HANSA Klimasysteme GmbH
Stockweg 19
26683 Saterland / Strücklingen

Hochschule Emden / Leer
Constantiaplatz 4
26723 Emden

Übergeordnete selbstlernende energieeffiziente Regelung mehrerer zusammenwirkender komplexer Raumlufttechnischer Geräte „Smart-RLT Net“

Abschlussbericht
gefördert durch die Deutsche Bundesstiftung Umwelt
Az: 35561/01-24/0

Prof. Dr. Sven Steinigeweg
Dr. Ing. Matthias Lamping

Strücklingen, 29.08.2023

HANSA Klimasysteme GmbH
Stockweg 19
26683 Saterland / Strücklingen

Hochschule Emden / Leer
Constantiaplatz 4
26723 Emden

Übergeordnete selbstlernende energieeffiziente Regelung mehrerer zusammenwirkender komplexer Raumlufttechnischer Geräte „Smart-RLT Net“

Abschlussbericht
gefördert durch die Deutsche Bundesstiftung Umwelt
Az: 35561/01-24/0

Prof. Dr. Sven Steinigeweg
Dr. Ing. Matthias Lamping

Strücklingen, 29.08.2023

Contents

List of Formulas

List of Figures

List of Tables

Terms, Abbreviation and Definitions

A

AHU .... Air Handling Unit

C

cloud
  Computer or data Cloud (IT-Infrastructure) 37

G

GmbH
  Gesellschaft mit beschränkter Haftung 1, 2

GRU
  Gated Recurrent Unit 20

L

Levenberg-Marquardt
  A training Method used to solve non-linear least squares problems 24

LSTM
  Long Shor Term Memory 20

LTI
  Linear Time Invariant 22

M

MPC
  Model Based Predictive Control 42

N

NARX.. *Non Linear Auto Regressive Exogenious*

P

PID
  Proportional-Integral-Derivative.. R

PSO
  Particle Swarm Optimization 55

PWW
  Pumpe Warm Wasser .... P

Q

$\dot{Q}$ .... *Heat Flow*

R

regression
  A statistical technique that relates a dependent variable to one or more independent (explanatory) variables 25

Regularization
  A technique that can prevent overfitting in neural networks 24

RMSE
  Root Mean Squared Error 66

S

Saia
  Saia-Burgess Controls AG .... W
SPS
  Sperichерprogramierbare Steuerung 13
Step Response
  The time behaviour of the outputs of a general system when its inputs change from zero to one in a very short time 19

T

training algorithms
  An algorithm that builds a model by examining many examples and attempting to find a model that minimizes loss 24

V

VPNs
  Virtual Private Network 64

Summary

In the previous project funded by the DBU with the AZ 33401/01 "Selbstlernende energieeffiziente Regelung komplexer raumlufttechnischer Gerätet", it was basically proven that the approach of model-based control of a swimming pool air handling unit has a clear potential for reduced energy consumption. A short test phase showed a reduction of about 20%. However, the swimming pool is ventilated by two devices and it became apparent that both devices must be equipped with the appropriate control approach and, in addition, the default values for the devices must be generated via a superordinate coordination level. This was implemented in this project.

The project was divided into 7 phases, which were partly worked on in parallel by HANSA and the cooperation partner Eden/Leer University of Applied Sciences. After the project conception, the sensor and data concept was updated and the necessary sensor technology was reduced to the required minimum on the basis of the results. The models developed in the previous project were transferred to two new AHU's in the swimming pool and validated. Based on the data collected in the first project, a higher-level neural control model was developed. After the end of the Corona phase, more data could be collected to further train the models. Approaches could be developed to normalize the models created in one building in order to transfer them to other buildings. The control quality is sufficient for this purpose.

When adapting the existing gray and black box models to the second device, it also became apparent that a physically based model offers advantages because sufficient data must be available for each operating state in order to train the neural networks, but in some cases this data was only available in small numbers. Furthermore, a neural learning model requires a larger number of sensors. This is negative for the later marketing, because the acquisition costs increase. In principle, it was possible to adapt the device models very quickly via scaling. In further operation, further learning can then be implemented quickly.

In this project, the basic possibility was created to continuously use the collected data for the optimization of the NN model during the operation of the devices and to continuously train it automatically. The automatic linearization required in the second step for the MPC was investigated, but could not yet be achieved. A manual step is still necessary here.

A higher-level neural building model and a two-level MPC cascade controller for the building and the devices were developed. This makes it possible to coordinate multiple devices acting on one room to be conditioned.

It was possible, after the end of the Corona restriction, to compare the energy consumptions for more than one year. The results show a saving of energy demand of 25% each of electrical energy and thermal energy in the swimming pool Ramsloh. To verify this, three other swimming pools were equipped with this technology. The results are comparable. A detailed description of these pools is omitted in this final report.

The developed controller has the further advantage that the control quality is better and thus also the comfort feeling of the visitors as well as the employees. A corresponding analysis is presented in the report.

All project goals were achieved. The controller was further developed to such an extent that a market launch was possible. The controller has already been sold in several new projects

(Vahrenwalder Bad, Sylter Welle, Schwimmbad Delitzsch, Ernst-Thälmann Platz) and also as a retrofit in various pools where HANSA equipment is installed (e.g. Kusel, Erlangen).

We will now further develop this control approach for other applications and hire several more engineers for this purpose.

We would like to thank the DBU for their support and assistance. Without these projects, we would not have been able to develop this forward-looking technology.

1. Introduction

Neural networks have found growing success in diverse industrial control applications. In comparisons with convenional control systems neural-network controllers have shown better performance when the plant characteristics are poorly known.
In previous project we implented a neural netwok based control system with one single model containing a single air handling unit and the building. The study showed that this change in control system comparing to the classical control system has a great potential in energy saving. In this follow up project we considered the second unit in the swmming hall Ramsloh as well. The second unit is different in capacity and structure.
As every new project has its own specific unit with different design and capacity in chapter three we studied the transferability of the models for different units with different size and design. This saves a lot of time and cost specifically for building models to use an already trained model for other buildings with different specifications, size, usage profile and weather condition and this makes the overal approach economically more feasible.
In chapter four we study the structure of a two layer MPC controller and training a building model for the supervisory level. When more number of AHUs condition one swimming hall at the same time they must be controlled by a higher level controller in a cascade control design. This allows to distrubte the load evenly among the AHUs and ensures a smooth operation and avoids operating AHUs against each other. The higher level controller is a neural network controller based on the building thermal and humidity model that predicts the required load by the units and divides the load between them. For this reason a two layer cascade control achitucture is designed.
In chapter five we describe the platform for online training which receives data online and updates the pretrained neural network with new data. Although this new model is still not ready to be used for control pourpose as it should go under linireazation and more stability analysis and sensitivity analysis to ensure a stable control strategy. We tried other method called state space neural network where we train a neural network hows weight form the state space matrices. This way we can bypass the linearization process although this method has its own limitations which were explained in chapter five as well.

2. Literature research – state of the art

HVAC industry is experiencing a significant transformation due to the integration of digital technologies. Integration of sensors, actuators, and connectivity technologies has enabled HVAC systems to become smart. These advances are in different areas from IoT and connectivity in central management systems to analytics and machine learnings for optimizing energy consumptions as well as cloud computing in real time data processing for predictive maintenance and automation as well as digital twin.
In digital twin applications a cloud based system platform can be developed which is used in energy monitoring, predictive maintenance, fault detection. At university Dresden [1] a digital twin application for heat pump system is used for optimizing the process in operation and fault detection.
Machine learning has other applications in building services such using machine learning methods for planning software [2]. Here the user provides the data of location, room specifications, cooling system and control system and the software designs the system according to standard norms.
Neural networks have also been used for fault detection in ventilation systems. TROX GmbH [3] has developed a method for using neural network based fault detection. The system monitors control systems as well as the ventilation system by monitoring duckt pressure and temperatures. Similarly Fraunhofer ISE has worked in this area [4] focusing more in fault detection of control system to detect inefficient control instances of the ventilation system. This way neural networks are trained for not directly controlling the system but to monitor how the system is controlled and find energy loss potentials in existing control systems.
In HVAC automation also neural networks have found their way to optimize the automation system. In this area we can mention a series of works in overall building heat management using model based predictive controllers. Among them we can mention the work by Viessman group who has developed a solution for interconnected single-room with the help of a model based control design with two degrees of freedom [5]. Here a model for multiple room temperatures is used to optimize a thermostat control system using weather condition and other informations. This is a rather simple application of model based control with few degrees of freedom for a heating system with thermostat control system.
A similar model based control system was developed with Otto Meyer Technik GmbH [6] for optimizing energy consumption of a big multi-zone office building in Hamburg using building grey box model. The MPC controller controls two parameters of hot water supply temperature and circulation pump on/off signal. This system uses weather forecast information and considers a long prediction horizon of 48 hours with control intervals of 30 minutes. With such a long control horizon the model can take into account the thermal mass of the building to store energy in certain times and release it in the rest of the day. A similar work has been done at University of Stuttgart [7] for load management in energy flexible buildings considering variable energy prices throught the day and weather condition. Similar applications can be found in this area such as in [8] for predicting the heat load using neural networks. These kind of control systems are suitable for an overall building energy management with limited control

commands while our case is a multiple input/output MPC model with 7 control signals to control an air handling unit which should react fast with sample time of 1 minute and has a control horizon of 60 minutes. Unike the previouse cases who only control room temperature in our case we have 3 major interlinked control variables of hall temperature, humidity and minimum recirculation rate. A combination of these control systems mentioned above with a large control horizon and control action intervals and our work can be applied to optimize the overall energy consumption at both air handling systms and heating/cooling systems in a swimming pool.

3. Modelling the new unit in Ramsloh

3.1 Unit description

The second swimming hall unit is composed of a double plate heat exchanger for energy recovery, supply and return air ventilators, outside and return air filter and a hot water air heater.

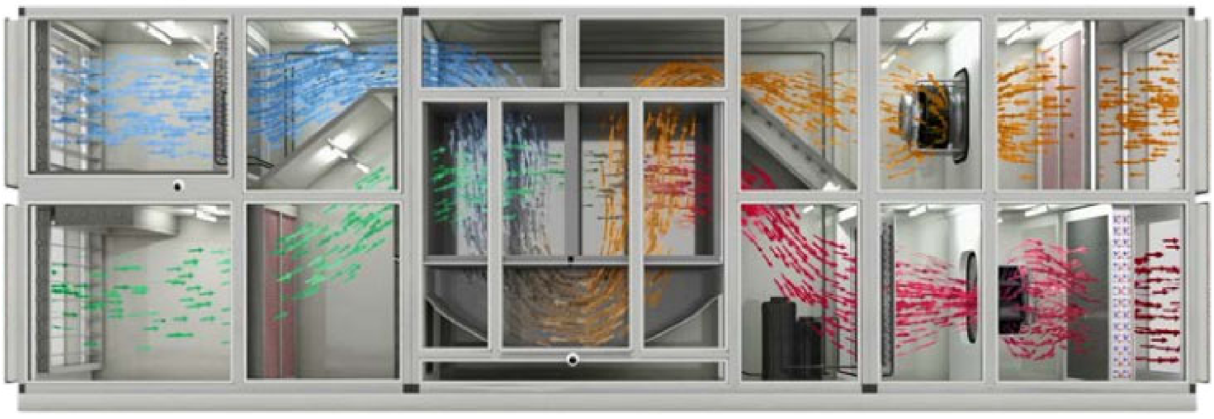


Figure 31 Partial Recirculation in winter mode

Figure 31 Partial Recirculation in winter modeFigure 31 shows an operation mode that typically is in outside temperatures between 12 and 20°C selected. By reducing the outside temperature a portion of the return air will be mixed with the outside temperature agter heat recovery over the recirculation damper.

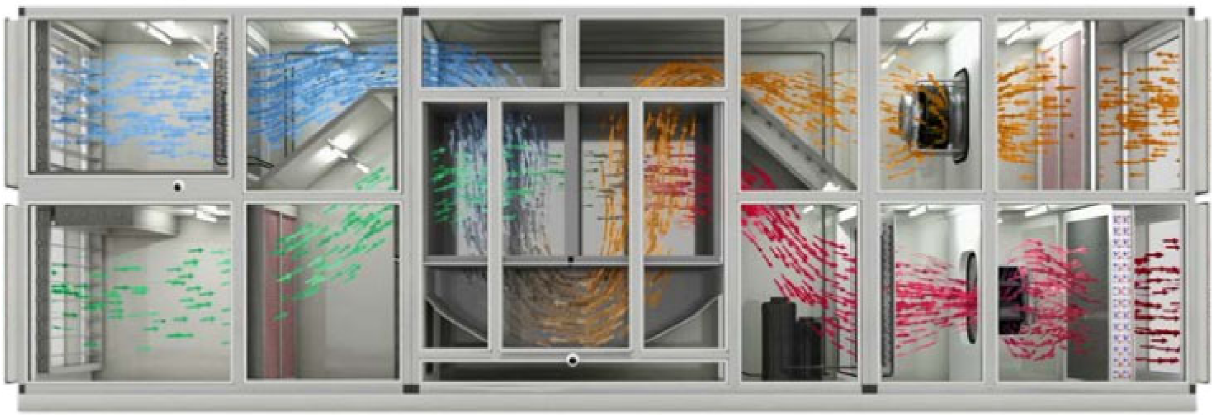


Figure 2 Dehumidification through outside air

When no fresh air and no dehumidification through the outside air damper are required (for example in night operation mode), the return air can be directed to the hall over the recirculation damper.

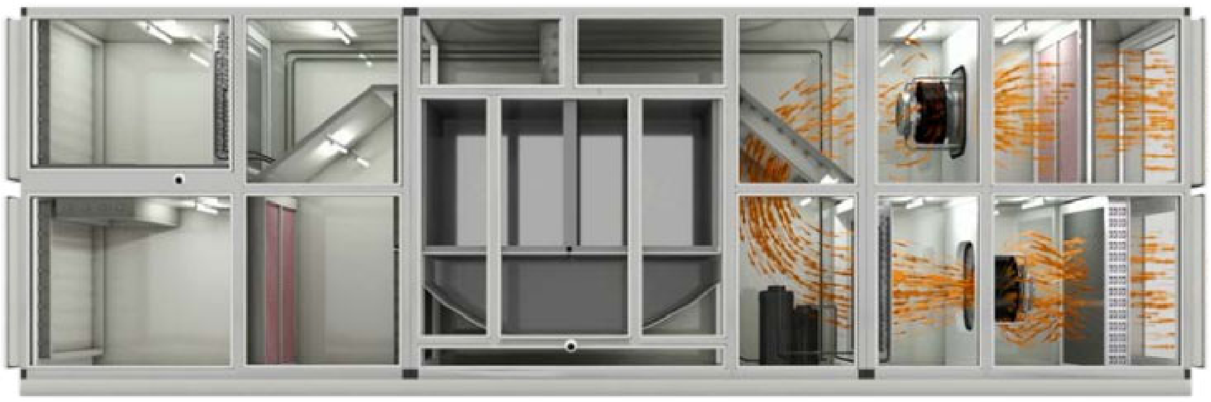


Figure 3 Recirculation without dehumidification

The heat exchanger is equipped with a bypass damper to bypass the heat exchanger in summer operation mode when no heat recovery is required. The overall energy consumption of the unit is composed of electricity consumption of the ventilators and the thermal energy consumption of the air heater.

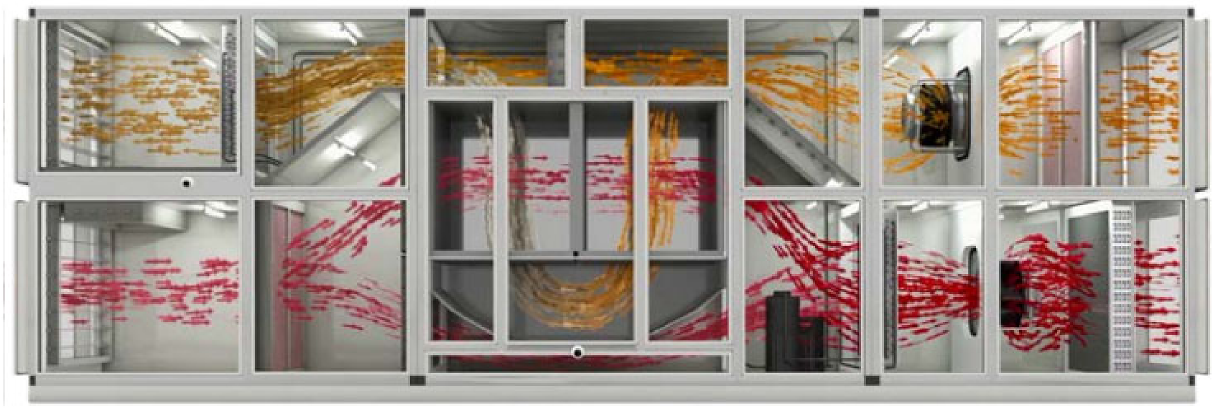


Figure 4 Bypass in summer mode

In summer dehumidification is carried out through the outside air and by reducing the recirculation rate and increasing the outside air portion. In order to avoid the swimming hall to be too warmed up in summer the outside are bypasses the heat exchanger using the bypass damper. The bypass damper can be adjusted such that the desired temperature in the hall is achieved.

The air handling unit is controlled at the moment with a DDC which contains several PI controllers for supply and return volume flow rates and hall temperature and humidity where the return air temperature and humidity is used as feedback. There are also two ristrictions for the minimum and maximum temperatures which should not be exceeded.

In comparison to the unit modelled and investigated in the first project, this unit has parallel flow arrangement (outside air – supply air @ lower side and return air – exhaust air @ upper side) and now cooling circuit as heat pump.

3.2 Developement of data and sensor installations

Figure 35 shows the control diagram of the air handling unit with input and output signals and sensors that we installed. We installed temperature, humidity and pressure sensors at the etrerance and output of each damper and heat exchanger in a way that we can model each component seperately.

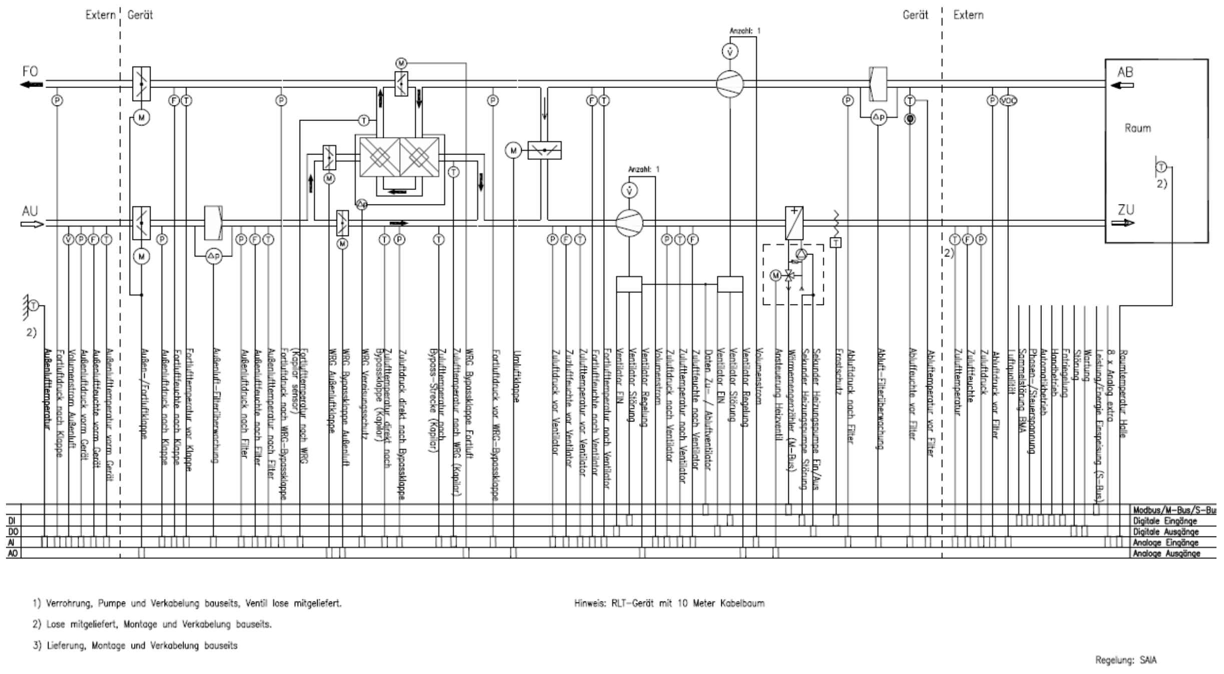

Extern
Gerät
Gerät
Extern
FO
AU
AB
Raum
ZU
Modbus/M-Bus/S-Bus
Digitale Eingänge
Digitale Ausgänge
Analoge Eingänge
Analoge Ausgänge
1) Verrohrung, Pumpe und Verkabelung bauseits, Ventil lose mitgeliefert.
2) Lose mitgeliefert, Montage und Verkabelung bauseits.
3) Lieferung, Montage und Verkabelung bauseits
Hinweis: RLT-Gerät mit 10 Meter Kabelbaum
Regelung: SAIA

Figure 35 Regelschema and sensor installation for data concept

At this stage, the sensor equipment is similar to that of the unit investigated in the first project. During our investigations we found out, that a significant reduction of sensors is possible.
As the sensors required for modelling need to be installed inside the unit to capture pressure, temperature and humidity variations while the air passes through the components. But the sensors required to let the control system run are only those which are needed to provide the required feedbacks for MPC controller such as outside/supply and return temperature and humidity, electricity/thermal energy counters and outside/supply volume flow rates.

3.3 Physics based model for AHU

A series of pressure resistance experiments was designed and performed on the AHU in order to evaluate the pressure resistance coefficients of the dampers and other components inside the AHU. The figure below shows the schematic of the flow resistance model setup for evaluating the resistance coefficients for dampers and other components.

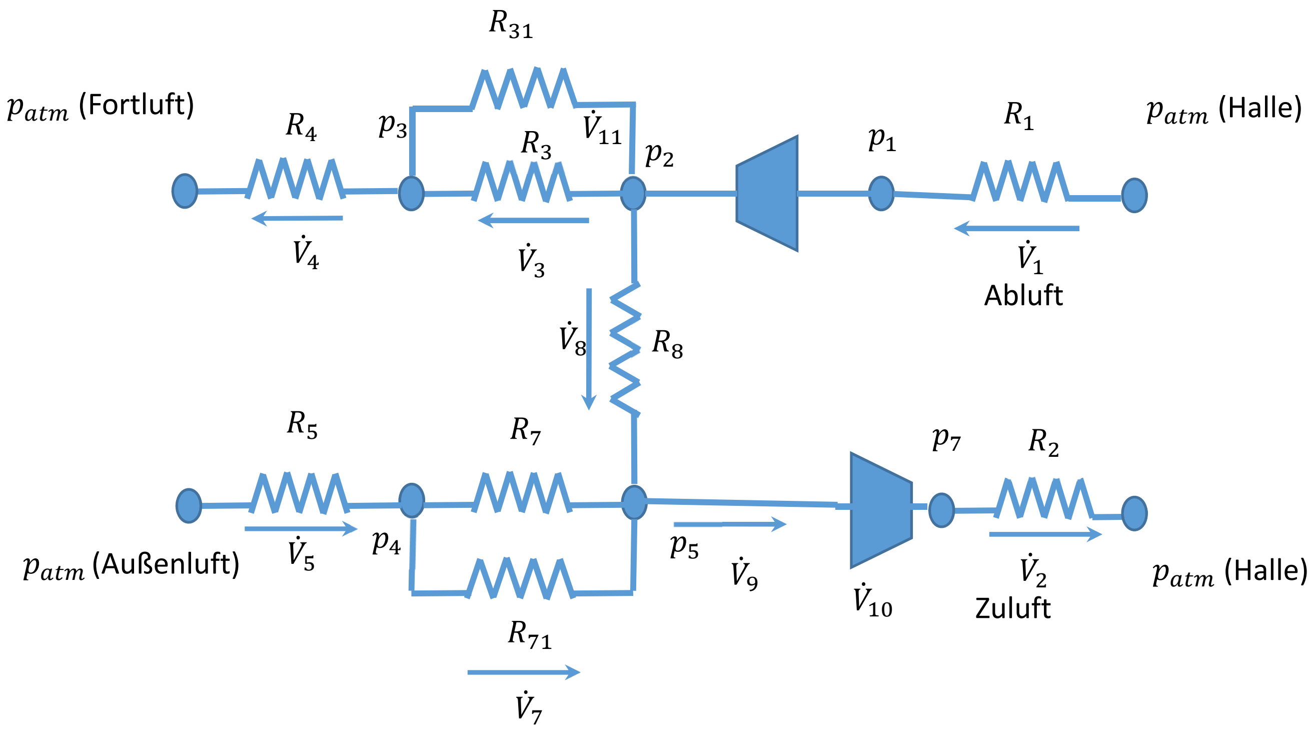


Figure 6 Flow resistance model

Figure above shows the flow resistance model with analogy to electrical circuit. The resistances represent dampers and other component's flow resistance as below:

R1:	Return duct and filter
R2:	Supply duct and heating register
R3:	Plate heat exchanger return side
R31:	Bypass Heat Exchanger return side
R4:	Exhaust duct and damper
R5:	Outside Duct, filter and damper
R7:	Plate heat exchanger and face damper
R71:	Bypass heat Exchanger supply side
R8:	Recirculation damper

Similar to unit number one in the previeous project we derived the characterisics of the dampers and ventilators using experimental data. More details and theory applied here can be found in previous report. The plots below show the characteristics of the fans and exemplarily the recirculation damper.

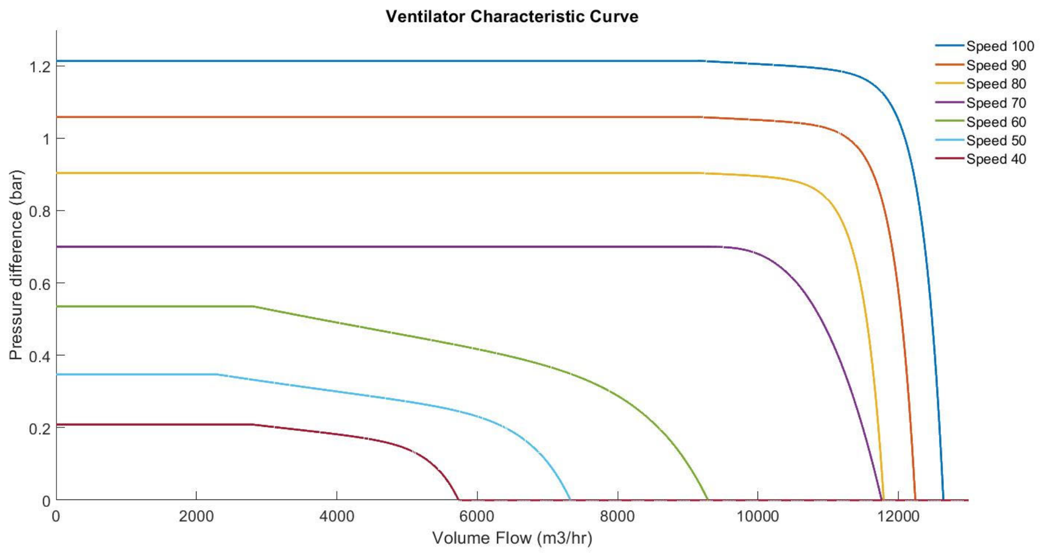


Figure 37 Characteristic curve of the Ventilator

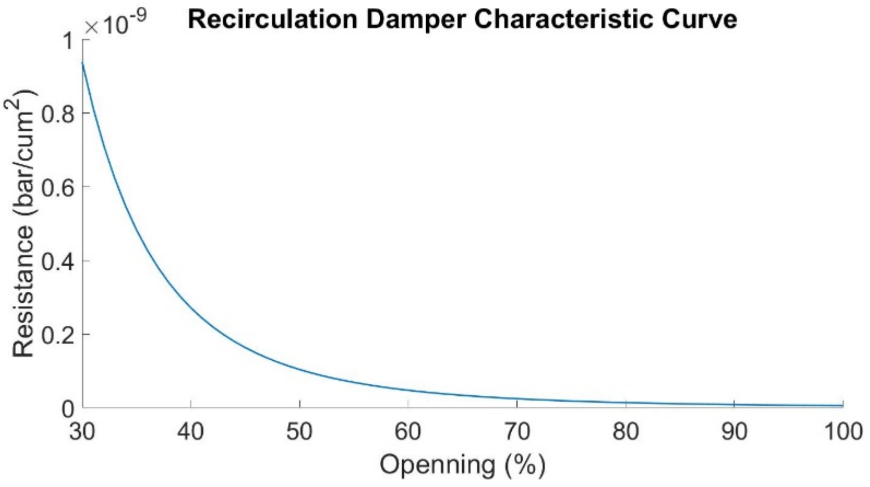


Figure 38 Recirculation Damper Characteristic

3.4 AHU Break Down Model

The Schematic in Figure 9 shows the neural network configuration which are used in modelling the air handling unit.

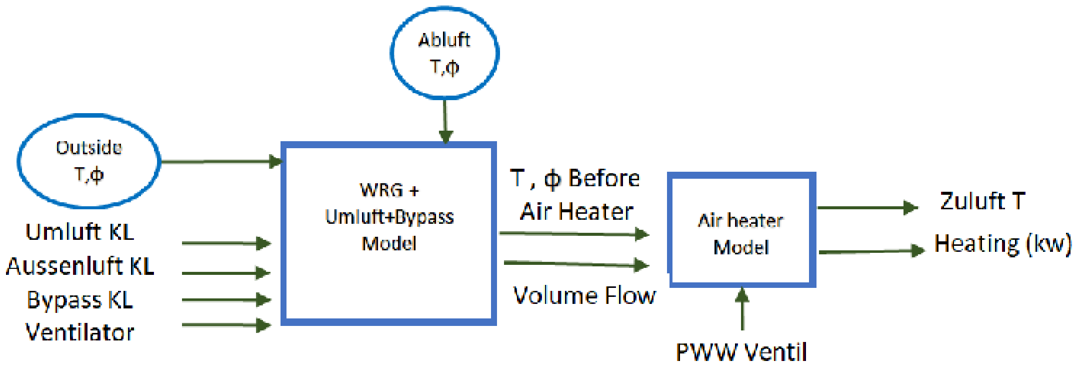


Figure 39 Schematic of the neural network configuration

Unlike the unit number one which had a mixing damper for mixing the outside air and the exhaust air after the evaporator which made a loop for calculating the air temperature and humidity this air handling unit does not have a heat pump and consequently no mixing damper which makes it possible to run the calculations without initial guess and iterations.
Here we have one neural network for calculating vlume flow rates as follows:

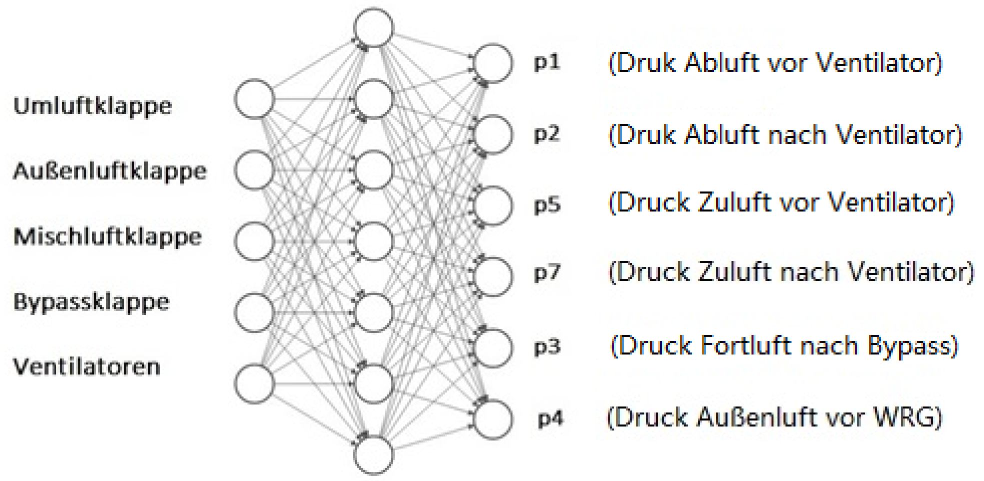

Umluftklappe
Außenluftklappe
Mischluftklappe
Bypassklappe
Ventilatoren
p1 (Druk Abluft vor Ventilator)
p2 (Druk Abluft nach Ventilator)
p5 (Druck Zuluft vor Ventilator)
p7 (Druck Zuluft nach Ventilator)
p3 (Druck Fortluft nach Bypass)
p4 (Druck Außenluft vor WRG)

Figure 310 Pressure Model Neural Network

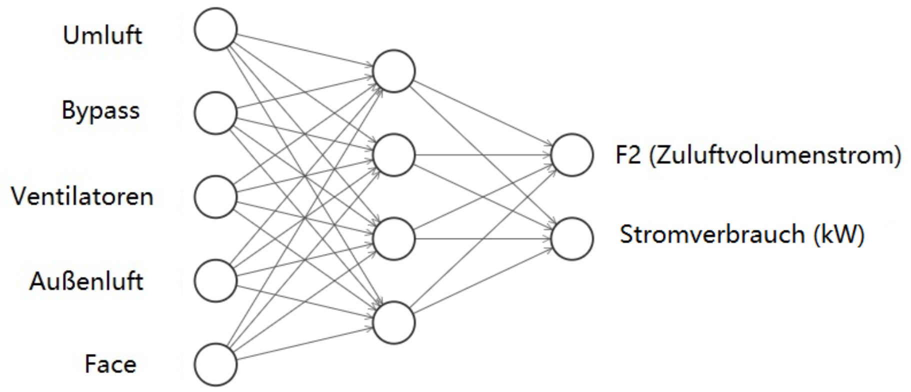


Figure 311 Volumeflow Model Neural Network

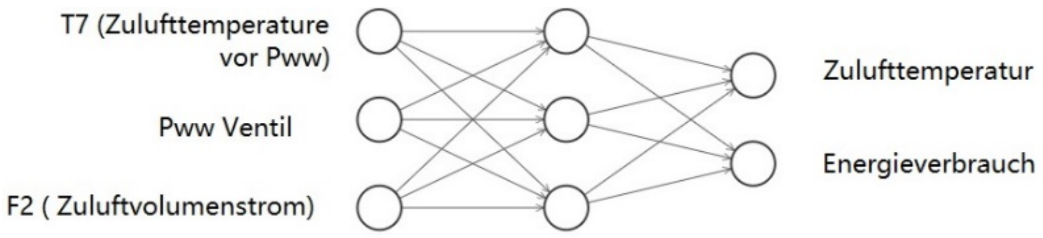


Figure 312 Supply temperature and energy consumption model

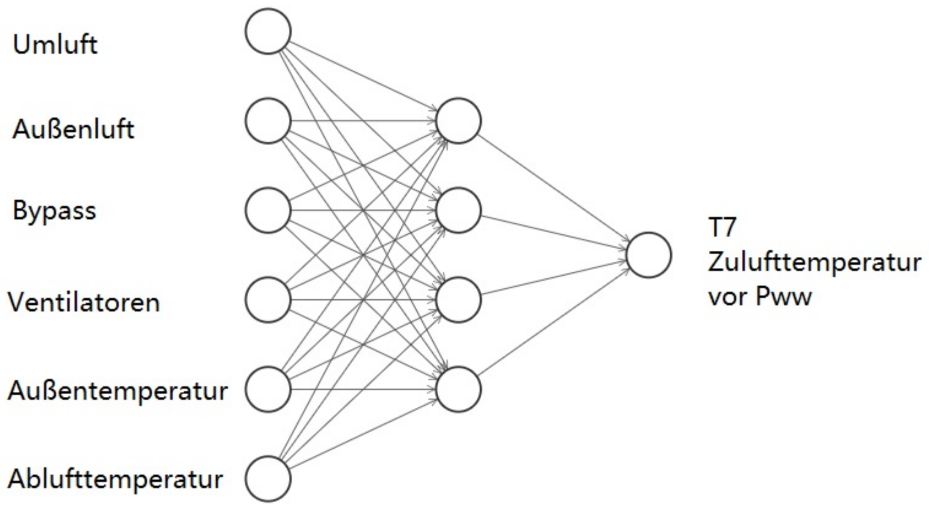


Figure 313 Temperature model

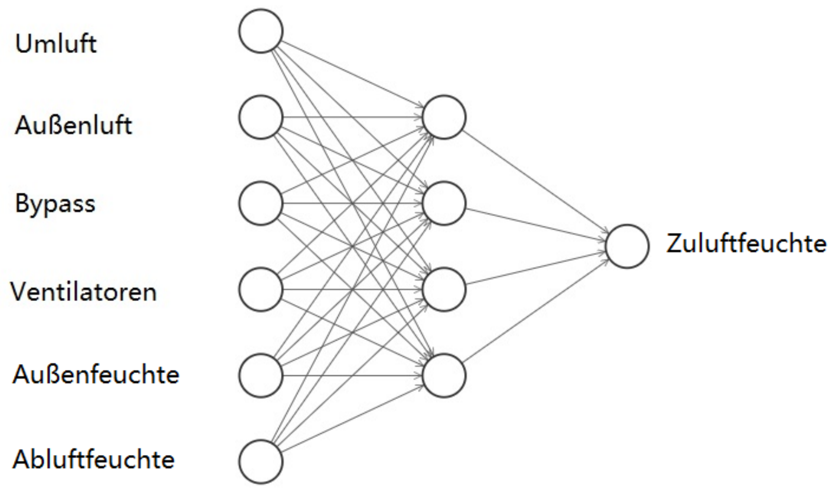


Figure 14 Supply Humidity Neural Network

As it can be seen in the figures above the system is broken down to subsystems. Each subsystem is a shallow neural network that calculates pressures, temperatures and humidities which will be used by other models as input. At the end the whole systems is calculated. We have the following subsystems:

- The pressure model neural network: Feedforward network that calculates different pressures at different positions in the unit
- Volume flow neural network model: Feedforward neural network model that calculates the supply volume flow rate
- Supply temperature model: Feedforward neural network model that uses the output from the tmeperature model and together with outside and return temperature and damper positions calculates the supply temperature
- Supply humidity model: Feedforward network model that calculates supply humidity using outside and return absolute humidity and damper positions

Model Training and validation

The models are trained once with data collected in 2020 and then retrained again two times for data in 2021 and 2022. The simulation results for a sample day are plotted in figure 19 versus historical data.

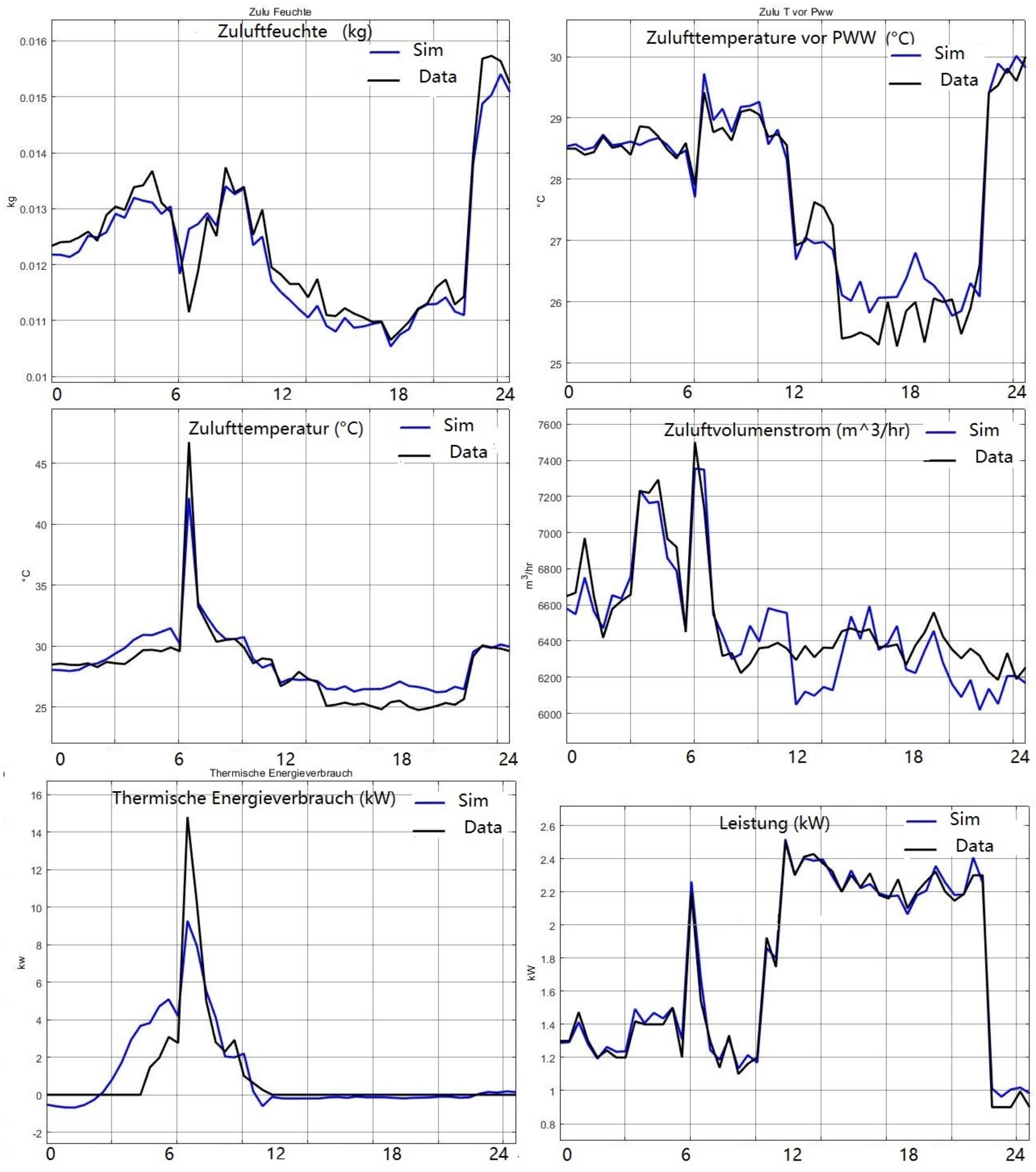


Figure 315 Model Validation

3.4.1 AHU Single Model for the AHU number 1

The AHU break down breaks the system down to several simple models. The drawback is it needs many extra sensors to be installed inside the AHU so we can model each component separately. An alternative approach would be to model the system with a single but more complex neural network model. We tried this using a NARX neural network structure for the bigger AHU with heat pump like below:

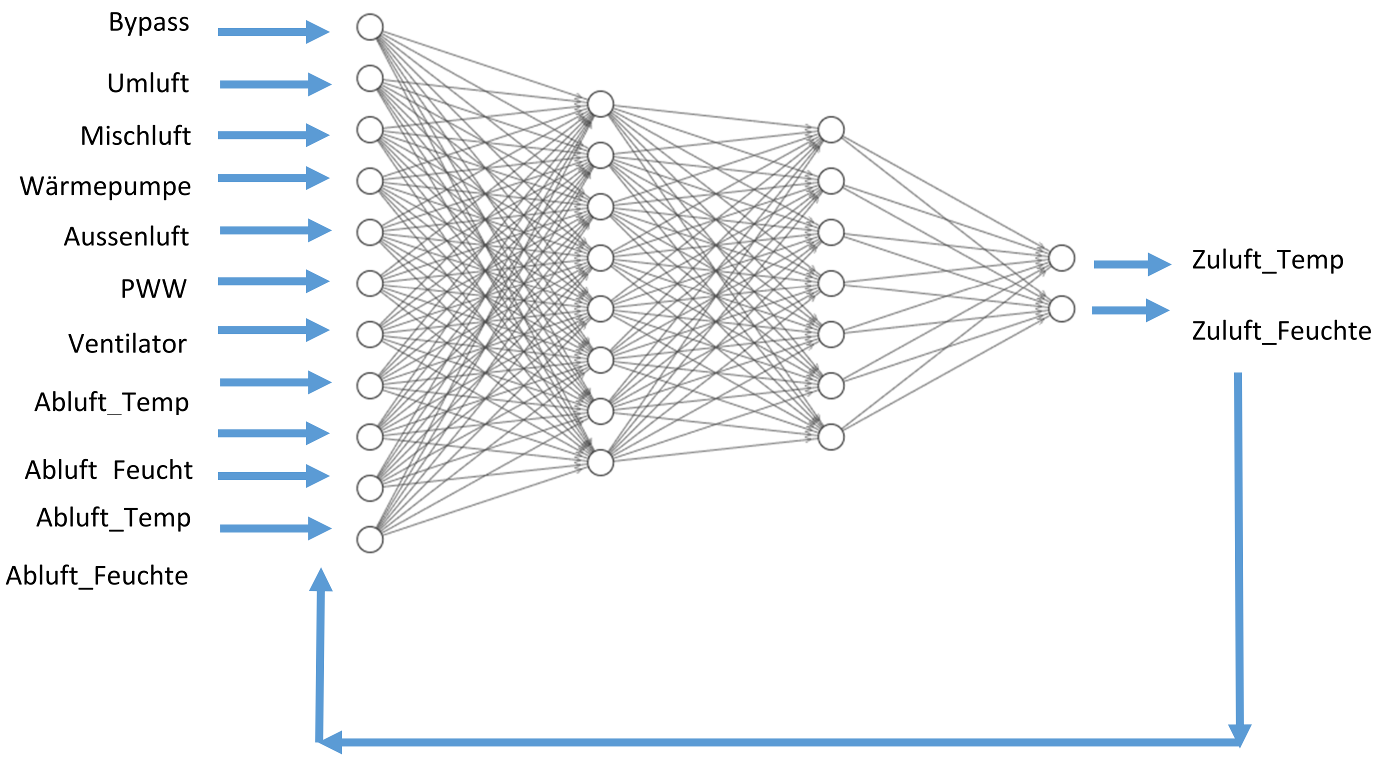


Figure 316 Single Neural Network Structure for AHU

Several neural network structures are trained 10 times with 10 different random initial values. Whichever has the best performance is selected. Here is a summery of the structures and performances:

Table 1 Training Results for AHU Single Model

Network Name	**Network Structure and number of hidden neurons**	**Performance (mse)**
AHUTphi11x2Ramsloh_7_2022	7	0.0062
AHUTphi11x2Ramsloh_15_2022	15	0.0063
AHUTphi11x2Ramsloh_20_2022	20	0.01
AHUTphi11x2Ramsloh_87_2022	[8,7]	0.0077
AHUTphi11x2Ramsloh_1010_2022	[10,10]	0.0158

Table 1 summerizes neural network structures which are trained for the air handling unit with their number of hidden layers and hidden neurons. Their performance is measured based on mean squared error (mse) of the target and simulation values. As seen the best performance is achived here with one hidden layer and 7 hidden neurons.

Plot below shows the step resonse for the AHU using the neural network with best performance. Comparing to physics based model although the neural network model captures generally the functions of the dampers relatively good but neural network is not reliable in capturing the operating modes that rarely happen and the number of samples in our training dataset is too few comparing to other operating modes.
For example in the plot below we see that function of the bypass damper is not captured by neural network as the bypass damper openes only for a limited time period in summer.
Heat pump function is also not easy to capture properly because heat pump has a certain range for safety reasons and inverter problem. For example Heat pump can not run below 30% so we are missing heat pump data in this region.

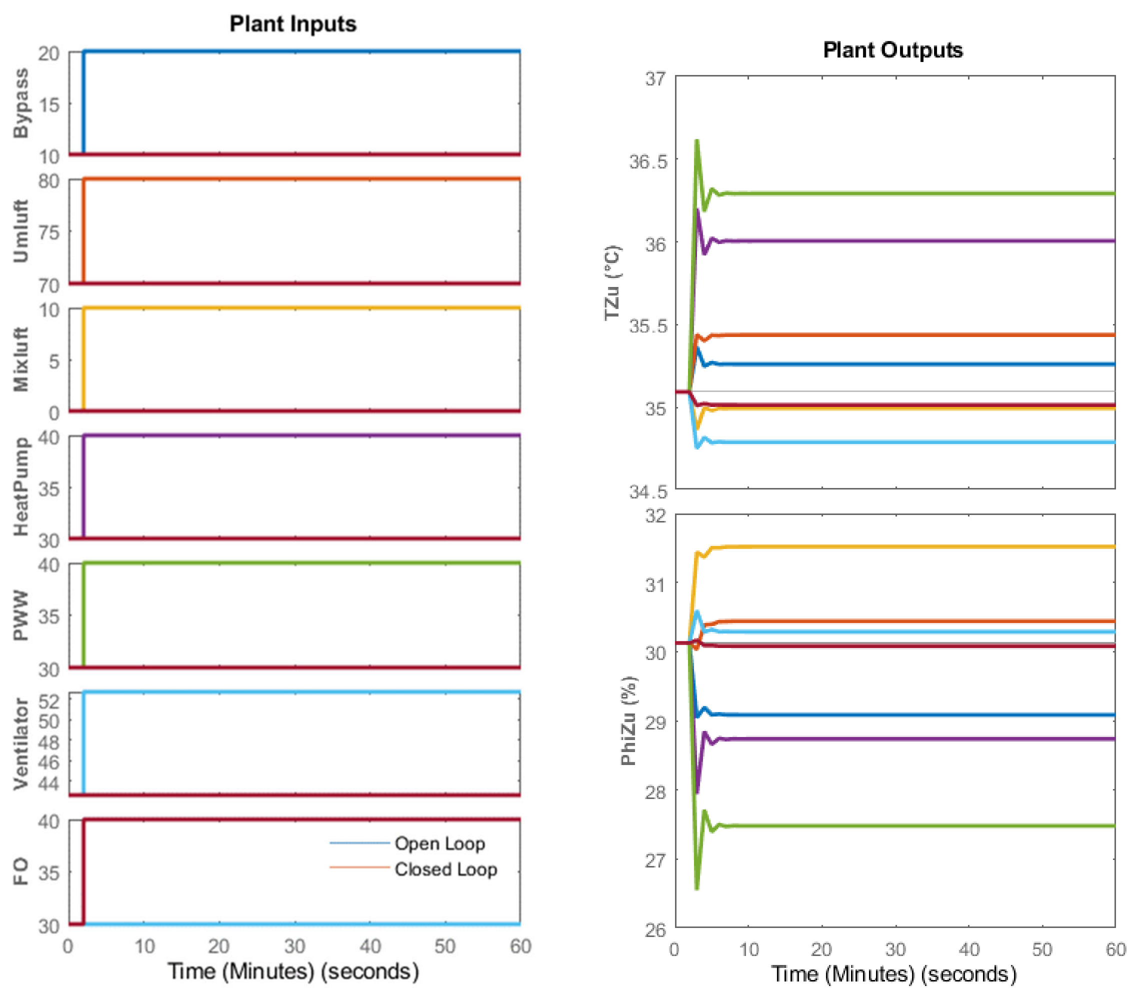


Figure 17 Step Response for AHU Single Model

The step input to the model shows an overshooting response which makes the model not suitable for control purposes. So still the physics based model is preferred here for creating MPC control. Further improvement is still needed for this neural network model so it can be employed for control purpose.

3.4.2 Training LSTM and GRU models for AHUs

Multiple other neural network structures were also tried in pyhon for creating a single model for AHU such as:

- LSTM: Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is capable of capturing long-term dependencies in data. LSTMs are specifically designed to address the vanishing and exploding gradient problems in traditional RNNs. This allows them to effectively capture and maintain dependencies over longer sequences, making them well-suited for tasks involving time-series data.

LSTMs can selectively retain important information and forget irrelevant information from past time steps. This memory retention capability makes them better at capturing long-term patterns and dependencies in sequential data. These features allows the model to handle noise and also the delayed effect of outputs to inputs properly.

- GRU: GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. [2]

The table below summarizes the results:

- Outputs:
  - Thermal Energy,
  - Electrical Energy,
  - Heating,
  - Dehumidification
- Inputs:
  - Return Temperature,
  - Return Humidity,
  - Heat pump Speed,
  - Ventilator Speed,
  - Outside Humidity,
  - Outside Temperature,
  - Recirculation Damper,
  - Outside Damper,
  - Bypass Damper,
  - Mixing Damper

Table 2 Hyper Parameters

Optimizer	Adam
Criterion	MSE
Early stopping patience	10 epochs
Learning rate at the start	0.01
Reducing learning rate patience (mode = min)	5 epochs
Epochs	100
Gradient clipping	1.0
Batch_size	64
Lookback	3, 1(feedforward)

Opimizer: Adam optimizer short for Adaptive Moment Estimation optimizer, is an optimization algorithm commonly used in deep learning. It is an adaptive method for learning rate adjustment strategy and most commonly used algorithm in LSTM method.

Early stopping method is used to avoid overfitting which stops training where the difference between training results and test results is diverging.

With a reducing learning rate feature the training process starts with a higher learning rate and reduces the learning rate when learning stagnates. The learning rate works like relaxation factor in numerical calculations where the increment of the results comparing to previous step is relaxed by a factor.

The neural network has a look back of 3 time steps meaning inputs of s previous time steps are considered.

Table 3 Mean absolute error scores for each model and its respective output

	Architecture	Thermal Consumption	Electical Consumption	Heating	Dehumidification
LSTM	[16], relu	19.782	0.233	209.739	0.2868
	[32], relu	21.664	0.324	250.302	0.2834
	[64], relu	21.613	0.217	264.436	0.2980
	[32, 16], relu	24.096	0.363	207.833	0.3135
	[64, 16], relu	24.950	0.617	257.326	0.3071
GRU	[16], relu	22.306	0.257	262.461	0.2886
	[32], relu	18.917	0.158	224.947	0.2891
	[64], relu	20.807	0.345	216.834	0.3166
	[32, 16], relu	28.130	0.254	254.874	0.3826
	[64, 16], relu	20.380	0.631	238.647	0.3807
Feed-Forward	[16], relu	40.381	0.448	705.909	0.3530
	[32], relu	36.539	0.355	668.100	0.3357
	[64], relu	33.627	0.369	685.258	0.3540
	[32, 16], relu	33.345	0.487	902.309	0.3365
	[64, 16], relu	38.381	0.703	1108.822	0.4103

3.5 Developing a neural network model for the building

In this chapter we focus on training a neural network model to simulate dynamics of hall thermal and humidity balance. Training a neural network for the building is different from air handling units in that we are dealing with a large volume swimming hall with non homogenious temperature and humidity. More ever the building itself has a thermal capacity that absorbs and holds heat and releases that slowly. That is why hall temperature and humidity are not only dependant on the present values of heating, dehumidification and weather condition but also the history of these values in previous time steps affects the present hall condition. In this regards we tried to model the building using appropriate neural network structures such as time series neural networks, NARX and deep learning neural networks LSTM with appropriate look back at input data will be shown in this chapter. These networks have their own specifications and capabilities and are used in different applications.

3.5.1 Training a Feed Forward Network to Predict a Time Series Data

In principle for training a neural network for a system like a building we need to train a nonlinear autoregressive with external input (NARX) neural network that predicts on time series data. Predicting a sequence of values in a time series is also known as multistep prediction. The autoregressive model specifies that the output variable depends linearly on its own previous values. Closed-loop networks can perform multistep predictions. In NARX prediction, the future values of a time series are predicted from past values of that series, the feedback input, and an external time series.
The defining equation for the NARX model is [10]:

$$1)\quad y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u))$$

The building model needs to predict hall temperature and humidity which both of them depend on the past values of hall temperature and humidity as well as past values of other inputs.
One simplification in training NARX network is to substitute it with a simple feed forward neural network and use the moving average of the last n time step inputs instead of n delay values. The main advantage would be that we do not need to have information about n data points for each input to predict the future but only an average of n data points is enough. This is important in making a linear time invariant (LTI) model from the neural network function that at the linearization point only one data point for each input values can define the operating point because in case we include look back values in neural network then one delay function should be added to each input values for each look back resulting 10 delay functions for a look back to 10 previous time steps. Figure below shows the layout for such an input.

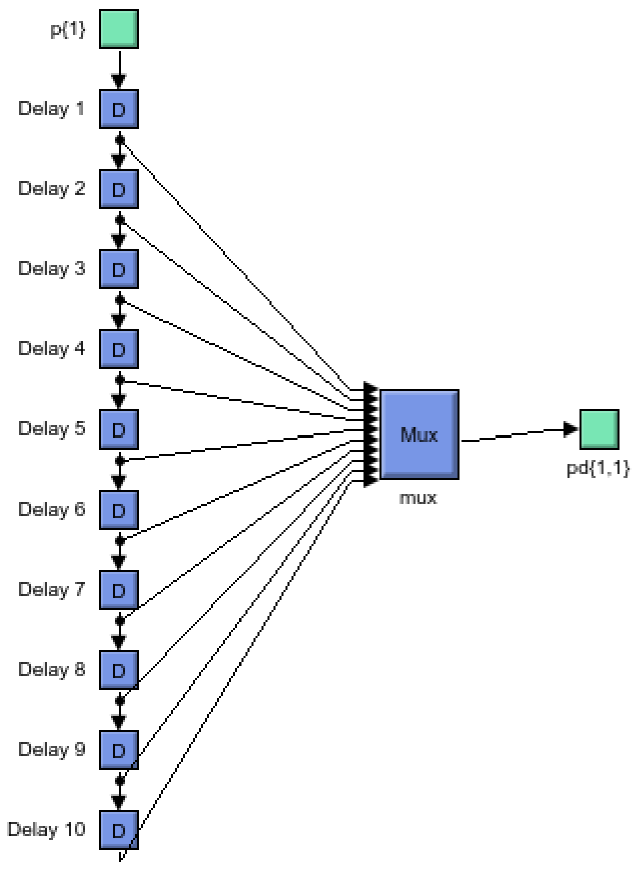


Figure 18 Model input with 10 look backs

In linearization process Simulink assigns one new state value to each delay block. So for a model with 4 inputs and 10 times look back we would have 40 state values for look backs and two more states for output delays. Wich results in a large 42x42 states matrics “A”.
As we know in order to linearize a model we need to find the steady state operating point for the given inputs. This is done in Simulink using a numerical iterative algorithm which would be difficult to come up with a solution when the number of states are too much. In order to avoid this problem we try to make the states matrics “A” simple enough by taking average of previous 10 minutes datapoints instead of adding 10 steps look back feature as shown below:

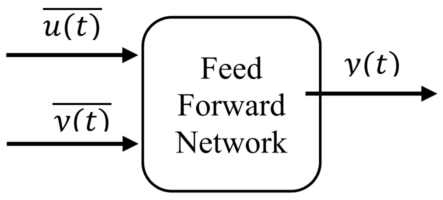


Figure 19 Schematic of Time Averaging of input Data

Where:

2) $\overline{u(t)} = \frac{1}{T}\int_{t_0}^{t_0+T} u(t)dt$

In oder to set up a neural network structure for the building temperature we can consider either feeding pure input data as temperatures and get output as hall temperature as depicted in schematic below :

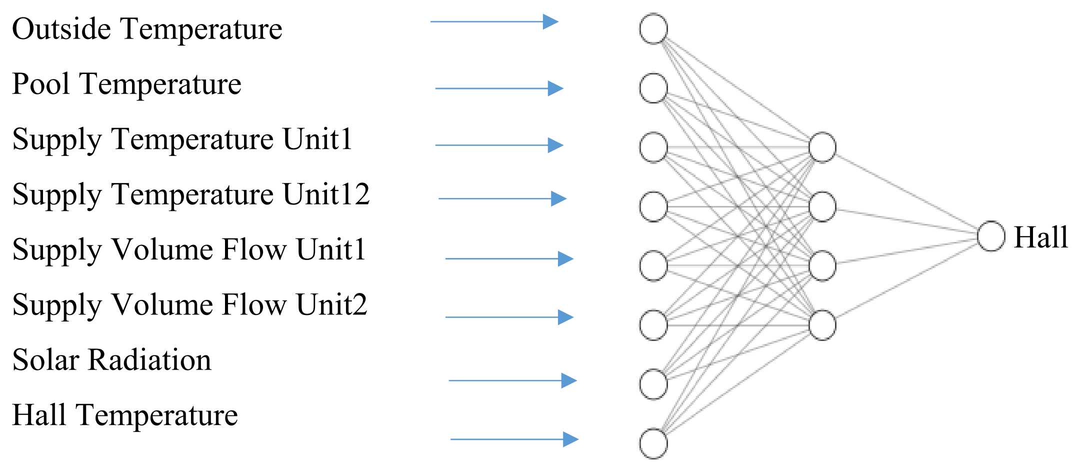


Figure 320 Feedforward Structure for Hall

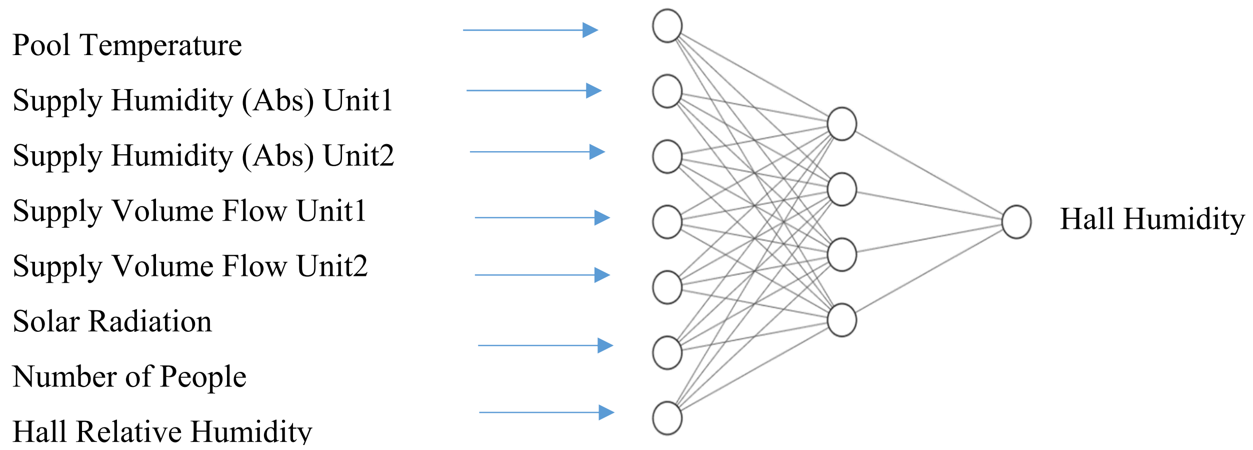


Figure 321 Feedforward Structure for Humidity

Experiance shows designing such a feed forward structure with too many inputs and one output and a feedback from that single output although can be trained to predict the output but fails to recognize the effect of all inputs.
In order to overcome this problem a different neural network structure is set up trying to create new variables with mxing multiple input variables and used them as inputs [11].

In this regard we created a neural network with fewer number of inputs based on physics of the problem in which the temperature rise is a function of heat transfer to the hall from the AHU, outside, pool surface and solar radiation.
This is the so called feature engineering which is a machine learning technique that leverages data to create new variables that aren't in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of simplifying and speeding up data transformations while also enhancing model accuracy.

3) $\frac{dT}{dt} = f(\dot{Q}_{Anlage}, \dot{Q}_{Außen}, \dot{Q}_{Becken}, \bar{I})$

Where:

$\dot{Q}Anlage$:	Overall heat flow through Air Handling Units
$\dot{Q}_{Außen}$:	Heat flow through the walls
$\dot{Q}_{Becken}$:	Heat flow through pool surface
$\bar{I}$:	Solar radiation

4) $\dot{Q}_{Anlage} = f(\sum_1^n (T_{Zuluft} - T_{Halle}).q)$

5) $\dot{Q}_{Außen=f(T_{Außenluft} - T_{Halle})}$

6) $\dot{Q}_{Becken=f(T_{Becken} - T_{Halle})}$

Where:
q: Volume Flow Rate

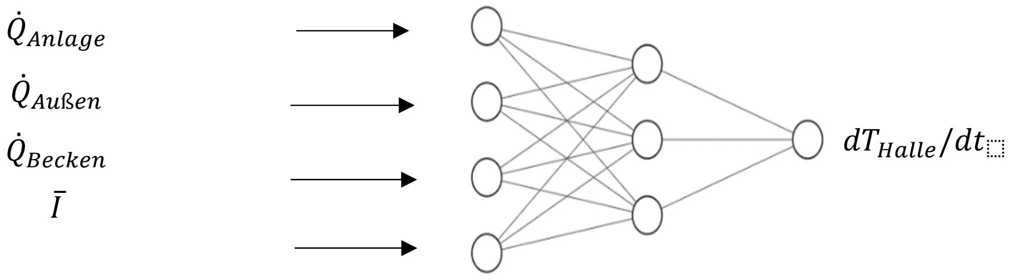


Figure 322 Feed-forward network for temperature variations

Similarily the hall humidity is a function of dehumidification through the AHU, number of people in the hall, pool surface evaporation, solar radiation and hall relative humidity.

7) $\frac{d\varphi}{dt} = f(\dot{\emptyset}_{Anlage}, \dot{\emptyset}_{Leute}, \dot{\emptyset}_{Becken}, \bar{I}, Feuchte_{Halle})$

Where:

$\dot{\emptyset}_{Anlage}$:	Overall Dehumidification through Air Handling Units
$\dot{\emptyset}_{Leute}$:	Evaporation from wet bodies of the people in the hall
$\dot{\emptyset}_{Becken}$:	Pool surface evaporation
$\overline{RH}_{Halle}$:	Hall relative humidity

And:

8) $\dot{\emptyset}_{Anlage} = f\left(\sum_1^n (\bar{\varphi}_{Zuluft} - \bar{\varphi}_{Halle}\right).q$

9) $\dot{\emptyset}_{Becken} = f(T_{Becken}, RH_{Halle}, I)$

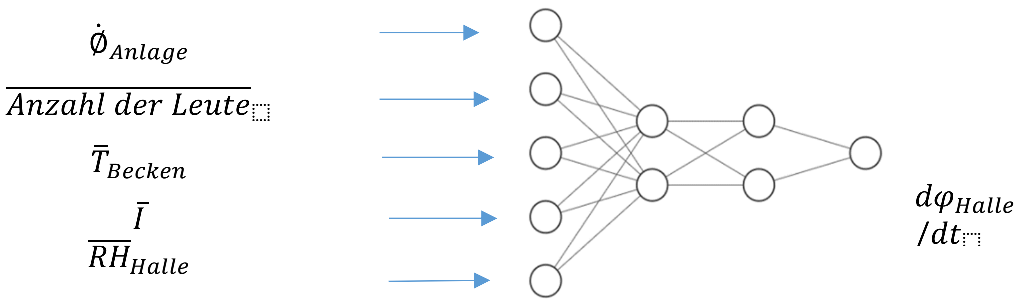


Figure 323 Feed-forward network for humidity variations

By integrating the outputs from the networks above, we get the hall temperature and humidity. The neural network is first trained using data from 2020. The training results are shown in pictures below for Levenberg-Marquardt and Bayesian Regularization training algorithms

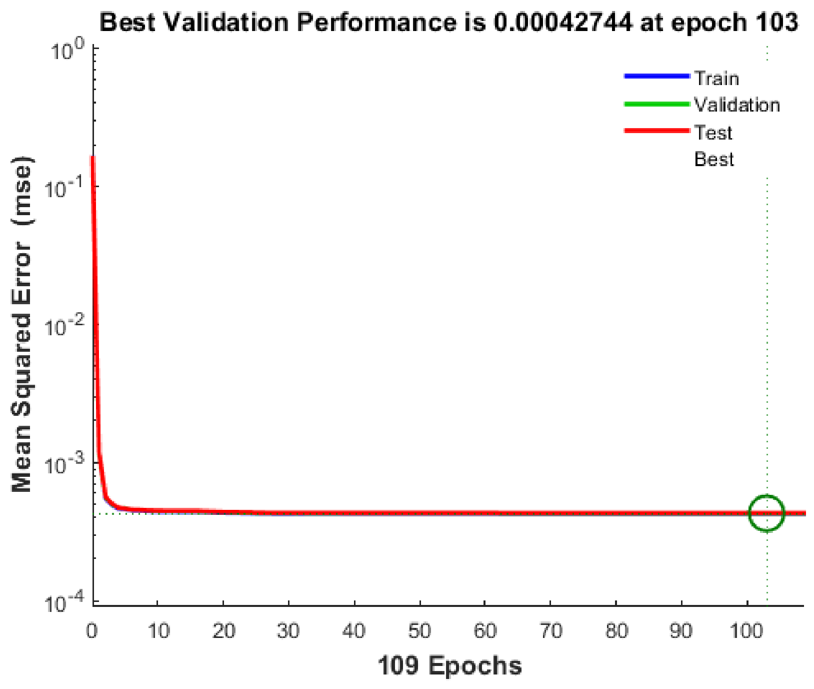


Figure 324 dTdt Performance with Levenberg-Marquardt algorithm 2020 data

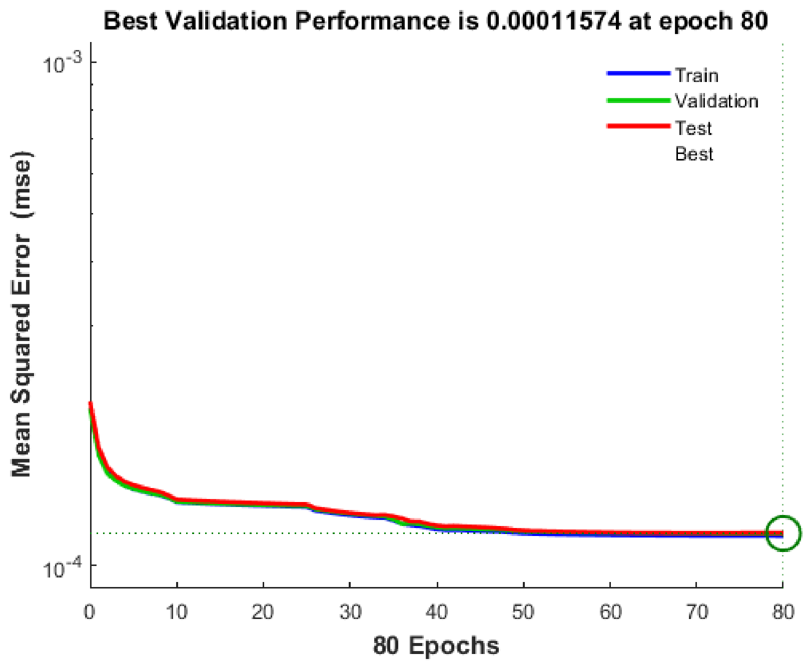


Figure 325 dTdt Performance with Levenberg-Marquardt algorithm retrained with 2021 data

The plots above show the improvement of the neural network performance for training, validation and test datasets versus number of epochs. In each epoch the neural network is trained with a complete set of input datasets for the entire time history. Then training continues until either training performance criteria is met or maximum amount of epochs are reached. Although with training data from 2020 the performance does not improve any more after mse=0.000.4 but retraining the model further improves the performance in figure 25.

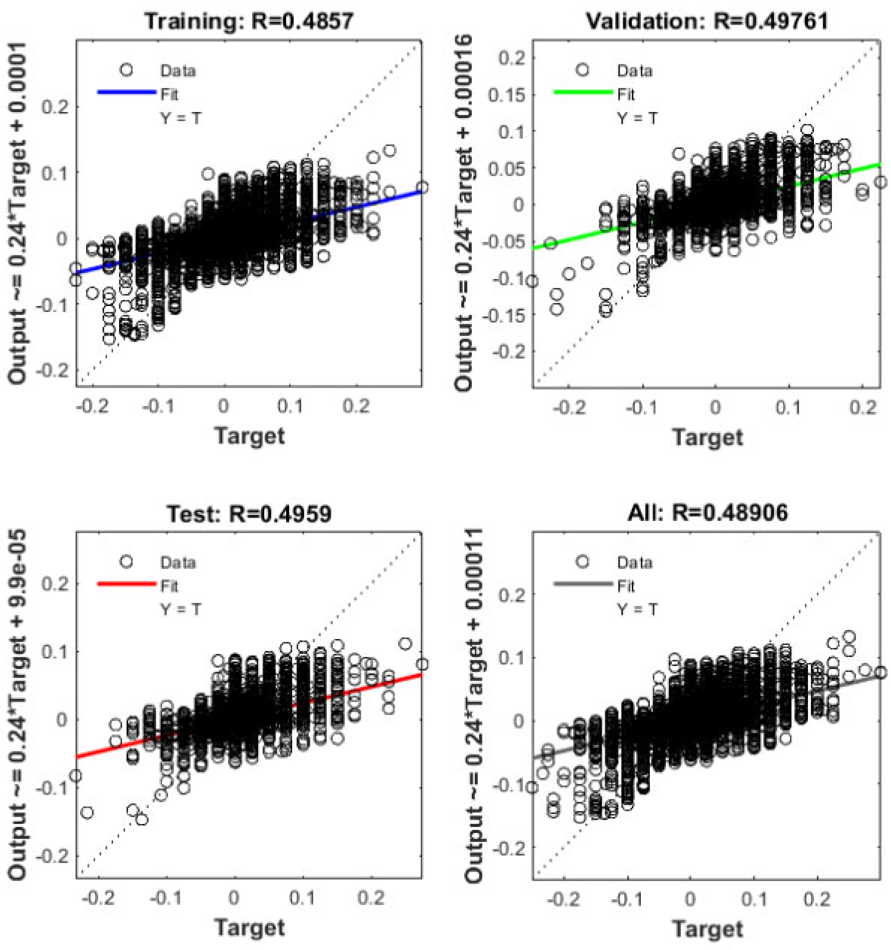


Figure 326 dTdt Regression with Levenberg-Marquardt algorithm 2020 data

The regression plots are one of the measures of training performance. They plot neural network outputs versus target values. It is obvious that for a perfect enural network performance we have to see a line with slope 1. Plots above show that regression does not reach a performance better than 0.48 which is considered as poor performance.

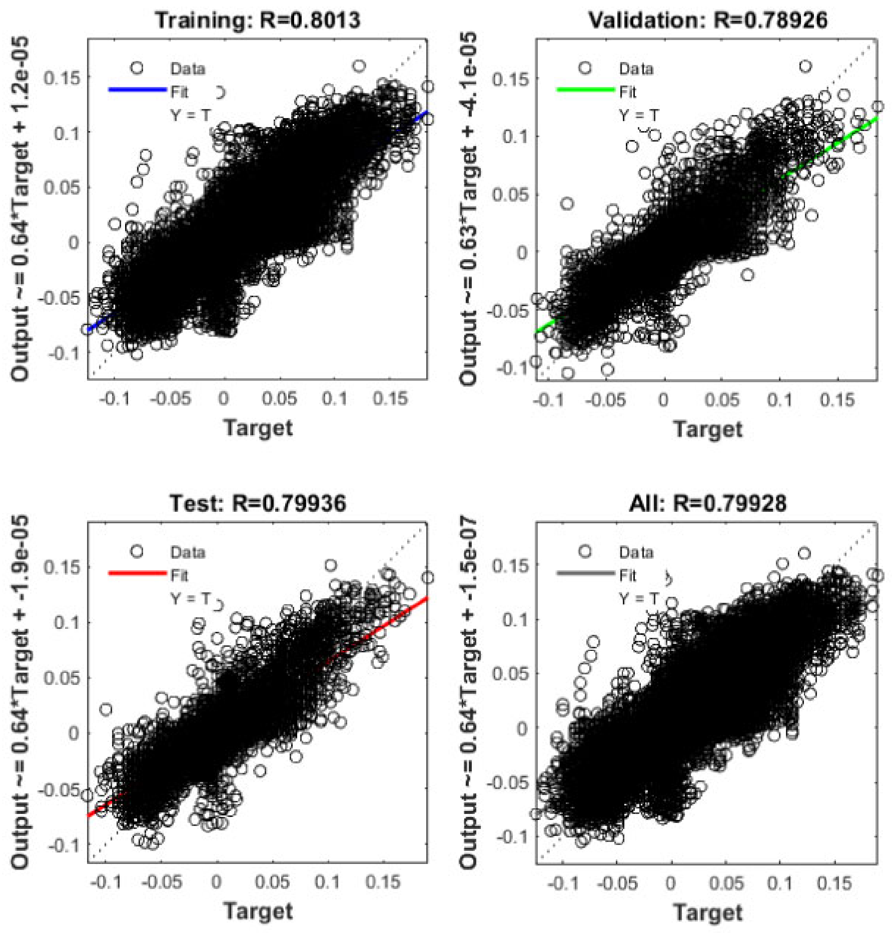


Figure 327 dTdt Regression with Levenberg-Marquardt algorithm retrained with 2021 data

The training performance is slightly better with Bayesien algorithm although the regression parameter shows poor results after training with data collected in 2020. From the regression plots it can be seen that for the majority of data wich lay within the normal data range we get good match between simulation and target values but for the data with higher and lower values there is no good match.
A significant improvement in training performance and regression is observed after retraining the network from 2020 with 2021 data. This could be because of lack of enriched training data points in 2020 because of corona lock down.

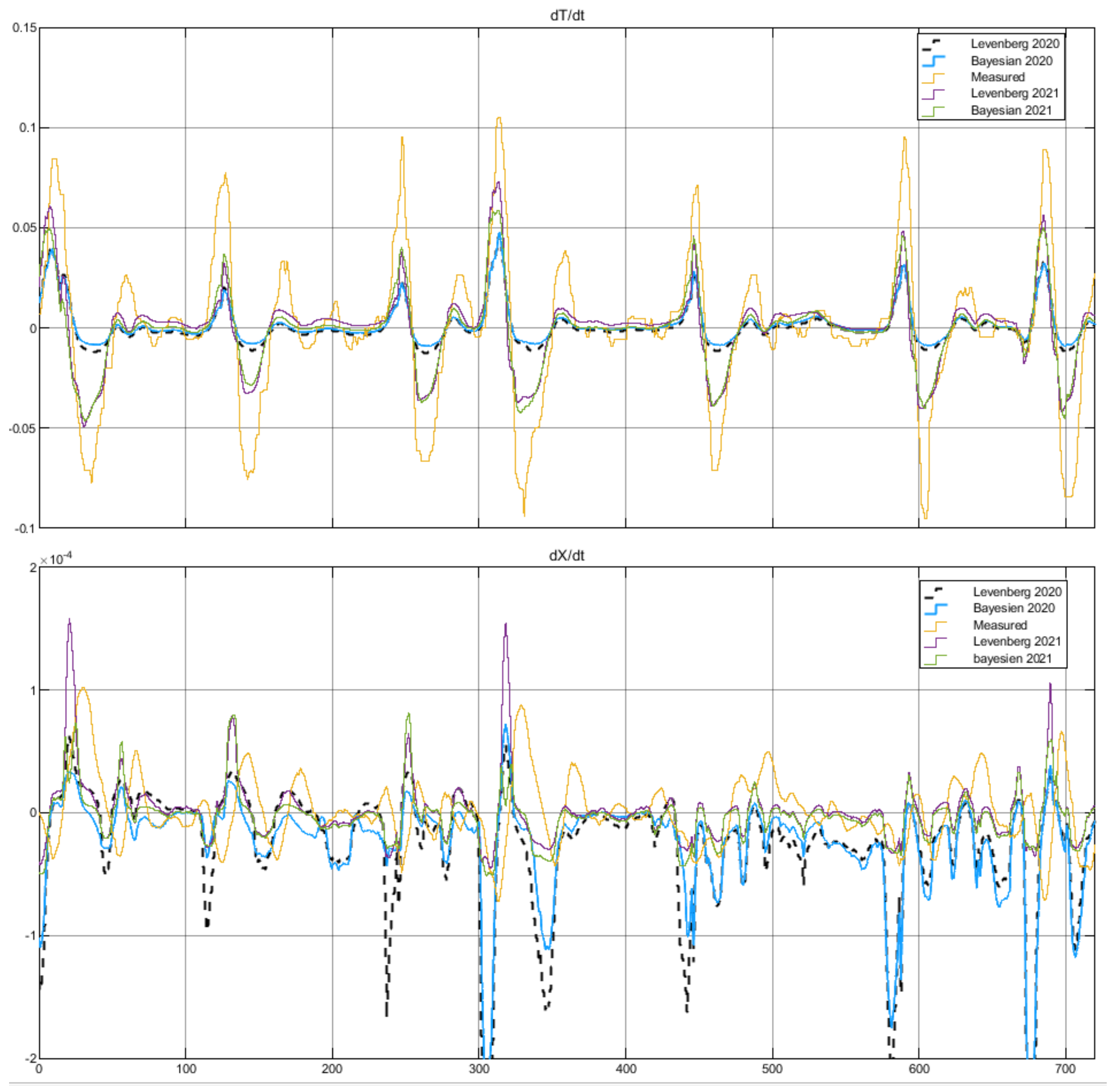


Figure 328 Validation of network (trained with 2020 data, validation with 2021 data)

3.5.2 Training a NARX Network to Predict a Time Series Data

Creating a feed forward network to predict hall temperature and humidity variations over time has the drawback that predicting variation within one minute would include big errors because the sensors are not that precise to capture temperature and humidity variation in the hall within only one minute precisely.
And predicting hall temperature and humidity in a time series needs that these values are known in previouse time steps and be fed to the model. So a recursive neural network needs to be trained in the following structure.
Two structures are created for buildings with two units and building with one unit. In the first one the heating and dehumidifications are added up from two AHUs and used as one single overall heating and dehumidification and one average temperature and humidity is calculated

as the plant output. In the double heating and dehumidification each heating and dehumidification from separate AHU is considered as separate inputs. This accounts for different sensor errors and calibrations and different heat losses in the supply ducts for each AHU. In case we simplify the model by adding up the two heatings and dehumidifications we are actually adding up the errors and heat losses of different kinds which makes the overall heating and dehumidification imprecise. Further more in some buildings with two AHU a small portion of the air conditioning from one AHU is redirected to different regions than the swimming hall as well. In this case adding up the outputs from two AHUs does not make sense and they should be considered separately.

On the other hand considering the model with double heating and dehumidification and two separate temperature and humidity outputs for each hall area provides two separate control signals for the AHUs. This makes the model too complex in terms of control and stability and avoiding the two units to run against each other.

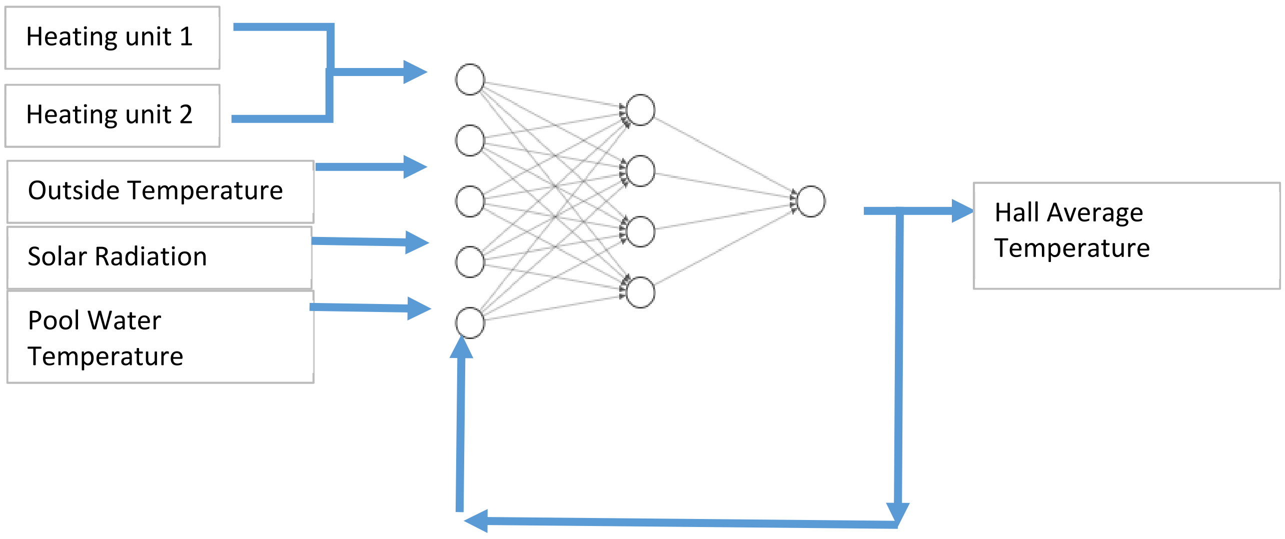


Figure 329 NARX Structure for Hall Temperature model with Unified Heating

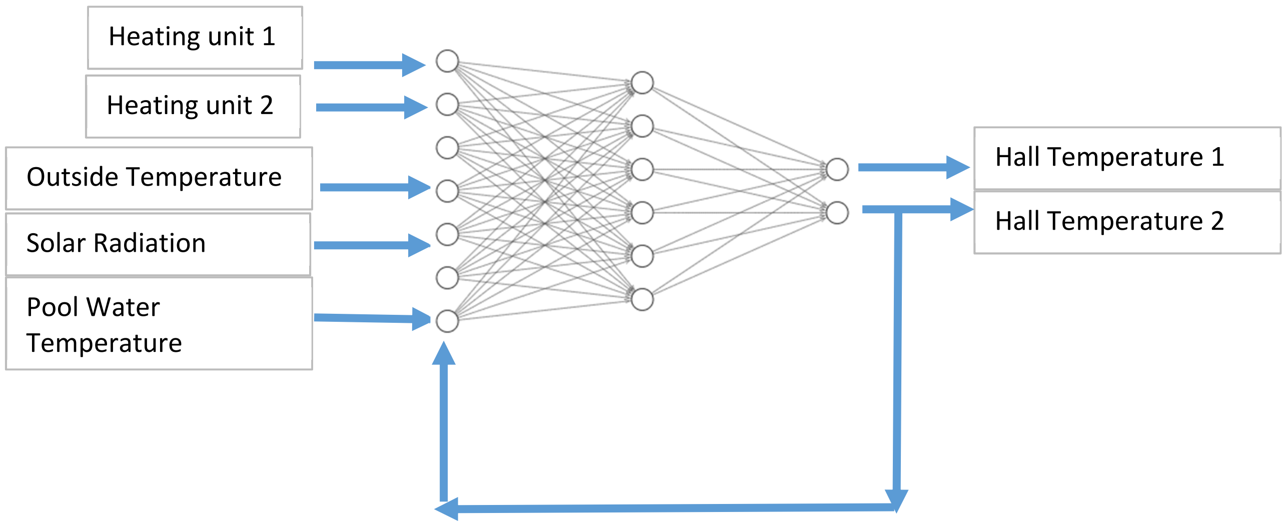


Figure 330 NARX Structure for Hall Temperature model with Double Heating

This is a nonlinear autoregressive model which has exogenous inputs. This means that the model relates the current value of a time series to both past values of the same series current and past values of the driving (exogenous) series — that is, of the externally determined series that influences the series of interest.

In this case we create two different models for hall average temperature and hall average absolute humidity. The exogenouse inputs are moving average of the last ten minutes of data. For predicting hall temperature only solar radiation, pool water temperature and outside temperature are considered as exogenouse inputs aside from heating. The number of people does not show a sensible effect on the hall temperature from data so it was not considered.

For hall absolute humidity only number of people and pool water temperature was considered as exogenouse inputs aside from dehumidification through AHUs. The solar radiation was initially considered as an input to this model but it does not show to have a sensible effect on humidity so it was not considered any more.

For the exogenouse inputs no look back is considered because they are moving averages of the last ten minutes data so they already contain information about the last ten data sequence. The recursive value is considered with only one delay meaning the present values of the hall temperature and humidity is predicted using the present value of exogenouse inputs and hall temperature and humidity of the previouse time step:

10) $T_t = f(\overline{Heating}_t, \overline{Tout}_t, \overline{I}_t, \overline{Tpool}_t, T_{t-1})$

Where:

$\overline{Heating}_t$:	Average heating for the past 10 time steps at present time (t)
$\overline{Tout}_t$:	Average outside temperature for the past 10 time steps at present time (t)
$\overline{I}_t$:	Average solar ration for the past 10 time steps at present time (t)
$\overline{Tpool}_t$:	Average pool water temperature for the past 10 time steps at present time (t)
T_t:	Hall temperature at present time (t)
T_{t+1}:	Hall temperature at next time (t+1)

Several neural network structures are trained 10 times with 10 different random initial values. Whichever has the best performance is selected. Here is a summery of the structures and performances:

Struture:	NARX
Activation function:	tansig
Initialisation function:	Nguyen-Widrow layer initialization function
Learning algorithm:	Levenberg-Marquardt

Table 4 Training Results for Hall T with Unified AHU Heating

Network Name	Network Structure	Performance (mse)
Building6x2T_6_2022	6	0.0157
Building6x2T_7_2022	7	0.0137
Building6x2T_8_2022	8	0.0126
Building6x2T_9_2022	9	0.0116
Building6x2T_10_2022	10	0.0123
Building6x2T_74_2022	[7,4]	0.0132

Table 5 Training Results for Hall Humidity with Unified AHU Dehumidification

Network Name	Network Structure	Performance (mse)
Building6x2X_6_2022	6	0.0253
Building6x2X_7_2022	7	0.0253
Building6x2X_8_2022	8	0.0254
Building6x2X_9_2022	9	0.0248
Building6x2X_10_2022	10	0.0242
Building6x2X_74_2022	[7,4]	0.026

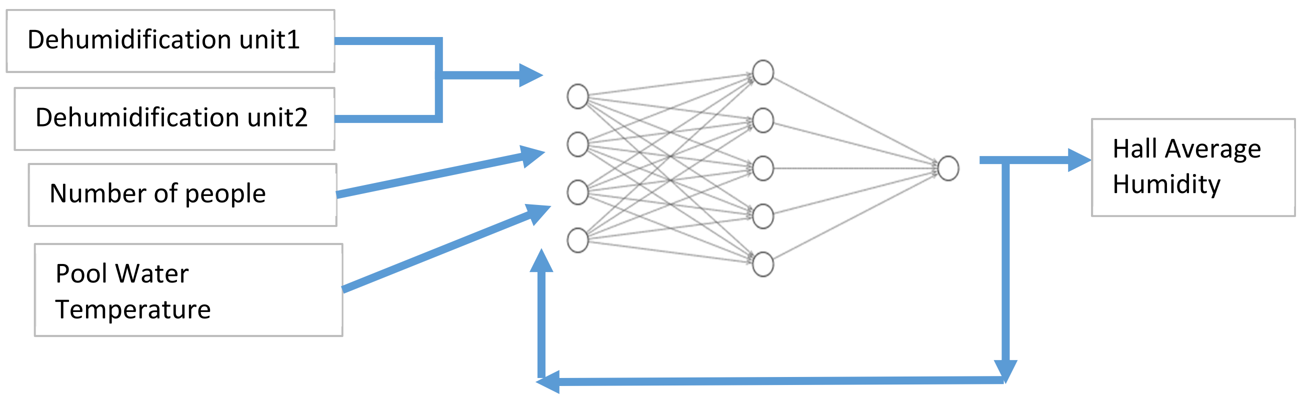


Figure 331 NARX Struacture for Hall Humidity model with Unified Dehumidification

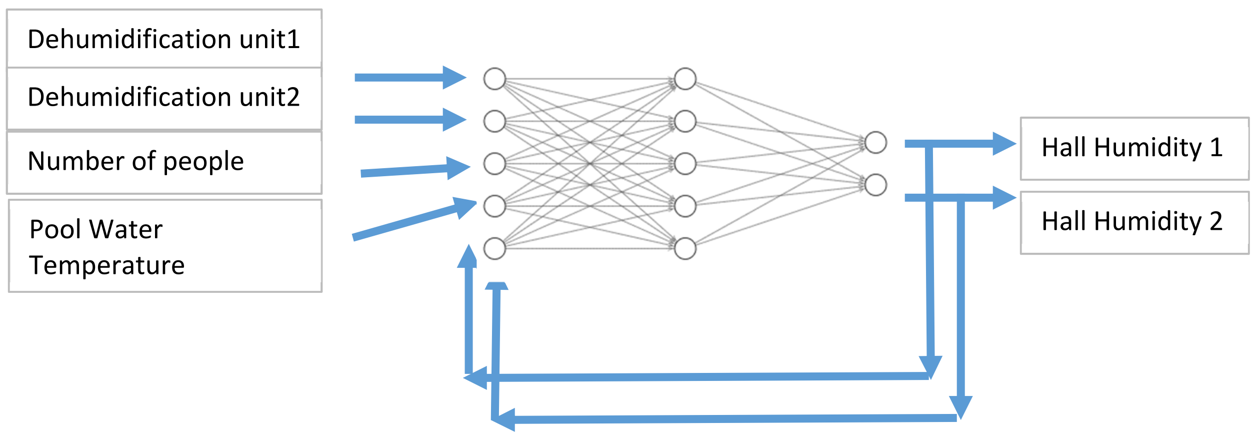


Figure 332 NARX Structure for Hall Humidity model with Double Dehumidification

Table 3 Training Results for Hall T with Double AHU Heating

Network Name	**Network Structure**	**Performance (mse)**
Building8x4T_6_2022	6	0.0052
Building8x4T_7_2022	7	0.0049
Building8x4T_8_2022	8	0.0042
Building8x4T_9_2022	9	0.0052
Building8x4T_10_2022	10	0.0041
Building8x4T_74_2022	[7,4]	0.0048

Table 4 Training Results for Hall Humidity with Double AHU Dehumidification

Network Name	**Network Structure**	**Performance (mse)**
Building8x4X_6_2022	6	0.0263
Building8x4X_7_2022	7	0.0264
Building8x4X_8_2022	8	0.0249
Building8x4X_9_2022	9	0.0263
Building8x4X_10_2022	10	0.0249
Building8x4X_74_2022	[7,4]	0.0242

As it can be seen in the tables above we can reach a best performance of 0.0041 for hall temperature and 0.0242 for hall humidity with double heating and dehumidification and a best performance of 0.0116 and 0.0242 for hall temperature and humidity in unified model. This is because in unified heatig model we add the amount of heating and dehumidifications from the two AHUs and consider it as the overall heating and dehumidification to the building. In this case the errors and uncertainties of each AHU measurement are added and make our training data poor. The advantage of considering the overall heating and dehumidification to the building as single input is that this makes the model a more general platform for the model which is adaptable to different swimming halls regardless of the number of AHUs.

The next step is making a LTI (Linear Time Invariant) out of the model to represent it as state space model:

$$dx = \boldsymbol{A}x + \boldsymbol{B}u$$

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{D}$$

Matrices **A**, **B** are the Jacobians of the system, evaluated at the operating point:

$$11) \quad A = \frac{\partial x_i}{\partial x_j}$$

$$B = \frac{\partial x_i}{\partial u_j} = \begin{bmatrix} \frac{\partial T}{\partial \dot{Q}_{Anlage}} & \frac{\partial T}{\partial \dot{\emptyset}_{Anlage}} & \frac{\partial T}{\partial \bar{T}_{Außen}} & \frac{\partial T}{\partial \bar{I}} & \frac{\partial T}{\partial \overline{Leute}} & \frac{\partial T}{\partial \bar{T}_{Becken}} \\ \frac{\partial \emptyset}{\partial \dot{Q}_{Anlage}} & \frac{\partial \emptyset}{\partial \dot{\emptyset}_{Anlage}} & \frac{\partial \emptyset}{\partial \bar{T}_{Außen}} & \frac{\partial \emptyset}{\partial \bar{I}} & \frac{\partial \emptyset}{\partial \overline{Leute}} & \frac{\partial \emptyset}{\partial \bar{T}_{Becken}} \end{bmatrix}$$

Jacobians are computed by block linearization which is computed by numerically perturbing the states and inputs of the block at the operating point of the system.
The system perturbation algorithm introduces small perturbations to the nonlinear system and measures the response to this perturbation. The algorithm uses this perturbation and the resulting response to compute the linear state-space of this system. The jacobian elements are found using the neural network function and perturbing inputs as below:

$$12)\quad \underbrace{\frac{\partial T}{\partial \dot{\emptyset}_{Anlage}}}_{\text{Neural network Output at the operating point}} = \frac{dT}{dt}\bigg|_{\dot{\emptyset}_{Anlage0},\bar{T}_{Außen0},I0,\bar{T}_{Becken0}} \cdot \underbrace{\frac{dt}{d\dot{\emptyset}_{Anlage}}}_{\text{Numerical perturbation}}$$

3.5.3 Training LSTM and GRU Neural Networks for the Building Model

LSTM and GRU and RNN are more complex neural network models that are used in deep learning. These are specific neural networks for sequence learning where a sequence of data in time are fed to the network and and they can keep track of long term or short term dependancies between a sequence of data in time. Their ability to use internal state (memory) to process arbitrary sequences of inputs makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. [11].
Our previous NARX neural networks are time series modelling networks where the outputs are dependant on the past values of the same serie as well as current and past values of other independent inputs with a feedback conection from the output to the input. NARX models are commonly used in time series forecasting when historical values of a sequence (such as stock prices, weather conditions, or sensor readings) and external factors (economic indicators, holidays, etc.) jointly impact future values.

Table 6 Hyperparameters

Optimizer	Adam
Criterion	MSE
Early stopping patience	10 epochs
Learning rate at the start	0.01
Reducing learning rate patience (mode = min)	5 epochs
Epochs	500
Gradient clipping	1.0
Batch_size	1
Lookback	3

	Architecture	**AbTmp**	**AB_Ahum**
LSTM	[16], tanh	0.061	9.829e-05
	[32], tanh	0.047	6.331e-05
	[64],tanh	0.035	5.349e-05
	[64, 16], tanh	0.054	5.643e-05
	[64, 32], tanh	0.280	5.537e-05
GRU	[16], tanh	0.081	11.412e-05
	[32], tanh	0.056	7.969e-05
	[64],tanh	0.037	6.195e-05
	[64, 16], tanh	0.066	5.415e-05
	[64, 32], tanh	0.033	5.536e-05
RNN	[16], relu	0.035	5.910e-05
	[32], relu	0.030	5.021e-05
	[16, 16], relu	0.261	5.631e-05

Table 7 Mean absolute error scores for each model and its respective output

We tried several RNN, LSTM and GRU models and as seen in table above did not achieve promising performance results specifically for hall temperature the mse values are too high. So we finally continued with NARX models.

Recurrent neural networks handle the previous time steps of inputs in such a way that inputs are acted on by the hidden state of the cell to produce the output, and the hidden state is passed to the next time step.

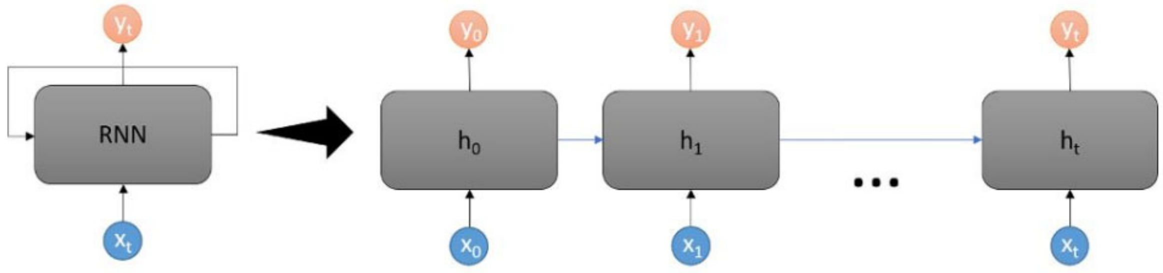


Figure 33 Unrolling a single cell of an RNN [13]

As the figure 33 shows each RNN cell has a hidden feedback which relates the sequence of data. This makes the recurrent neural networks more complex for training and also as in literatures mention suitable in applications such as unsegmented, handwriting recognition or speech recognition because of their ability in pattern recognition in a sequence of data.
While what makes NARX networks suitable for our work is to consider a feedback from output as shown in figure 34 which relates the output to its own past values that makes the neural network capable of predicting time series data such as weather condition or stock prices.

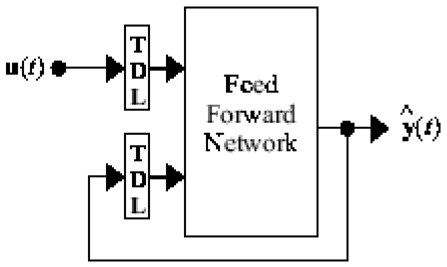
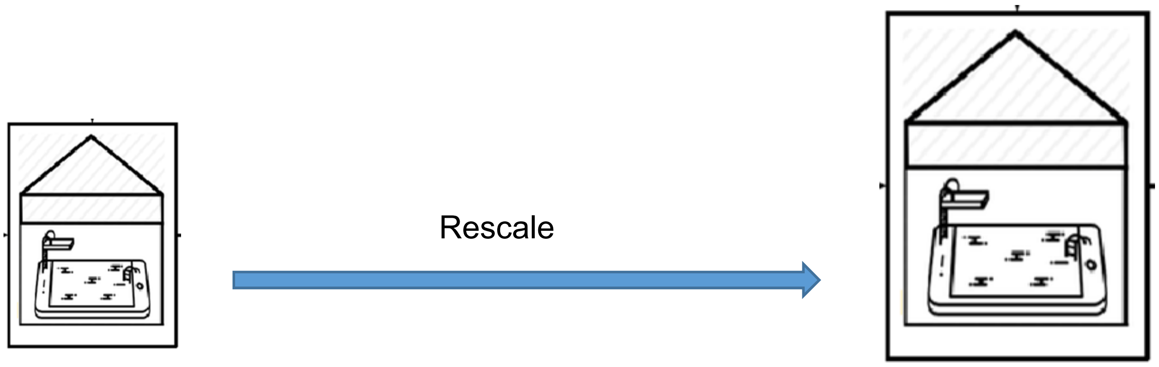


Figure 34 NARX neural Network Architecture [10]

4. Transfer learning and adaptability

In order to adapt the model for different projects with different specifications we need to adapt the current trained model to the new data. Two approaches are taken. First one scaling up and down the linearized model to adapt to the existing linearized model. Second approach would to retrain the model by new data starting the initial values from the already trained model instead of starting from random data.

4.1 Scaling up and down the models to adapt to the new projects



$$dx = \mathbf{A1}(p)x + \mathbf{B1}(p)u$$
$$y = \mathbf{C1}(p)x + \mathbf{D1}(p)$$

$$dx = \mathbf{A2}(p)x + \mathbf{B2}(p)u$$
$$y = \mathbf{C2}(p)x + \mathbf{D2}(p)$$

Figure 35 Rescaling Models

As it would be time consuming and costly to create models for every new project we are interested to use the already trained models to use them for new projects. There are two approaches to achieve this goal. One ist to use the trained model and collect new data from the new swimming hall and retrain and update the model which would be still time consuming. The second option would be to derive a transformation matrix that can be applied to the LTI model state space matrixes. The transformation matrixes $\begin{bmatrix} \propto_1 \\ \propto_2 \\ \propto_3 \end{bmatrix}, \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$ will contain dimensionless factors that can be applied to the state space matrices of the reference model.

$$dx = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{23} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \times \begin{bmatrix} \propto_1 \\ \propto_2 \\ \propto_3 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{23} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \times \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

In theory for analytical model for the building we have:

$$dT_{Hall} = \frac{\phi_{Ventilation}}{c_p.V_{Hall}} + \frac{A_{wall}.h.\left(T_{Wall_i} - T_{Hall}\right)}{c_p.V_{Hall}} + \frac{\phi_{Radiation}.A_{window}}{c_p.V_{Hall}} + \frac{A_{Pool}.h.(T_{Pool} - T_{Hall})}{c_p.V_{Hall}}$$

$$dX_{Hall} = \frac{\phi_{Ventilation}}{V_{Hall}} + \frac{\gamma 1.\phi_{Radiation}.A_{window}}{V_{Hall}} + \frac{\gamma 2\,.Nuser}{V_{Hall}} + \frac{A_{Pool}.g(X_{spool} - X_{Hall})}{V_{Hall}}$$

$$\text{B}=\begin{bmatrix} \frac{1}{c_p.V_{Hall}} & 0 & \frac{A_{wall}.h}{c_p.V_{Hall}} & \frac{A_{window}}{c_p.V_{Hall}} & 0 & \frac{A_{Pool}}{c_p.V_{Hall}} \\ 0 & \frac{1}{V_{Hall}} & 0 & \frac{\gamma 1.\ \ A_{window}}{V_{Hall}} & \frac{\gamma 2}{V_{Hall}} & \frac{A_{Pool}.\gamma 3}{V_{Hall}} \end{bmatrix}$$

$$A=\begin{bmatrix} -\frac{2\times A_{wall}.h}{c_p.V_{Hall}} & 0 \\ 0 & \frac{A_{Pool}.\gamma 4}{V_{Hall}} \end{bmatrix}$$

$$B2 = \beta \times B_{ref}$$

$$\beta = \frac{B_2}{B_{ref}} = \left[\frac{A_{wall}}{V_{Hall}} \Big/ \frac{A_{wall}}{V_{Hall}}_{ref} \quad \frac{A_{window}}{V_{Hall}} \Big/ \frac{A_{window}}{V_{Hall}}_{ref} \quad \frac{1}{V_{Hall}} \Big/ \frac{1}{V_{Hall}}_{ref} \quad \frac{1}{c_p.V_{Hall}} \Big/ \frac{1}{c_p.V_{Hall_{ref}}} \quad \frac{A_{Pool}}{c_p.V_{Hall}} \Big/ \frac{A_{Pool}}{c_p.V_{Hall_{ref}}} \right]$$

$$\alpha = \frac{A_2}{A_{ref}} = \left[\ \frac{A_{window}}{V_{Hall}} \Big/ \frac{A_{window}}{V_{Hall}}_{ref} \quad \frac{A_{Pool}}{V_{Hall}} \Big/ \frac{A_{Pool}}{V_{Hall_{ref}}} \right]$$

Where the subscript ref referes to the reference model.
So analytically it can be shown that the α, β matrices are actually the ratios of thermal properties and geometrical values of the new model to the reference model.

In the example below the step response of the reference building and a new building with hall volume 0.8 times the reference value is shown.
The state space matrices for the building model are:

```
B =
          WPEQnet      WPEXnet         Tout            I        Nuser        TPool
 T       3.04e-07            0     0.001328      4.6e-06            0      0.00453
 X              0    1.603e-06            0   -2.365e-09    6.441e-07    5.749e-06
 phi   -7.761e-07     0.005018    -0.003389   -1.915e-05     0.002016     0.006435
```

```
A =

            T        X      phi
  T    0.9941        0        0
  X         0   0.9966        0
  phi   -2.82     3467        0
```

$$\frac{V_{hall}}{V_{hall_{ref}}} = 0.8$$

$$\beta = \frac{B_2}{B_{ref}} = 1.25$$

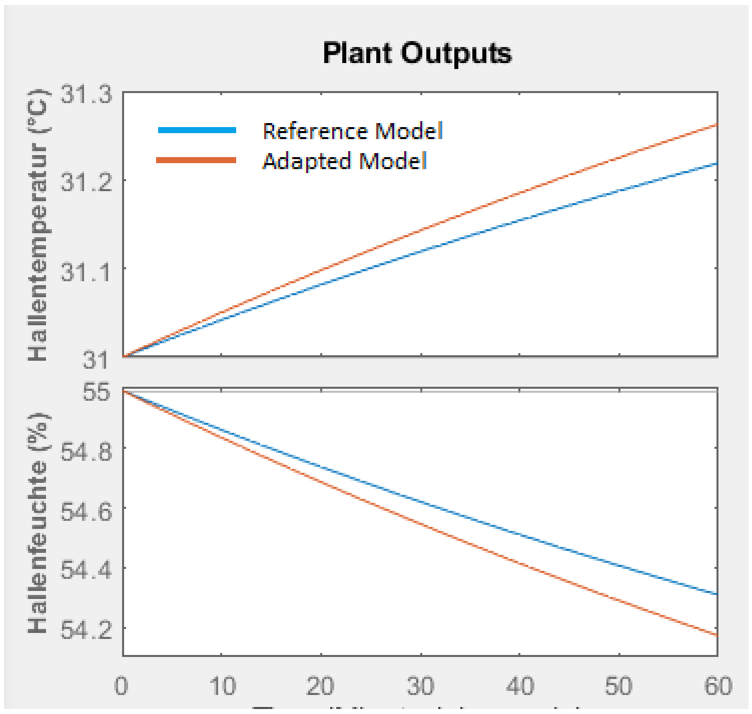


Figure 36 Step response to the heating from the air handling unit

Now assume the new model has window surface of 0.8 times of that of the reference model

$$\beta = \frac{B_2}{B_{ref}} = \begin{bmatrix} 1 & 1 & 1 & 0.8 & 1 & 1 \\ 1 & 1 & 1 & 0.8 & 1 & 1 \\ 1 & 1 & 1 & 0.8 & 1 & 1 \end{bmatrix}$$

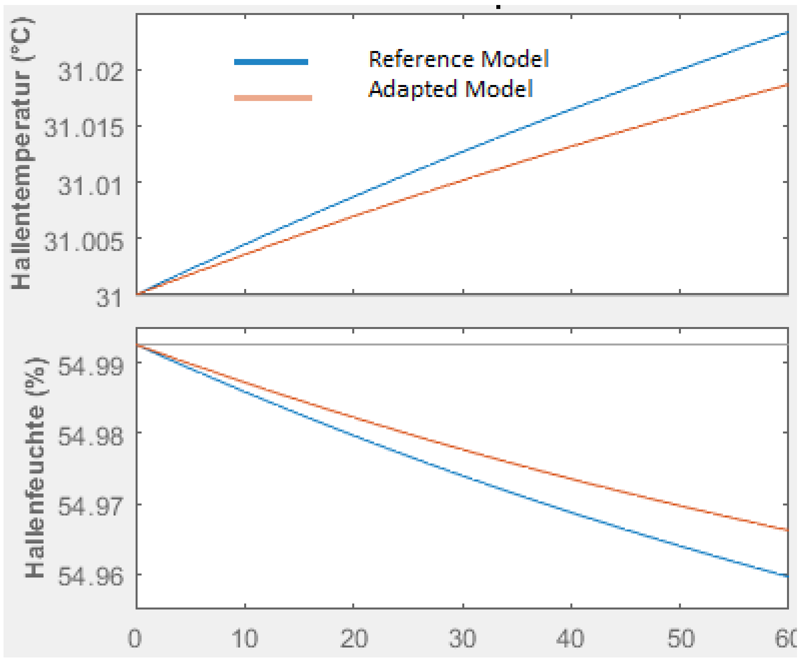


Figure 37 Step response to the solar radiation

Figures 36 and 37 show how applying adjustment factors changes the model response to inputs as expected. Figure 36 shows a building with smaller volume reacts more to increase in heating for a single step because of less heat capacity. Figure 37 shows a building with smaller windows area to the reference model reacts less to the solar radiation. Which is reasonable.

For the air handling unit mode the story is different because we do not have an analytical formulation for the state space representation.

We have the following state space representation for the air handling units:

```
B =

                  Bypass       Umluft      Mixluft     HeatPump          PWW   Ventilator           FO
 DelayeP               0       0.0119    4.847e-08       0.0276            0       0.1082      0.02105
 DelayeTZu        -0.133       0.1044     -4.4e-07       0.2371       0.2081     -0.02425     -0.04585
 WQnet            -653.2        294.3    -0.003724         1165         1022       -469.5       -583.8
 Wxnet         -0.003368       0.2373    9.781e-07   -1.647e-06            0      -0.8086       -1.029
```

Assuming that the new model is similar to the reference model and the only difference is the nominal capacity. We apply this simplification that the jacobians of the reference model is multiplied by the ratio of the nominal capacities of the new model and the reference model. The rest of the dynamics of the system remains the same.

$$13) \quad \beta = \frac{B_2}{B_{ref}} = \frac{Nominal\ Capacity_{\ 2}}{Nominal\ Capacity_{\ ref}}$$

As described this is a workaround. The small discrepencies between the AHUs and the reference model are compensated by manually enerting adjustment factors inside the program.

The other solution to that on which we are still working and will be addressed in chapter cloud connection and online training.
In this method the Edge systems are connected togeather over a cloud system forming a network of AHUs. Model-based control strategies are adapted online using accumulatd and shared data according to collected data.

4.2 Retraining neural networks

The previous method of scaling up and down the linearized models work better for the air handling unit models than for the buildings. The air handling units for swimming halls of different projects have same principles but they are different in size. So the linearized model can be adjusted according to the scaling factors.
In case of different buildings they can have totally different specifications because of thermal properties of the walls and windows orientations, different solar radiation intakes and also different pool water surface to the hall volume ratio. So in this case adapting the neural network is a more reasonable approach.
Because different buildings can have different data range the training data are normalized to (-1, 1) before training so retraining can be carried out with compareable input data for two different buildings.
In retraining we initialize the weights and biases from the pre trained neural network instead of initialzing from random data.
Here is some rsults comparing retraining a swimming hall in Bad Zwischenahn with two months of training data and traing a network from scratch. The neural network is trained first in Ramsloh using one year of data from 2022 then retrained for Bad Zwischenahn for two months of February and March 2023.

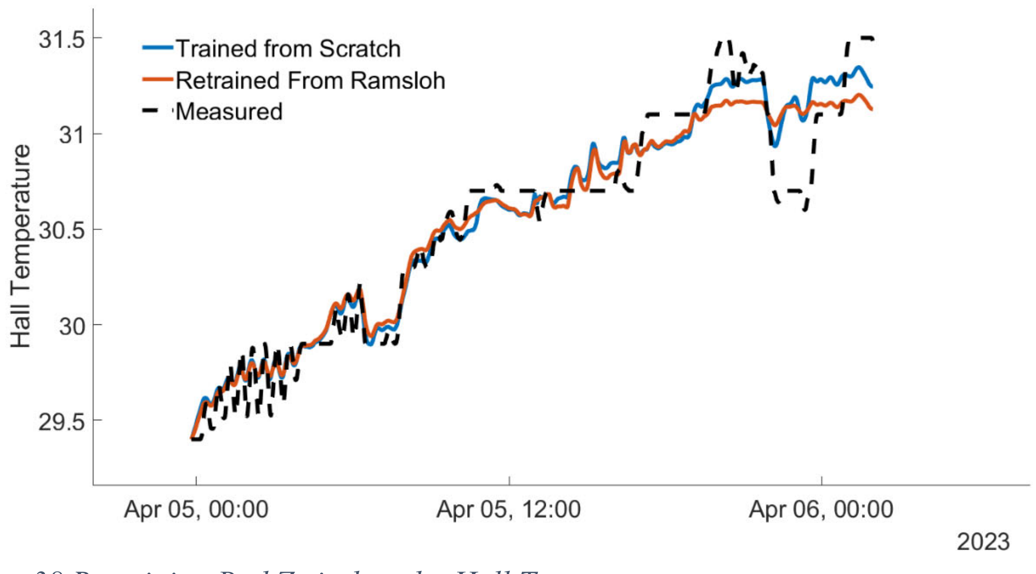


Figure 38 Retraining Bad Zwischenahn Hall Temperature

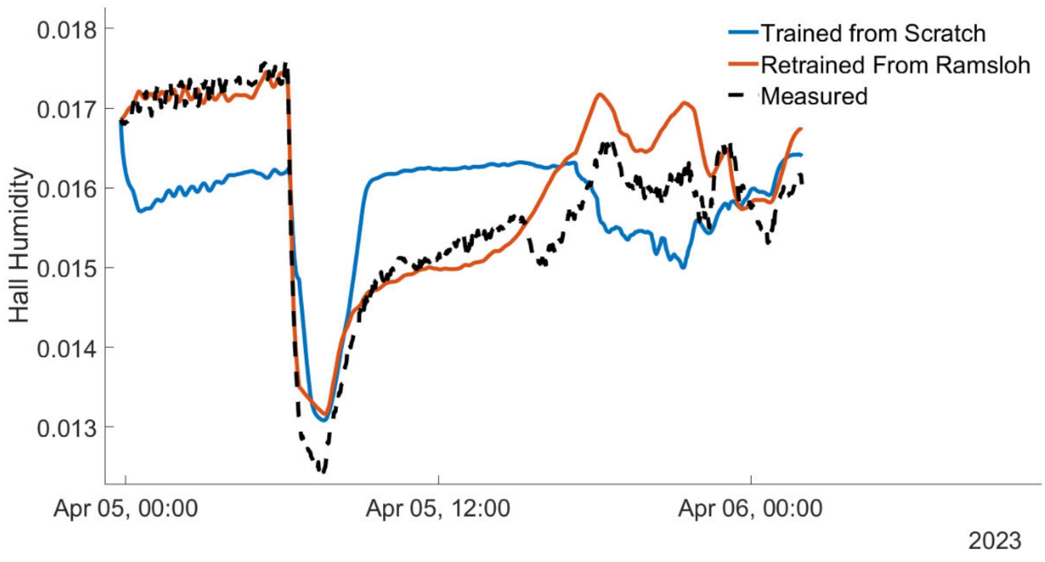


Figure 39 Retraining Bad Zwischenahn Hall Humidity

Plots 38 and 39 show improvement in neural network performance when retrained from reference model rather than training from scratch. Although two months of data is not enough for training a neural network to capture all dynamics and specification of a building thermal and humidity balances but initializing a neural network with weights of the trained reference model could save time in training the building model.

5. Neural network based control system

The main application of neural networks in control problems are to use them as function approxiamtors [14]. As shown in the figure below when we have an unknown system that its physics is unknown we need to train a neural network that can generate the same response as the real system when the same input is applied into them.

In our application the unkown functions correspond to the airhandling unit and the thermal model of the building. The unkown function can represent either the function of the system or the reverse function of the system that we want to control. In the latter case the neural network can be used to control the system.

There are variety of methods that can apply neural networks in control problems. The main typical ways are model predictive control and NARMA-L2 control.

There are typically two steps involved when using neural networks for control: system identification and control design. In the system identification stage, we develop a neural network model of the plant that we want to control. In the control design stage, we use the neural network plant model to design (or train) the controller. In each of the control architectures, the system identification stage is identical. The control design stage, however, is different for each architecture.

5.1 Neural Network Predictive Control

The neural network predictive controller uses a neural network model of a nonlinear plant to predict future plant performance. The controller then calculates the control input that will optimize plant performance over a specified future time horizon. The first step in model predictive control is to determine the neural network plant model (system identification) [15]. Next, the plant model is used by the controller to predict future performance.
The first stage of model predictive control as well as the other control is to train a neural network to represent the forward dynamics of the plant. The prediction error between the plant output and the neural network output is used as the neural network training signal. The process is represented by figure below.

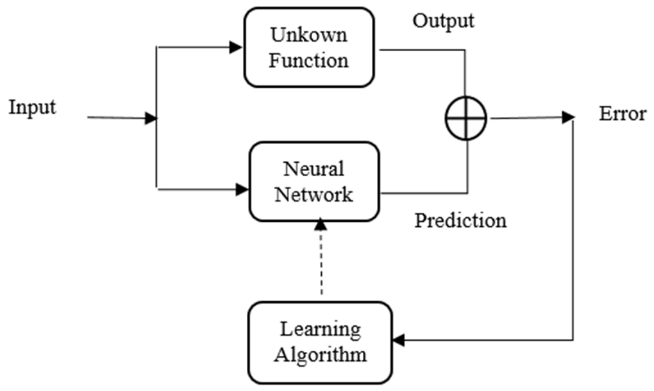


Figure 40 Neural Network as Function Approximator

The model predictive control method is based on the receding horizon technique. The neural network model predicts the plant response over a specified time horizon. The predictions are used by a numerical optimization program to determine the control signal that minimizes the following performance criterion over the specified horizon.
The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of that minimize, and then the optimal is input to the plant.

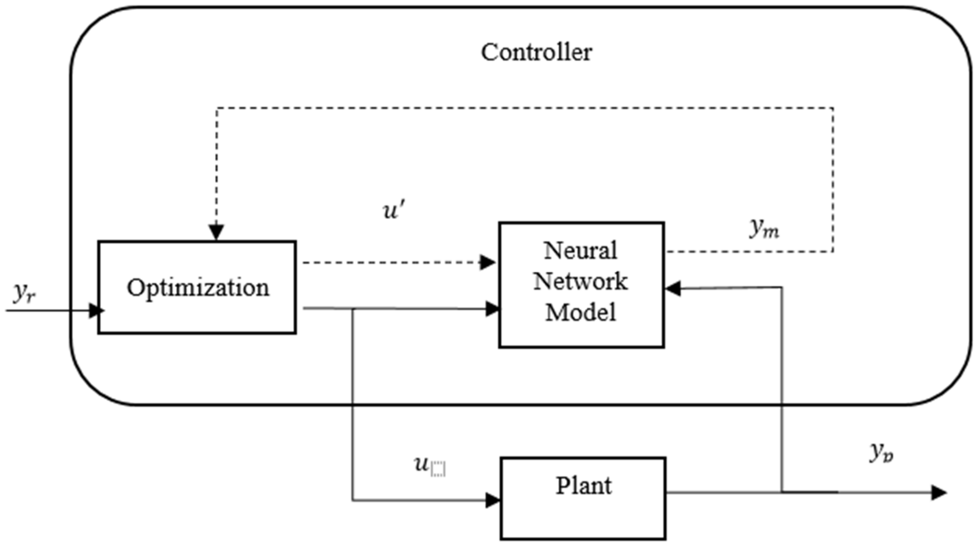


Figure 41 Neural Network Predictive Control

5.1.1 NARMA-L2 Controller

This neural network controller is referred to by two different names: feedback linearization control and NARMAL2 control. The central idea of this type of control is to transform nonlinear system dynamics into linear dynamics by canceling the nonlinearities [15]. This section begins by presenting the companion form system model and demonstrating how a neural network can be used to identify this model. Then it describes how the identified neural network model can be used to develop a controller.
As with model predictive control, the first step in using NARMA-L2 control is to identify the system to be controlled. The. The NARMA-L2 approximate model is given by:

14) $y(t+1) = f[y(t), u(t)] + g[y(t), u(t)].u(t+1)$

This model is in companion form, where the next controller input is not contained inside the nonlinearity. Below figure shows the structure of a neural network NARMA-L2 representation. The advantage of the NARMA-L2 form is that you can solve for the control input that causes the system output to follow a reference signal. The resulting controller would have the form:

15) $u(t) = \frac{y_r(t+1) - f[y(t), u(t)]}{g[y(t), u(t)]}$

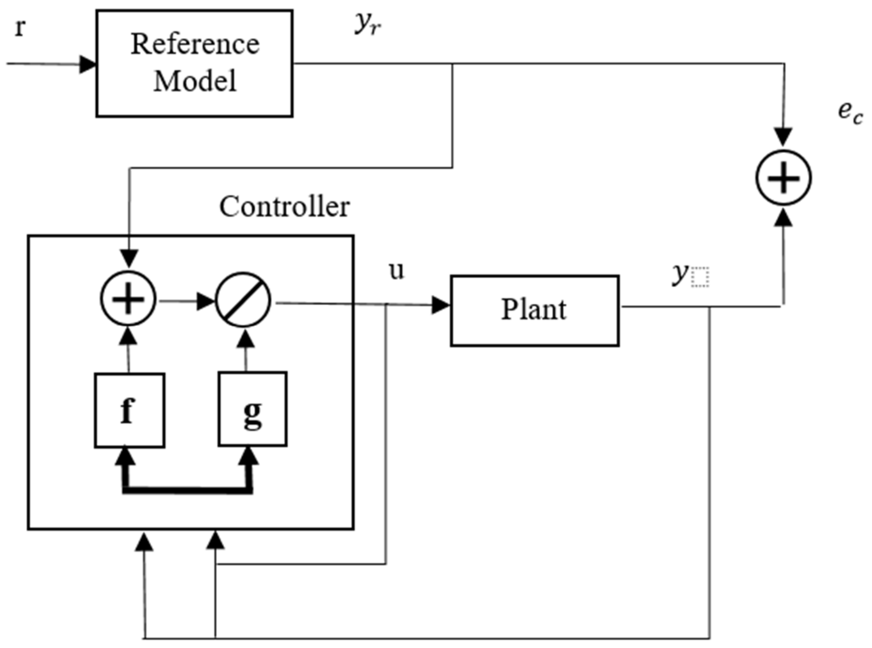


Figure 42 Narma-L2 Control Layout [15]

As it can be seen unlike the previous method, here a neural network is trained that computes the required inputs to meet the desired outputs. So the neural network function does the task of controlling the system alone. In order to train this neural network we need to either have a mathematical physics based model and use it to generate the required training data or we need to train a seperate neural network that works as a function estimator for the system. This neural network then will be used to generate the requited data for training the contoller.

The drawback is that the controller here can only be trained for a single input single output controller. A multipe input multiple out system which is our case can not use this approach because then there is probably no unique set of inputs that produces the desired outputs. So for our specific case with a complex air handling unit with many number of inputs and outputs where the coupled dynamic of the system should be considered the model based predictive controller is the best option because this controller uses an optimization algorithm to find the optimum set of system inputs that meets the desired outputs considering defined weight factors for all outputs.

5.2 Developing a two layer cascade model based control system

In the first project there was one air handling unit involved in our study. So the model of the unit and the building were unified such that the model predicted the hall temperature and humidity by receiving control signals and outside disturbance as inputs.

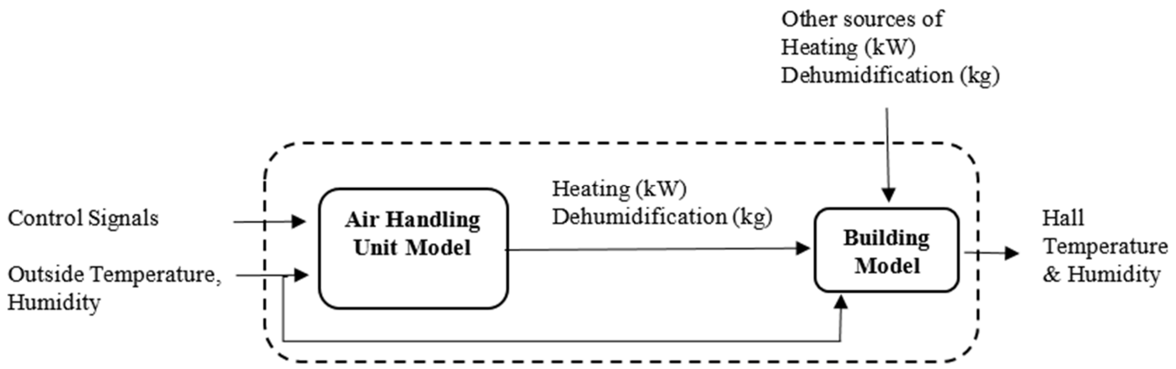


Figure 43 The unified model used in the controller

The controller algorithm was formulated such that it calculated the optimal control signals to meet the desired hall temperature and humidity while using minimum energy consumption possible.

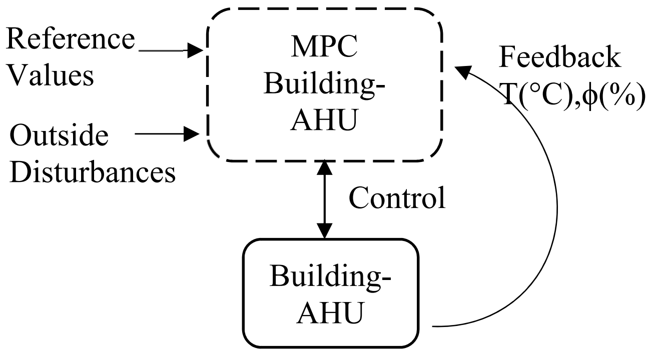


Figure 44 MPC with Unified Building-AHU Model

The drawback with this control design is that when more than one unit are in operation for conditioning the hall the bilateral effect of the units is not considered and controllers can go unstable. On the other hand the controller does not have information about the future moves of the other units. So there could be also some occasions that the units work against each other. So in the best scenario all heating devices contributing in conditioning the hall should be included in the control algorithm where an upper layer controller does the task of distributing the load among them.

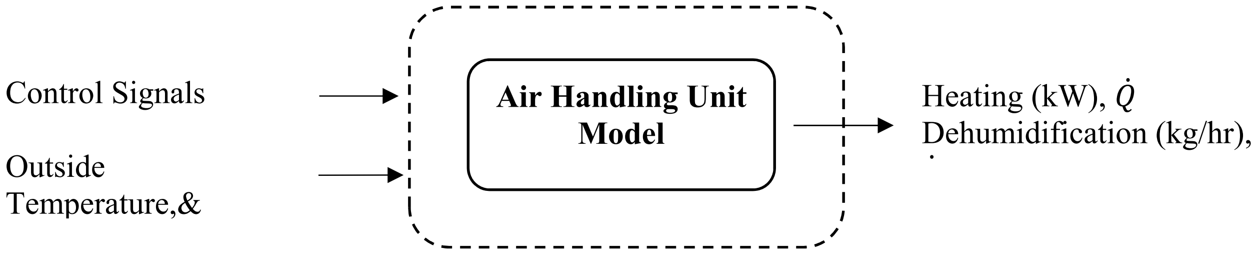
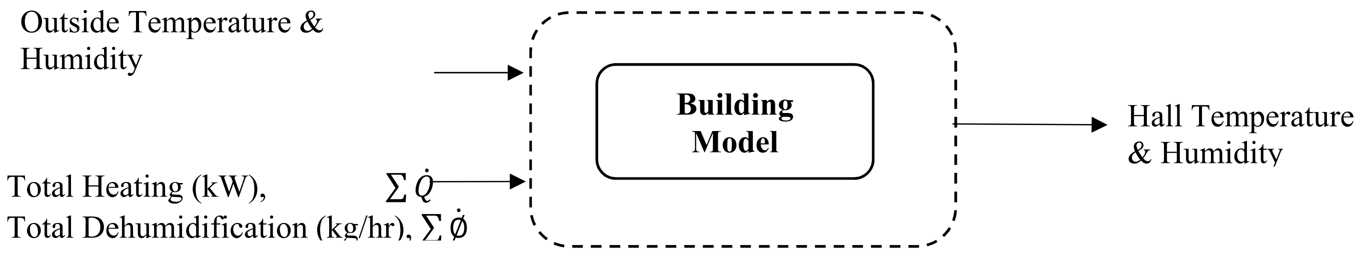


Figure 45 Unit model used in lower level



Outside Temperature & Humidity
Building Model
Hall Temperature & Humidity
Total Heating (kW), $\sum \dot{Q}$
Total Dehumidification (kg/hr), $\sum \dot{\phi}$

Figure 46 Building model used in Supervisory Level

As depicted above in the two layer controller the unit model calulates the amount of heating and dehumidification fort hat specific unit and the building model calculates the hall temperature and humidity by taking into account the total heatings and dehumidifications from all devices conditioning the hall.

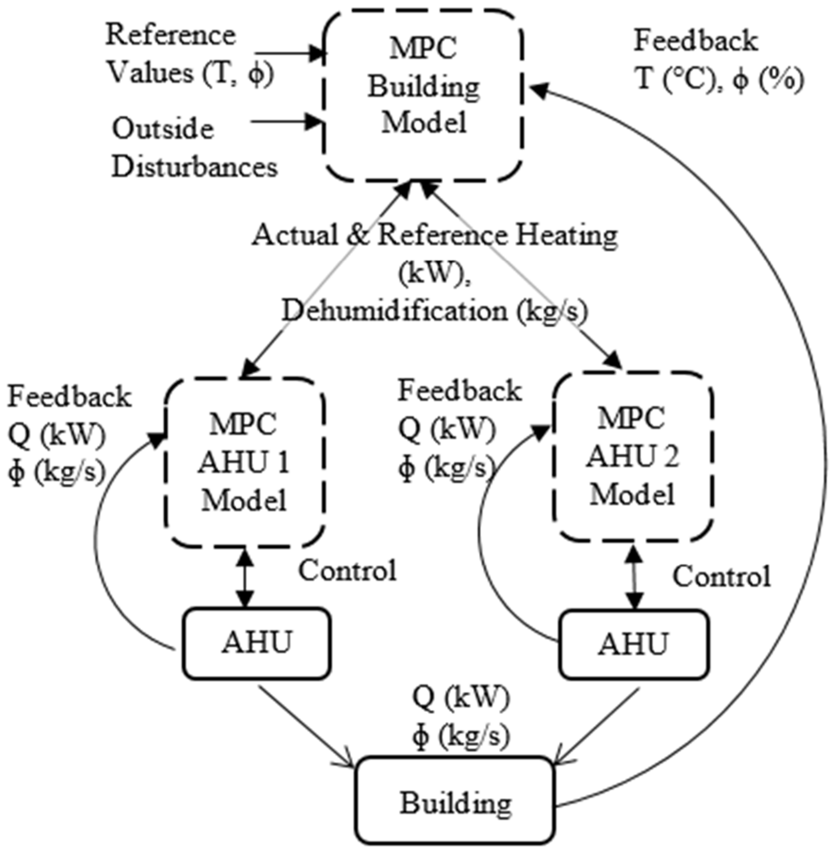


Figure 47 Two Layer MPC Algorithm

In the system above, each AHU can be stopped or started independently or its model can be updated without interrupting the whole system. For the supervisory level, we outline a MPC algorithm that outputs the required hall heating and dehumidification loads to achieve the desired comfort condition based on the current indoor temperature and humidity and outside weather condition and number of people present in the hall. The prediction time horizon is one hour and is divided in to 60 steps of one minute. The program runs in real time computing the new values of the required load every one minute.

5.2.1 Formulating the supervisory model based predictive controller

The model based controller for building layer is formulated based on the neural network function estimator described before. The controller has the following inputs and outputs:

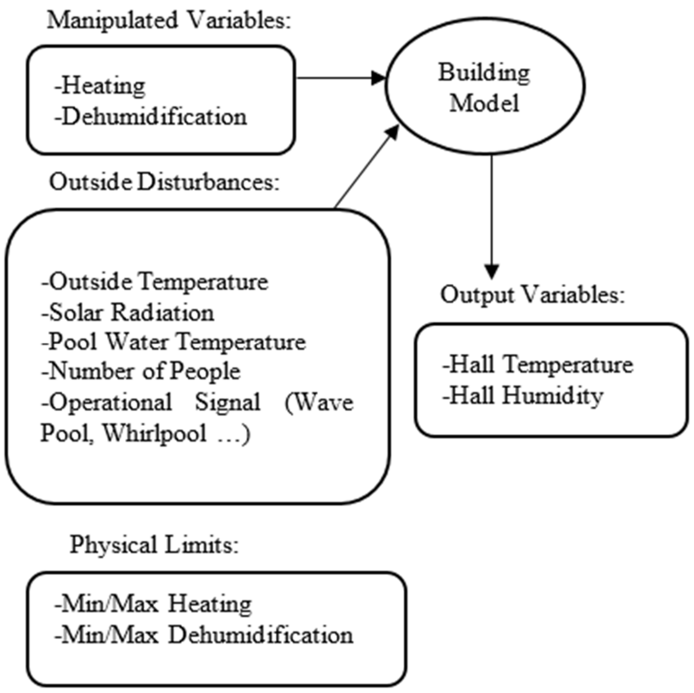


Figure 48 Building MPC Layout

The minimum and maximum limits of heating and dehumidification are calculated based on the AHU capacities and outside weather condition. This way the controller has a realistic insight of the maximum amount of heating and dehumidification that it can demand from the AHUs within the next one hour.

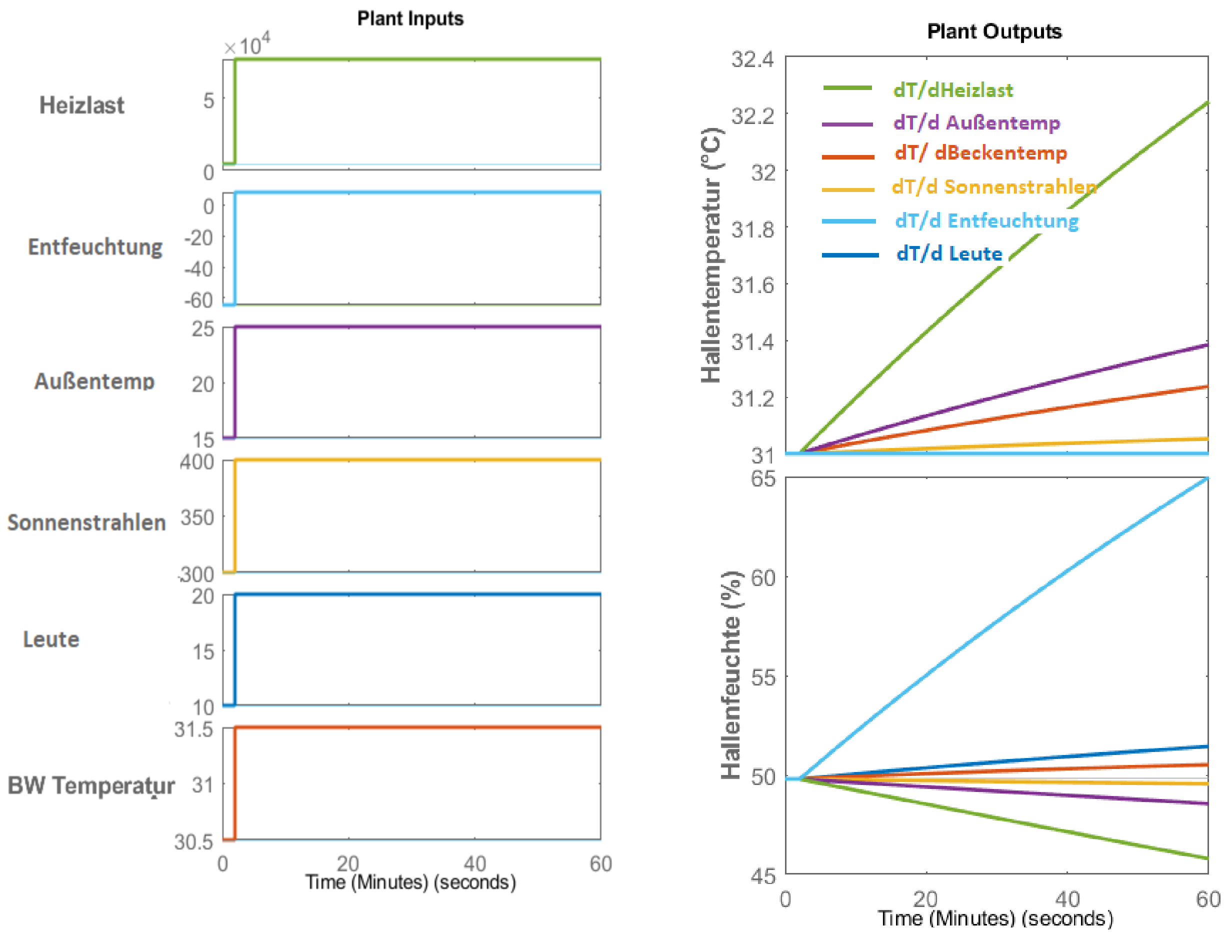


Figure 49 Open loop step response building model

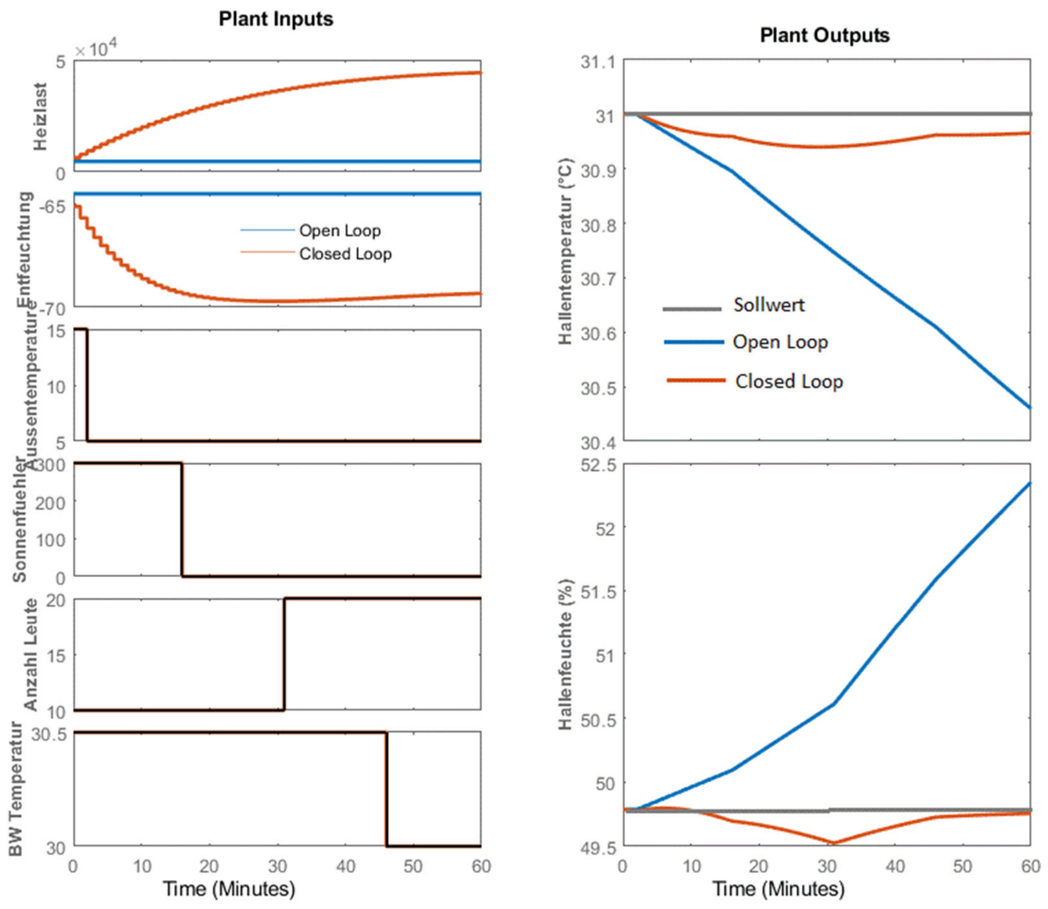


Figure 50 Closed loop step response, Building model

Figure 49 shows the building model step reponse for each of the input disturbances to the system and figure 50 shows the closed loop reaction of the controller to the inputs disturbances and computes the optimal moves to track the reference values for thermal comfort in the hall.
As the pictures show in the open loop response an increase in heating reduces hall humidity while increases hall temperature as expected.
A decrease in dehumidification increases hall humidity and has no effect on hall temperature. Here dehumidification values are considered negative so more negative values mean higher dehumidification.
The third step input shows the response to outside temperature increase which increase hall temperature and decreases hall humidity.
An increase to the number of people present in the hall from 10 to 20 rasies hall humidity from 50% to 65% within one hour and a one degree rise in pool water temperature raises both hall temperature and humidity which makes sense because the higher the pool water temperature the higher the rate of surface evaporation rate.
Figure 50 shows the optimal control moves calculated by MPC in order to keep hall temperature and humidity on the reference values in accordance with outside disturbance variations.

5.3 Evaluating Energy Consumption Optimization and Comfort Condition

We evaluated the performance of our control system by comparing to our existing PLC control. Obviously for the control pattern changes by weather condition throught the year so for having a realistic comparison we set the control system to switch between MPC and PLC automatically on a daily basis. We collected data within this period for over one year and analysed the results. The reference values for temperature and humidity are set by our PLC program so both MPC and PLC have the same reference values for comfort condition, minimum ventilation and minimum outside air rate. Then we compared the performance both in energy consumption and comfort condition to have a comprehensive comparison between the two control systems.

5.3.1 Energy Saving Swimming Hall Ramsloh

- Hauptbecken 25 x 10 m
- Babybecken
- 57 m Röhrenrutsche
- Anzahl Geräte (Bad): 2
- Gerät 1: 12.000 m^3/h mit Wärmepumpe
- Gerät 2: 10.000 m^3/h
- Modellansatz: Modellierung beider Geräte und Gebäude

- Hauptbecken 25 x 10 m

The swimming hall Ramsloh was the first swimming hall for which the model based control system was implemented. The unit is frequently switching from conventional control system in a daily basis.

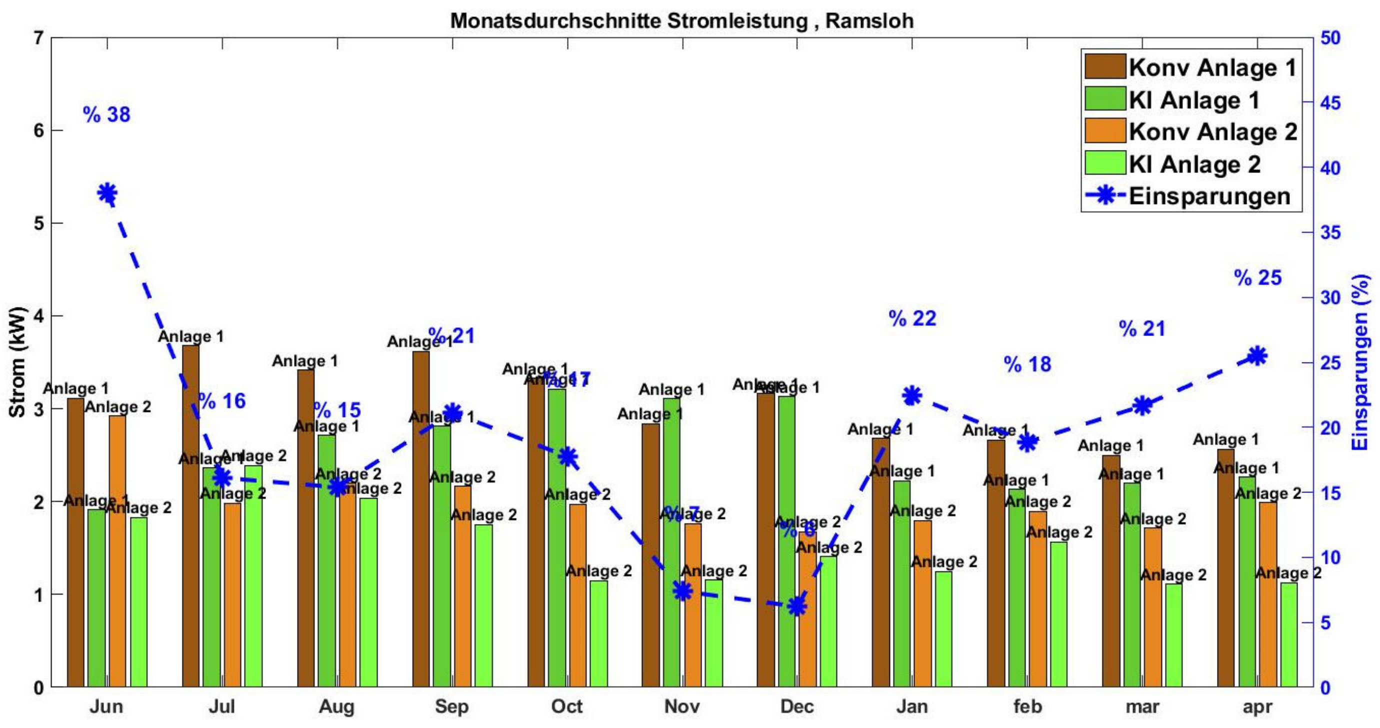


Figure 51 MPC Performance Test in Ramsloh Electrical Energy

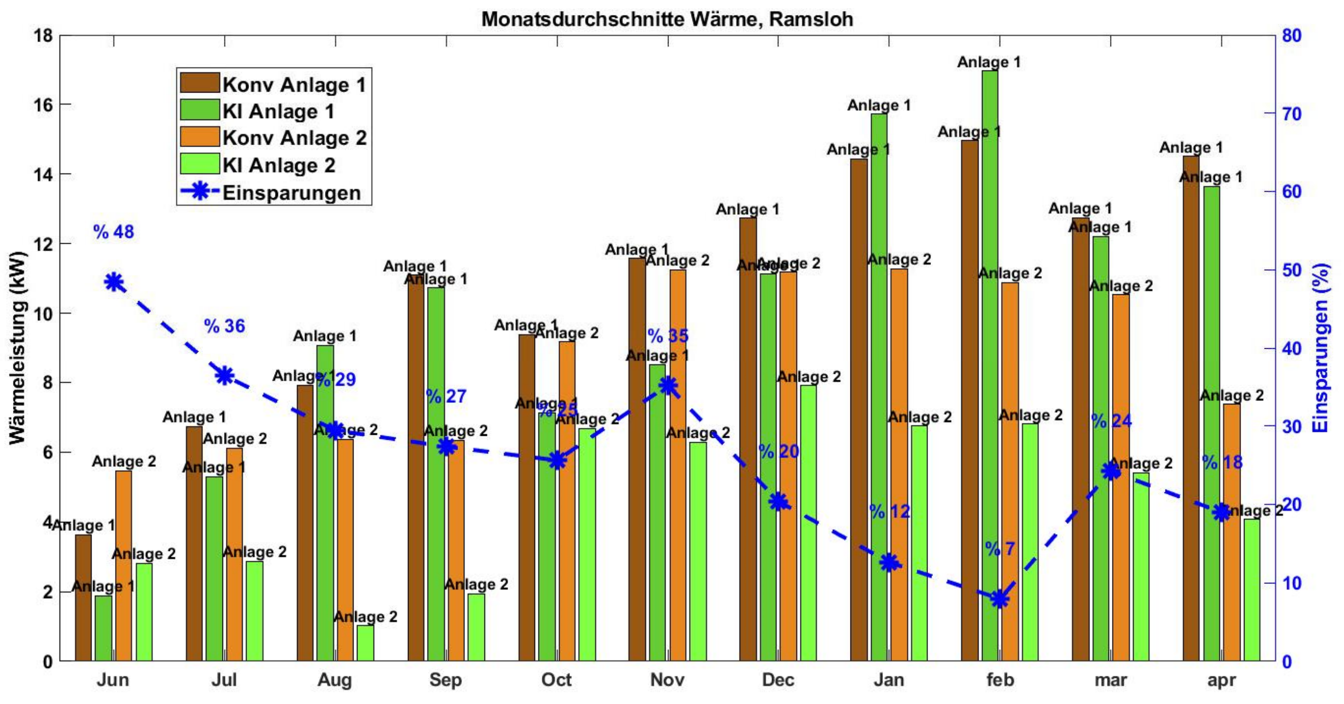


Figure 52 MPC Performance Test Ramsloh Thermal Energy

The results show around 23% energy saving in average throughout the year. The energy saving potential was more in summer because of conventional control system putting much effort in cooling in summer. As it can be seen the average electicity consumption is higher in summer because of higher ventilation rate.

5.3.2 Comfort Condition Swimming Hall Ramsloh

The index for evaluating the hall comfort condition is the reference tracking error for hall temperature and humidity. The controllers minimize the deviation from the reference value.

$$Temperature\ Reference\ Tracking\ Error = \frac{|T_{hall} - T_{ref}|}{T_{ref}} \times 100$$

$$Humidity\ Reference\ Tracking\ Error = \frac{|\emptyset_{hall} - \emptyset_{ref}|}{\emptyset_{ref}} \times 100$$

As introduced eralier the supervising control layer which is a MPC with the building model is responsible for maintaining the comfort condition in the swimming hall. The cost function for the model predictive cntroller of the supervising layer contains two terms for temperature and relative humidity and tries to minimize the devation from the references at the same time.

$$\sum_{i=0}^{p} = \left[\boldsymbol{w0} \times \left(\boldsymbol{T_{hall}} - \boldsymbol{T_{ref}}\right)^2_{\ i} + \boldsymbol{w1} \times \left(\boldsymbol{\varphi_{hall}} - \boldsymbol{\varphi_{ref}}\right)^2_{\ i} \right]$$

Graph below shows the evalution of the comfort condition in the swimming hall Ramsloh. The graph compares the reference tracking error for the hall temperature and humidity in conventional and predictive control mode.

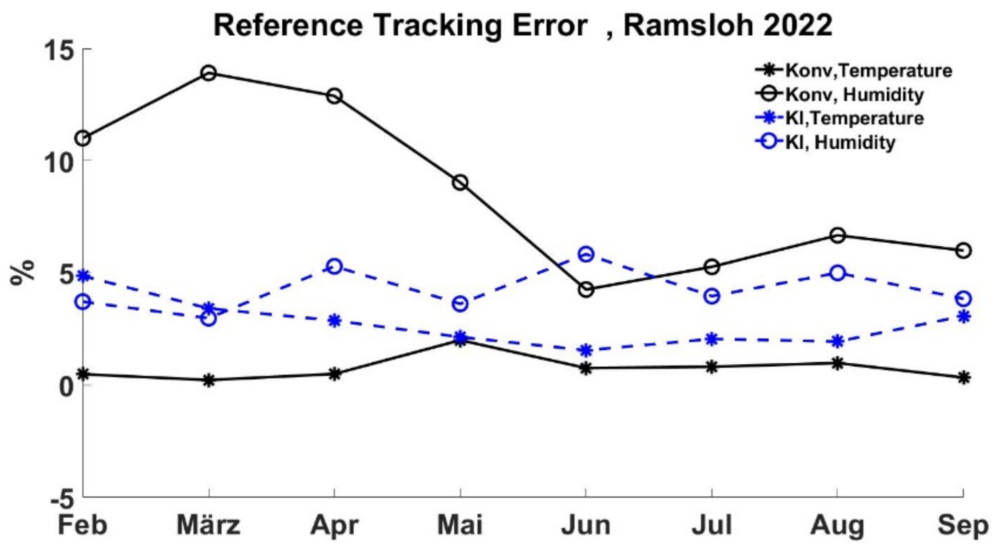


Figure 53 Comfort Condition Test Results

As depicted in the picture the conventional conrol system has a higher error in tracking the reference value of the hall humidity and lower error in tracking the hall temperature comparing to predictive controller. Good “comfort feeling” of visitors is mainly dependent on humidity, so the better humidty control is beneficial for the comfort. The Ramsloh pool manager gives us the same feedback about better comfort with KI control.

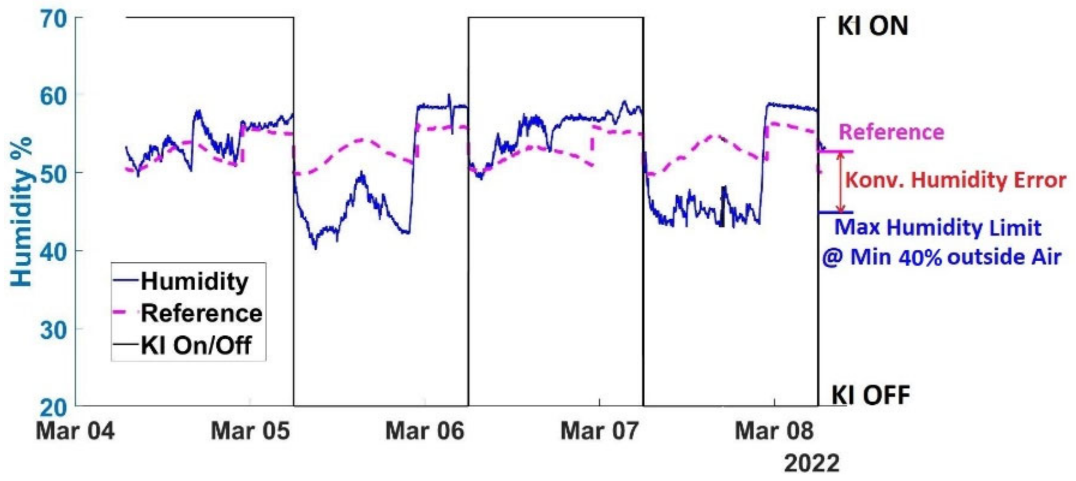


Figure 54 Hall Humidity

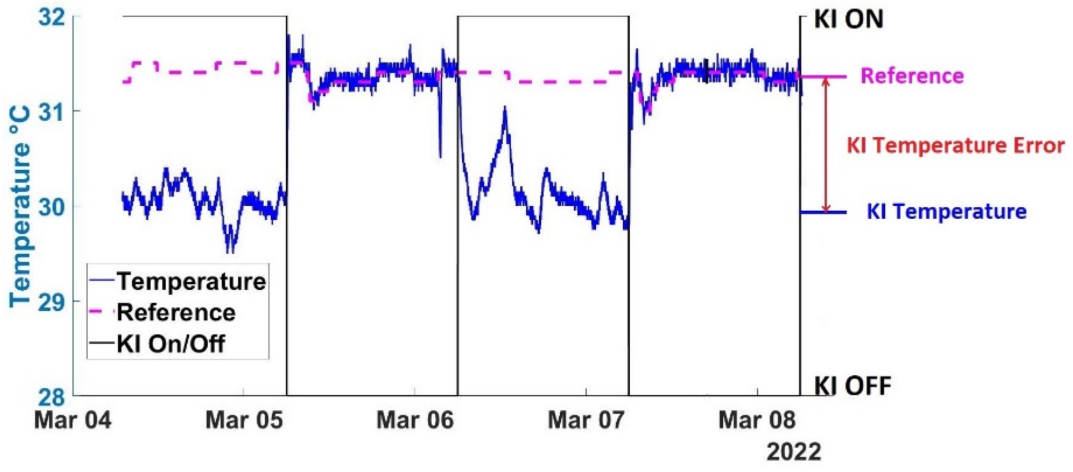


Figure 55 Hall Temperature

Taking a closer look in the month March with greater devations from the referene value it can be seen that in conventional control mode the controller sticks to the hall temperature reference value of 31.5°C but fails to maintain the hall humidity reference value of 55%. This is because of the physical limit of minimum 30% outside air that is amposed on the system. In this case in order to minimize the objective function which is the summation of the tracking errors of temperature and humidity, the predictive controller keeps the hall temperature a bit lower at 30°C so increasing the error at temperature and compromises at temperature reference tracking so that hall humidity can be kept near 55%. This helps also to save energy without compromising or even improving comfort.

This behaviour depends on the currently adjusted MPC tunning parameters (objective weight and scale factors). It is obvious that increasing the weight factor fort he hall temperature would result in a similar control behaviuor as the conventional control system.
It is important here to note that the hall humidity is not only important as a comfort condition factor but also affects the rate of pool water evaporation. As the figure 56 shows pool water temperature and hall humidity have great effect in the rate of pool water surface evaporation. Lower hall humidity increases pool water evaporation.

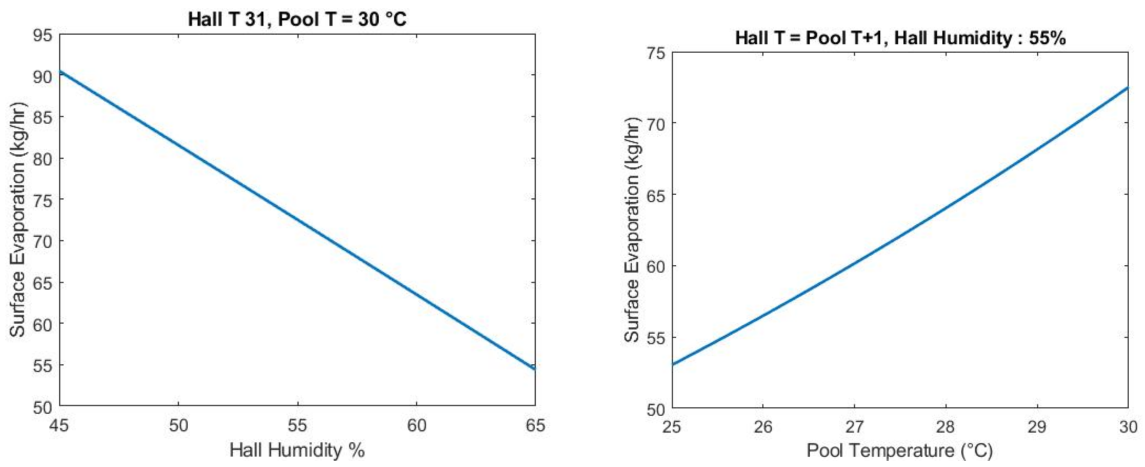


Figure 56 Pool Surface Evaporation vs Temperature

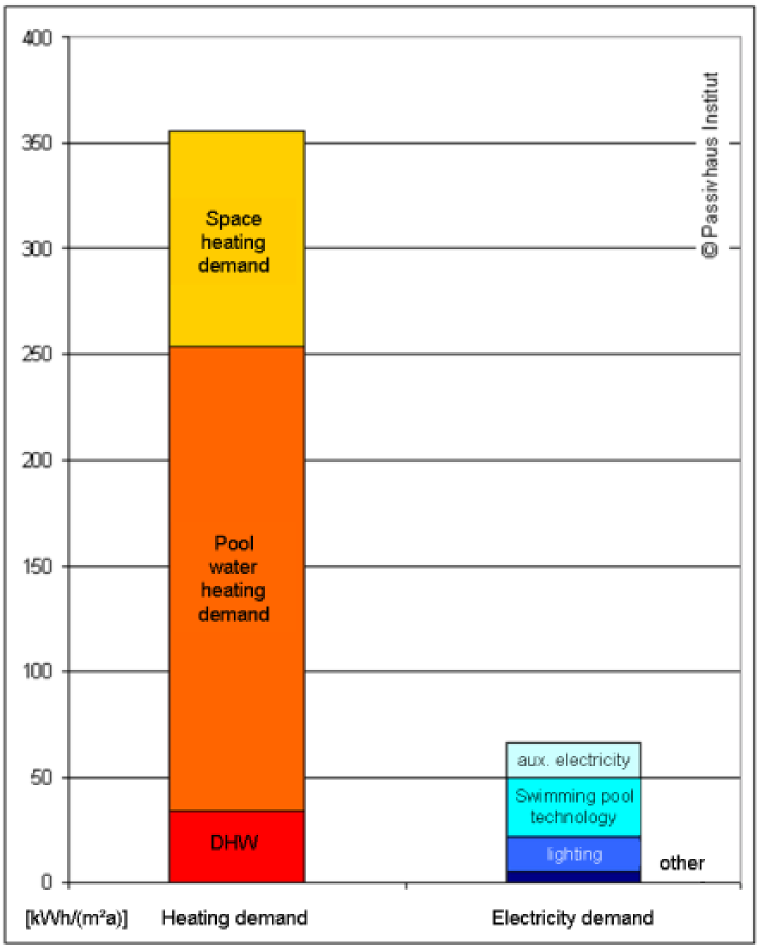


Figure 57 Energy Consumption Break down in Swimming Hall [16]

The figure above shows the breakdown of energy consumption in a typical swimming pool. As it can be seen pool water heating demand is a determining factor in energy consumption in an indoor swimming hall. Higher hall humidites result in higher corrosions and rusting in building facilities and lower humidity results in higher evaporation rate. So maintaining the hall humidity indirectly affects the energy costs which is reflected in pool water heating and we can not measure that because pool water heating is usually provided by other companies than air conditioning and ventilation systems.

5.3.3 Comfort Condition Swimming Hall Bad Zwischenahn

In swimming hall Bad Zwischenahn we used the same model trained in Ramsloh but the building is not equipped with people counter and solar radiation sensor. Here the number of people and solar radiation will be treated as unmeasured disturbances. These are independent inputs of which the controller has no direct knowledge, and for which it must compensate.

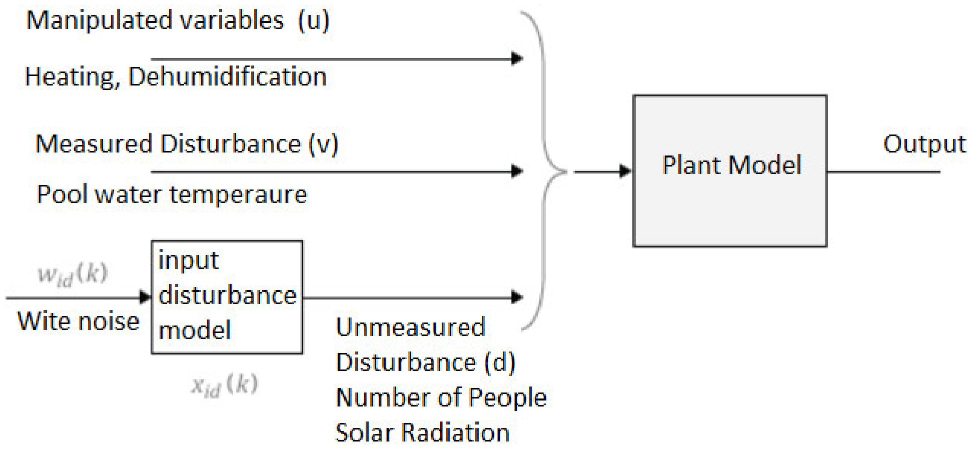


Figure 58 Disturbances at Plant Input

The input disturbance model is a key factor that influences the following controller performance attributes:

- Dynamic response to apparent disturbances — the character of the controller response when the measured plant output deviates from its predicted trajectory, due to an unknown disturbance or modeling error.
- Asymptotic rejection of sustained disturbances — If the disturbance model predicts a sustained disturbance, controller adjustments continue until the plant output returns to its desired trajectory, emulating a classical integral feedback controller.

Figure below shows the reference tracking error in the swimming hall Bad Zwischenahn. We observe similar pattern to that of swimming hall Ramsloh but with higher tracking erroros comparing to ramsloh. This can be attributed to the unmeasured distrubances in Bad Zwischenahn for number of people in the hall and solar radiation and also model imprecisions. But still the predictive controller has a better performance comparing to conventional control system.

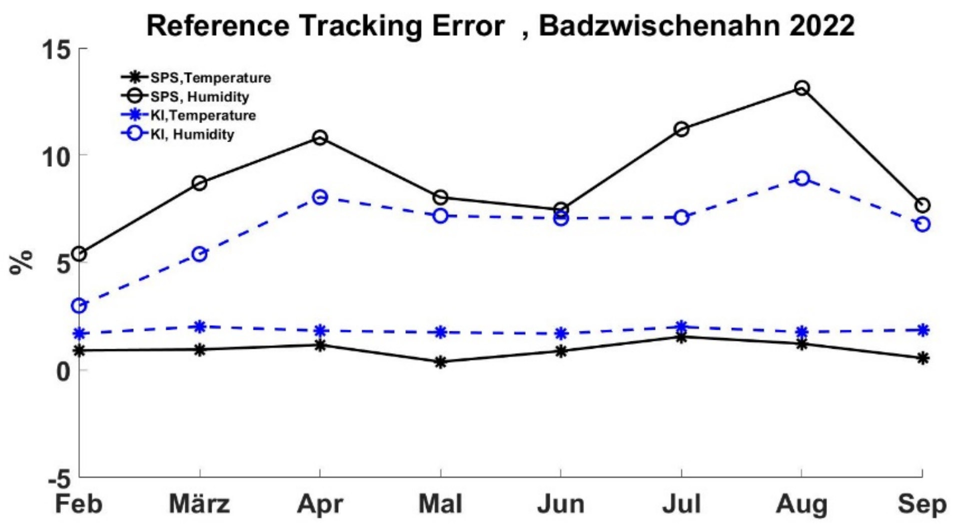


Figure 59 Reference Tracking Error Bad Zwischenahn

We have recently equipped swimming hall Bad Zwischenahn with people counter and solar radiation sensor to update the building and see how the performance of the control system changes. At the time of writing this report the swimming hall is not in operation and we lack enough data for that.

6. Cloud Connection

Cloud connection can benefit the field of machine learning by providing flexible and cost-effective infrastructure for developing, training, and deploying machine learning models. Here are some key applications of cloud connection in machine learning:

- Data Storage: Cloud storage solutions allow it to store and manage large datasets efficiently. Data can be stored in cloud-based storage systems, making it accessible from anywhere and facilitating collaboration among team members. This can be used in collecting data of fault detection, predictive maintenance of our AHUs or digital twin model training.
- Data Preprocessing: Before training a machine learning model, data often needs to be preprocessed, cleaned, and transformed. Cloud platforms offer tools and resources for data preprocessing, making it easier to handle these tasks at scale.
- Model Deployment: Once a machine learning model is trained, it needs to be deployed to serve predictions in real-world applications.

In this section we evaluated and studied the possibility of an all automated training and updating the model based predictive controller using cloud connection. The idea is that data is collected locally and is processed and used for training a new model on site. The training data are then sent to a cloud for further processing. The main challenge here is to develop an online training application that can process and train models online.

6.1 State Space Neural Network

One option for automatic online model adaptation is the so called model-free methods where we do not have a dznamic model.

In our classical method the dznamic model goes through lineariyation and the linearization process needs a numerical perturbation algorithm that sends numerical perturbation at the inputs of the model and measures the outputs. In this case if new set of data are collected and we want to update the mpc object the whole process should be repeated. One alternative meathod to create an mpc object only once and keep it and in case of updating the mpc object only train a state space neural network instead of a dynamic model. This would remove the dynamic model and bypasses the linearization process so the mpc controller can be updated whenever enough set of new data is collected making it a really automated self learning controller.

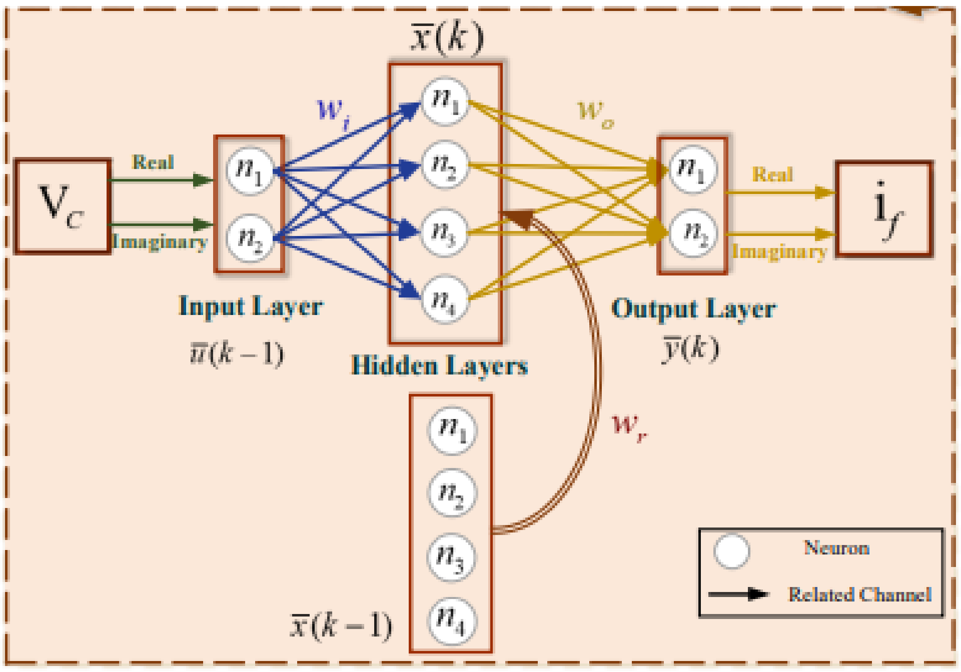


Figure 60 Proposed Neural Network Architecture [17]

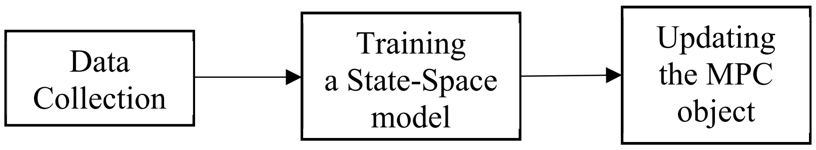


Figure 61 State Space Neural Network

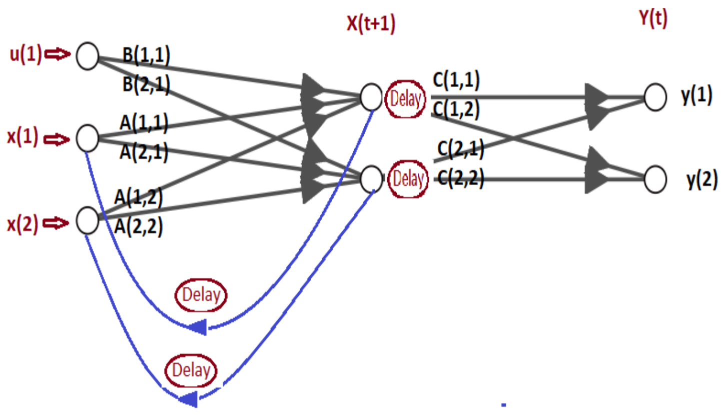


Figure 62 State space Neural Network Algorithm

Figure above shows a neural network structure that calculates the state space formulation of a set of linear partial differential qeuations

$$x(t+1) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

The differences between a SS-neural network and a regular feed forward network is that in training the feed forward network we are intrested in how well the neural network mocks the real system regardless of the neural network structure but in SS-neural network the structure of the neural network must be in a way that if the weights of the network are equal to state space matrixes the network can predict the system outputs.
So this should be a recurrent neural network that recurrents the outputs from the hidden layer to inputs. The number of hidden layers equals the number of states and the number of output layers equals the number of outputs.

6.2 Differences between Simple Neural Network and SS neural network

There are a couple of differences between a regular neural network and a state space neural network:

- Preprocessing function
  In regular neural network we apply preprocessing function to normalize the inputs but in state space neural network the normalization process is omitted so the neural network receives the actual inputs
- Bias
  As it is seen in figure 60 the state space neural networks function only as an operator that multiplies U and X matrices with A,B,C state space matrices. So the bias values are dropped here
- Activation Function
  In Regular neural network activation function maps the neuron output between -1 and 1 or 0 and 1while in state space neural network we are interested in the product outputs of matrices themselves. So there is no need to apply activation function.
- Number of neurons
  Unlike regular neural networks where the number of hidden layer neurons can be selected as much as needed in state space neural networks that must be equal to the number of states. So there we have a limit in selecting the number of neurons

6.3 Training an SS neural network for spring mass damper problem

The method is first verified using a simple spring mass damper problem. For this reason two neural networks are trained once as a regular feed forward neural network and second one as an SS neural network as described above. The results can be seen in **Fehler! Verweisquelle konnte nicht gefunden werden.**. As the figure shows the outputs from the Simulink dynamic model and the two neural network models match pretty well. This proves that our setup for SS neural network is valid and works for mass spring damper model.

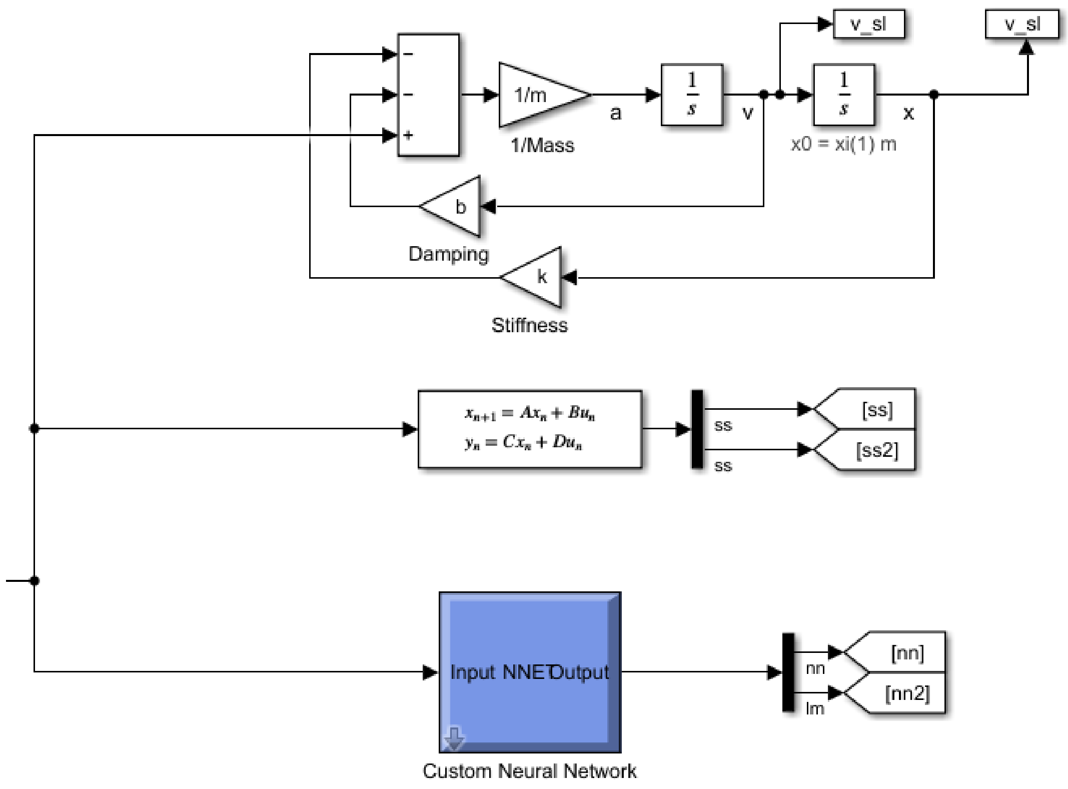


Figure 63 Simulink Setup for Spring Mass Damper

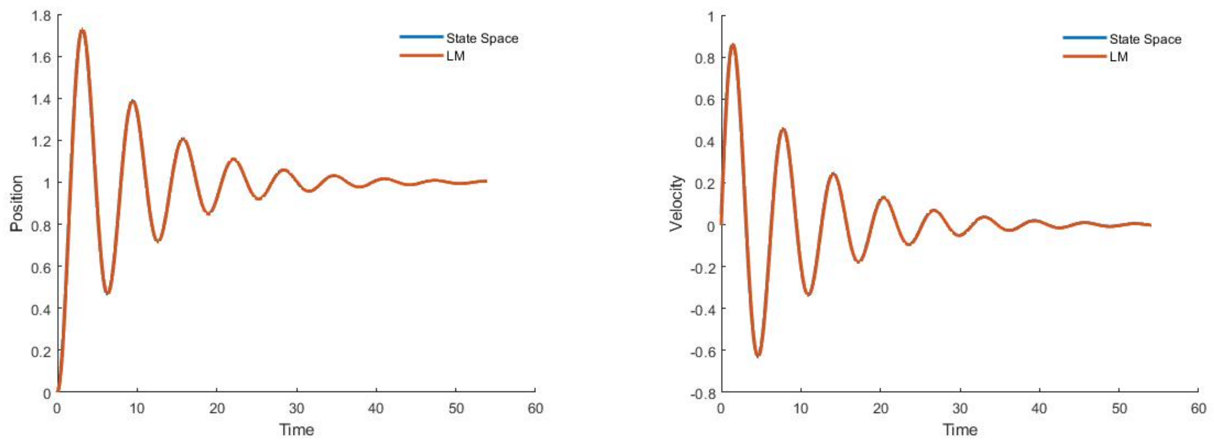


Figure 64 Regular Neural Network vs SS Neural Network

6.4 Training a SS neural network for Building Hall Temperature

In the mass spring damper we used the levenberg-marquardt algorithm as usual to train the neural network. Although this algorithm uses basically newton method to find the local minimums in this case we used PSO algorithm to train the SS neural network. In PSO algorithm the search area would be a multidimentional area whoes dimensions are the number of unkowns (here the number of weights. A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space. The movements of the particles are guided by their own best-known position in the search-space as well as the entire swarm's best-known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. This method is widely used in neural network training.

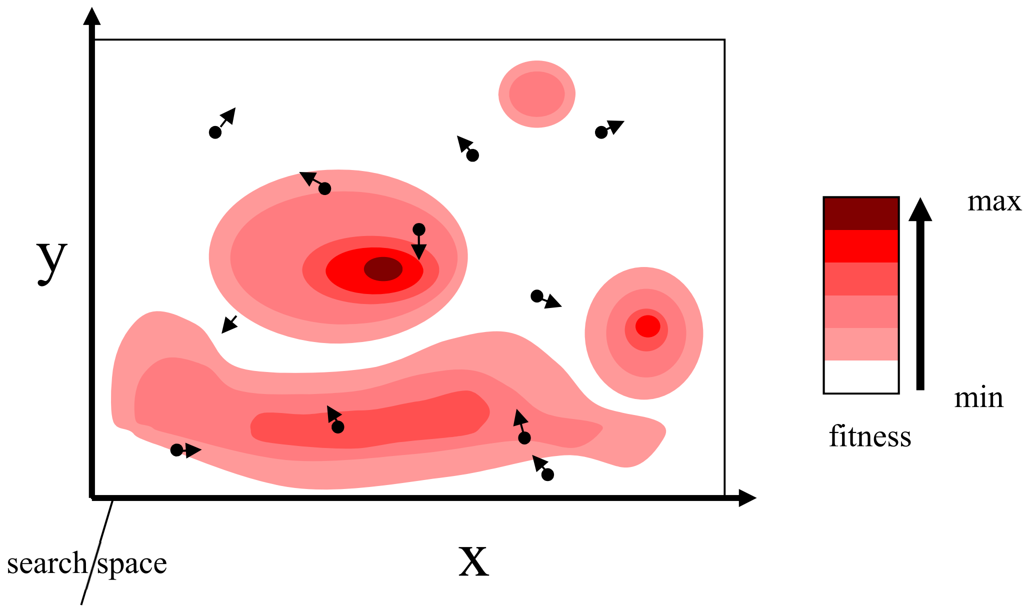


Figure 65 2D search area and particles in PSO method

We tried both LM and PS training methods to train an SS neural network for the building model but got poor results.

The challenges in training a neural network for the building model in general:

- Noise in Data

  The data always contains noise depending on the type of sensors we use. In order to avoid it affect our training we use different filters to remove noises such as median filter or moving average filters

- Phase difference between input and ouput data
  For the building model we observe that it takes a while until any change in input data is reflected in output data. This is eather because of the size of the hall and inhomogenity of temperature and humidity of the hall that any change in temperature and humidity needs some time to show itself in the overall average hall temperature and humidity. And secondly we measure hall temperature and humidity at the return duct at the AHU. So it takes some time until the air travels through the return duct and reaches the measurement point. For this reason we apply moving averages to shift the inputs and outputs in such a way that this delay in outputs does not affect the training

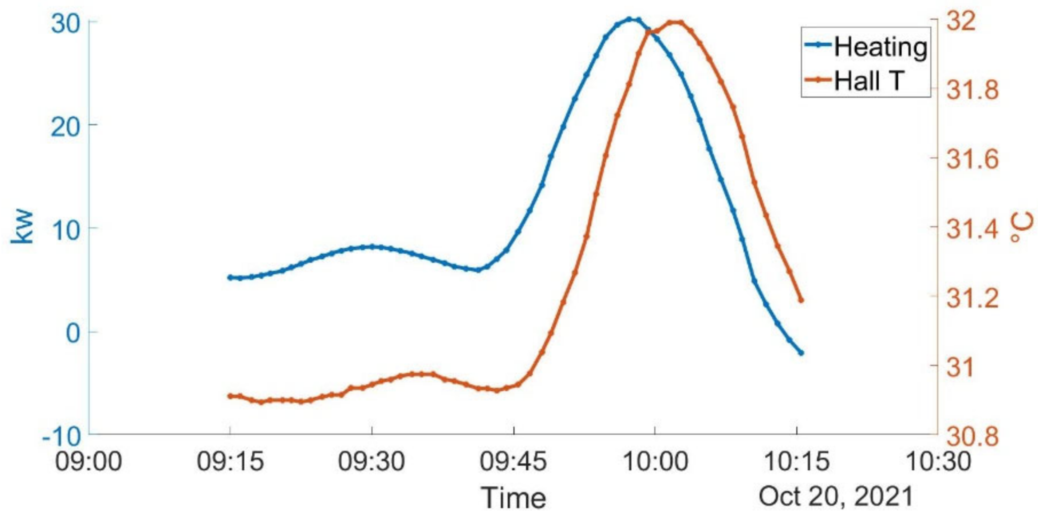


Figure 66 Input / Outputs 5 mintes delay

As in SS neural network the weight of the neural network are actually the A,B,C matrixes of the physics based model. Ghe elements of these matrixes are actually the physical properties of the building. So we can initialize the training process using the physical properties to speed up and improve the learning process.
But at the end this method did not end up in good results specifically for hall humidity. The reason is that SS neural network works only for the problems who have linear character. It means the state space matrices must be valid in all range of inputs and outputs such as the spring mass problem. For the nonlinear problems such as hall humidity or the AHU model this is impossible to find a set of A, B, C, D matrices which are valid in the whole range of training data and that is why we actually apply adaptive MPC with variable state space matrices.

6.5 Cloud connection

Cloud connection refers to the process of connecting to and accessing resources and services hosted on cloud infrastructure over the internet. The cloud is a network of servers, storage devices, and other computing resources that are accessible over the internet, and cloud connection allows users to access and utilize these resources from anywhere in the world, if they have an internet connection.
Cloud connection provides users with the flexibility and scalability to access computing resources and services quickly and easily on demand. It allows businesses and organizations to

save on infrastructure costs and eliminates the need for on-premises hardware and software installations, as all resources are hosted and managed by the cloud service provider.
Cloud connection can be achieved through various methods, including virtual private networks (VPNs), direct internet connections, and software-defined wide-area networks (SD-WANs). With the increasing adoption of cloud computing, cloud connection has become an essential component of modern-day IT infrastructure and is critical for businesses and organizations to stay competitive in today's digital age.

Raspberry Pi can be used as an MQTT (Message Queuing Telemetry Transport) broker. MQTT is a lightweight, open-source protocol that is commonly used in Internet of Things (IoT) applications to send and receive data between devices.

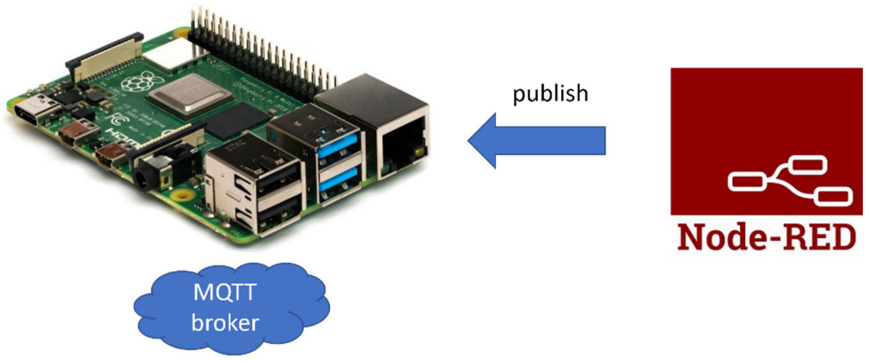


Figure 67 Setup of IO broker on Raspberry Pi

Setting up a Raspberry Pi as an MQTT broker involves installing an MQTT broker software like Mosquitto on the device, which acts as a message broker that receives and distributes messages between connected MQTT clients. The Raspberry Pi can then be used to connect IoT devices, sensors, and other clients to the MQTT broker, allowing them to communicate with each other using the MQTT protocol.

6.6 Online Training

Traditional machine learning techniques run in batch mode, where the complete training data is fed in advance to train a model by applying certain algorithms. Such an approach requires entire training data available prior to the learning task and the process is also in offline mode due to expensive training costs. Conventional techniques suffer from some critical drawbacks like low efficiency in both time and space cost; and poor scalability for large-scale applications because the model often has to re-train from scratch for new data.

Online machine learning model retraining involves updating a machine learning model continuously as new data becomes available. This process of retraining allows the model to adapt to changing conditions or new patterns in the data.

In online learning, the model is initially trained on a small subset of the data, and as new data becomes available, it is used to update the model's parameters. The model is then tested on a separate validation set to ensure that it is still performing well.

Machine learning models are trained on a specific set of data to make predictions or decisions. However, over time, the data distribution may change, or new data may become available that was not present during the initial training phase. As a result, the performance of the machine learning model may degrade, and it may no longer make accurate predictions. This problem of the changing underlying relationships in the data is called concept drift in the field of machine learning.

There are many ways to address concept drift; the most common way is to periodically update your model with more recent historical data. Online machine learning is a type of machine learning that allows a system to continuously learn and improve from new data as it becomes available, so that it can adapt to changes in the underlying data distribution and continue to make accurate predictions. This process can be done periodically, depending on the frequency and magnitude of changes in the data.

Online learning in machine learning has several advantages, including the ability to handle large and constantly changing datasets, adapt to new data quickly, and work well in real-time applications. However, it also has some unique challenges that must be addressed.

Here are some of the challenges associated with online learning:

- Data Quality: Online learning algorithms require high-quality data to produce accurate predictions. However, when dealing with large and constantly changing datasets, it can be challenging to ensure the quality of the data. Therefore, it is essential to develop robust data pre-processing techniques and quality control measures.
- Model Stability: Online learning algorithms are susceptible to sudden changes in the data distribution. These changes can cause the model's parameters to fluctuate, resulting in instability. Therefore, it is crucial to develop algorithms that can adapt to changes in the data distribution while maintaining stability.
- Overfitting: Online learning algorithms are prone to overfitting, which occurs when the model becomes too complex and starts to fit the noise in the data rather than the underlying

patterns. Overfitting can be mitigated by regularization techniques or by using ensembles of models.

Addressing these challenges is critical to developing effective and efficient online learning algorithms. Therefore, it is necessary to monitor the process throughout the time.

1.1 Online Training Results

The purpose of the app is to train a generic model that can be used to train any type of data. To evaluate the performance of the app, we developed and executed a feedforward neural network. This network was designed to predict the behaviour of the first air handling unit (AHU) in Ramsloh, based on the positions of various valves, ventilator speed, air heater and heat pump. The AHU's volume flow rate and electric consumption were chosen as the outputs for this

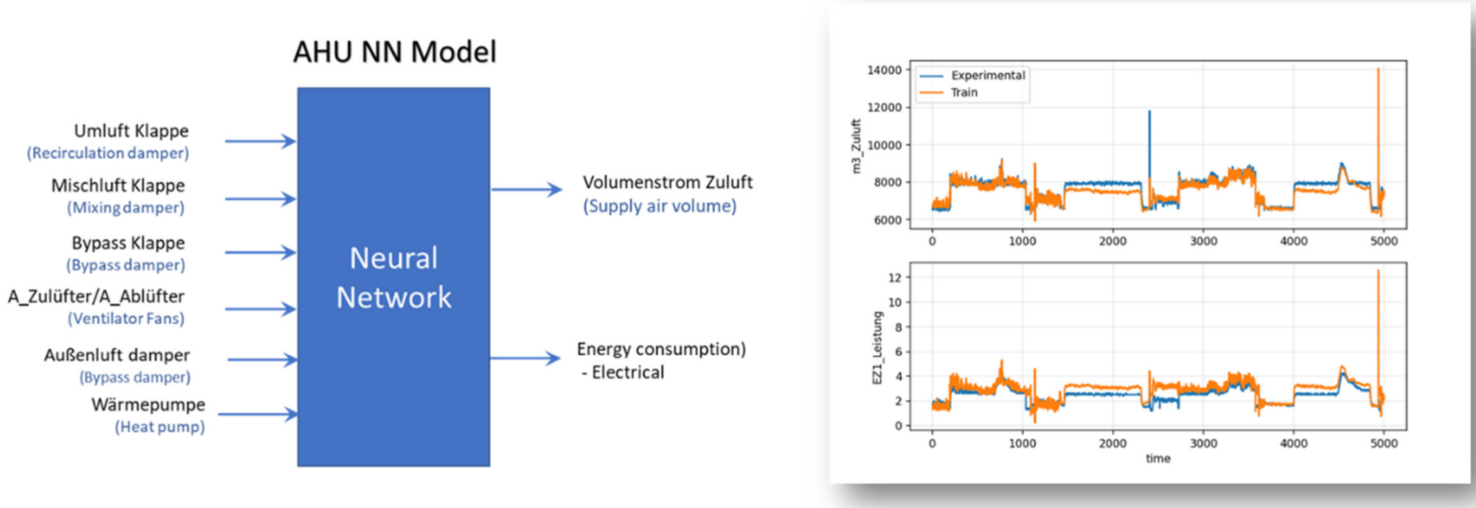


Figure 68 Actual vs. Predicted values using online training

network.

The model's hyperparameters can be configured using the function node in Node-red, as shown in the Appendix. By analysing the network's predictions, we were able to assess the effectiveness of the app in training a generic model.

Our evaluation revealed that the training procedure produced positive results. Specifically, the r2 score of the fit was 0.64. This indicates that the model was able to capture a significant portion of the variance in the data and thus make reasonably accurate predictions of the behaviour of the AHU.

To evaluate the online training capabilities of the app, we conducted a retraining exercise by sending data from Node-RED to the app. specifically, we compared the predictions of two models: the first prediction was made using the original model, while the second prediction was generated using an updated version of the model. The prediction generated by the original model showed what the model would have predicted had it not been updated. By comparing this with the prediction generated by the updated model, we were able to evaluate the effectiveness of the retraining process.

In the retraining process, we added 60,000 new data points at each iteration, while keeping the hyperparameters of the model constant. Our results indicate that the new model outperformed the old model at each iteration, as evidenced by the lower RMSE value. This suggests that the online retraining of the model led to an improvement in its overall performance. However in **Fehler! Verweisquelle konnte nicht gefunden werden.** it is obvious, that the new model prediction is on the same level as old level prediction. This is due to the fact, that the new training data is not significantly different to the old training data.

This application provides another feature, which is the ability to train a dynamic NARX

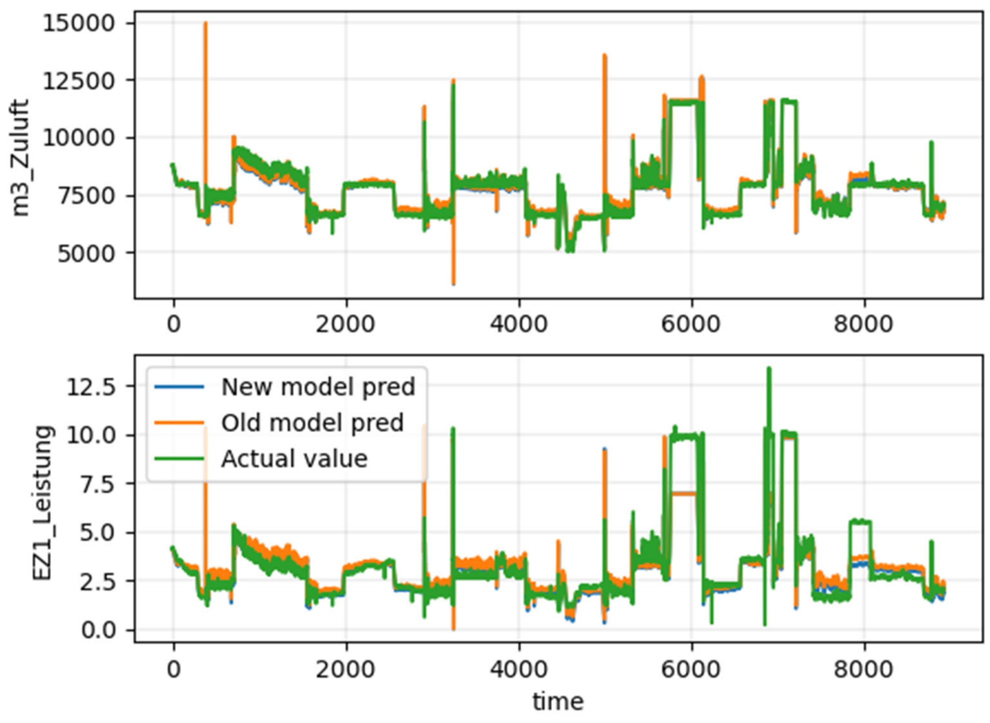
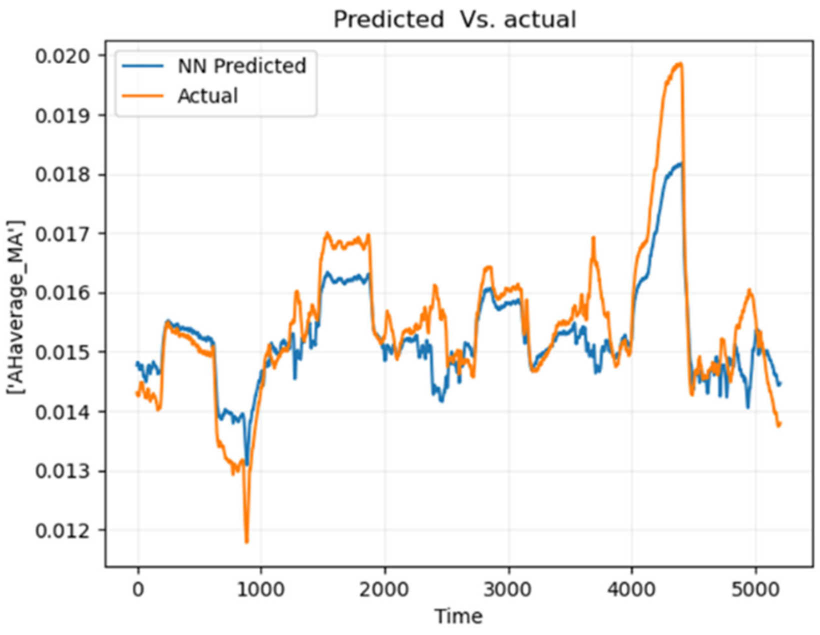


Figure 69 the actual values and predicted values using old and new updated model

(Nonlinear Autoregressive with eXogenous inputs) model. To perform NARX training in Python, we utilize the FireTS package, which incorporates a sklearn module. All the hyperparameters are defined via Node-red and the results of the training can be seen in the figure below.



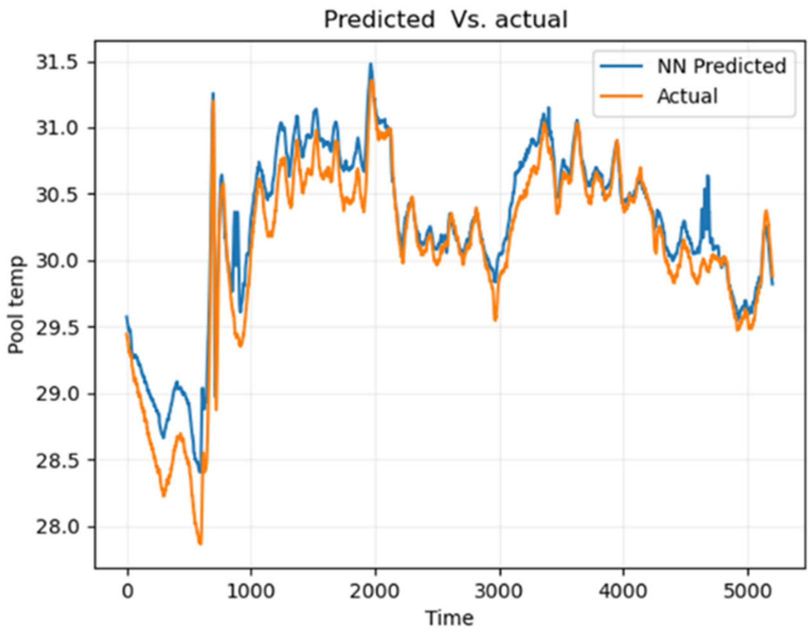


Figure 70 Predicted vs Actual value of Humidity and Temperature

Unlike the feedforward systems, online retraining is not achievable for NARX models. This is because there is currently no available package or library in Python that supports online retraining for NARX models. Therefore, implementing online retraining for NARX models is currently not feasible.

Through this, we were able to confirm that the app can facilitate online training and updating of models. This is particularly important in situations where new data is continually being generated, and the model needs to be adjusted to accurately reflect changes in the data distribution.

Using a Raspberry Pi as an MQTT broker has several advantages, including:

Low cost: Raspberry Pi is an affordable and accessible device, making it an economical option for setting up an MQTT broker.

Low power consumption: Raspberry Pi has low power consumption, which makes it ideal for running an MQTT broker without incurring high energy costs.

Flexibility: Raspberry Pi is a versatile device that can be easily customized to meet specific requirements and can be used to set up an MQTT broker that is tailored to specific IoT applications.

Scalability: Raspberry Pi can be used to set up a scalable MQTT broker that can handle many connected devices and clients.

Raspberry Pi can be used as a small-scale cloud server for personal use or for small businesses. Once set up, a Raspberry Pi can be used to store and access files, documents, and other data from anywhere in the world using an internet connection. This makes it a convenient and cost-effective solution for personal cloud storage or for small businesses that don't require the full scale of enterprise cloud services. It also provides users with greater control over their data and privacy, as they are not relying on third-party cloud providers to store their information.

Using a Raspberry Pi as a cloud server is a viable option for small businesses with modest storage needs and can provide users with greater control over their data and privacy while also being a cost-effective and energy-efficient solution.

Conclusion

In this project we used the experiances collected in the previouse project to further develop our design of model based predictive controller based on combination of neural network and physics based models.
In previous project the new control system was implemented on unit number one in swimming hall Ramsloh. In this project unit number two was also equipped with new sensors and was modelled both mathematically and data driven.
We applied more neural network strcutures such as NARX, LSTM and GRU and also more training technics such as early stopping and regularization to avoid overfitting and imroving the results. Though the results from neural network model of the AHU comparing to mathematical model shows still the physics based models are more reliable for control purposes. Because for complex and nonlinear multiple input multiple ouput models a physics based model which is based on mathematical formulation and is tunned using experimental data always gives outputs which are physically explainable. But because a neural network model has no physical formulation in background for such a complex system it can produce outputs in some certain ranges of input variables which are physically not expected and this can be because of either lack of training data in certain ranges, incapability of neural networks in interpolation and extrapolation where training data are missing, training methods applied and the neural network structures employed. In general we can say that although neural networks are powerfull tools for creating modls for functions with unkown mathematical formulations but for the systems whos physics are well know the physics based models still are more reliable.
Comparing to AHU models, the building thermal and humidity balance models are simpler in terms of linearity and fewer number of inputs and outputs. Furthermore creating a mathematical model of each individual buildings with different layouts and material properties would incur immense costs and time unlike the AHUs which all have the sample principle and specifications and only differ in size. So neural network models developed for building models are more practical here than physics based models.
In chapter three we studied transfer learning for air handling units and building models. We applied two methods; scaling up and down the models and retraining the neural networks with new data. In scaling up and down we apply some scale factors to the state space matrices correspondant to their nominal capacity comparing to the original model. These scale factors adjust the model to the new size. This is suitable for physics based models for the AHUs which have the same principle but are different in size and nominal capacity. In retraining we use the already trained building model for which we used one year of data history and retrained it with limited data collected in other swimming hall. The results show considerable improvement in adapting the model to the new data comparing to training the neural network from scratch.
In chapter four we studied different methods of model based predictive control and designed a two layer model based predictive controller in which the upper layer is a supervisory layer with building model that predicts the heating and dehumidification load demand and distributes it among the AHUs. It ensures a smooth operation of multiple AHUs with properly distributing the load among them according to their nominal capacity. Furthurmore it makes a modular

layout of the controller where different modules for the building and the AHUs can be seperatly updated.
We also made some improvements in out MPC controller to vercome some problems that we had with fixed speed heat pumps. As we know MPC controllers use optimization algorithms that find the optimums within a continuous range. With fixed speed heat pumps we had this problem that MPCs could not handle on/off functions. This problem was solved using a workaround with two parallel MPCs running one with heatpump always on and one with heat pump aways off. The controler switches between the two MPCs by comparing their final cost function value.
Furthurmore we added one more degree of freedom to the AHU by letting the outside air damper moving freely. In previous reort the outside damper was linked to the recirculation damper because we did not have information about the exact outside volume flow rate. In this study we developed a method to measre the exact outside volume flow rate and thereby made the outside damper an independent variable. This leads to better performance of the system by reducing the overall flow resistance and lower electrical energy consumption.
The performance test results from this control system comparing to our classical SPS system shows considerable imptovement both in energy saving and also comfort condition. The interesting thing is that we not only could save energy consumption but also improved hall comfort level by considering the coupled dynamics of hall temperature and humidity in building model. Our evaluation shows the overall deviation from the reference values for hall temperature and humidty was improved comparing to conventional control system. It is important to note that keeping hall temperature and humidity in desired level indirectly affects the overall energy consumption by affecting pool water surface evaporation rate. Normally hall temperature and humidty are kept in a range that minimizes pool water surface evaporation. Any deviation from these reference values would cause increased surface evaporation that leads to higher energy consumption for pool water heating. As in this study we only studied the ventilation systems we can not measure any changes in other energy consumptions. For a more accurate evaluation a more comprehensive study should be conducted considering all energy systems of the swimming pool including CHP system and pool water heating.
In chapter five we developed a platform for online training of the models. This is actually a python ptogram that collects data and retrains the original neural network and adapts it to the new data. The neural network structure and hyperparameters can be changed in the platform by user. The weights of the updated neural network can be sent to a remote cloud system for further processing. For this purpose we setup a remote server on a raspberry PI micro controller.
The transferred weights of the updated neural network must be further processed to ensure a stable model that can be used for control purpose. The model should then be linearized and create state space matrices which are used by the MPC controller. So a fully automated self learning system that can update itself from the collected data and updates the controller is not accomplished yet in this study. Although we studied other alternatives such and state space neural network training where the state space matrices of the MPC controller are directly updated using the collected data. Hence bypassing the further processing and linearization process. Although this method may be applicable for linear systems such as hall temperature but does not work for nonlinear systems such as hall humidity and AHU model. Other

alternative would be reinforcement learining which is a completely different topic. Reinforced learning is an unsupervised learning that does not require a model of the system to operate [13]. It learns directly from experience by interacting with the environment. This method had beed applied in robotics but its application in climate industry can be subject to furtur study.

Bibliography

[1] *Lars Haupt, TU Dresden, „Systemplattform für Digitale Zwillinge am Beispiel von Wärmepumpen – ein ganzheitlicher Ansatz," in TGA Kongress, Berlin, 2023.*

[2] *Thomas Oppelt, ILK Dresden, „Maschinenlernbasierte Module für intelligente TGA-Planungssoftware," in TGA Kongress, Berlin, 2023.*

[3] *Dorothea Völkerling, TROX GmbH, „Fehlererkennung und -diagnose für RLT-Geräte mit maschinellem Lernen und schwellwertbasierten Methoden," in TGA Kongress, Berlin, 2023.*

[4] *Sebastian Dietz, University of Luxembourg, „Anwendung einer Kombination von KI-Methoden für die automatische Fehlererkennung und -diagnose an raumlufttechnischen Anlagen," in TGA Kongress, Berlin, 2023.*

[5] *Jan Strubel, Viessmann Elektronik GmbH, „Modellbasierter Entwurf und simulationsgestützte Evaluation einer vernetzten Einzelraumregelung," in TGA Kongress, Berlin, 2023.*

[6] *Svenne Freund, Rud. Otto Meyer Technik GmbH & Co. KG , „Ergebnisse der Implementierung Modellbasierter Prädiktiver Regelung in einem großen Verwaltungsgebäude zur Optimierung der Energieeffizienz und des Komforts," in TGA Kongress, Berlin, 2023.*

[7] *Christian Karczewski, Uni Stuttgart, IGTE, „Direktes Lastmanagement energieflexibler Gebäude durch den Einsatz von Zustandsbeobachtern," in TGA Kongress, Berlin, 2023.*

[8] *Alexander Neubauer, TU Berlin, HRI, „Heizleistungsprognosen mit Hilfe von maschinellem Lernen," in TGA Kongress, Berlin, 2023.*

[9] *WILDML, „Wildml.com," 27 10 2015. [Online]. Available: https://web.archive.org/web/20211110112626/http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/. [Zugriff am 2023].*

[10] *Mathworks, „Design Time Series NARX Feedback Neural Networks," [Online]. Available: https://de.mathworks.com/help/deeplearning/ug/design-time-series-narx-feedback-neural-networks.html. [Zugriff am 2023].*

[11] *A. C. Alice Zheng, Feature Engineering for Machine Learning, O'Reilly Media, Inc., 2018.*

[12] *S. Dupond, „A thorough review on the current advance of neural network structures," Annual Reviews in Control, pp. 200-230, 2019.*

[13] *Mathworks, „Recurrent Neural Network (RNN)," [Online]. Available: https://www.mathworks.com/discovery/rnn.html. [Zugriff am 2023].*

[14] *H. T. T. Bernt M. Åkesson, „A neural network model predictive controller," Journal of Process Control, pp. 937-946, october 2006.*

[15] *Mathworks, „Design Neural Network Predictive Controller in Simulink," 2023. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/design-neural-network-predictive-controller-in-simulink.html. [Zugriff am 2023].*

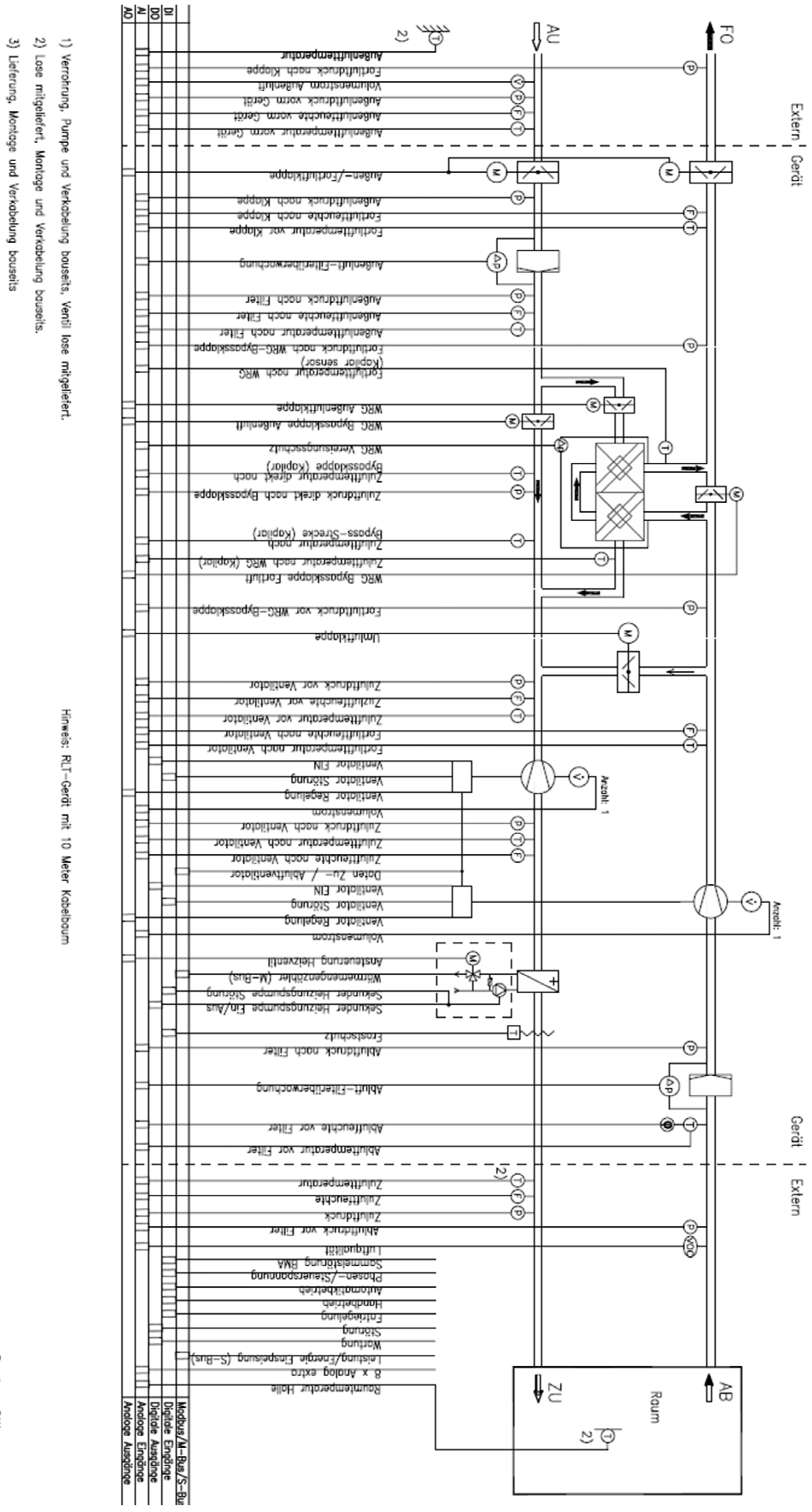
[16] *P. H. Institute, „Passi Haus Institute," [Online]. Available: https://passiv.de/.*

[17] *e. a. Sanaz Sabzevari, „Model FRee Neural Network-Based Predictive Contol for Robus Operation of Power Converters," Energies, p. 14(8), 2021.*

[18] *L. P. L. M. L. Kaelbling, „Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research, 1996.*

[19] *U. C. Birkenfeld, „Studie zur Entwicklung des Energiebedarfs zentraler Raumlufttechnischer Anlagen in Nicht-Wohngebäuden in Deutschland," 15.06.2014.*

I Appendix Control Diagram Unit number 2



AU
FO
ZU
AB
Raum
Extern
Gerät
Außenlufttemperatur
Fortluftdruck nach Klappe
Volumenstrom Außenluft
Außenluftdruck vorm Gerät
Außenluftfeuchte vorm Gerät
Außenlufttemperatur vorm Gerät
Außen-/Fortluftklappe
Fortluftdruck nach Klappe
Fortluftfeuchte nach Klappe
Fortlufttemperatur vor Klappe
Außenluft-Filterüberwachung
Außenluftdruck nach Filter
Außenluftfeuchte nach Filter
Außenlufttemperatur nach Filter
Fortluftdruck nach WRG-Bypassklappe
Fortlufttemperatur nach WRG (Kapilar sensor)
WRG Außenluftklappe
WRG Bypassklappe Außenluft
WRG Vereisungsschutz
Zulufttemperatur direkt nach Bypassklappe (Kapilar)
Zuluftdruck direkt nach Bypassklappe
Zulufttemperatur nach Bypass-Strecke (Kapilar)
Zulufttemperatur nach WRG (Kapilar)
WRG Bypassklappe Fortluft
Fortluftdruck vor WRG-Bypassklappe
Umluftklappe
Zuluftdruck vor Ventilator
Zuluftfeuchte vor Ventilator
Zulufttemperatur vor Ventilator
Fortluftfeuchte nach Ventilator
Fortlufttemperatur nach Ventilator
Ventilator EIN
Ventilator Störung
Ventilator Regelung
Volumenstrom
Zuluftdruck nach Ventilator
Zulufttemperatur nach Ventilator
Zuluftfeuchte nach Ventilator
Daten Zu- / Abluftventilator
Ventilator EIN
Ventilator Störung
Ventilator Regelung
Volumenstrom
Anzahl: 1
Ansteuerung Heizventil
Wärmemengenzähler (M-Bus)
Sekunder Heizungspumpe Störung
Sekunder Heizungspumpe Ein/Aus
Frostschutz
Abluftdruck nach Filter
Abluft-Filterüberwachung
Abluftfeuchte vor Filter
Ablufttemperatur vor Filter
Zulufttemperatur
Zuluftfeuchte
Zuluftdruck
Abluftdruck vor Filter
Luftqualität
Sammelstörung BMA
Phasen-/Steuerspannung
Automatikbetrieb
Handbetrieb
Entriegelung
Störung
Wartung
Leistung/Energie Einspeisung (S-Bus)
8 x Analog extra
Raumtemperatur Halle
DI
DO
AI
AO
Modbus/M-Bus/S-Bus
Digitale Eingänge
Digitale Ausgänge
Analoge Eingänge
Analoge Ausgänge
1) Verrohrung, Pumpe und Verkabelung bauseits, Ventil lose mitgeliefert.
2) Lose mitgeliefert, Montage und Verkabelung bauseits.
3) Lieferung, Montage und Verkabelung bauseits
Hinweis: RLT-Gerät mit 10 Meter Kabelbaum
Regelung: SAIA

II Appendix Function Description of the Simulink Control models

Building Model Based Control

This MPC is created by batch linearizing the discrete building model „DModelBuildingRamsloh6x2“with fixed sample time = 0.1 seconds. The model has 6 inouts and 2 outputs.

Inputs :

1)WPEQ (Represents the total amout of heat exchanged by the air handling units)

WPEQ = $\left(T_{Abluft} - T_{Zuluft}\right) \times Volumeflow\ [\frac{m^3}{hr}]$

2)WPEX (Represents the total amout of humidity exchanged by the air handling units)

WPEQ = $\left(x_{Abluft} - x_{Zuluft}\right) \times Volumeflow\ [\frac{m^3}{hr}]$

3)AussenT (Outside Temperature)
4)I (Solar Radiation [kw/m^2])
5)Nuser (Number of people)
6)TPool (Pool Water Temperature)

Outputs:
1)Hall Temperature
2)Hall Humidity

C

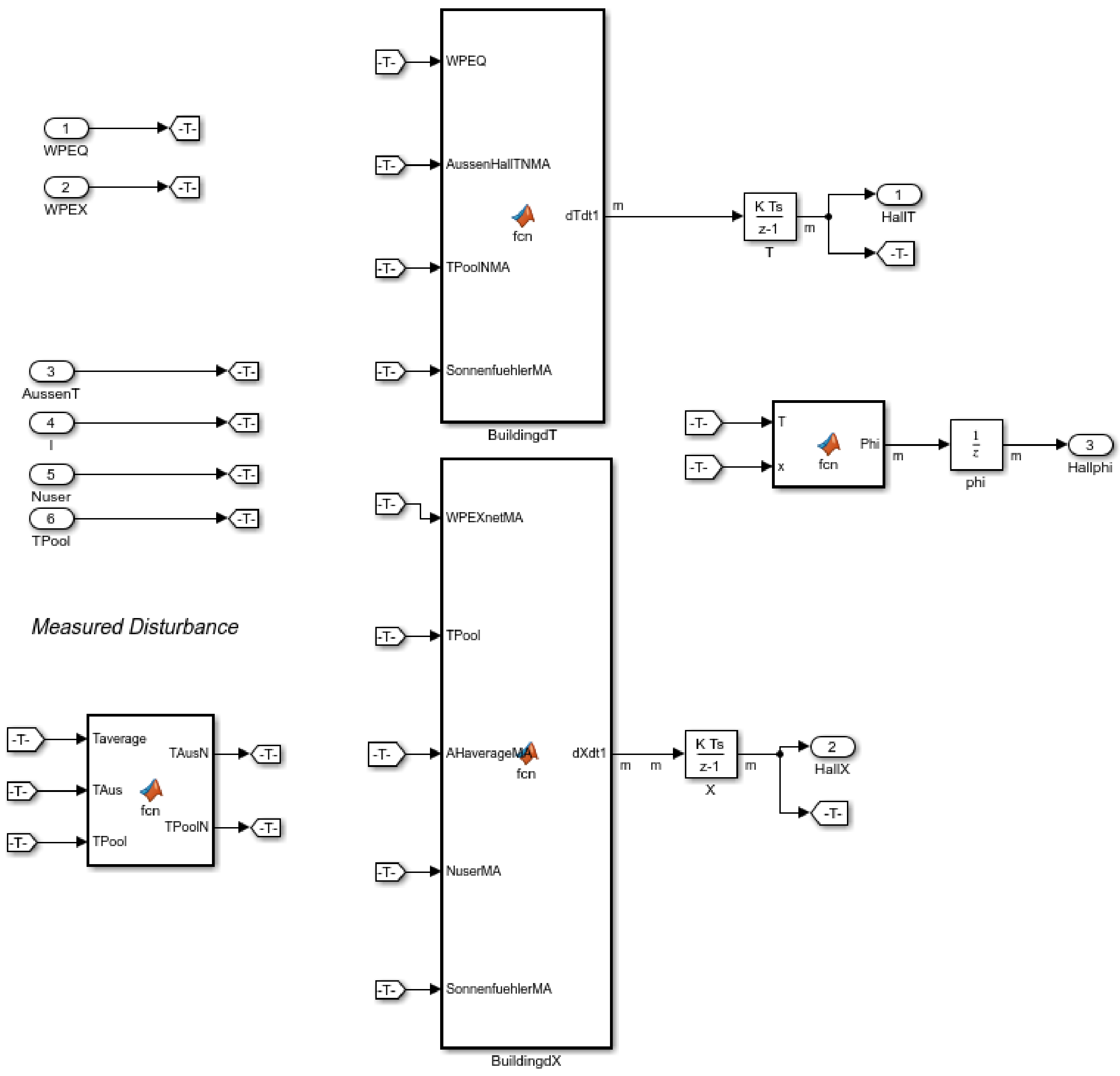


Figure III-0-1 Discrete Building Model for Linearization

The model is linearized at the following steady state operating points:
Outside T = 15°C
Radiation = [300 , 400] $\frac{kw}{m^2}$
Number of people = 10
Pool water T = 30.5°C
Hall Absolute Humidity = [0.014, 0.0155, 0.0165] kg/kg

As it can be seen the model is linearized on different values of solar radiation and hall humidity in ordert o cover the dynamics of the problem when the criteria changes. The linearized model will be used in the following simulink model for the MPC controller.

D

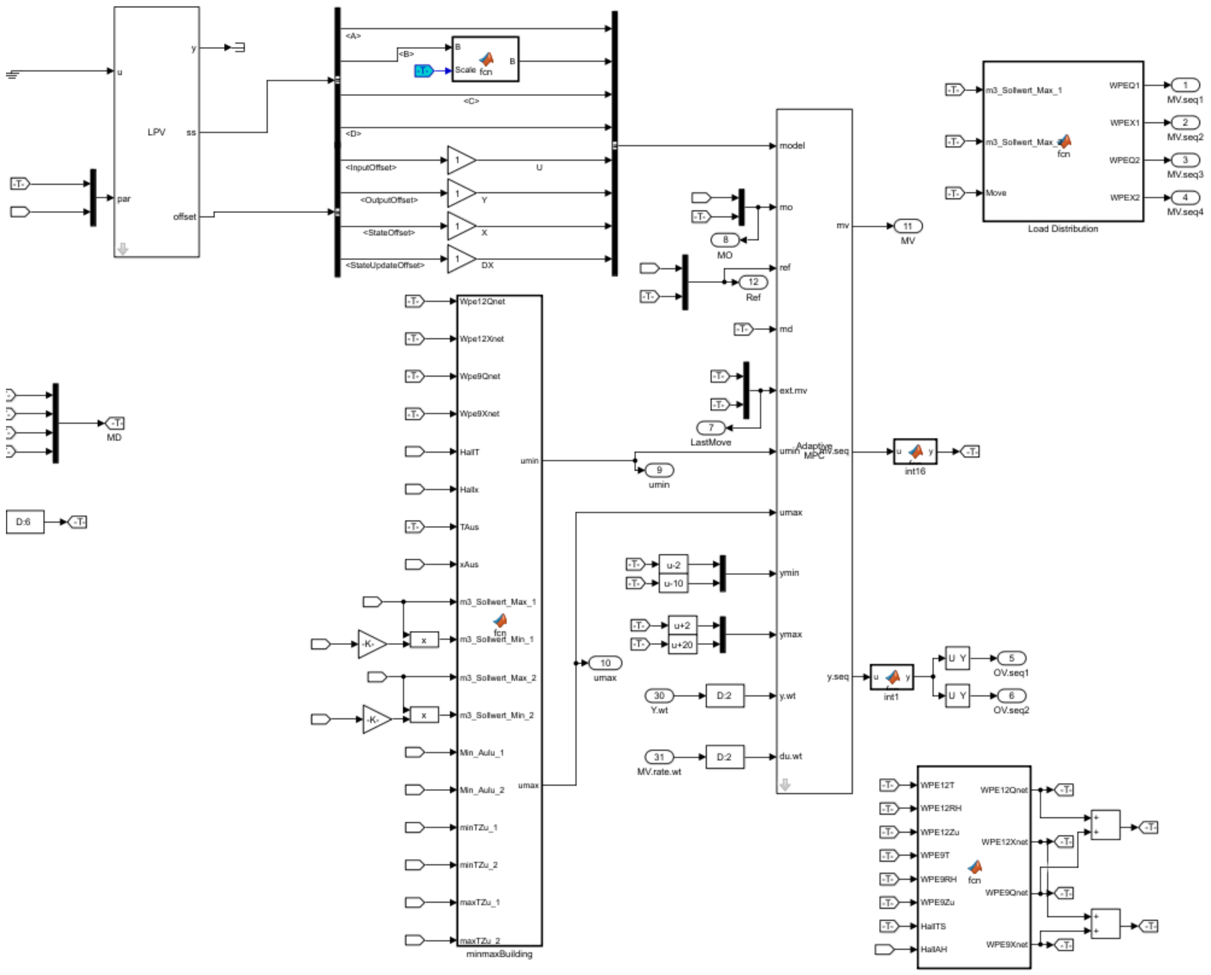


Figure III-0-2 Final MPC Simulink Model for the Building

The inputs to the adaptive MPC block are:
Mo : feedback hall temperature, Humidity
Ref: current reference values for hall temperature and humidity from Saia program (AktAblSW, AktFeuchtesollwert)
Md: measured disturbances (Outside Temperature, Solar Radiation, Number of People, Pool Water Temperature)
Ext.mv: The last move of the MPC controller. The currrent measured values of WPEQ and WPEX
Umin,umax: (Minimum and maximm values allowed for the next move)
These values are calculated in Matlab function „minmaxBuilding_Unified.m“ based on the minimum and maximum supply temperatre, minimum and maximum volume flow rates and minimum outside volume flow rate allowed in saia program as „Sollwerte“ as well as the outside absolute humidity.

The minimum and maximum heatning and dehumidification potentials are calculated as below:

- Minimum Cooling Potential:

16) $Qnet_{min} = Volumeflow_{max} \times (TZulu_{min} - T_{Hall})$

- Maximum Heating Potential:

17) $Qnet_{max} = Volumeflow_{max} \times (TZulu_{max} - T_{Hall})$

- Maximum Dehumidification Potential:

18) $Xnet_{max} = Volumeflow_{max} \times (x_{Aus} - x_{Hall})$

- Minimum Dehumidification Potential:

19) $Qnet_{min} = Volumeflow_{min} \times (x_{Aus\_mixed} - T_{Hall})$

20) $x_{Aus\_mixed} = \frac{Aussenluft_{min}}{100} \times x_{Aus} + (1 - \frac{Aussenluft_{min}}{100}) \times x_{Hall}$

Ymin, ymax: the minimum and maximum allowed taret values for the MPC
y.wt: weight parameters for target values
du.wt: move rate values for the manipulated variables

The Load distribution matlab function calculates the load tob e distributed among the units based of their maximum volume flow rate

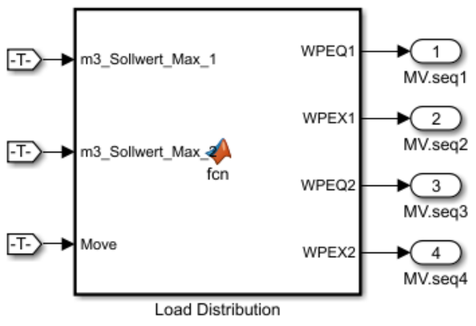


Figure III-0-3 Load distribution Among Units

Move : 2x60 matrix containing the next 60 moves for WPEQ and WPEX demand for the building.
MV.seq1: next 60 moves of WPEQ for unit number 1
MV.seq2: next 60 moves of WPE for unit number 1
MV.seq3: next 60 moves of WPEQ for unit number 2
MV.seq4: next 60 moves of WPE for unit number 2

Unit Model Based Control without Heat Pump

F

For the air handling unit without heatpumps the following 13 inputs 4 outputs simulink model is batchlinearized:

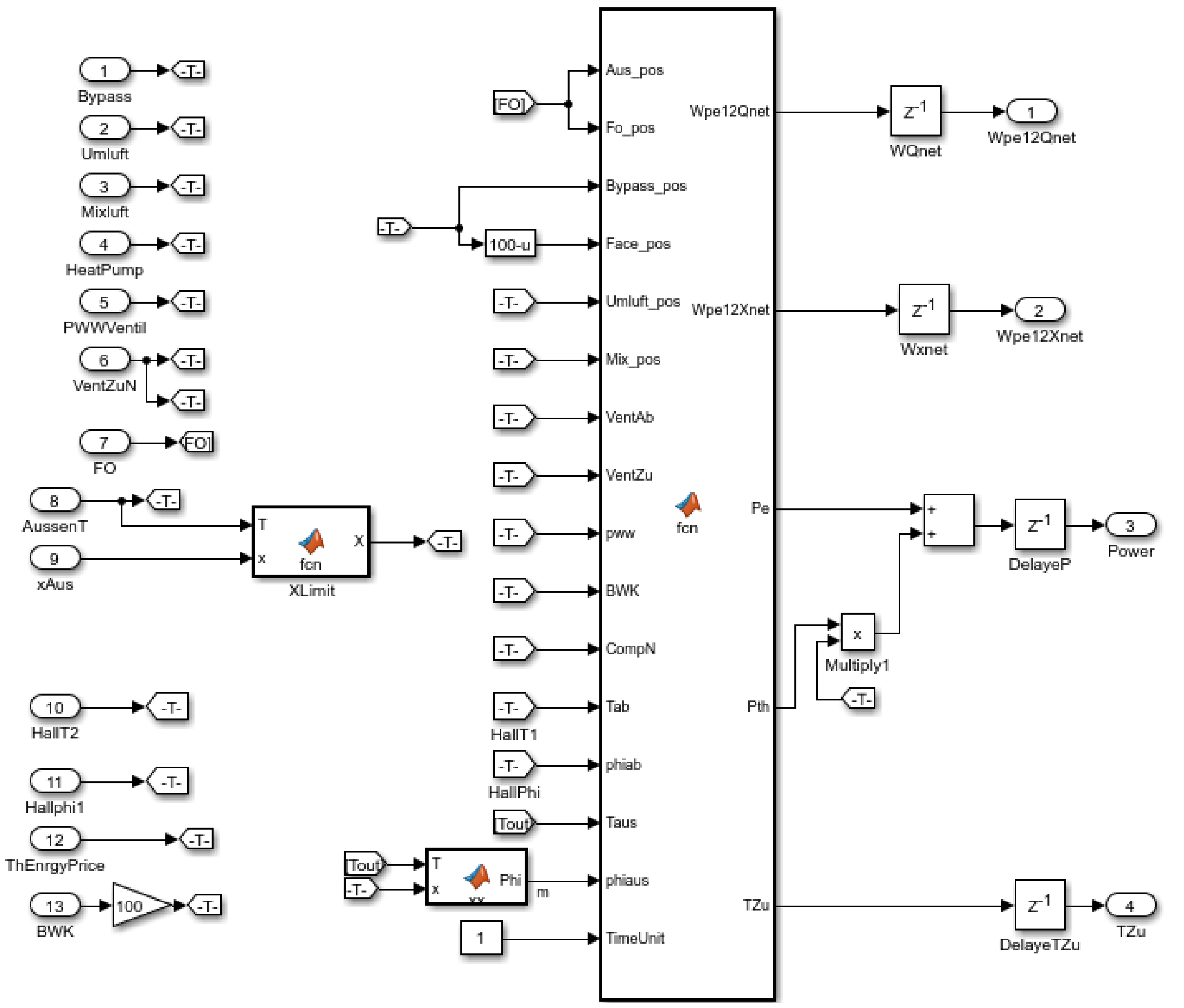


Figure III-0-4 Discrrete Model for Unit

ThEnergyPrice = Gas price / ElectricitaPrice is the ratio of gas to electricity energy prices

The model is linearized on the following steady state operating points:
Bypassklappe = 0
Umluftklappe = [30, 45, 60,75]
Mixluftklappe=0
Wärmepumpe= 0
PWWventil= [10,30,50]
Ventilator=50
Aussenluftklappe= [30,60]
Aussentemperatur= [5,10,20,25]
Aussenabsolutefeuchte= [0.004,0.006,0.008,0.01,0.012]
Halltemperatur=31.5;
Hallfeuchte=50;

ThEnrgyPriceRange= [0,0.5,1]
Beckenwasserkondensator=0

The model is linearized in different set points of Umluftklappe, PwwVentil, Aussenluftklappe, Aussentemperatur, Aussenfeuchte, and energy price. This will make a 6-dimensional grid of LTI objects that can be used with the following adaptive MPC Simulink model.

Wpe12OhneMi6DFOphys_Scaled.slx

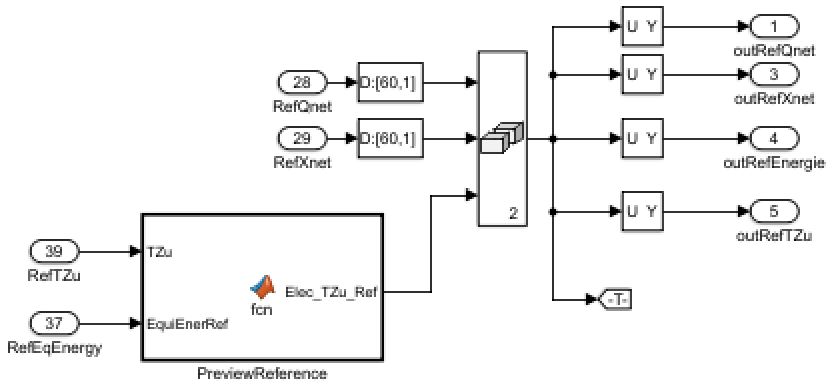


Figure III-0-5 Set References for Unit

Reference previwing function makes a vector of 1x60 for reference Zulufttemperatur and equivalent energy cost. The other two references RefQnet and RefXnet are each vectors of 1x60 which come from the uppler layer Building MPC and they represent the heating and dehumidification demand for this unit.

These values will be set in node red:

RefTZu: 32 °C
RefEqEnergy: 1 kw (energy consumption is set to be minimum)
The other inputs to the MPC toolbox are:
Mo: measured outputs: feedbacks for Heating, Dehumidification, equivalent Energy Cost and ZuluTemperatur
Ext.mv: the actual position of the control signals
Umin: minimum values for control signals

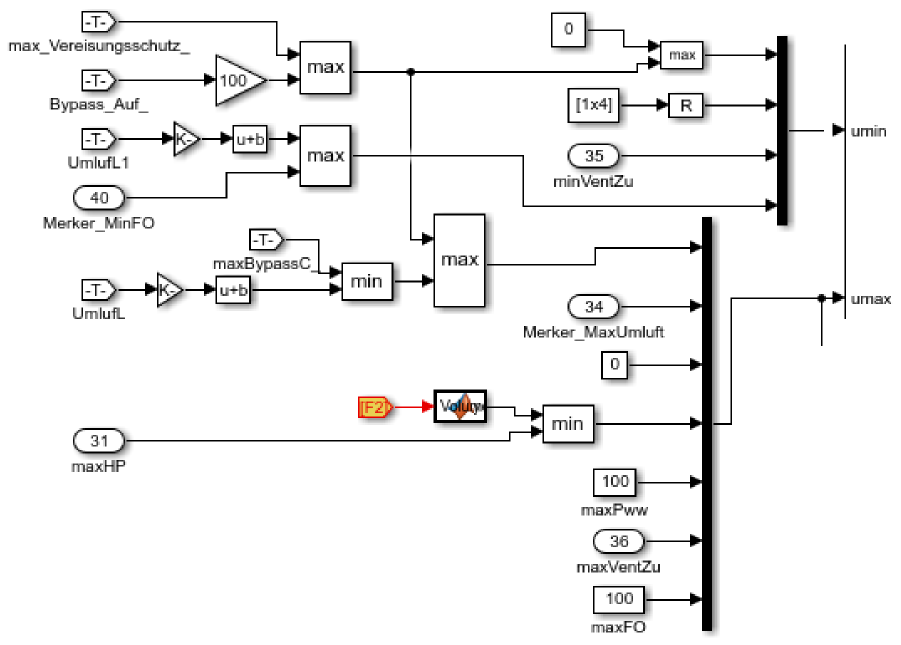


Figure III-0-6 Set Minimum and Maximum Values for AHU

Minimum Bypass is normally zero except there is a command for heat exchanger freezing protection (max_Vereisungsschutz or Bypass_Auf).
The minimum values for Umluft, Mixluft, Heatpump and air heater is set to zero.
The minumin values for the ventilator is set by a PID controller which ensures the minimum allowed volume flow rate.

Maximum Bypass is a function of Umluft and airheater so that the air heater and bypass do not open at the same time. And also bypass and Umluft can not open at the same time.
Maximum Mixluft here is zero because these units without heat pump do not have mixing damper.
Maximum heat pump is eather set manually (here zero) or by volume flow rate whichever is minimum is taken.
Maximum air heater is set to 100.
Maximum ventilator is set by PID controller to avois exceeding maximum volume flow rate
Maximum Außenluft/Fortluft is set to 100.

Ymin: the minumum values for the controller objectives:
These parameters will be set in node red

Ymin(1) = $(T_{Min\_Zuluft\_Kaskade} - 32) \times m3\_Zul$
Ymin(2) = $(0.005 - 0.0165) \times m3\_Zul$
Ymin(3) = 0
Ymin(4) = $T_{Min\_Zuluft\_Kaskade}$

Ymax: the maximum values for the controller objectives:
Ymax(1) = $(T_{Max\_Zuluft\_Kaskade} - 29) \times m3\_Zul$
Ymax(3) = $EnergyPriceRatio \times maxThermal \times maxElectircal \times SF$
Ymax(4) = $T_{Max\_Zuluft\_Kaskade}$

Where:

$EnergyPriceRatio : \frac{Thermal\ Energy\ Price}{Electrical\ Energy\ Price}$

$maxThermal : Maximum\ Thermal\ Energy\ Capacity$

$maxElectrical : Maximum\ Electrical\ Energy$

SF : $\frac{Maximum\ VolumeFlow\ Rate}{12000}$

Ywt: weight parameters for the target values

Du.wt: weight parameters for moving rate

mv.seq: 60x1 vectors of manipulated variables move within control horizon
y.seq: 60x1 vectors of output variables within prediction horizon

PID control for setting min/max values of Umluft, Ventilator and Außenluft/Fortluft

J

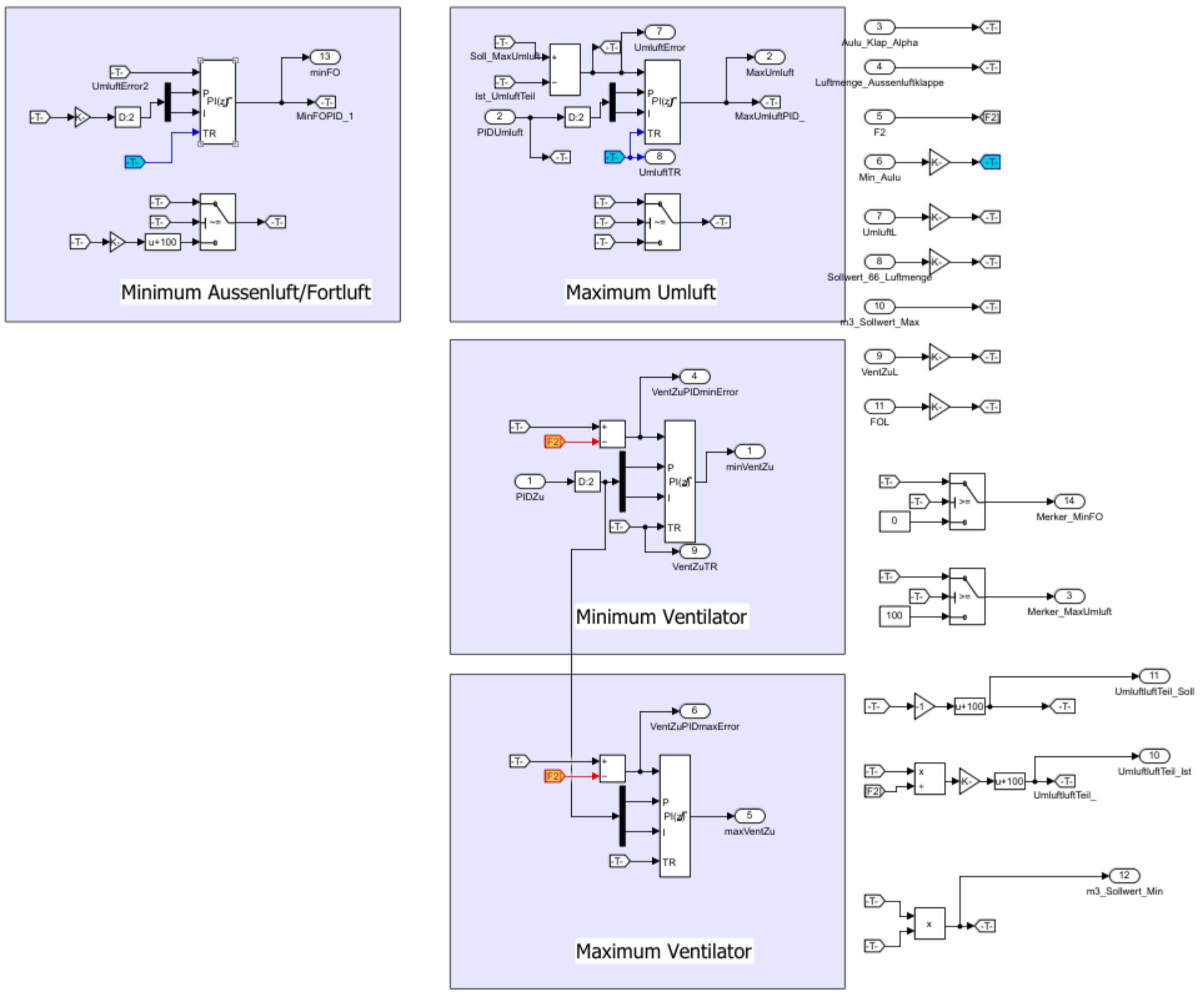


Figure III-0-7 Set Minimum and Maximum Values using PIDs

These are the reference values (Sollwerte) used for the PID controlers:

Minimum set point for the ventilator : minim volume flow allowed (m3_Sollwert_Min)
Maximum set point for the ventilator : minim volume flow allowed (m3_Sollwert_Max)

Maximum set point for the Umluft : maximum allowed recirculation (100- Min_Aulu)
The actual value (Istwert) for PID is calculated based on measured outside volume flow rate:

Istwert =100- $\frac{Luftmenge\_Aussenluftklappe}{m3\_Zuluft} \times 100$

In case no outside volume flow rate is available
Minimum Außenluft/Fortluft will be calculated similar to Umluft but with negative tunning parameters.

Unit Model Based Control with Heat Pump

MPC Switching

The units which are equipped with heat pump have a different type of MPC control which switches controllers based on optimal costs. The problem with heat pumps ist hat although they are supoosed to be variable speed in practice they have problem running in low speeds. So we want that the heat pump does not run below 50% capacity. On the other hand we can not define discrete values as MPC outputs. That’s why a workaround is used here where two MPC blocks work in parallel.

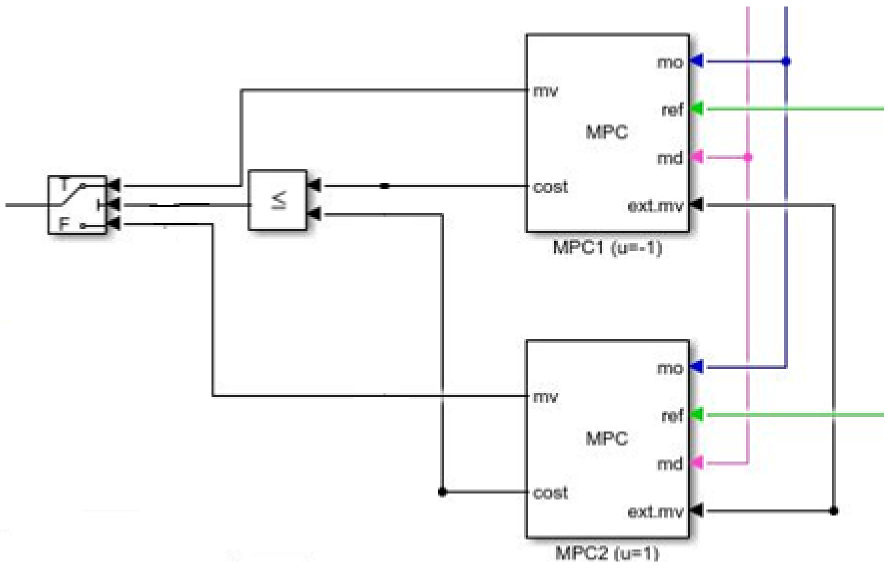


Figure III-0-8 MPC Switching Method

In this case the heat pump control range in MPC1 is between 50 – 100 and in MPC is 0 – 0. So in one MPC the heat pump is permanentaly off and in the other one it can run above 50.
One problem in this workaround would be that because of the manipulated variable rate tuning weights which limit big movements of the manipulated variables a jump from MPC1 to MPC2 and vice versa creates a big value in cost function which in practice avoids the switch between the two MPC blocks. In ordert o resole this problem the move rate weights of the PWW and eat pump are set to a small value „mvratePww“ (0.005) for the other MPC whos cost function is higher. This way the big move rate because of MPC switch does not count in the cost function and enables a MPC switch only based on target values.

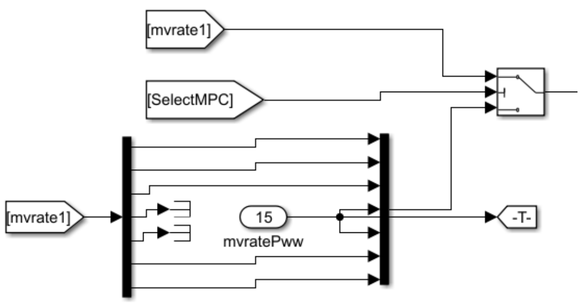


Figure III-0-9 Variable move rates for MPC Switching Method

Heat Pump Stop Delay Counter Function

The function is used to apply a delay for switching between MPC2 (heat pump on) to MPC1 (Heat pump off). The delay time „HPStopCounter“ parameter is set in node red in settings node. This ensures enough time for the heat pump to warm up before early stopping.

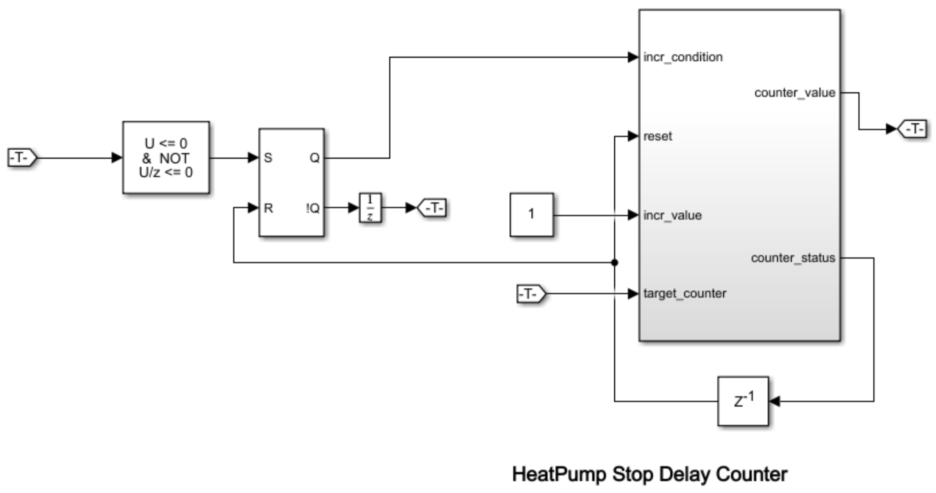


Figure III-0-10 Heat Stop Delay Counter Function

Heat pump start/stop based on heating load demand

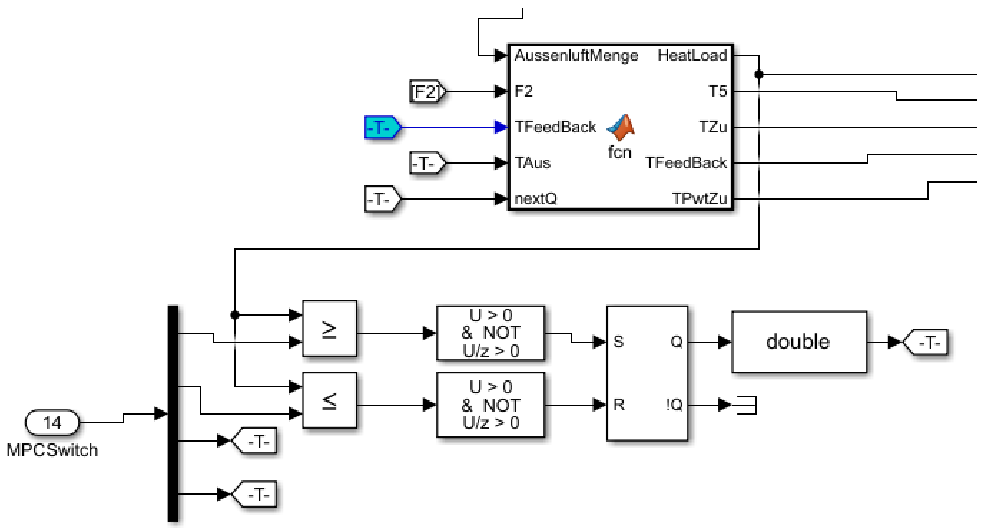


Figure III-0-11 Heat Pump Start/Stop Function

This function estimates the heat load required by the system based on the building load demand.The heat pump does not start if the heat load demand is less that a certain amount. This helps to avoid frequent MPC switching when load demand is low.

$$T_{PwtZu} = \left(T_{Hall} - T_{Außenluft}\right) \times 0.7 + T_{Außenluft}$$

M

$$T_{mischluft} = (Flow_{Außen} \times T_{PwtZu}) + (Flow_{Zulu} - Flow_{Außen}) \times T_{Hall}/Flow_{Zulu}$$
$$T_{Zu} = \frac{Q}{Flow_{Zulu}} + T_{Feedback}$$
$$HeatLoad = (T_{Zulu} - T_{Mischluft}) \times Flow_{Zulu} \times 1.15/3600$$

MPCSwitch(1): Minimum load heat pump start
MPCSwitch(2): Minimum load heat pump stop

Switsching between MPC blocks based on the costs

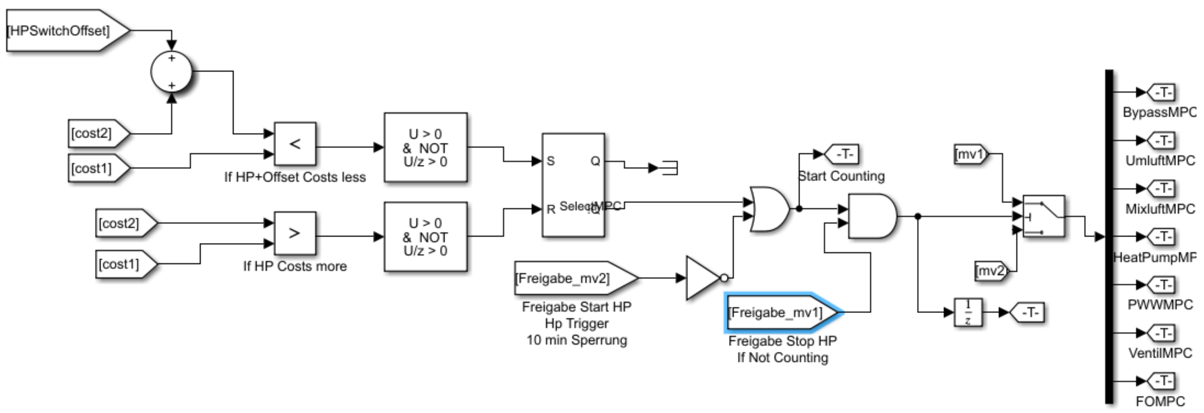


Figure III-0-12 MPC Switching Based on Cost Function Value

HPSwitchOffset: used for mpc stwitch if the cost of MPC2 + offset is less that cost of MPC1.

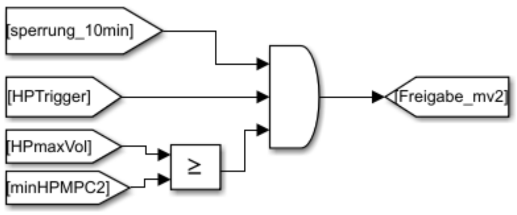


Figure III-0-13 MPC Switching Ofsset Value

Freigabe_mv2: allows to switch to MPC2 when Sperrung_10min from Saia is off, heat load demand criteria is fullfilled and volume flow rate over the evaporator is enough to start the heat pump.

Freigabe_mv1: allows to switcht to MPC1 if the counter is not counting

N