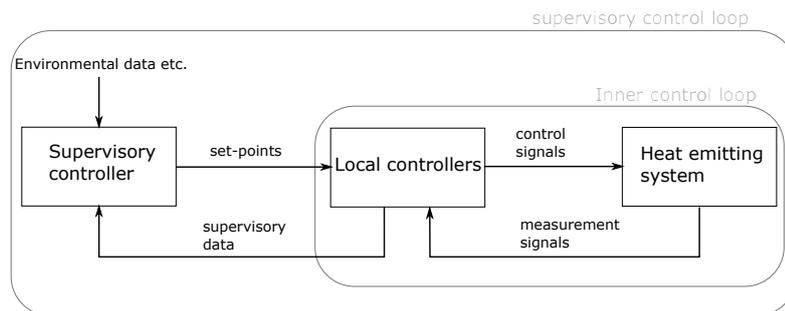


Appendix E



Control strategies, algorithms, and simulation results

Tomasz Minko
Jan Bendtsen
Søren Østergaard Jensen
Research carried out at Aalborg University



August 27, 2018

Title: Control strategies, algorithms, and simulation results

Authors:

Tomasz Minko, AAU

Jan Bendtsen, AAU

Søren Østergaard Jensen, DTI

May 2018 1st edition 2018

The present document is part of the documentation of the *Underfloor heating and heat pump optimization* project financed by the Danish Energy Agency through the EUDP project no. 64014-0548.

Front page: Conceptual block diagram of supervisory control interacting with underfloor heating system

E.1 Introduction

As outlined in the project description, the efficiency of heat pumps depends significantly on its operating conditions. Typically, an increase of 1 degree C in the temperature difference leads to a decrease of 2-3 per cent in terms of COP. A too high supply temperature also has consequences for the floor heating system itself, as the volume flow through the system fluctuates due to uncoordinated opening and closing of valves. The fluctuating flow causes the heat pump to fluctuate in produced heat power, resulting in a sub-optimal COP at the actual temperature level.

This document gives an overview of the control algorithms tested in simulation within the project. The reader will first be introduced to the general control architecture and the notion of a supervisory controller (Section E.3). Then, the system dynamics is simulated in closed loop with a simple (but standard) PI controller in a configuration where the flow is controlled via on/off valves, and a proposal for improvements is given. Next, some simplified mathematical models (Section E.5) and a model based predictive control algorithm using these simplified models are presented (Section E.6).

Finally, a different strategy based on Artificial Neural Network modeling will be outlined (Section E.8 and onward). The results are detailed in the attached paper, “Predictive Control of Hydronic Floor Heating Systems using Neural Networks and Genetic Algorithms” (Vinther et al., 2017), but the methodology is included here as well for completeness.

E.2 PI comparison test evaluation

E.2.1 Heat Pump and Floor Heating Systems Description

The four-room house model described in Appendix C and its interaction with the installed heat pump through the floor heating system as described in Appendix D are considered.

The indoor air room temperature T_a is affected by the ambient conditions, such as ambient air temperature T_{amb} , sunlight etc. Moreover, it is affected by the heat generated by people living in the building and the devices they use, such as stove, appliances etc. From a control perspective, all ambient effects on the T_a are classified as a disturbances. In order to maintain T_a within specified comfort limits, the heat pump has to maintain a satisfactory temperature T_f of hot water, which is transported and used for in-house purposes. The amount of water delivered to each room by the floor heating

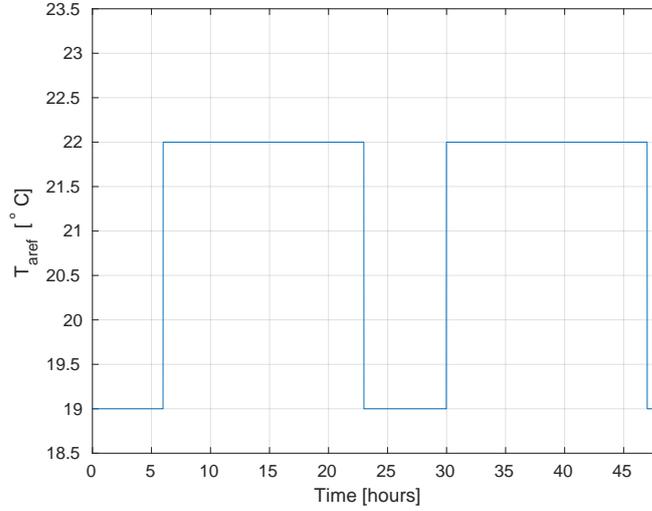


Figure E.1: Indoor Air Room Temperature Reference signal

system is regulated via the valve position d_a for that room. While flowing through the under-floor pipes, the hot water delivers heat to the surrounding floor and to the room, causing the hot water to be cooled instead to a lower temperature than T_f . The temperature of the return water is denoted T_r ; the difference between T_f and T_r describes the amount of heat the heat pump needs to deliver to the system. Unfortunately, there is no simple way of controlling the heat pump; the only possibility is to overwrite its temperature sensor readings in a way that will cause the heat pump to change its operation.

E.2.2 Indoor Air Room Temperature References

To obtain a fair basis for comparison, a setup where the air temperature in the house is governed by a simple PI controller is first simulated. One weekend is simulated (48 hours) with the indoor air temperature references $T_{\text{a,ref}}$ being equal for all four rooms. The references are shown in Figure E.1. Night set back has been defined to start at 23:00 and last until 6:00, during which $T_{\text{a,ref}} = 19^{\circ}\text{C}$. During the daytime (after 6:00), $T_{\text{a,ref}}$ has been chosen to 22°C .

E.2.3 Dynamics evaluation

Figures E.2, E.3, E.4 and E.5 present the dynamics of indoor air temperature of rooms 1, 2, 3 and 4, respectively. The rooms differ in size, and this is

T_{a1} [%]	T_{a2} [%]	T_{a3} [%]	T_{a4} [%]
34.0	51.0	74.0	97.0

Table E.1: Percentage of time the room temperature (6:00-16:00) is outside the allowed hysteresis band around the set-point

clearly reflected in the dynamics of their air temperature; Room 3 has the slowest dynamics, for example. After the change of set-point at 6:00, it takes almost 10 hours for the temperature to reach the set-point. In order to quantify the dynamics, a metric for calculating the percentage of time the room temperature is within allowed band around the set-point is chosen. Note that the PI controller hysteresis is $\pm 0.5^\circ\text{C}$. The dynamics is evaluated for the temperatures in between 6:00 and 16:00, and the results are listed in Table E.1.

As can be seen from the table, the air temperature of room 1 has been out of the hysteresis bound 34% of the time, and this is the best result for all the rooms. The PI controller managed to keep the room 1 temperature within the set-point bound more often than other rooms, because the dynamics of room 1 is the fastest, meaning it is easier to heat up and cool down the room to designated temperatures. Room 2 and 3 shows worse results, 51% and 74%, respectively, and it can likewise be concluded that their slower dynamics are mainly responsible for that fact. In room 4, the set-point has not been reached most of the time (97%), and two things can be concluded. First, room 4 dynamics is slow. Second, either the forward temperature has been too low and the room was not able to reach given set-point or the floor heating gains have been chosen wrongly during the modeling part. Taking into account that the forward temperature has been sufficient for the three other rooms, modeling errors seems the most likely explanation.

Furthermore, one can observe that even though the PI controller should keep the room air temperatures within the $\pm 0.5^\circ\text{C}$ bound, the controller very often closes/opens the valve too late, see Figure E.6. It happens very often that the valve is closed when T_{a1} violates the hysteresis bound by 0.1-0.2 degrees. A delay of up to two iterations (2 minutes) has been noticed between the control signal is sent and the valve responds; see also Appendix F, Figure 2.22.

E.3 Supervisory Controller

Following the PI control test, a supervisory controller will be designed in an attempt to improve the temperature control performance. A supervi-

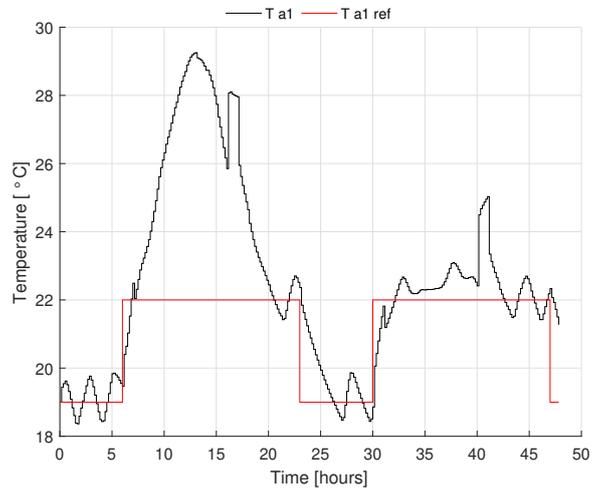


Figure E.2: PI Control Test Room 1 Indoor Air Temperature

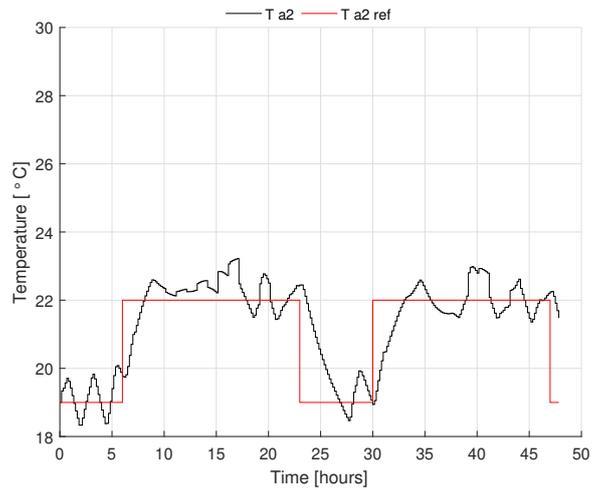


Figure E.3: PI Control Test Room 2 Indoor Air Temperature

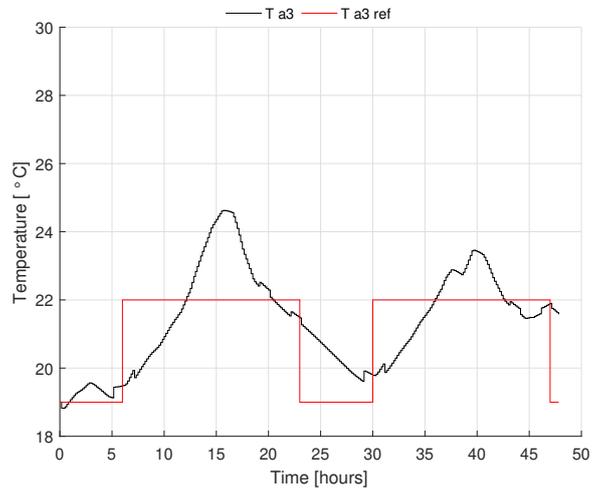


Figure E.4: PI Control Test Room 3 Indoor Air Temperature

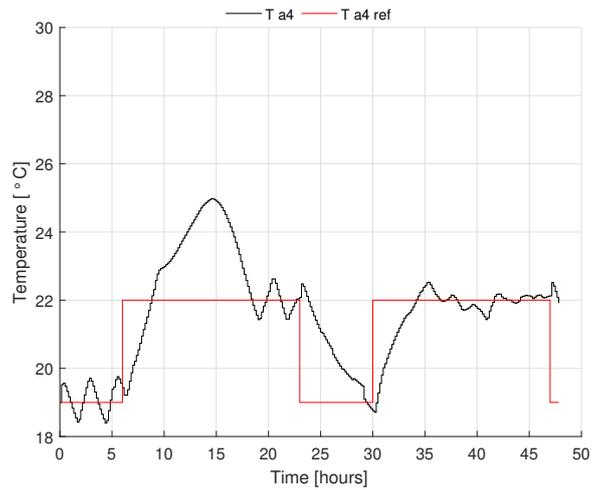


Figure E.5: PI Control Test Room 4 Indoor Air Temperature

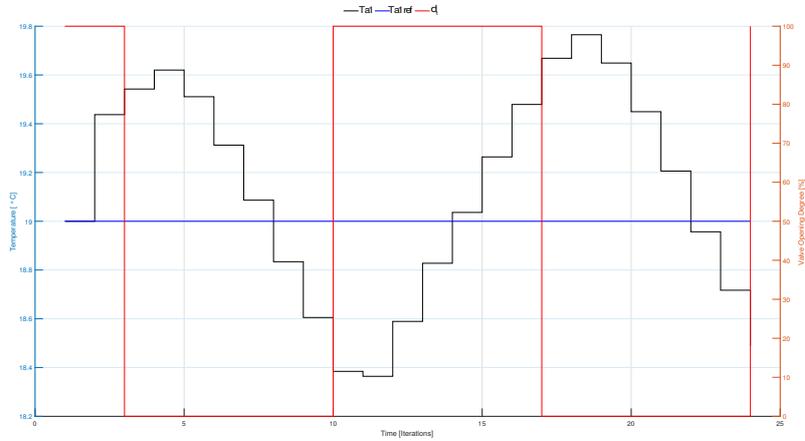


Figure E.6: PI controller hysteresis bound violation. Legend: black: T_{a1} , blue: T_{a1ref} , red: d_1 .

sory controller is a type of a controller that does not send the control signal directly to a given device or subsystem to be controlled; instead, it sends signals to the local controller of the device or subsystem. The local controllers are responsible for managing the fast dynamics of the device/subsystem, while the supervisory controller gathers the current states of the subsystems and governs the behavior of all local controllers. In the present context, the design starts out with the following three main assumptions:

- **Hypothesis 1:** Through supervisory control action it is possible to bring down the power consumption of the heat pump.
- **Hypothesis 2:** Introducing a supervisory controller will lead to better set point following.
- **Hypothesis 3:** Variable valve position control is beneficial in terms of power consumption and set point following, for the operation of the system.

Often, supervisory controller dynamics is dictated by physical constraints of the system. In this case, the supervisory controller sends control signals every 10 minutes, which corresponds to the opening/closing time of the valves and minimum safe time between heat pump switches. In order to improve upon the PI control mentioned in the previous chapter, the supervisory controller will be model-based. Furthermore, it is chosen to make it predictive, since it is known that predictive controllers are able to deal effectively with constraints; they are able to estimate when a constraint violation is likely

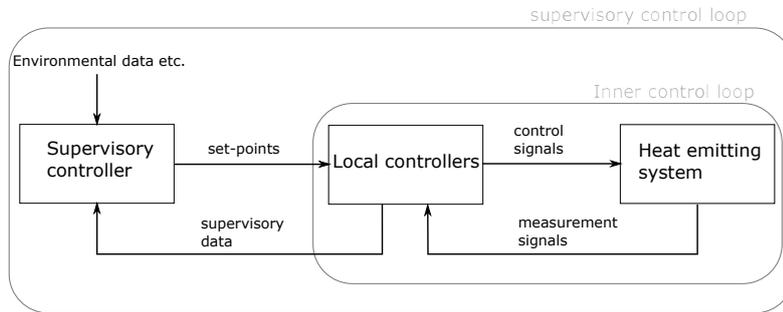


Figure E.7: Classification of the examined structure according to the inner and supervisory control loops

to occur and thereby be able to avoid it in time. Another advantage of predictive control is that future expectations, such as weather forecasts, occupancy schedule etc., can be incorporated into the control scheme, allowing for even better performance. One of the few drawbacks of using model predictive control is computation requirements; a predictive controller normally requires far more computation resources than e.g., a PI controller.

In the following, the supervisory controller will thus be referred as a Model Predictive Controller (MPC). The general structure of the supervisory controller is depicted in Figure E.7.

E.4 Methodology

In the new controller formulation, the previously set $\pm 0.5^\circ\text{C}$ hysteresis is replaced by an optimization algorithm; Figure E.8 illustrates how the optimization algorithm has been fit into the simulation framework.

Before the first iteration a mathematical model of the optimized system needs to be built (see Section E.5). When choosing the model a trade-off between complexity and simplicity has to be made. Usually, more complex models lead to higher accuracy, which is needed when making satisfactory predictions of system behavior. On the other hand, simpler models offer faster computation times and typically have fewer parameters to tune. Next, the model needs to be tuned, i.e., parameters have to be chosen such that the modeled dynamics mimics the true system behavior as much as possible. In this report it is chosen to use a very simple model with heuristically chosen parameter values, see Table E.2.

Once the model has been chosen, one has to find a suitable objective func-

tion, which is a scalar function of system states, inputs and disturbances (if measurable) that encapsulates the most desirable system behavior at an optimum. The aim of the optimization is then to drive the system towards the optimum. The objective functions investigated in this project are presented in Section E.6.2. The optimization is repeated during each iteration of the control loop, repeatedly recomputing control signal trajectories in the presence of new disturbances etc. The states, control variables and disturbances together with their initial values, and constraints are presented in Table E.3, E.4 and E.5, respectively. In order to integrate the systems dynamics, a 4th-order Runge-Kutta integrator has been used in the numerical studies.

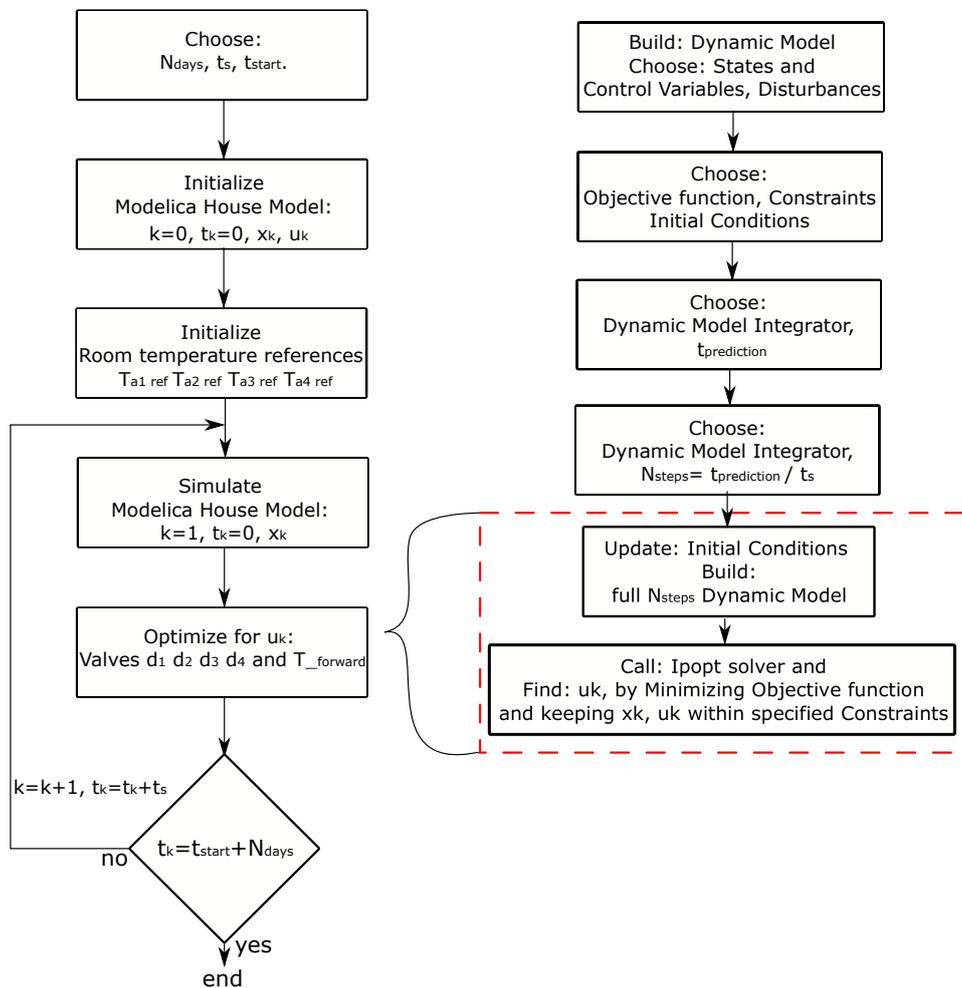


Figure E.8: Flow diagram presenting the optimization algorithm steps

Symbol	Description	Value	Units
M_{z1}	Room 1 thermal mass	650	kg
M_{z2}	Room 2 thermal mass	650	kg
M_{z3}	Room 3 thermal mass	2500	kg
M_{z4}	Room 4 thermal mass	2000	kg
UA_{hi}	Heat exchange coefficient	200	J/kgK

Table E.2: Model parameters; see Appendix C.

E.5 Mathematical Modelling

Each room’s indoor air temperature dynamics is expressed using the following ordinary differential equation (ODE).

$$M_z c_{\text{pair}} \dot{T}_{\text{ai}} = d_{\text{ai}} \dot{m}_{\text{ai}} UA_{\text{hi}} ((T_{\text{f}} + T_{\text{ri}})/2 - T_{\text{ai}}) + Q_{\text{inti}} + Q_{\text{shi}} + UA_{\text{ci}}(T_{\text{amb}} - T_{\text{ai}}) \quad (\text{E.1})$$

where $i \in \{1, 2, 3, 4\}$ represent the room number, M_z are thermal masses of each room, c_{pair} are specific heat capacities of each room, d_{ai} are valve positions, \dot{m}_{ai} are hot water flows for each room floor, UA_{hi} are heat transfer coefficients for each floor to the rooms, Q_{inti} and Q_{shi} are interior and external heat sources in each room, UA_{ci} are heat transfer coefficients for heat exchange with the ambient (walls, windows etc.), and T_{amb} is the ambient temperature. As usual, T_{ai} denotes the air temperature of room i , while \dot{T} denotes derivative of T with respect to time.

The thermal masses M_{zi} , together with the heat transfer coefficients used to describe the heat exchanger of each zone have been chosen; model parameter values can be found in Table E.2. The heat transfer coefficients in (E.1) are assumed to be linearly dependent on the circulated water mass flow.

Note also that since the heat pump cannot be controlled directly – it is equipped with its own built-in controller – the heat pump is manipulated by adding an artificial offset signal $T_{\text{amboffset}}$ to its temperature measurement feedback, thereby ‘tricking’ it into following a behavior dictated by the supervisory controller.

The following variables were chosen for the analysis.

- **state variables:** Air temperature, room 1 - 4, i.e., T_{a1} , T_{a2} , T_{a3} and T_{a4} (see (E.1)). Initial values, constraints and units can be found in Table E.3.
- **control variables:** Valve opening degree, room 1 - 4, i.e., d_1 , d_2 ,

Symbol	Description	Initial	Minimum	Maximum	Units
T_{a1}	Room 1 air temperature	22	18	24	$^{\circ}\text{C}$
T_{a2}	Room 2 air temperature	22	18	24	$^{\circ}\text{C}$
T_{a3}	Room 3 air temperature	22	18	24	$^{\circ}\text{C}$
T_{a4}	Room 4 air temperature	22	18	24	$^{\circ}\text{C}$

Table E.3: System states: Initial values and constraints.

Symbol	Description	Initial	Minimum	Maximum	Units
$T_{\text{amboffset}}$	Temperature offset	0	-5	5	$^{\circ}\text{C}$
d_1	Room 1 valve	1	0	100	%
d_2	Room 2 valve	1	0	100	%
d_2	Room 3 valve	1	0	100	%
d_2	Room 4 valve	1	0	100	%

Table E.4: Control signals: initial values and constraints.

d_3 and d_4 , as well as heat pump temperature sensor offset $T_{\text{amboffset}}$. Their initial values, constraints and units can be found in Table E.4.

- **disturbances:** Ambient air temperature T_{amb} ; insolation in room 1 - 4, i.e., $Q_{\text{sh}1}$, $Q_{\text{sh}2}$, $Q_{\text{sh}3}$ and $Q_{\text{sh}4}$; and internal heat sources in room 1 - 4, i.e., $Q_{\text{ind}1}$, $Q_{\text{ind}2}$, $Q_{\text{ind}3}$ and $Q_{\text{ind}4}$.

Note that the presented minimum and maximum values for the disturbances are not used in the optimization procedure as actual constraints. They are

Symbol	Description	Initial	Minimum	Maximum	Units
T_{amb}	Ambient air temperature	10	10	26	$^{\circ}\text{C}$
$Q_{\text{sh}1}$	Room 1 solar heat	22	22	22	J
$Q_{\text{sh}2}$	Room 2 solar heat	22	22	22	J
$Q_{\text{sh}3}$	Room 3 solar heat	22	22	22	J
$Q_{\text{sh}4}$	Room 4 solar heat	22	22	22	J
$Q_{\text{ind}1}$	Room 1 internal heat	29.17	0.05	32.04	J
$Q_{\text{ind}2}$	Room 2 internal heat	29.17	0.05	32.04	J
$Q_{\text{ind}3}$	Room 3 internal heat	29.17	0.05	32.04	J
$Q_{\text{ind}4}$	Room 4 internal heat	29.17	0.05	32.04	J

Table E.5: Disturbances: initial values and constraints.

presented in order to give the reader an impression of the variations in ambient conditions.

E.6 Optimization

This section briefly presents the optimization problem to be solved at each iteration. Let $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ be time-dependent states and input signals, respectively. Given a system described by a set of coupled first-order ordinary differential equations $\dot{x} = F(x, u)$, a set of constraints on inputs and states $h(x(t), u(t)) \geq 0$ and some appropriate cost function $L(x, u)$, the Optimal Control Problem (*OCP*), defined for a given initial state \bar{x}_0 at time t_0 , is to find the control policy $u^*(\tau)$ over the time horizon $\tau = [t_0, t_p]$ that achieves the minimal integrated cost over τ .

The optimization problem can be formulated conveniently as follows:

$$\text{minimize}_{x(\cdot), u(\cdot)} \int_{t_0}^{t_p} L(x(t), u(t)) dt \quad (\text{E.2a})$$

$$\text{subject to } \dot{x}(t) - F(x(t), u(t)) = 0, \quad t \in [t_0, t_p], \quad (\text{E.2b})$$

$$x(0) - \bar{x}_0 = 0, \quad (\text{E.2c})$$

$$h(x(t), u(t)) \geq 0 \quad t \in [t_0, t_p], \quad (\text{E.2d})$$

where (E.2b)–(E.2d) are constraints. The first constraint states that the system states must follow the dynamics F given the candidate input u ; this is essentially what makes the optimization above a predictive controller. The second constraint states that the optimization must start at the current value of the states each time it is executed; this introduces feedback into the optimization. The third constraint is a general statement of constraints, such as upper and lower limits on forward temperature, rate of change of valve positions etc.

E.6.1 Multiple Shooting

Multiple Shooting (MS) is a method for solving nonlinear MPC problems such as (E.2a) numerically. The basic idea is as follows. For each iteration, a prediction horizon is defined, setting the current time to 0 and the end time to the current time plus t_p . Assuming a sequence of control samples covering the prediction horizon is given, MS then divides the integration of the system dynamics into short time intervals $[\tau_i, \tau_{i+1}]$ where

$$0 = \tau_0 \leq \tau_1 \leq \dots \leq \tau_N = t_p.$$

Throughout each of the short intervals, the control signal is assumed to be kept constant; see Figure E.9.

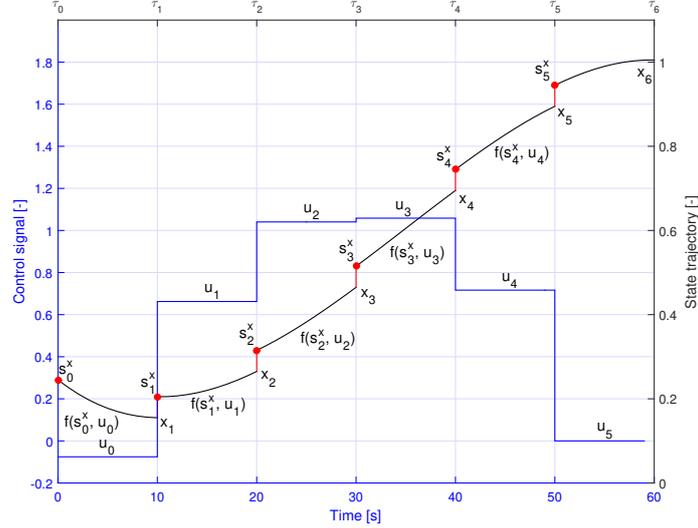


Figure E.9: General idea of Multiple Shooting discretization. Note that this is a generic example; see e.g., Kirches et al. (2012).

In the example shown, $t_p = 60[s]$ and the prediction horizon is divided into $N = 6$ intervals. The ODE state trajectories are first divided into N intervals. Next, additional optimization variables s_i^x are introduced, which represent the initial state value at the start of each interval. The corresponding final value is represented by the variable x_i evaluated at (τ_{i+1}) . Looking at Fig. E.9, it is easy to see that the state trajectory at the end of each of the intervals needs to connect with the initial value of the following one. This can be done by introducing so-called "continuity constraints":

$$x_i(\tau_i) = s_i^x. \quad (\text{E.3})$$

The objective function is similarly divided into N intervals and evaluated on each one of them:

$$L_i(s_i, u_i) = \int_{\tau_i}^{\tau_{i+1}} L(x_i(\tau), u_i) dt, \quad (\text{E.4a})$$

subject to initial value constraints (E.4b), continuity constraints (E.4c) and path constraints (E.4d) evaluated only at the introduced nodes,

$$s_0^x = \bar{x}_0 \quad (\text{E.4b})$$

$$s_i^x = x_i(\tau_i) \quad (\text{E.4c})$$

$$h(s_i^x, u_i) \geq 0., \quad (\text{E.4d})$$

where \bar{x}_0 contains the current state value (i.e., at the beginning of the prediction horizon).

Thus, the MPC problem (E.4a)–(E.4d) can be written in the following compact form:

$$\min_w \Phi(w) \quad (\text{E.5})$$

subject to

$$g(w) = \begin{bmatrix} s_0^x - \bar{x}_0 \\ s_1^x - f(s_0^x, u_0) \\ \dots \\ s_N^x - f(s_{N-1}^x, u_{N-1}) \end{bmatrix} = 0 \quad (\text{E.6})$$

$$h(w) = \begin{bmatrix} h(s_0^x, u_0) \\ \dots \\ h(s_{N-1}^x, u_{N-1}) \\ h(s_N^x) \end{bmatrix} \leq 0 \quad (\text{E.7})$$

where $\Phi(w) = \sum_{i=0}^N L_i(s_i^x, u_i)$ is the objective function and w contains the discrete state and control variables:

$$w = \{x_0, u_0, \dots, x_{N-1}, u_{N-1}, x_N\}.$$

For further details and notes on application, see for instance (Kirches et al., 2012).

E.6.2 Objective function

In order to validate the hypotheses from Section E.3, the objective function (E.2a) with integrand

$$L = \sum_{i=1}^4 r_i (T_{\text{airef}} - T_{\text{ai}})^2 + r_f T_{\text{forward}}^2 \quad (\text{E.8})$$

has been used. Here, i represents the room number and r_i, r_f are scalar weights. The first term in the objective function is responsible for driving the room temperature toward the set-point, while the last term penalizes high forward temperature values.

E.7 Simulation Results

The PI controller has been compared with the model predictive controller (MPC) with different lengths of prediction horizons. All simulations have

been done for a 48-hour period. The purpose of the simulations is not to present optimal performance of the MPC as such, but rather to show the differences between the MPC and PI controller and try to verify the hypotheses in Section E.3.

E.7.1 Room Temperature Dynamics Evaluation

Figures E.10, E.11, E.12 and E.13 present simulations of the indoor air temperature for each room governed by the PI controller, plotted against its reference (set-point of the room air temperature), and the corresponding room temperature achieved by the MPC controller. The MPC is assumed to have access to future temperature references and all disturbances throughout the prediction horizon, in this case one sample (10 minutes). By simple observation of the plots it can be noticed that the MPC yields better set-point following. Table E.6 summarizes the performance of each controller; the first row shows the PI controller performance and the second row shows the corresponding MPC results. The first four columns present the convergence metric for the whole 48h test period. The last three columns shows the total power consumption of heat pump, the mean forward temperature, and the mean coefficient of performance value, which has been calculated by Equation (E.9):

$$COP = \frac{Q_{hp}}{P_{hp}} \quad (E.9)$$

$$Q_{hp} = \dot{m}_{tot} c_{cpw} (T_f - T_{rt}) \quad (E.10)$$

where \dot{m}_{tot} describes the total mass flow in the floor heating system and P_{hp} represents the measured power consumption of the heat pump.

As can be seen from Table E.6, the model predictive controller outperforms the PI/hysteresis controller in all columns. The air temperatures in room 1, 2, 3 and 4 are within the allowed hysteresis bound for 15.2, 17.4, 10.8 and 7 percentage points more of the relevant time intervals, respectively. The power consumption is lowered by 2 kW by the MPC, and the mean heat pump forward temperature T_f is 0.9 degree lower while at the same time improving the comfort through more accurate temperature set-point following.

E.7.2 Valve Dynamics Evaluation

In order to lower the power consumption and T_f while maintaining the correct delivered heat, it is required to increase the mass flow. The following analysis will give an attempt to understand the influence of the valves as

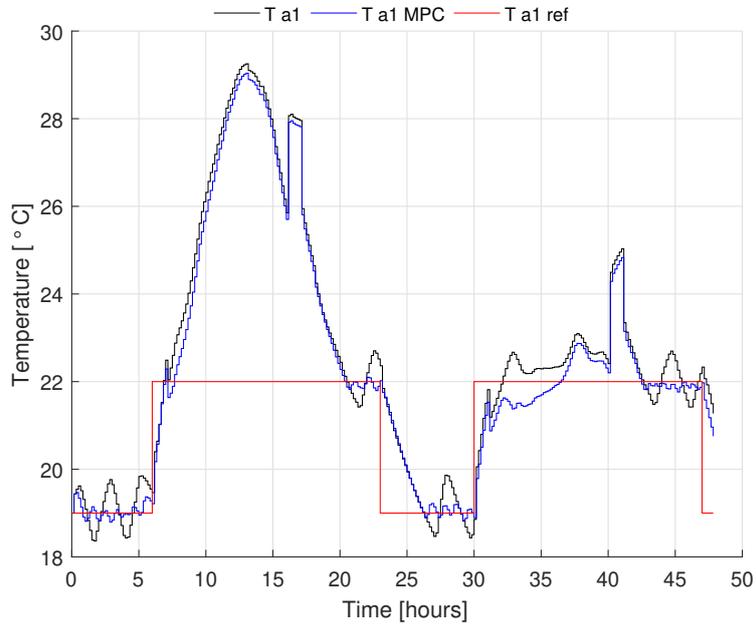


Figure E.10: MPC and PI Test Comparison: Room 1 Indoor Air Temperature.

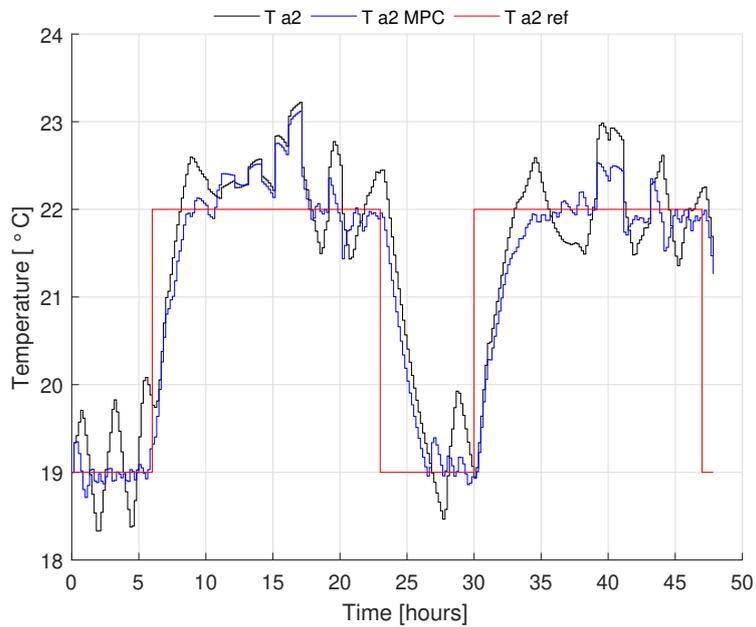


Figure E.11: MPC and PI Test Comparison: Room 2 Indoor Air Temperature.

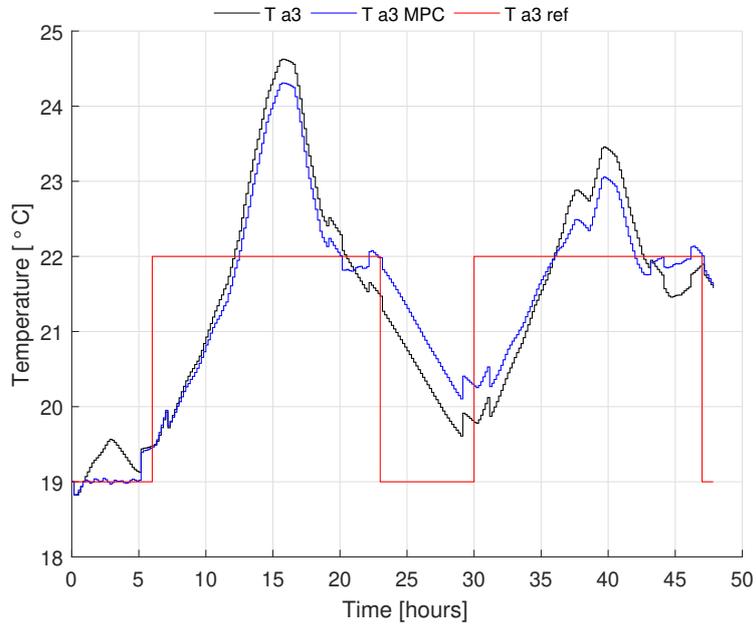


Figure E.12: MPC and PI Test Comparison: Room 3 Indoor Air Temperature.

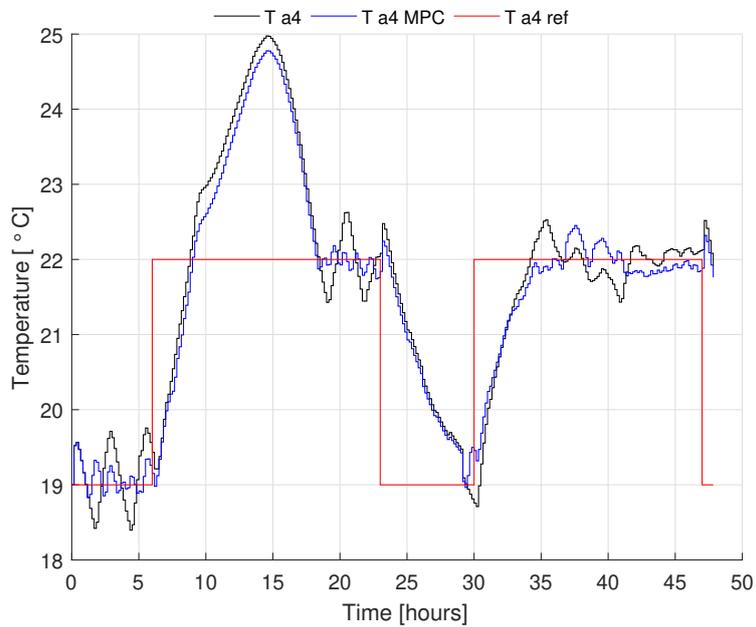


Figure E.13: MPC and PI Test Comparison: Room 4 Indoor Air Temperature.

Test	T_{a1} [%]	T_{a2} [%]	T_{a3} [%]	T_{a4} [%]	P_{hp} [kWh]	T_f [°C]	COP []
PI	37.2	57.6	34.7	47.9	23.1	35.5	3.855
MPC	52.4	75.0	45.5	54.9	21.1	34.6	3.935

Table E.6: Performance evaluation. The heat pump power consumption is lowered by 8.1%.

d_{t1} [h]	d_{t2} [h]	d_{t3} [h]	d_{t4} [h]	d_{m1} [%]	d_{m2} [%]	d_{m3} [%]	d_{m4} [%]
10.8	18.5	18.7	19.3	22.6	38.5	38.9	40.3
7.8	15.3	20.7	18.0	16.3	31.8	43.2	37.4

Table E.7: Performance evaluation for PI (1st row) and MPC (2nd row)

input, in particular whether continuous control over the valves position contributes to achieving higher COP. In this simulation, the valve positions d_i and the forward temperature T_f are thus chosen as control variables. In order to evaluate the valve dynamics the following four metrics have been chosen:

- **Total Opening Time [h]** : Table E.7, columns: d_{t1} , d_{t2} , d_{t3} , d_{t4}
- **Mean Valve Opening Degree [%]** : Table E.7, columns: d_{m1} , d_{m2} , d_{m3} , d_{m4}
- **Mean Mass Flow [$\frac{kg}{s}$]** : Table E.8, columns: m_1 , m_2 , m_3 , m_4
- **Total Number of Valve Position Changes[-]** : Table E.8, columns: d_{j1} , d_{j2} , d_{j3} , d_{j4}

Evaluations of the four metrics are listed in Table E.7 and E.8.

Looking at Table E.7, it is seen that the total valve opening time for room 1, 2 and 4 is decreased by the MPC compared to the PI controller by 3, 3.2 and 1.3 hours, respectively. Furthermore, the mean opening degrees were decreased by 6.3, 6.7 and 2.9 percentage points for room 1,2 and 4,

m_1 [kg/s]	m_2 [kg/s]	m_3 [kg/s]	m_4 [kg/s]	d_{j1} []	d_{j2} []	d_{j3} []	d_{j4} []
0.0388	0.046	0.0123	0.0186	17.0	21.0	7.0	15.0
0.0288	0.0403	0.0141	0.018	79.0	112.0	39.0	64.0

Table E.8: Performance evaluation for PI (1st row) and MPC (2nd row) (cont.)

respectively. According to Table E.8, the mean mass flows were lowered by 0.01, 0.0057, and 0.0006 kg/s for room 1, 2 and 4, respectively. On the other hand the number of position changes increased significantly, by 62, 91 and 49 times for room 1, 2 and 4, respectively. As for room 3, its total opening time got increased by 2 hours, but mean valve opening degree, mean mass flow and number of position changes increased, by 4.3 percentage points, 0.0018 kg/s and 32 times, respectively.

Concluding, it can be said that MPC tends to open/close valves much more often and keep the valves open for shorter period of time than in the case of a baseline controller. It is also worth noting that the total mass flow was in fact slightly *lower* than for the PI controller.

E.7.3 Forward Temperature

One of the primary goals of this project is obtaining a higher COP for the heat pump through optimization of its operating conditions. As can be seen from Equation (E.9), in order to achieve this, one should control the forward temperature and valves such that either the power consumption is decreased while the total heat input is maintained at the same level, or increase the heat input while keeping the power consumption constant, or some combination of the two.

Figure E.14 presents a comparison of the forward temperature T_f commanded by the PI and MPC controllers. In addition to the sample-by-sample values of T_f (PI: thin black line, MPC: thin blue line), the figure shows hourly mean values of T_f (PI: thick black line, MPC: thick blue line). Red dots indicate where the MPC demanded higher T_f values.

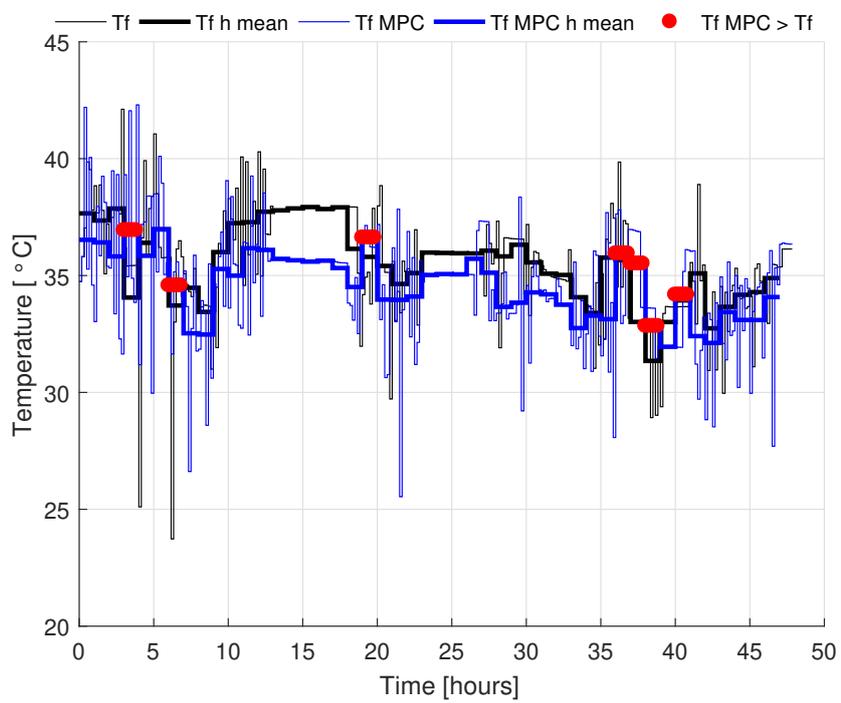


Figure E.14: MPC and PI Test Comparison: Heat pump Forward Temperature. Thin lines show sample-by-sample (ten-minute) values, while thick lines show hourly averages.

Overall, the MPC succeeded in lowering the mean T_f by 0.9 degree. Throughout the 48h simulation, the MPC kept T_f lower for about 75 % of time.

E.7.4 Power consumption and COP evaluation

This section investigates the differences in power consumption and COP between the PI and MPC controllers. For this purpose, traces of the differences in power consumption demanded during the simulations have been plotted together with the difference of sum of all room valve opening degrees in Figure E.15.

The figure should be read in the following way. The black line indicates the power differences, meaning that if it takes positive values, then the PI controller resulted in larger power consumption than the MPC, and if the power difference is negative the MPC resulted in larger power consumption. The red dots indicate the difference of the sum of all valve opening degrees. It turns out that there is a significant correlation between the two data sets presented, with linear correlation coefficient $r=0.47$ and p-value equal to 0. In conclusion it can be said that the ability to control the valves gives the MPC controller a large advantage in terms of power consumption savings. Furthermore, looking at the power difference trace, one can notice that the MPC has shifted the power consumption to different time periods.

E.7.5 Prediction Horizon Length

So far it has been shown that it is possible to achieve lower power consumption and higher COP by introducing a predictive controller, even if it has only a short prediction horizon. This section focuses on the MPC itself and contains simulation results for controllers with various prediction horizons (1, 2, 3, 4, 5 steps, where each step corresponds to 10 min). Table E.9 is an extension of Table E.6 and presents convergence metric values for the four rooms, the heat pump power consumption, the mean forward temperature and the mean COP for the PI controller as well as for the MPC with the listed horizon lengths. The table also lists the relative improvements in heat pump power consumption for each MPC simulation relative to the PI controller.

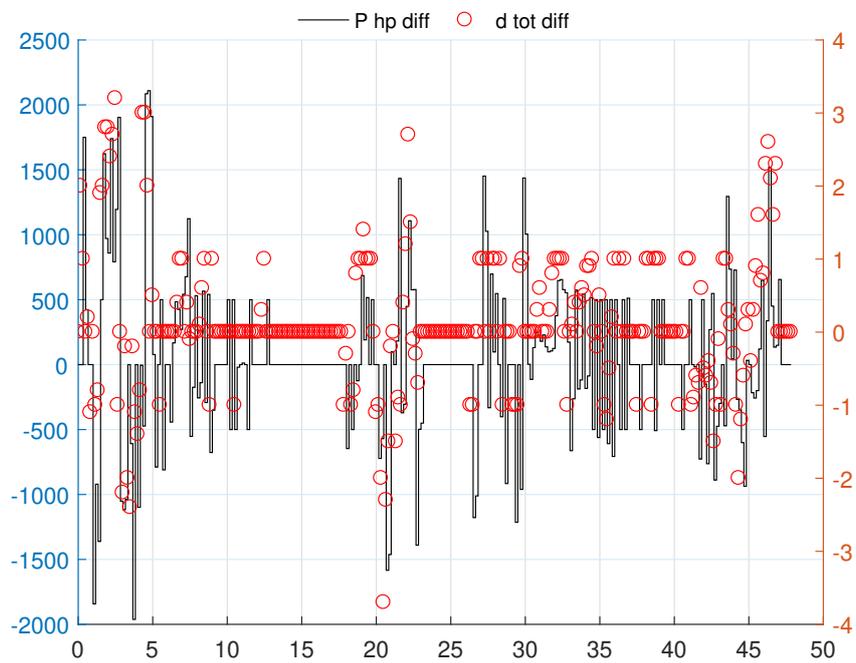


Figure E.15: MPC and Baseline Test Comparison: Power and valve differences. Left scale: Power (W), Right scale: Sum of opening degree, where 1 corresponds to 100%.

	T_{a1} [%]	T_{a2} [%]	T_{a3} [%]	T_{a4} [%]	P_{hp} [kWh]	ΔP_{hp} [%]	T_f [°C]	COP
PI	37.2	57.6	34.7	47.9	23.1	-	35.5	3.855
1	52.4	75.0	45.5	54.9	21.1	8.66	34.6	3.935
2	51.7	77.1	46.2	55.9	20.8	9.96	34.7	3.939
3	50.7	77.4	46.5	56.6	20.4	11.7	34.1	3.804
4	50.7	76.4	47.2	55.9	20.2	12.6	35.1	3.942
5	51.0	76.0	46.5	55.9	20.1	13.0	35.2	3.942

Table E.9: Performance evaluation for control with different horizon lengths. Higher convergence metric scores indicate that the temperature stays within the comfort band for a greater ratio of the simulation.

Clearly, increasing the prediction length results in decreased power consumption and increased COP, but the improvements quickly become small as the horizon is increased. The simulation with the 5-step-ahead MPC resulted in slightly higher forward temperature than the one-step ahead MPC. Small differences in the convergence metric can be noticed, but the comfort is not compromised noticeably, which is crucial.

Further MPC controller comparisons have been made for the MPC with one and five steps ahead prediction horizons. For the sake of consistency with the previous comparisons, a valve dynamics evaluation will be done next; see Tables E.10 and E.11.

The tables indicate that the previously observed trend continues; by increasing the prediction horizon, the total opening time, mean opening degree and mean mass flow have all been decreased even further for room 1, 2 and 4, whereas they were increased for room 3.

d_{t1}	d_{t2}	d_{t3}	d_{t4}	d_{m1}	d_{m2}	d_{m3}	d_{m4}
7.817	15.283	20.733	17.967	16.3	31.8	43.2	37.4
6.35	14.85	21.317	17.867	13.2	30.9	44.4	37.2

Table E.10: Performance evaluation for 1-step MPC (top row) and 5-step MPC (bottom row)

m_1	m_2	m_3	m_4	d_{j1}	d_{j2}	d_{j3}	d_{j4}
0.0288	0.0403	0.0141	0.018	79	112	39	64
0.0231	0.0399	0.0145	0.0179	72	113	38	63

Table E.11: Performance evaluation for 1-step MPC (top row) and 5-step MPC (bottom row)

The number of valve position changes was decreased by the 5-step ahead MPC by nearly 10% in case of room 1, while maintaining approximately the same activity for the other rooms.

E.7.6 Partial conclusion

In conclusion it can be stated that introduction of predictive control gives rise to significant performance increases in terms of heat pump power consumption – up to about 13% – which can undoubtedly be improved further than illustrated above by better tuning of the model and control optimization. The observed increases in COP factor were more modest than expected, however.

It is also noted that the decrease in electricity consumption in the heat pump was actually followed by a slight increase in comfort (in terms of better set-point following), which must be due to better flow control into the underfloor heating sections in each room.

It was furthermore noted that increasing the prediction horizon in the MPC yielded modest, but consistent performance improvements.

However, the aforementioned improvements come at the expense of having to have a model of the house and heating system. In the following, it will be investigated if the rather knowledge-expensive model can be avoided while still achieving some of the advantages of the predictive control scheme.

E.8 Neural network-based predictive control

In addition to the model-predictive controller detailed above, a data-driven approach has been investigated as well. The idea is that by employing a ‘learning’ controller that does not rely on a pre-tuned and potentially complicated model of the house, floor heating system and heat pump, the control concept can more easily be rolled out to new houses. In principle, a ‘learning’ controller does not require any a-priori knowledge of layout or materials of the building (flooring, walls, windows etc.), the type of heat pump, and so forth.

In this project, it has been chosen to investigate a solution based on Artificial Neural Networks (ANN). The work was published at the European Control Conference 2015 in Linz, Austria, in the attached paper (Vinther et al., 2017). Only an overview of the methodology and the main results is given here; the reader is referred to the attached paper for details.

E.9 System configuration

The overall setup is shown in Figure E.16. During operation, in- and output data is collected from the system (‘House’ in the figure) and used to train a prediction model (‘NN’ in the figure). Using the trained ANN model to predict the system dynamics in response to hypothetical inputs, it is then possible to perform nonlinear model predictive control in a similar fashion as in the previous supervisory control approach.

As in the previous case, the most important values to include in the model are the room indoor air temperatures (T_{ai}). Unlike the previous case, it is now assumed that the forward temperature can be controlled (within pre-set limits), whereas valve settings are dictated by setting room temperature references. Various combinations of these and other measurements (disturbances) were considered as inputs to the ANN model to investigate its prediction capabilities under various conditions; an example is shown in Figure E.17. See Table 1 in (Vinther et al., 2017) for the complete list.

E.9.1 Artificial neural network model

A general fully connected Multi-Layer Perceptron ANN structure is illustrated in Fig. E.18. Inputs to the network are composed of augmented state measurements $x \in \mathbb{R}^n$ (actual states, i.e., current and previous temperature data) and disturbances (solar irradiation and other incident heat)

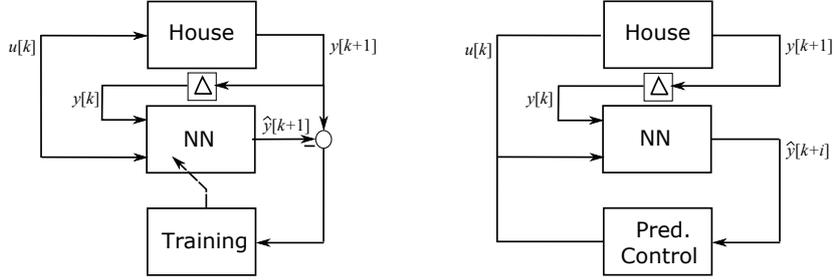


Figure E.16: Learning control setup. Left: training configuration; u is provided externally, for instance by a controller or as a pseudo-random signal, and a training algorithm adjusts the ANN model weights to minimize the network’s prediction error. Right: control configuration; the trained ANN model provides predictions of future outputs for a predictive controller, which in turn generates control signals. Δ is a delay operator.

along with controllable inputs $u \in \mathbb{R}^m$. The outputs from the network are calculated via the formula

$$\hat{y}[k + 1] = W_2 \tanh(W_1 \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} + b_1) + b_2 \quad (\text{E.11})$$

where the hyperbolic tangent function is a common choice of neuron function, often referred to as the ‘tangent-sigmoid’ in neural network literature. Any differentiable function could in principle be used, however. k is sample number (assuming a fixed sampling period of one hour). W_1, W_2, b_1 and b_2 are weight and bias-matrices containing the parameters of the model. These $nm + n + pn + p$ parameters need to be *trained*, to give a mapping from u and x to y that approximates the measured data well. Simply put, the training process consists of adjusting the weights and biases to minimize the sum of squared prediction errors

$$J(W_1, W_2, b_1, b_2) = \sum_{k=1}^N (y[k] - \hat{y}[k + 1])^T (y[k] - \hat{y}[k + 1]) \quad (\text{E.12})$$

The celebrated Levenberg-Marquardt algorithm (Hagan and Mehraj , 1994) with bayesian regularization, which minimizes a combination of squared errors and weights in order to produce an ANN that generalizes well, was chosen for training of the ANN (Matlab function `trainbr`). An example of prediction performance of trained ANN models is shown in Figure E.19.

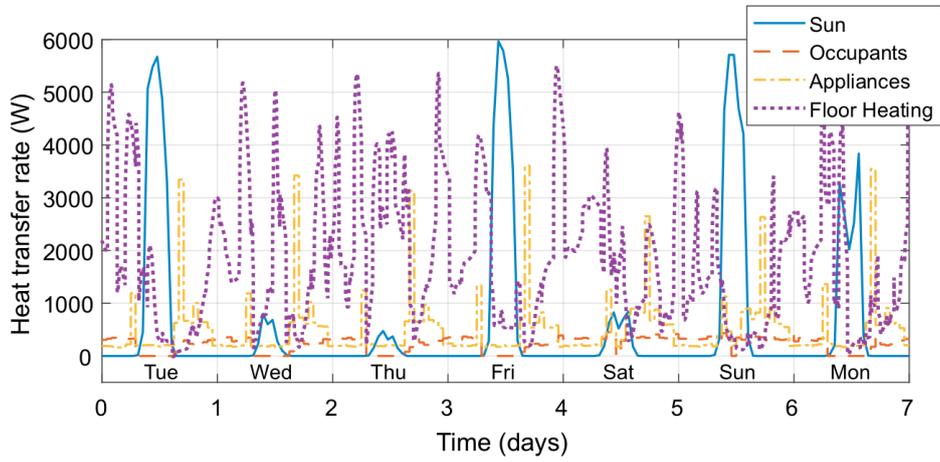


Figure E.17: Example of inputs used for training.

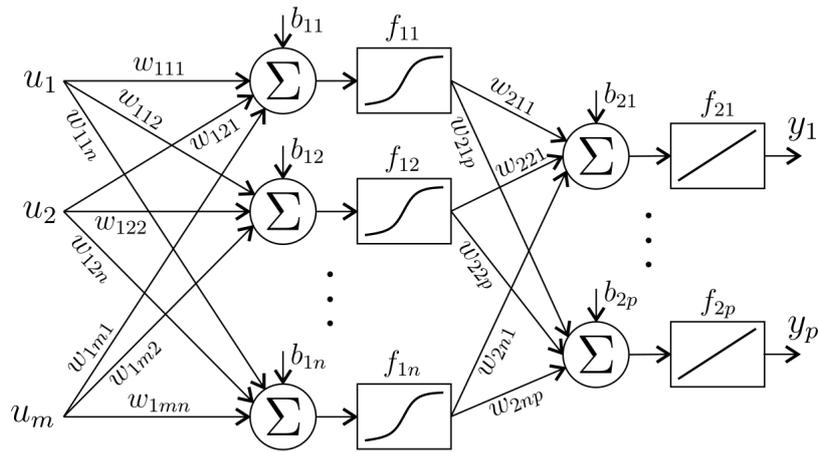


Figure E.18: Multi-layer perceptron architecture.

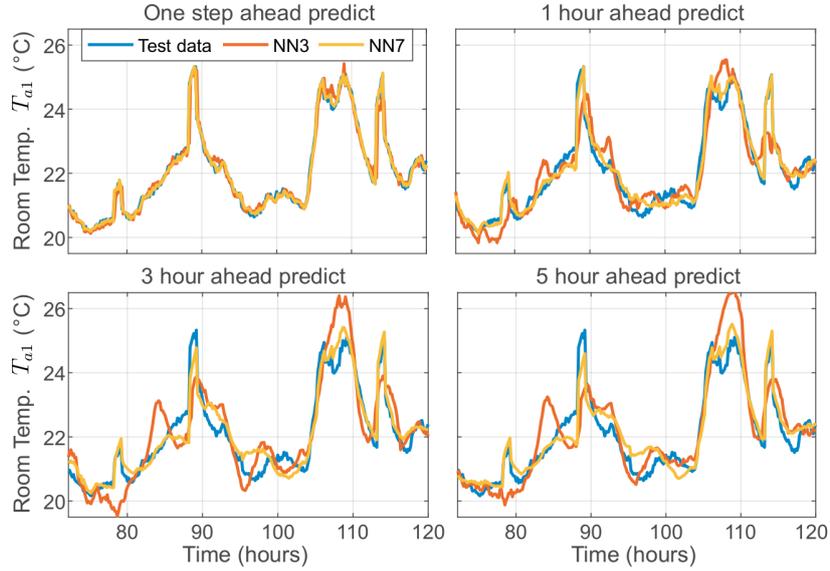


Figure E.19: Example of prediction performance. ‘NN3’ and ‘NN7’ refer to representative ANN models with different amounts of information available; see Section E.9.2 for further explanation.

E.9.2 ANN-based predictive control

Once the prediction model is trained to yield satisfactory performance, it can be used in the intended role for predictive control. The basic idea is largely the same as in the previous setup, namely to lower the forward temperature while keeping the correct room temperatures. The minimization problem is in this case defined as

$$\underset{u(\cdot)}{\text{minimize}} \quad \sum_{i=0}^{N_h} T_f[k+i] + r(y_{\text{ref}}[k+i] - y[k+i])^T (y_{\text{ref}}[k+i] - y[k+i]) \quad (\text{E.13a})$$

$$\text{subject to} \quad \hat{y}[k+i] = \mathcal{N}(u[k]), \quad (\text{E.13b})$$

$$u_{\min} \leq u[k] \leq u_{\max}, \quad (\text{E.13c})$$

$$y_{\min} \leq y[k] \leq y_{\max} \quad (\text{E.13d})$$

where the vector inequalities should be interpreted element-wise and $\mathcal{N}(\cdot)$ is shorthand for the ANN model (E.11).

Due to the presence of the nonlinear ANN model in (E.13b), the problem is likely to be non-convex and not suitable for standard solvers. It was therefore investigated, if the optimization problem could be solved with

less conventional methods; in this case, Genetic Algorithms, e.g. (Renders et.al., 1992). Genetic Algorithms basically employ a number of randomly selected candidate control trajectories, whose performance can be evaluated using (E.13a). The best-performing candidates are kept and combined (so-called *crossover*), while poorly performing candidates are eliminated from the population. The candidate pool is then re-populated with new, randomly chosen candidates, and the selection process can be repeated.

Figure E.20 shows simulations of four different control strategies: just keeping the forward and room air temperature set-points constant, using heating curves to calculate the forward temperature, and optimization based on two different neural network models, one with an unrealistic amount of input information available (NN7) and one with a more realistic amount of information available (NN3). The heating curve sets the forward temperature based on the outdoor ambient temperature and is adjusted to provide 35°C when the ambient temperature is 0°C to provide sufficient heating of all the rooms. The ANN-based concepts are allowed to adjust the forward temperature.

The main simulation results are summarized in Table E.12, which lists mean values of forward temperature and deviations from the setpoints in each room along with the mean thermal heat input to the underfloor heating system.

From the table, it is seen that all the strategies use the same average amount of floor heating Q_f during the test period to keep the room temperatures close to 22°C. The deviations from the setpoint for each room are very similar; slightly worse for the ANN approaches, but nothing that would be felt by a person in the room. On the other hand, most of the large peaks in the temperature in room 1 have been reduced a bit by turning off the heating preemptively, causing the room to be pre-cooled. This is possible due to the ability of the NN to predict ahead (see the zoomed plot in Figure E.20). Furthermore, the forward temperature was lowest with the ANN strategies – between 0.7 and 1.5°C – which can result in operation at better COP values, as explained previously.

E.9.3 Partial conclusion

The investigation show a decrease in the forward temperature of between 0.7 and 1.5°C, depending on the applied ANN controller. This reduction of the forward temperature was achieved without jeopardizing the comfort in the house. As a result, it was found that the ANN-based predictive control strategy was feasible, at least in simulation. The Genetic Algorithm-based optimization was able to provide sub-optimal, but satisfactory, solutions

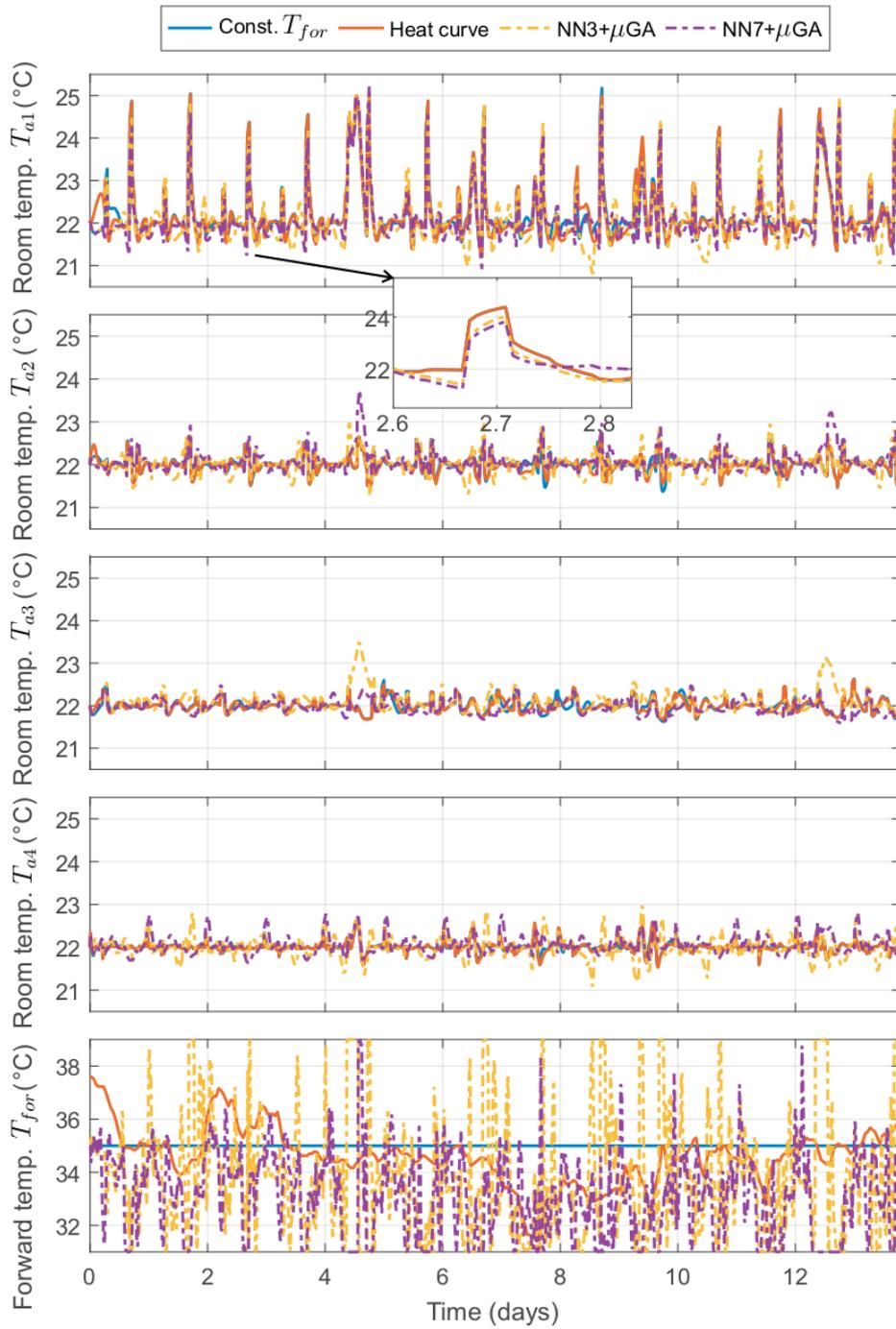


Figure E.20: Example of control performance.

Variable	Constant	Heat curve	NN3	NN7
Mean T_f [°C]	35	34.5	33.9	33.1
RMSE T_{a1} [°C]	0.71	0.72	0.72	0.63
RMSE T_{a2} [°C]	0.19	0.20	0.26	0.27
RMSE T_{a3} [°C]	0.16	0.15	0.27	0.17
RMSE T_{a4} [°C]	0.12	0.13	0.25	0.23
Mean Q_f [W]	1875	1876	1876	1875

Table E.12: Performance evaluation, ANN control. RMSE values basically means standard deviations of deviations from the setpoint in the corresponding rooms (22°C).

to the control problem once the ANN was trained well within the time allocated between samples. The method requires less modeling effort than the previous control scheme, but may also be less reliable if the training data is insufficiently rich.

The time unfortunately did not allow for development of an ANN controller optimizing both the performance of the heat pump and the heating system, however, it is anticipated that similar good results as obtained for the MPC controller may be obtained.

E.10 Conclusions

This Appendix has shown that model based predictive control yields decreased heat pump total power consumption and increased mean coefficient of performance, at least in simulation. This has been achieved by decreasing the mean heat pump forward temperature as well as the mean mass flows for room 1, 2 and 4, while increasing the mass flow for room 3. The predictive control makes more active use of the valves and is able to shift heat input to advantageous periods of the day.

More specifically, the investigation showed that significant savings are achievable when controlling the heat pump and the heat emitting system together. Savings in the electricity demand of the heat pump of up to 13% were found for the investigated period. These savings were obtained even when the overall COP system was as high as 3.9 with the traditional PI controller.

The hypotheses of savings up to 25% for a heat pump installation with traditional control and a mean seasonal COP of around 3 thus do not seem unrealistic.

The downside is the need for a model, which needs to be tuned to the individual house, heat pump and system, which can be very expensive if it has to be done by hand for each house installation. Using artificial neural networks may offset some of these costs, but it has not yet been investigated how good data availability for training may be guaranteed in a given situation.

Simulation-based investigations with a developed Artificial Neural Network (ANN) controller showed a reduction of the forward temperature for the investigated period of up to 1.5°C for the same heat pump installation as investigated with the MPC, again without compromising comfort.

Both model/supervisory control configurations require more computational resources than traditional PI controllers, but certainly not prohibitively so. All the control simulations documented above were carried out on a completely standard PC using Matlab, Modelica and off-the-shelf Python libraries. It is estimated that the supervisory controller, in either nonlinear MPC or ANN configuration, can easily be implemented and executed on a Raspberry Pi with 0.5 GB RAM or similar industry-standard hardware. Furthermore, the supervisory control configuration makes it reasonably easy to interface with the heat pump and underfloor heating subsystems.

The above results are very encouraging and leads to the conclusion that MPCs and ANNs are promising candidates for optimized control of heat pump installations where the performance of both the heat pump and the heat emitting system are optimized jointly. So the problems with poorly performing heat pump installations documented in (Poulsen et al, 2017) may, therefore, likely be solved by switching from traditional PI control to advanced combined control.

The performance of the advanced controllers were unfortunately not tested in the OPSYS test rig, nor were their annual performance determined. The development of the two OPSYS tools took more time than anticipated, leaving less time to develop and test new control strategies.

An application for a follow-up to the OPSYS project has, therefore, been submitted with the aim of developing both software and hardware of a controller capable of optimizing the performance of heat pump installations. The software will be based on the above described findings.

It is clear that further improvements can be made and that the optimization algorithm should be developed in the following directions:

- Model identification: Good model parameter identification is necessary in order to achieve accurate predictions.

- Model structure improvements: Extending the nonlinear model used in the MPC, for instance adding 4 more differential equations describing the return temperature of each room might give further improvements. It is also very likely that including return temperature measurements will give additional performance improvements.
- Different choice of objective function: Introducing the coefficient of performance directly in the objective function might be an interesting choice for evaluation.
- Tuning the objective function: Finding the correct tuning parameters in the objective functions are obviously the key to good overall performance; finding a consistent way that does not require extensive simulation would be very valuable.
- Integration with local/in-house energy sources: Should additional sources of cheap/renewable energy be available, e.g., in the form of photovoltaics or a household wind turbine, system-wide price-based optimization could be considered as well.

Bibliography

- K. Vinther, T. Green, S.Ø. Jensen and J. Bendtsen: Predictive control of Hydronic Floor Heating Systems using Neural Networks and Genetic Algorithms. *Proc. IFAC World Congress*, 50:1, pp. 7381–7388, 2017.
- C. Kirches, L. Wirshing, H.G. Bock and J.P. Schlöder: Efficient direct multiple shooting for nonlinear model predictive control on long horizons *Journal of Process Control*, 22:3, pp. 540–550, 2012.
- M.T. Hagan and M.B. Mehnaaj: Training Feedforward Networks with the Marquardt Algorithm *IEEE Transactions on Neural Networks*, 5:6, pp. 989–993, 1994.
- J.M. Renders, J.P. Nordvik and H. Bersini: Genetic algorithms for process control: a survey *Proc. IFAC Symp. Artificial Intelligence in Real Time Control*, pp. 323–328, 1992