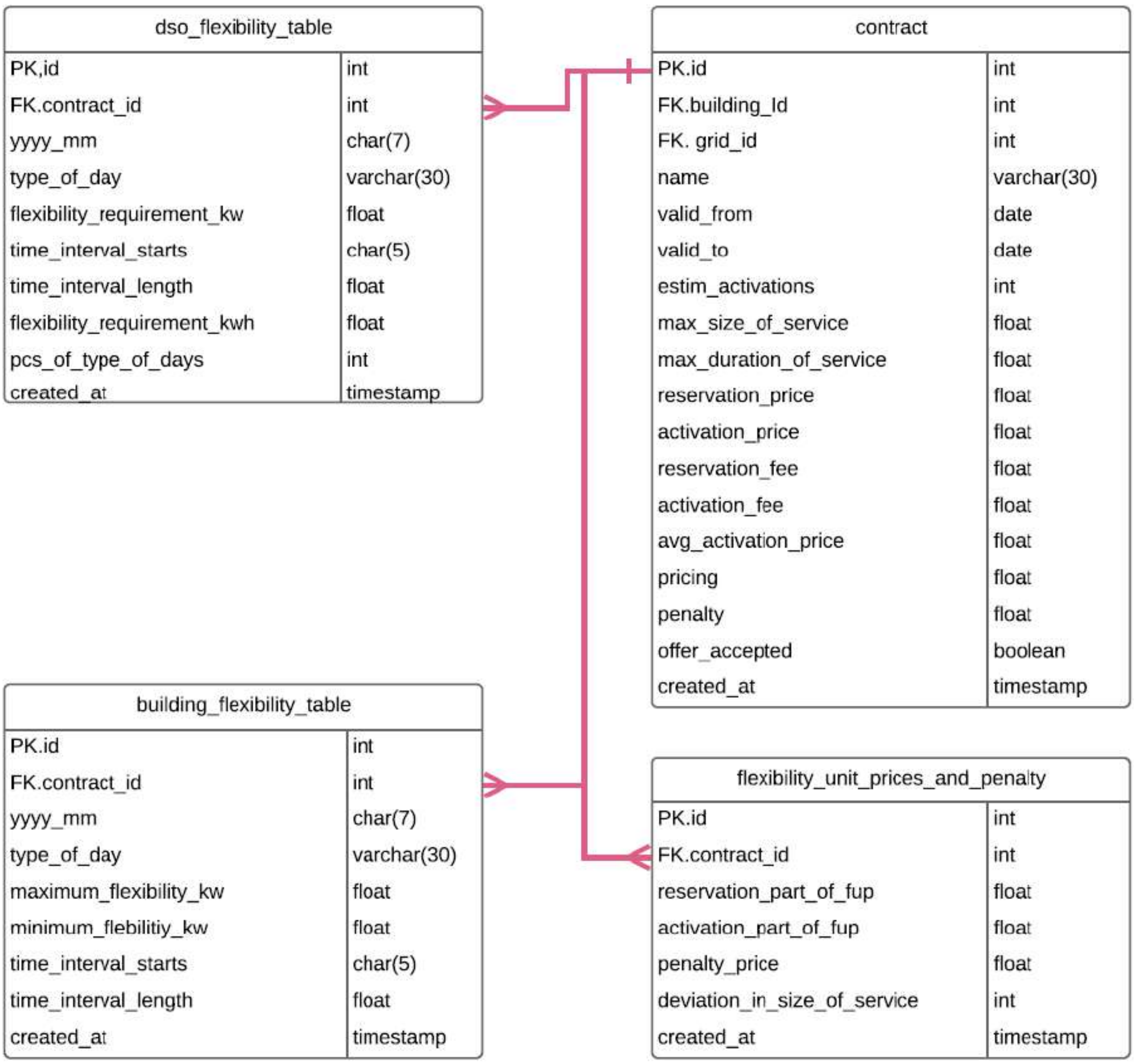


Figure 1 – Long-term database tables and relations

2. Communication events between the modules related to long-term grid-side operation

The below table describes the communication mechanism of the LT workflow which contains steps regarding both Annual and Multiannual modules. The main characteristic of the description is to focus on Grid-Building interaction and related database manipulations.

Table 3. Chronological communication events with detailed exchange structure

ID	Time (UTC)	Data exchange/ activity	D.5.3.1 (Annual and Multiannual) Nomenclature	module	Reads data	Puts data at disposal	Tri-gger
1	till December, before contract agreement	Calculation of flexibility needs, prices, penalty and quality of service by using " 3Smart_LT module_v3.xlsm"	Result: DSO Flexibility table; Flexibility unit prices,penalty; Output for long term contract sheets	LT module	DSO (staff)	DSO (staff)	0
2	till December, before contract agreement	Importing results of " 3Smart_LT module_v3.xlsm"	Result: DSO Flexibility table; Flexibility unit prices,penalty; Output for long term contract data base tables	LT module	DSO (LT)(script 1)	DSO (staff)	0
3	After step 2	Building EMS Microgrid module is fetching data from LT database		Microgrid	Building	DSO (LT)	0
4	After step 3	Building calculate flexibility offer	Result: Building Flexibility database table, (Microgrid database)	Microgrid		Building	0
5	After step 4	DSO (LT) module is fetching data from Microgrid database		LT	DSO (LT) (script2)	Building	0
6	After step 5	Generating file from Building Flexibility table	Result: Building Flexibility table in Excel	LT	DSO (staff)	DSO (LT) (script3)	0
7	After step 6	Contract preparation by DSO, inserting Building Flexibility table into " 3Smart_LT module_v3.xlsm"	Result: Output for long term contract sheet	LT		DSO (staff)	
8	After step 7	Acceptance/Rejection of Building offer	Result: Offer acceptance sheet (Yes/No)	LT		DSO (staff)	
9	After step 8	Importing Output for long term contract sheet of " 3Smart_LT module_v3.xlsm"	Result: The details of contract in Database	LT	Building	DSO (LT) (script4)	

Annex II

Logic of Open software module for long-term level of grid-side energy management - Multiannual module

1. Multiannual load forecasting: Method of load forecasting

1.1 The used method of load forecasting

Beside the annual planning, where the details are considered, the DSO derives the needed equipment upgrade from the multiannual load forecasting. Based on the load forecasting the DSO can run load flow calculation and can decide whether and what kind of technical intervention should be elaborated (e.g. increase of the cross section of the line, or introducing some voltage deviation mitigation technology, etc.).

Long-term load forecast is based on time series forecasting. Series of past measurements are available for trend analysis. The estimation is then modified based on the correlation between economic development and the change in electricity demand. The change in electric loads shows determinate correspondence with the development of the economy and its qualitative and quantitative changes. Long-term load estimation represents an output of the estimates forecasting the state of the economy.

Exceptional events (e.g., new HV/MV station, large consumer or distributed generation unit) and the effect of photovoltaic generation on the summer daytime peak are taken into consideration during the process. The load estimation is revised annually.

The span of long-term load estimation is 10 years at most and is needed for determining development of the main distribution network. The Transmission System Operator regularly prepares the national long-term forecast for electricity usage and consumer peak load. E.ON needs regional data for its area of operation as this differs from the national average, also different regions of the operating area show significant difference from each other. To lay the regional foundation for development of the main distribution network, the difference in characteristics of each region from the national average needs to be considered.

1.2 Load forecast for Debrecen HV/MV and given MV line

The method described for main distribution networks can be extended to medium voltage distribution backbones.

Load estimation for branch lines of distribution networks, big city medium voltage distribution networks, and low voltage networks would require a finer resolution of load distribution data within a region, which can only be determined with a great degree of inaccuracy. The appearance of concrete power demands following the values defined in the regional estimate only in average is to be expected. The concrete, local appearance of a part of these demands cannot be defined with the required accuracy even for a short term. Demands “not covered” should be taken into account when planning investment resources, but exactly how and at which network parts these have to be considered can only be determined once the specific demand has arisen. The concrete technical tasks and investments should be chosen and worked out with consideration for the technical-economic principles. A quick reacting planning and investment organization is required to achieve this.

The optimistic (high) load prognosis for the supply area of Debrecen Délkeleti 132/22 kV substation is 2.27% and the low prognosis is 2.07%.

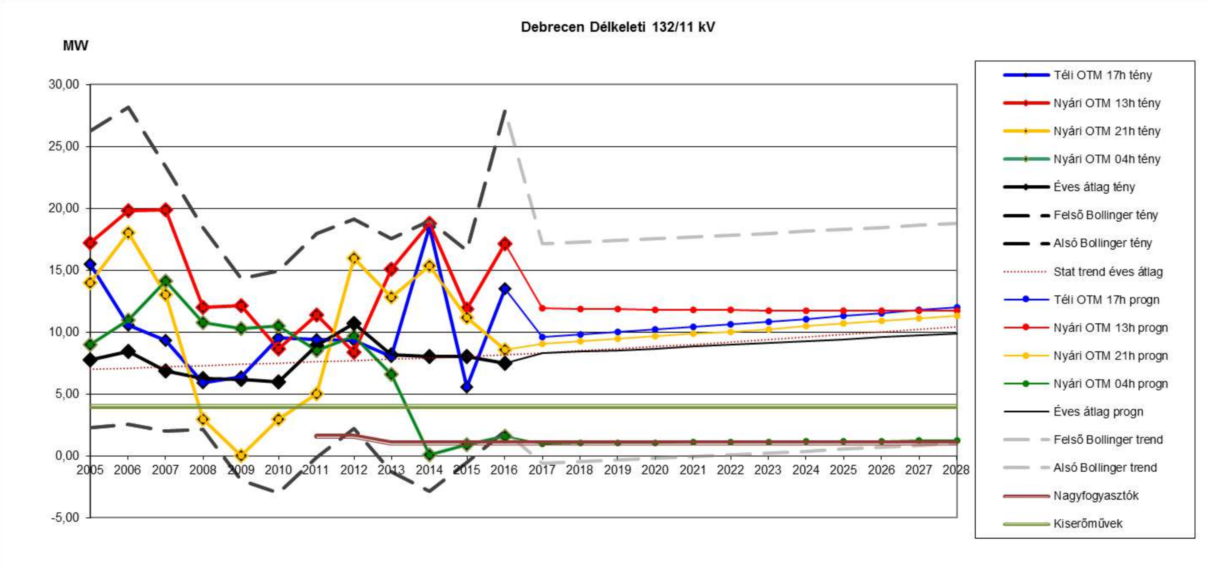


Figure 2 – The diagram of the low load forecast for the HV/MV substation

2. Method for preparing and applying long-term contract in simulations

Here we describe the preliminary steps that are needed for a long-term contract, namely, the technical and economic parameters that need to be considered. These parameters include frequency of activation, activation time, avoided cost, the method of deriving the flexibility price from avoided cost. The chapter contains long-term contractual framework for the DSOs. The steps of price calculation will be detailed in chapter 5 with an example.

2.1 Considered factors

Derivation of the price calculation

The needed flexibility price should be derived from such investment alternative that are necessary when the DSO wants to mitigate the possibility of supply quality insufficiency (e.g. voltage fluctuation, deviation) or avoid physical deterioration of its equipment (due to overloading). Here we consider the case of overloading.

The needed amount of flexibility can be determined from the scenarios. The DSO should carefully examine the circumstances. Based on the load flow results and in accordance with its technology policy, the DSO can estimate how many times and for how long the set limit would be exceeded and should decide whether the equipment would be jeopardized or not. (For example, if the load of a line or a transformer exceeded the operational limit for longer than X hours more than Y times a year, then an upgrade would be necessary. However, in some instances the limit is exceeded only for a short time once or twice a year.)

The price will be derived from the alternative DSO equipment upgrade investment cost, in our case the MV line upgrade. We should determine the total price of the flexibility for a given period, for the time the contract is valid between the building and the DSO. Next, the whole price of the flexibility has to be divided in two and the ratio between the sum of reservation fees and the sum of activation fees has to be determined.

Frequency of activation, time of activation, size of services, penalty

The frequency of activations, the time interval of each activation and the needed size of flexibility for each time interval can be derived from the load flow calculations for each scenario. The results will change yearly due to the load increase (or decrease), therefore the DSO should run the scenarios with appropriate load forecast yearly. If the load flow calculation based on load forecast shows that in the next year the load will exceed the operational limit several times, the DSO will decide whether a line upgrade is necessary or not. If necessary, the maximum amount of the needed flexibility will be determined.

After that, the DSO decides about the ratio of the reservation fee to the activation fee. When this background is determined, the number of necessary activations and the needed flexibility amount for each activation has to be determined. The time interval of activations will be calculated from the long-term LFs. When we have the needed flexibility per activation, the activation frequencies and durations, the ratio between reservation and activation, then the DSO will be able to calculate the basic parameters of the long-term contract for a given customer: the reservation fee (unit price of reservation*available flexibility in kW) and the activation price, which is described below. Of course, prior to “signing” the contract, the customer has to report its available flexibility. Nevertheless the unit prices will not be affected by the report. Based on above parameters the DSO can calculate the reservation fee (unit price of reservation*available flexibility in kW) and the activation price, which is described below.

(Indicative) activation price calculation

If we already have the ratio between the reservation and activation, then we can calculate the activation price per kWh as well if we know the needed flexibility per activation and the number of activations. The activation price calculation will be based on the following equation:

sum cost of activation part of flexibility for the whole line (whole cost - reservation cost) in EUR /
sum of needed flexibility in kWh,

The dimension will be EUR/kWh/activation. The sum of needed flexibility is based on the flexibility table which informs the building of the needed flexibility size and duration for each typical day in each month (the maximum needed kW*duration will give the needed flexibility in kWh, it is needed to sum up all activations within the year),

The building can calculate the possible revenue based on its flexibility service in the following way:

E -- overall kWh of flexibility which grid needs in the period for which contract is made(whatever the period is);

N – overall number of 15-minute time intervals of flexibility in the contracting period;

Y – money in EUR which the grid invests into flexibility in the contracting period;

0<a<1 – part for activation;

0<r=1-a<1 – part for reservation

Price of activation (EUR/kWh): a * Y / E

Needed overall average power for flexibility (kW): P = E / (0.25 h * N)

Price for reservation in one 15-minute time period (EUR/(kW in one 15-minute period)): (r * Y / P) / N

Reservation revenue= Price for reservation in one 15-minute time period (EUR/kW)* sum of provided flexibility by Building (N)

Activation revenue= activation price in EUR/kWh* provided amount of flexibility in the year in kWh

Table 1 does not inform about the accurate service provision of the building; the Building flexibility table should be attached to the contract because it describes the used service in the contractual period in a more accurate way.

2.2 Contractual framework

The basis of the long-term contract will be the table below:

Table 4- Contract framework

Service name		Flexibility service
1)	**Contract valid from**	dd.mm.yyyy.
2)	**Contract valid until**	dd.mm.yyyy.
3)	**Est. no. of activations during period**	The estimated number of activations in the Contract period
4)	**Size of service in power (kW)**	This is the maximum value of flexibility in kW in the contractual period

5)	**Max. duration of service per activation (h)**	e.g. 3 hours
6)	**On - Trigger**	Signal from the DSO or according to DA AC OPF calculator
7)	**Off - Trigger**	Maximum: see "Max.duration of service per activation" from “on”-signal, or by earlier signal from the DSO
8)	**Maximum allowed activation time**	15 min (but it depends of the capability of the Customer process technology)
9)	**Quality of Service**	- Deviation in max. duration: +/- 15 min. - Deviation from, On - Trigger: +/- 15 min. - Deviation in size of service: Max. +/- 10% deviation - Acceptable no. of unsuccessful activations: x
10)	**Unit price of Reservation (EUR/kW)**	Based on flexibility table of DSO, the unit price is derived from the provided flexibility in kWh and sum of provided flexibility by Building (N)
11)	**Unit price of Activation (EUR/kWh)**	Based on flexibility table of DSO, the unit price is derived from the needed flexibility in kWh (this is the activation part of the service)
12)	**Reservation fee for the contractual period**	The whole amount of the reservation fee for the contractual period
13)	**Activation fee for the whole contractual period**	The whole amount of the activation fee for the contractual period
14)	**Average activation price/activation**	Price of one activation: Since during the whole contractual periode the duration and size of activation varies day by day, the entire Activation fee ("Activation fee for the whole contractual period") will be diveded by the number of activations from "Est. no. of activations during period "
15)	**Pricing**	Reservation fee for the whole contractual periode (in EUR)+ number of activation*Average activation price (inEUR)=EUR
16)	**Penalty if failed supply**	Calculation of penalty in case of failed delivery of one activation(zero activation): This is the multiplication of the activation fee, in this case we used 2 multiplicator. - Y times of failed delivery → termination of the contract
17)	**Building offer accepted**	The value can be No/Yes which is coming from LT workflow and reflects to the acceptance of DSO.

3. Suggested method for setting price incentives in simulations

In this chapter, we identify the financial elements that need to be considered when reservation, activation and penalty are priced. To be able to calculate the competitiveness of a flexibility with the avoided investment which should be needed to prevent the line from overloading next year (or exceeding the operational limit), we will use the approach of future value (FV) or in more general, the basics of net present value (NPV) calculation.

3.1 Net Present Value (NPV) and inflation

Net present value (NPV) is the difference between the present value of cash inflows and the present value of cash outflows over a period. It determines future net cash flows of an investment, discounting those cash flows using a discount rate reflecting the risk level of the project and then subtracting the net initial outlay from the present value of the net cash flows. It helps in identifying whether a project adds value or not.

Inflation is a phenomenon that results in a decrease in purchasing power and an increase in revenue and costs. It affects estimates of future cash flows. In order to make better decision, accurate capital budgeting calculations are important, which are possible only when all the financial variables are taken care of.

There are two ways of how inflation can be accounted for while calculating net present value. The final net present value it is same under both methods.

1. Nominal method: converting real cash flows to nominal cash flows and discounting them using nominal discount rate.
2. Real method: estimating real cash flows and discounting them using real discount rate.

Under the nominal method, net cash flows in time t are calculated by the following formula:

$$\text{Nominal Cash Flows at Time t} = \text{Real Cash Flows at Time t} \times (1 + \text{Inflation Rate})^t$$

Under the real method, real cash flows and real discount rate are used. Relationship between nominal discount rate, real discount rate and inflation is given below:

$$\text{Nominal Discount Rate} = (1 + \text{Real Discount Rate}) \times (1 + \text{Inflation Rate}) - 1$$
$$\approx \text{Real Discount Rate} + \text{Inflation Rate}$$

Example 1: Inflation adjustment using nominal cash flows

A company is considering a project that is expected to generate $10 million at the end of each year for 5 years. The initial outlay required is $25 million. A nominal discount rate of 9.2% is appropriate for the risk level. Inflation is 5%.

Nominal cash flows are calculated for each year as follows:

- Year 1 = \$10 million $\times (1 + 5\%)^1$ = \$10.5 million
- Year 2 = \$10 million $\times (1 + 5\%)^2$ = \$11.0 million
- Year 3 = \$10 million $\times (1 + 5\%)^3$ = \$11.58 million

- Year 4 = \$10 million × $(1 + 5\%)^4$ = \$12.16 million
- Year 5 = \$10 million × $(1 + 5\%)^5$ = \$12.76 million

These nominal cash flows are to be discounted using nominal discount rate, which is 9.2%

Table 5 – Inflation adjustment using nominal cash flows

Year	1	2	3	4	5	Total
Nominal cash flows	10.50	11.03	11.58	12.16	12.76	
PV discount rate at 9.2% nominal	0.916	0.839	0.768	0.703	0.644	
PV of cash flows	9.62	9.25	8.89	8.55	8.22	44.52

Net present value = \$44.52 million – \$25 million = \$19.52 million

Example 2: Inflation adjustment using real cash flows and real discount rate

The relationship between nominal discount rate, real discount rate and inflation can be rearranged as follows:

$$\text{Real discount rate} = (1 + \text{nominal discount rate}) \div (1 + \text{inflation rate}) - 1$$
$$\approx \text{nominal discount rate} - \text{inflation rate} = (1 + 9.2\%) \div (1 + 5\%) - 1 = 4\%$$

Table 6 – Inflation adjustment using real cash flows and real discount rate

Year	1	2	3	4	5	Total
Real cash flows	10.00	10.00	10.00	10.00	10.00	
PV discount rate at 4% real	0.962	0.925	0.889	0.855	0.822	
PV of cash flows	9.62	9.25	8.89	8.55	8.22	44.52

Net present value = \$44.52 million – \$25 million = \$19.52 million

3.2 Calculation of the total price

In our case we calculate a real investment deferral value, i.e. a monetary benefit if we defer the investment (it is just like putting the money in the bank).

The maximum price on flexibility products for the DSOs will be set from the DSOs' alternative costs in reinforcement. This will form a sort of price-cap on flexibility products for the DSO. The final price will depend on what price the Aggregator offers its flexibility products at. If it is sufficiently low, the DSOs are likely to use the offered flexibility product.

If the DSO's only alternative to buying this flexibility product is to upgrade its grid components (cables, transformers, etc.), the price setting could be done based on the 1st year value of these upgrades.

For example, if the upgrade of a 10 kV feeder costs 65.000 EUR/km, the life expectancy of this upgrade is 40 years, the inflation is 2,5 % and an interest rate (in our case the recognized WACC by the regulator) of 4,69% is considered, the value of the grid upgrade deferral will be the following, of which some will be spent on the necessary flexibility product un-locking the possibility of the deferral.

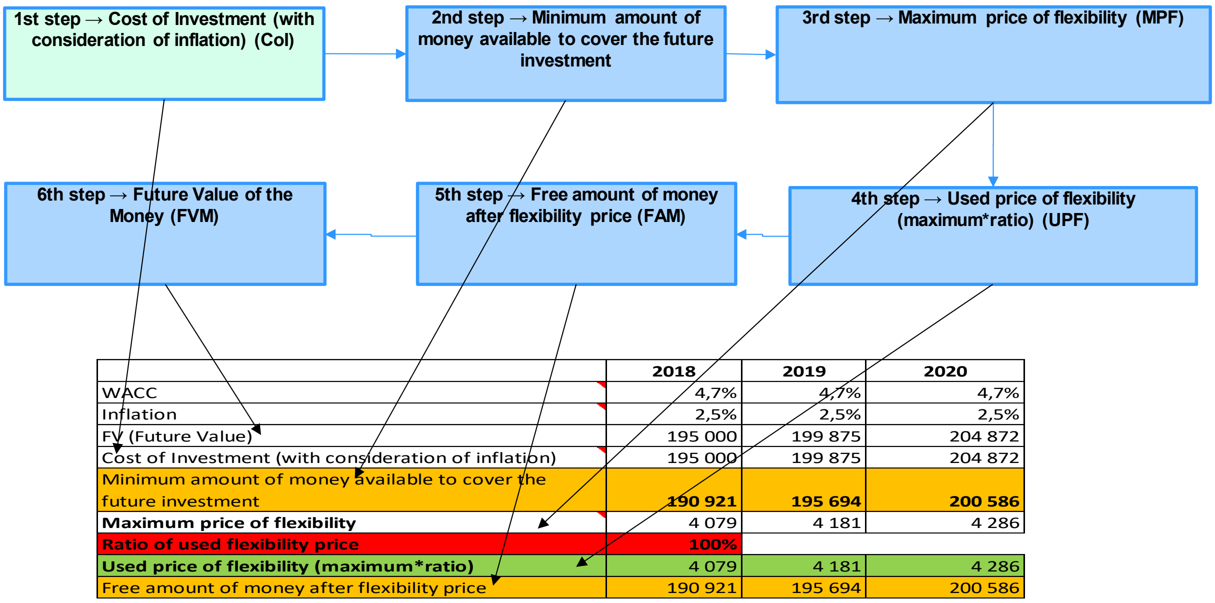
Table 7 – Considered variables

WACC	4.69%
Inflation	2.5%
Useful lifetime	40 years
Cost of 1 km 10 kV cable upgrade	65,000 EUR
Cable length	3 km
Real interest rate	$\frac{1+\text{WACC}}{1+\text{Inflation}}-1=3.65\%$

Table 8 – Calculation of maximum price

	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
WACC	4,7%	4,7%	4,7%	4,7%	4,7%	4,7%	4,7%	4,7%	4,7%	4,7%	4,7%
Inflation	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%
FV (Future Value)	195 000	199 875	204 872	209 994	215 244	220 625	226 140	231 794	237 589	243 528	249 616
Cost of Investment (with consideration of inflation)	195 000	199 875	204 872	209 994	215 244	220 625	226 140	231 794	237 589	243 528	249 616
Minimum amount of money available to cover the future investment	**190 921**	**195 694**	**200 586**	**205 601**	**210 741**	**216 009**	**221 410**	**226 945**	**232 618**	**238 434**	**244 395**
Maximum price of flexibility	4 079	4 181	4 286	4 393	4 503	4 615	4 731	4 849	4 970	5 094	5 222
Ratio of used flexibility price	**100%**										
Used price of flexibility (maximum*ratio)	4 079	4 181	4 286	4 393	4 503	4 615	4 731	4 849	4 970	5 094	5 222
Free amount of money after flexibility price	190 921	195 694	200 586	205 601	210 741	216 009	221 410	226 945	232 618	238 434	244 395

The logic of the calculation is depicted in Figure 3:



	2018	2019	2020
WACC	4,7%	4,7%	4,7%
Inflation	2,5%	2,5%	2,5%
FV (Future Value)	195 000	199 875	204 872
Cost of Investment (with consideration of inflation)	195 000	199 875	204 872
Minimum amount of money available to cover the future investment	**190 921**	**195 694**	**200 586**
Maximum price of flexibility	4 079	4 181	4 286
Ratio of used flexibility price	**100%**		
Used price of flexibility (maximum*ratio)	4 079	4 181	4 286
Free amount of money after flexibility price	190 921	195 694	200 586

Figure 3 – Logic of the price calculation

1st step → Cost of Investment (with consideration of inflation) (CoI):

This value represents the needed amount of money for the investment that we would like to avoid. It is important to mention that the valorization is necessary for each year, since in the next year we have to spend more money on the investment because of the inflation.

2nd step → Minimum amount of money available to cover the future investment (MAM):

This value represents the amount of money which should be put into the bank to cover the next year investment cost. It results from a reverse calculation from the „Cost of Investment” (which is valorised in each year with the inflation). In the reverse calculation we use the WACC as a „bank interest rate” because of the energy industry. Nevertheless, WACC can differ industry by industry and country by country. MAM= next year CoI/1+WACC.

3rd step → Maximum price of flexibility (MPF):

It is calculated form the Future Value of the Money (FVM) and Minimum amount of money available to cover the future investment (MAM). MPF=FVM - MAM. The DSO can spend this amount of money for the flexibility. Only the first year should be considered, the subsequent years calculation can inform us only about what happens if we planned long-term and DSO would require the flexibility only e.g. in the 3rd year. Based on the time series calculation the DSO can consider what should be put into the medium-term plan.

4th step → Used price of flexibility (maximum*ratio) (UPF):

This flexibility price is derived from the Maximum price of Flexibility (MPF), if the DSO does not intend to use the whole amount of the MPF then it can use a ratio (%) by which this MPF will be multiplied, and only a given portion of the Maximum price of Flexibility will be used. If we set the Ratio of Used Flexibility price (RUF) e.g. 80%, then DSO will use only the 80% of the MPF. The remaining part of money will increase the Free amount of Money after flexibility price. In this way DSO will have more money which will increase the Future Value of the Money in the next year.

5th step → Free amount of money after flexibility price (FAM):

This amount of money in the first year is the difference of the Future value of the money (which equals Cost of investment in the first year) and Used price of flexibility. This money theoretically can be put into the Bank and is the basis of the next year Future Value of the Money.

6th step → Future Value of the Money (FVM):

This amount of money in the first year will be the Cost of Investment. In the subsequent years it will be calculated from the previous year Free amount of money after flexibility price (FAM)*(1+WACC), since this amount of money will be in DSO’s hand and it can be put in Bank theoretically.

3.3 Reservation, activation and penalty fee

Let us assume that maximum total price of the flexibility service is the calculated for first year value at 4.079 EUR. From the DSO’s point of view, it would make sense to consider the price calculation bases as kW, but from the Building optimisation point of view the most comfortable unit is the kWh (the optimisation uses this unit), therefore the DSO should set the flexibility price according to this request in case of activation, but in case of reservation the considered dimension is kW. In order to be able to give the most accurate calculation for the long term contract of the DSO, it should calculate the possible use cases of flexibility services based on load curve examination (calculation which is already described in the 5.3.1.1 Annual specification document). Here we give an example based on assumed DSO calculation which – in this case - consist of 36 scenarios. For each month the DSO will calculate typical daily load curves for weekdays, Saturday and Sunday. For the sake of calculation simplicity, we consider a typical day and it will be multiplied by 365 days only for sake of example, nevertheless it is well known that workdays are less than 365 in a year.

Table 9 – Flexibility requests

02.jan	workday	1	down, 150	07:15-09:45
02.jan	workday	1	down, 80	13:30-17:15

Table 9 contains one day's flexibility request. From network point of view, it is necessary to reduce the load by 150 kW for 2,5 hours in the morning, and by 80 kW for 3,75 hours in the afternoon. That means two activations per day. It has to be inverted into kWh.

150 kW * 2.5 hours + 80 kW * 3.75 hours = 375 kWh + 300 kWh = 675 kWh/day. The whole amount of flexibility service is 675 kWh * 365 = 246 375 kWh (246,375MWh) for the network for the year.

Of course, we have to consider that the flexibility service is split into two parts, Reservation and Activation part. The Building should know the constant part of the income which is derived from the Reservation part, and of course it will calculate the not constant part of the income, but it should be aware that it has some uncertainties (due to the fact that DSO will not surely activate the needed flexibility on the given day because the short-term calculation/measurement will calculate/measure more precise flexibility needs from network point of view).

We assume a 50% ratio for the Reservation in terms whole available amount of free money, i.e. 50% of the 4.079 EUR = 2.039,5 EUR.

The Reservation unit price is derived from the needed flexibility in kWh and converted into EUR/kWh/15 min, e.g. 2.039,5 EUR / 246.375 kWh/4 = 0,002069 EUR / kWh/15min.

Activation unit price: 2.039,5 EUR / whole amount of activation in kWh in the year, e.g. 2.039,5 EUR / 246.375 kWh = 0,008278 EUR / kWh

Take an example with a Building which can provide only a fragment of the needed flexibility for the whole network. In order to be able to calculate the flexibility income of one building, it is necessary to have a Building schedule prior to the Long term contract. The schedule should be provided by the Building in the same structure as the flexibility table of DSO for ease of comparison.

Table 7 – Building flexibility table

Day	Type of day	Month	DSO flexibility requirement [kW]	Time	Building available flexibility[kW]	Time	kWh
02.jan	workday	1	down, 150	07:15-09:45	down 15	07:15-09:45	37,5
02.jan	workday	1	down, 80	13:30-17:15	down 8	13:30-17:15	30

The DSO in this way could consider the available flexibility from the Building side. In this example, the Building can provide only one tenth of the needed flexibility (in kW), but the DSO should manage to gather more flexibility from other Buildings to satisfy the network constraints.

Based on the above Building data, the flexibility income of the Building will be the following:

Reservation: 0,002069EUR/kWh/15min*67,5kWh*365*4 = 203,95EUR

Activation: at the beginning of the service period we assume that all planned activation will be called, i.e. two times per day with the given time period, 365*(37.5 kWh+30 kWh)*0,0083 EUR= 204,49 EUR.

But the Building should be aware that there are uncertainties in terms of calling planned Activation, e.g DSO will call only 2 hours from the above mentioned 3,75 hours, and it can differ day by day (of course it is worth mentioning that in practice the calculated load curve for an MV line is well compared to the real load curve, either measured or calculated by DA AC OPF).

What is really necessary to stipulate in the long term contract between DSO and Building is that Building should provide commitment not only in kWh and time interval, but in kW as well. Taking the above example, the Building could deliver the needed service in kWh during the 2,5 hours period with 7,5 kW in the first 1,25 hours and with 22,5 kW in the second 1,25 hours. This kind of service is not appropriate if it was not stipulated in the contract because it means uncertainties for the network management. So, if the building will give information only in kWh for the given period, the DSO will assume that this service will be delivered with a constant size of service in power, e.g. 37,5 kWh for a 2,5 hours period will be considered with 15 kW, and the contract will contain it (e.g. as a flexibility table with the service provided by the Building in an appendix).

The below example explains these restrictions in a visual way:

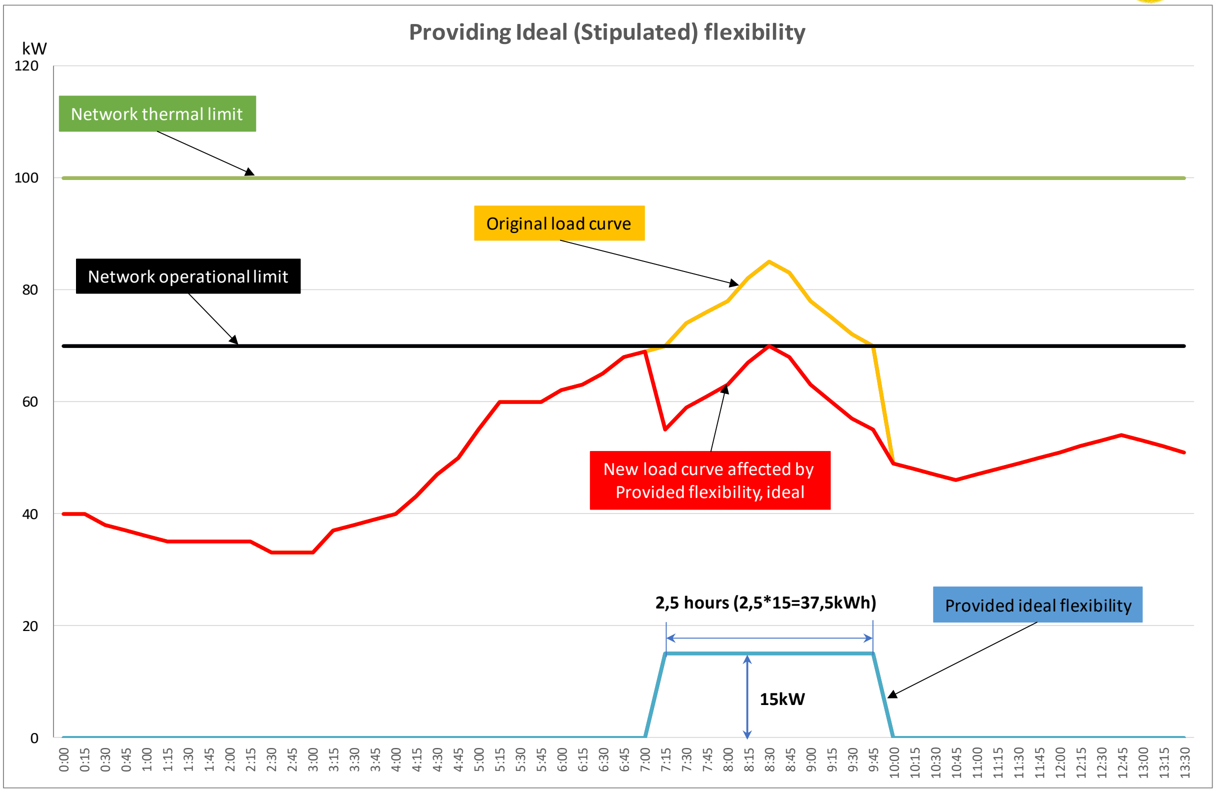


Figure 4 – Appropriate service

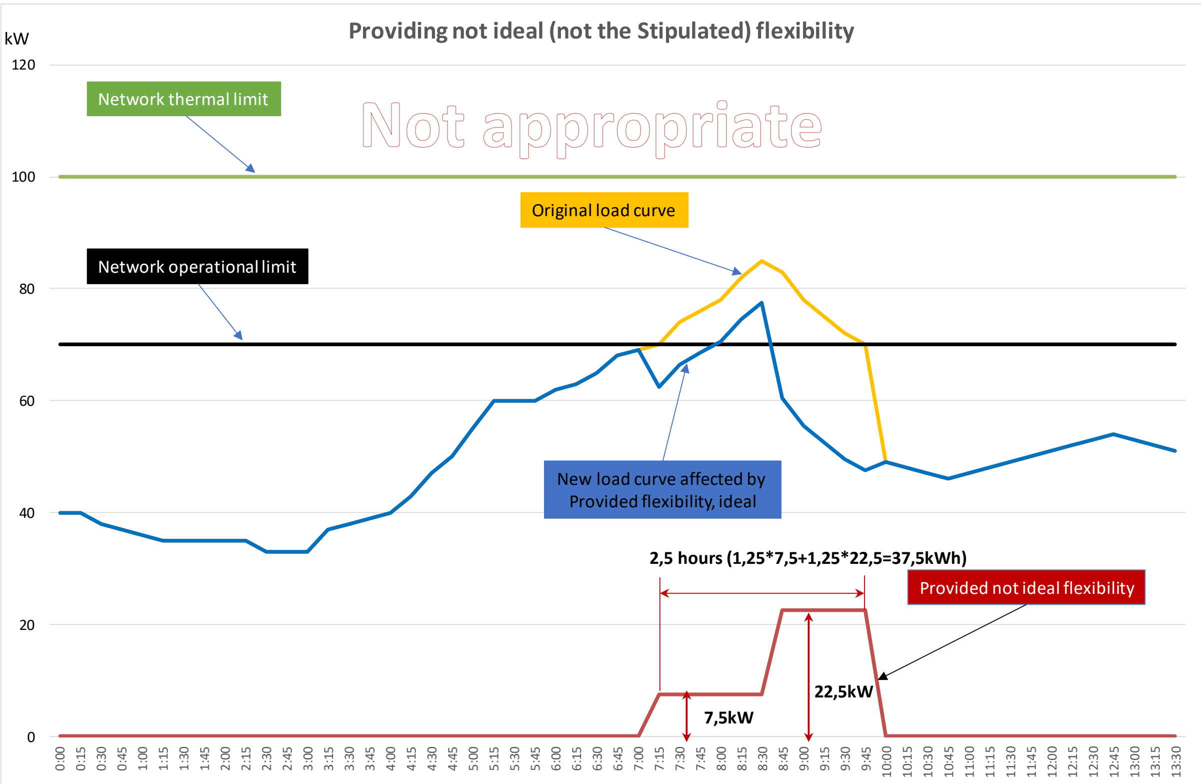


Figure 5 – Inappropriate service

The penalty in case of flexibility service failure must be considered. The proposed penalty in case of not delivering the needed flexibility service equals the Multiplicator*Activation unite price (the Multiplicator can be chosen by DSO but of course if Building will not accept either it should be iterated or rejected the contract.

4. Multiannual Software module

Based on the above description and referring to 5.3.1.1 Annual software module specification, Multiannual software module description and the module itself can be found here.

The module consists of three main parts:

1. Price and penalty
2. Flexibility unit prices, penalty
3. Building Flexibility table
4. Output for long term contract

Each part of module consists of three parts: input for the specific calculation function, the calculation function itself and the output of the specific calculation function.

- WACC
- Inflation
- Ratio of used flexibility price (see explanation in chapter 5.2)
- The cost of investment (this is the cost of investment if we could not use flexibility)

Here the subpart of the module calculates the available amount of money for the flexibility service in terms of the specific network part (e.g if an MV cable line is concerned then its upgrade cost should be given as input). The logic of calculation was described in chapter 5.2.

Here the subpart of the module gives the available money of the DSO for the flexibility service (both the maximum available and the used price, see explanation in chapter 5.2). It will be the basis of the further calculation of unit price of reservation and activation.

Price and penalty

Price and penalty

- Reservation ratio
- Activation ratio
- DSO Flexibility table from Annual module

The ratio between reservation and activation depends on the DSO policy, here we set up a fifty-fifty ratio but it can be e.g 20-80%.

The DSO flexibility table gives information about the scenarios: of which month, of which specific day, how many timesflexibility is needed for how long. Beside of this there is calendar in the Annual module which helps to determine the type of days in accordance with the special dates of the year. It is crucial that the Building knows on which date of the year what kind of flexibility is needed. The building will answer for this request.

- Used price of flexibility (maximum*ratio)
- Unit price of flexibility for reservation
- Unit price of flexibility for activation
- Reservation part of Flexibility unit price
- Activation part of Flexibility unit price

This is the central point of the calculation, the already described Reservation and Activation unit price here will be calculated

Flexibility unit prices, penalty

Here the subpart of the module gives Reservation part of Flexibility unit priceand Activation part of Flexibility unit price as an output which will be used by BEMS (Building side).

Building Flexibility table

DSO Flexibility table from Annual module serves as an input. The Building gives an answer to DSO for this flexibility need by this Building Flexibility Table which is one of the basis of the Long term cotract

Long term contract preparation

Module subpart: **Input for long term contract_1**

- Contract valid from
- Contract valid until
- The proposed penalty in case of not delivering the needed flexibility service (per activation) equals the X% of the whole activation cost

Input for long term contract_2

- Building Flexibility table

The Building flexibility table gives information about the Building answers to DSO Flexibility table: of which month, of which specific day, how much flexibility is provided for how long. Beside of this there is calendar in Annual module which helps determine the type of day accordance with the specific dates of the year. It is crucial that the Building knows on which date of the year what kind of flexibility is provided. This is the basis of the long term contract, this is the „promise“ of the Building.

Module subpart: **Long term contract preparation**

- The provided amount of flexibility by Bulding in kWh
- Reservation part of Flexibility unit price
- Activation part of Flexibility unit price
- Reservation fee for the contractual period
- The fee of Sum of Activations
- The maximum size of Flexibility in kW
- The maximum duration of Flexibility
- Number of activations during period

The provided amount of flexibility in kWh:The sum of activation service in kWh from calculation table based on Building Flexibility tale (the module summarizes the activations in kWh based on typical days and the number of the typical days in the year).

Reservation fee for the contractual period: based on EUR/kWh/15 min unit price, the provided flexibility in kWh and the overall number of 15-minute time intervals of flexibility in the contracting period

The fee of Sum of Activations: The product of the Activation part of Flexibility unit price and The provided amount of Flexibility by Building in kWh

The maximum size of Flexibility in kW: Based on the provided flexibility table of the Building here we indicate the maximum size of provided flexibility.

The maximum duration of flexibility: Based on the provided flexibility table of the Building here we indicate the maximum duration of provided flexibility, the calculator seeks the activation in the year with the maximum duration.

Number of activations during period: The number of all activation counted based on the Building flexibility table.

Output for long term contract

Here we have to mention that part of the Output for long term contract needs some editing activities, these activities were remarked with "Manual entry":

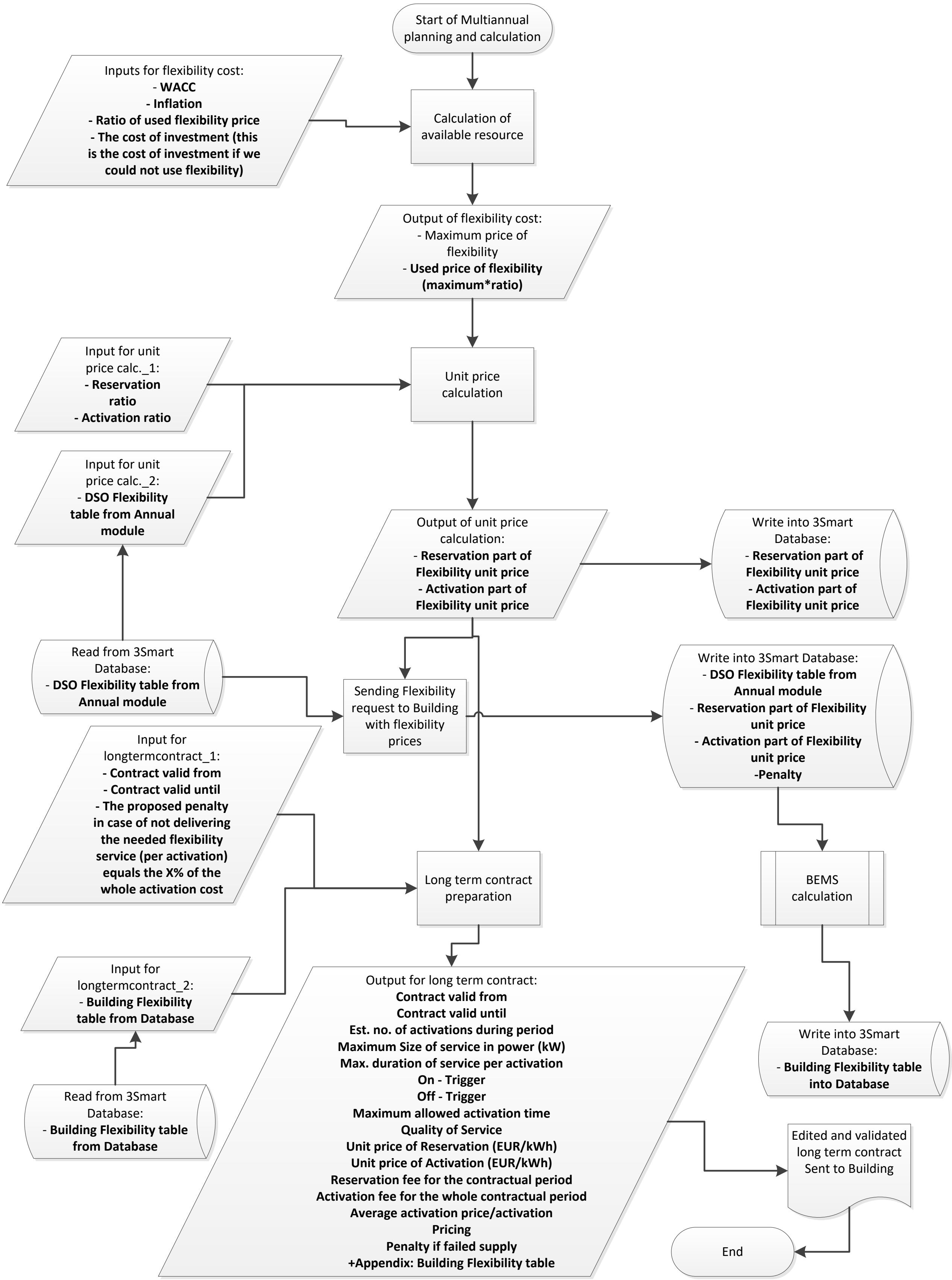
Output for long term contract

Service name		Flexibility service			
1)	Contract valid from	01.01.2019.	dd.mm.yyyy.		
2)	Contract valid until	31.12.2019.	dd.mm.yyyy.		
3)	Est. no. of activations during period	227	Practically here will the number of activation be calculated within the contractual periode, i.e. the number of activations from Provided flexibility table by the Building		
4)	Maximum Size of service in power (kW)	97,72913636	The algorithm seeks the maxumum power within the Flexibility table provided by the Building (i.e. the maxumim in the column "Provided flexibility by Building [kW]")		
5)	Max. duration of service per activation (h)	5,00	The algorithm seeks the maxumum duration within the Flexibility table provided by the Building (i.e. the maxumim time interval)		
6)	On - Trigger	Signal from the DSO or according to DA AC OPF calculator	Manual entry		
7)	Off - Trigger	Maximum: see "Max.duration of service per activation" from "on"-signal, or by earlier signal from the DSO	From "Max.duration of service per activation(h)" and partly Manual entry		
8)	Maximum allowed activation time	15 min (but it depends of the capability of the Customer process technology)	Manual entry		
9)	Quality of Service	Deviation in max. duration: +/-	min		Manual entry
		Deviation from, On - Trigger: +/-	min		Manual entry
		Deviation in size of service: +/-	% of kW		Manual entry
		Acceptable no. of unsuccessful activations (above it terminate contract):	pcs		Manual entry
10)	Unit price of Reservation (EUR/kW/(15min))	0,016245729			
11)	Unit price of Activation (EUR/kWh)	0,064982918			
12)	Reservation fee for the contractual period	1835,633776			
13)	Activation fee for the whole contractual period	1835,633776			
14)	Average activation price/activation	8,086492405	Since during the whole contractual periode the duration and size of activation varies day by day, therfore the whole amount of Activation fee ("Activation fee for the whole contractual period")will be diveded by the number of activation from "Est. no. of activations during period "		
15)	Pricing	3671,267552	Reservation fee for the whole contractual periode (in EUR)+ number of activation*Average activation price (inEUR)=EUR		
16)	Penalty if failed supply	0,129965836	Calculation of penalty in case of failed delivery of one activation(zero activation): The fee of Sum of Activations* Percentage of the Activation fee for the whole contractual period. In case of partial service provision the slope of penalty curve can be found in "Input for longtermcontract_1" sheet of the Multiannual module		
			Manual entry	- Y times of failed delivery → ter	
17)	Building offer accepted	no	Manual entry	yes/no	

The entire LT module can be found below as an embedded file, the Multiannual subpart can be found in the excel:

3Smart_LT module_v4.xlsm

Flow chart for Multi annual software module



From the above flow chart, it can be deduced where is necessary a Database interaction.

Bibliography

[1] 3Smart deliverable D5.1.1. Grid-side EMS concept and information exchange interfaces definition. December 2017.

[2] 3Smart deliverables D5.4.1. Grid models for grid simulation in professional simulation tools. December 2017.

[3] iPower: Development of a DSO-market on flexibility services. http://orbit.dtu.dk/en/publications/development-of-a-dsomarket-on-flexibility-services(2c6422eb-5190-4421-a79c-2692db0567ca).html March 2013.

[4] AccountingExplained.com. NPV and Inflation. https://accountingexplained.com/managerial/capital-budgeting/npv-and-inflation March 2018.

[5] "Energie Koplopers: Flexibility from residential power consumption: A new market opportunities"

Output Quality Report

Output title: Modular cross-spanning energy management tool	
Type of output:	☐ Documented learning interaction ☐ Strategy/ Action Plan ☑ Tool ☐ Pilot action
Contribution to PO indicator:	**P24 No of tools to improve energy security and energy efficiency developed and/or implemented**

Summary of the output (max. 1500 characters)

Please describe the output in terms of content, objective, scope and main characteristics.

The output represents a tool for integrated energy management of buildings and energy distribution grids. It is constituted of software modules, some meant for buildings and some for the grids. Interaction of particular software modules on the building and on the grid side is established, which enables integrated energy management both in near-real time and on longer time scales for the purpose of demand response service contracting, planning and benefits assessment.

On the building side, the modules are divided into three vertical levels covering three key parts of the buildings – level of comfort control in individual building zones, level of central heating, ventilation and air conditioning systems for preparation of heating/cooling media for zones, and level of major energy flows in the building, shortly named building microgrid level.

On the grid side, the modules encompass long-term planning of grid operation and the grid operation itself on time windows of day-ahead and intra-day markets.

Main features of the tool are:

- it is meant as an add-on to the existing automation systems in buildings and grids;
- it operates building and grid elements to minimize costs, including exploitation of demand response opportunities;
- it respects comfort and equipment constraints in buildings and grids;
- it is operable in different configurations which can be selected based on projected costs of needed interventions and expected benefits in operation.

Added value (max. 1500 characters)

For strategies and tools:

Please provide a comprehensive explanation regarding the added value of the output as compared to already existing strategies/ tools of similar type.

What is so far usually meant under building energy management tools are SCADA systems or integration tools that gather information from various systems for display or analysis.

However, the tools that process data on the entire building scale and act back towards the building systems with optimized feedback actions are very needed and practically up to the point of 3Smart tool inception non-existent. Such tools should exhibit optimal operation of buildings in a full variety of their possible configurations.

They are meant not just to save energy, but to make the building consumption signature on various distribution grids controllable and flexible under constraints of ensuring comfort, respecting equipment possibilities and under goal of inducing minimized building operation costs. Important is also the feature of optimal daily operation planning which can reveal how much power for flexibility provision is available and what would be the gains achievable by tool on-line exploitation within the building, which then guides the investments for the tool introduction.

From the grid side, the 3Smart tool is also unique as it determines the technical and financial conditions for flexibility services engagement. It interacts via simple energy-power-price data with the 3Smart tool on the side of buildings. Based on that interaction the tool contracts flexibility services from the buildings and operates the grid such that the contracted flexibility services are engaged optimally from the standpoint of grid distribution losses.

Applicability and replicability (max. 1500 characters)

Please provide a concrete description of how the project output is to be applied in real life and could be replicated in other geographical and sectorial areas or different environments.

The modularity of the tool and its universal building and grid operation/planning problem posed as a tractable mathematical optimization problem makes it applicable for various configurations of buildings and distribution grids. The applicability is practically without any geographical constraints, subject to the use of needed building automation equipment to be able to gather data and act back with controls through the 3Smart database. Other, new elements may also be introduced. E.g., currently the 3Smart tool does not have software modules for control of air handling units or for air-based heating/cooling elements in zones, but they can be well added by developing and introducing new software modules to the existing base.

The tool is applied on the grid and building side first through its operational planning modules which are used to assess gains in operation and flexibility provision possibilities (some a priori, hopefully with time catalogue-knowledge is needed for that). After the selected configuration is economically valorized from the investment point of view, the tool can be installed together with standardized 3Smart database for data transfer from/to the building or grid, designed to speed up integration of the software modules. The installation requires an expert work for adaptation of generic software modules to a particular building or grid configuration. This is a source of new high-added value jobs for SMEs in the coming era of energy transition.

Suggestions for improvement, if applicable (max. 1500 characters)

Please provide information on possible improvements that could be brought to the current output

considering the general context in which it is delivered. Possible improvements may be the following: • inclusion of additional software modules to cover air-based systems in buildings, • more advanced demand response planning on the building and the grid side, with wide activation time windows, maximum service activation time and minimum recuperation times between service calls.	
Output Quality Level	☐ Low ☐ Average ☐ Good ☑ Excellent

Names of the Quality Managers

Prof. Mato Baotić

Assoc. Prof. Hrvoje Pandžić

Signatures of the Quality Managers