Setting up a framework for model predictive control with moving horizon state estimation using JModelica

Mats Vande Cavey[a], Roel De Coninck[a,b], Lieve Helsen[a]
[a]KU Leuven, Department of Mechanical Engineering, 3001 Leuven, Belgium,
[b]3E nv., 1000 Brussels, Belgium,
mats.vandecavey@mech.kuleuven.be, roel.deconinck@3e.eu, lieve.helsen@mech.kuleuven.be

Abstract

A model predictive control framework for optimal heating of a residential building is proposed. The control inputs are applied to a virtual building emulator model using a limited amount of measurements. State estimation is implemented using moving horizon estimation to reinitialize the states of the controller model in every time step. To implement the moving horizon estimation, the Modelica equations had to be modified. A stochastic input is declared at the controller model state equations to represent the process noise (model error). The state estimation significantly improves the output matching between emulator and controller model. The JModelica optimization framework proves to be satisfactory for this first, limited case investigated here. Future work will focus on the extension to different models and prediction errors within the framework developed here.

Keywords: Model Predictive Control; Moving Horizon Estimation; State estimation; JModelica; Modelica

1 Introduction

Building heating systems are usually controlled using a heating curve that determines the supply water temperature based on the outside temperature. An increased interest in optimal control is encouraged by the widespread adoption of optimal control in other engineering domains. Model predictive control (MPC) is a general purpose control scheme that involves repeatedly solving a constrained optimization problem. Optimal control inputs are computed using a reduced order controller model and are applied to an emulator model or a real case. Measurements of (one or more) states in the emulator model or real case are used to reinitialize all states of the controller model.

This paper proposes a general framework for MPC with state estimation using Modelica models and JModelica. The main focus lies on the implementation of moving horizon state estimation.

In the literature, otften this step is either bypassed by using the controller model as an emulator model or by assuming all states can be measured and thus the controller model is updated perfectly. However, these assumptions do not hold when using MPC in real buildings where state estimation is thus needed to update all controller model states.

Because deterministic models cannot explain the differences between the system model output and the real system observations, stochastic models are needed. Therefore, the deterministic model equations are extended with a noise term, to overcome the simplifications in the model and the input uncertainty. State estimation computes this noise term based on statistical knowledge of this extra term and system observation.

For state estimation, often a Kalman filter is considered, which updates the states, by calculating a deterministic estimate, based on the covariance matrices of the noise. A classical Kalman filter can update the states of a linear time varying model. An extended Kalman filter can update the states of a non-linear time varying model, by linearizing the model equations in the working point. The framework described in the current paper uses Moving Horizon Estimation (MHE). MHE solves a least squares estimation that determines the optimal state estimates, based on covariance matrices of the noise.

The reason for choosing MHE over Kalman filter is twofold: first, the state estimation fits in a framework that is being developed for MPC with Modelica models to be used in real buildings. The MPC framework starts with parameter estimation of greybox controller models in Modelica using the greybox building models in a model library (*FastBuildings*) developed by De Coninck et al. [1]. MHE uses the Modelica-model differential algebraic equations (DAE) formulation directly and thus keeps the great flexibility of the greybox parameter estimation toolbox. In contrast, Kalman filter is applied to ordi-

DOI
10.3384/ECP140961295

nary differential equation (ODE) models. Second, the parameter estimation, the optimal control problem and the state estimation problem are all optimal control problems which are solved using JModelica [2]. JModelica allows solving non-linear problems using gradient-based optimization. The state estimation and optimal control problem encountered here, will always be initiated with an initial guess, based on a nearly equal former solution. This will ensure that the non-linear problem will be handled robustly. This is in contrast with often reported failing of an extended Kalman filter (EKF) when handling non-linear models [3].

In this study, the MPC is applied to a virtual, single zone residential building equipped with a floor heating system and fed by a heat pump. It is represented by a 'detailed' Modelica model, later called the emulator model which is presented in [4][4]. Only a limited number of states are measured. The next part of the paper explains the optimization framework step by step, after which the effects of state estimation are discussed.

2 Optimization framework

Figure 1 schematically shows one loop MPC, which is processed every 'open loop' time step. The disturbance inputs are ambient temperature, global horizontal solar irradiance and internal gains. The control input is the floor heating heatflux. Since a heatflux is not a physical decision variable, it is translated into a floor heating water supply temperature setpoint as a function of the measured return temperature. The temperature of the building zone is also measured, for feedback in the high level control.
The MHE problem looks at the past time window to estimate every new initial state variables of the controller model. The MHE optimization is initialized by a controller model simulation over that past period using observed control inputs. This initialization is an important part of solving the state estimation problem, especially when the model has non-linearities.

The optimal control problem(OCP) problem looks at the future time window to optimize the control inputs over the prediction horizon The OCP optimization. is initialized using the solution of the OCP from the past MPC-step. The shorter the open loop horizon, the closer the optimal control inputs will be to the initial values. Optimization using linear models will always find the global optimum for the corresponding linear system.

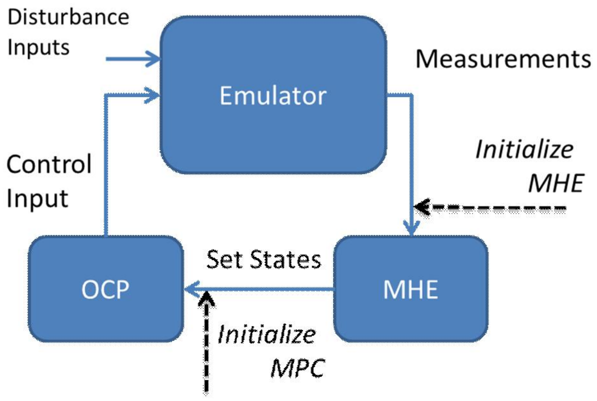


Figure 1: Outline of MPC

2.1 State estimation

Because of model mismatch and disturbance prediction errors, the controller model state values deviate from the emulator values. To prevent this, a moving horizon estimator is implemented to correct the states of the controller model based on measurements (or emulator model values). MHE can be seen as the dual of the MPC as it solves an optimal control problem over the past horizon to fit some measurement(s). The difference is that MHE takes the fixed past control inputs and optimizes the model error: it determines the process noise w over the optimization horizon for every state. To understand the concept of process noise, one must look at the model as a stochastic model. Take a look at the notation of a simple, explicit form of the model equations, with x the model states and y the output. w represents the model error (process disturbance/noise) and v the output error (measurement disturbance/noise):

$$\dot{x} = f(x,u) + w$$
$$y = g(x,u) + v$$

The explicit formulation of the model error w means that a stochastic model allows for a difference between the state change according to a deterministic model: $f(x,u)$, and the state change in the real system: $\dot{x}$. Solving a state estimation problem means finding the difference w over the past time, such that the stochastic model accurately represents the real system behavior.

To solve the state estimation as an optimization problem, the following least squares problem should be solved by determining w_k.

$$\min_{x0,\{w_k\}_{k=0}^{T-1}} \sum_{k=0}^{T-1} \|v_k\|^2_{R^{-1}} + \|w_k\|^2_{Q^{-1}} + \|x_0 - \overline{x_0}\|^2_{P_0^{-1}}$$

This is known as the 'full information problem', which minimizes the weighted sum of the output noise *v* (on some measured outputs *y)*, the model noise *w* and the initial state x_0 over the past time. The weights can be determined using the inverse covariance matrices of the model error (Q^{-1}), the measurement noise (R^{-1}) and the initial condition (P^{-1}). A larger covariance of a model state thus leads to a smaller weight in the objective function and a larger value for w_k as optimal solution. A model state with large covariance will have larger deviations from the real system, which is to be expected from a larger covariance.

As time progresses, this full information problem becomes computationally infeasible. Moving horizon estimation removes this difficulty by considering only the most recent N measurements. An arrival cost is formulated to represent the information about the initial states and the measurements prior to N. The arrival cost cannot be calculated exactly (full information problem) and therefore approximations are used. For stability reasons often the prior measurements are disregarded. The initial value x_0 is then set according to the solution of the last MHE update and the arrival cost is thus a constant value. Another approach is to calculate deterministic updates of the initial MHE states (k=T-N) using an extended Kalman filter (EKF). The first approach is followed here and prior measurements are disregarded. The MHE problem translates to:

$$\min_{x0,\{w_k\}_{k=T-N}^{T-1}} \sum_{k=T-N}^{T-1} \|v_k\|_{R^{-1}}^2 + \|w_k\|_{Q^{-1}}^2$$

or

$$\min_{\{w_k\}_{k=0}^{T-1}} \sum_{k=T-N}^{T-1} \left\|y_{meas_k} - g(x_k,u_k)\right\|_{R^{-1}}^2 + \|x_{k+1} - f(x_k,u_k)\|_{Q^{-1}}^2$$

With $\|.\|_A$ the 2-norm with weights A. As explained before, the weights are determined by the covariance matrices. The covariance matrix (*Q*) for the is a diagonal matrix with covariance of each model state on its diagonal. The covariance matrices can be considered a tuning parameter. This is shown Figure 2 where the R^{-1} values (*weight_meas* in the figure) takes three different values while keeping the Q^{-1} value constant. In order to get the stochastic model output to agree more with the measurements: increase R^{-1}/decrease Q^{-1}. And vice versa to agree more with the deterministic model state: decrease R^{-1}/increase Q^{-1}). In this work, the weights are not related to the covariance matrices yet. However as we estimate the greybox models using a parameter estimation, it is mathematically possible to estimate the covariance matrices along with the model parameters [5].

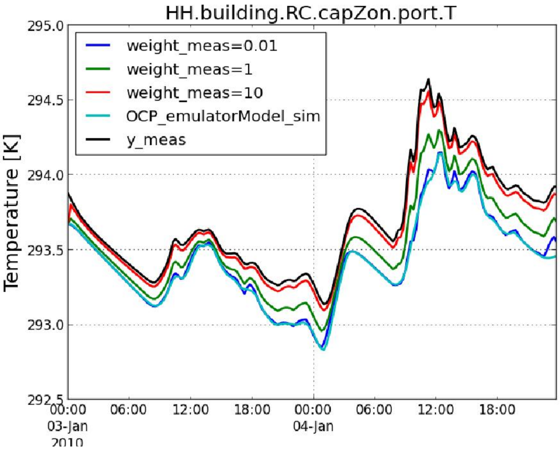
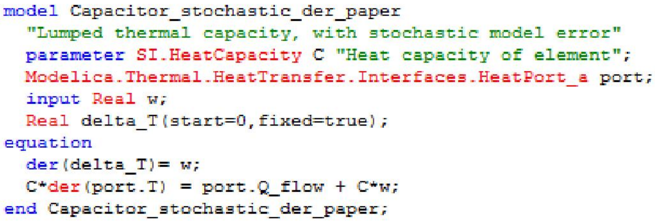


Figure 2: State estimation influence of the different weighting factors

To implement the MHE optimization problem using Modelica models, the model equations need to be adapted. Modelica model equations are by design deterministic and not stochastic. The DAE system of equations in the general implicit form is:

$$F(\dot{x}, x, u, d, p)$$

In this equation *d* represents the disturbance input and *p* the model parameters. Each equation holds and simulating a Modelica model generates a deterministic solution for the variables. To use MHE with Modelica models, we decide to add a normalized, stochastic variable *w* to every state equation (here for a heat capacity, with C=m*c). The stochastic Modelica model then looks like:



```
model Capacitor_stochastic_der_paper
  "Lumped thermal capacity, with stochastic model error"
  parameter SI.HeatCapacity C "Heat capacity of element";
  Modelica.Thermal.HeatTransfer.Interfaces.HeatPort_a port;
  input Real w;
  Real delta_T(start=0,fixed=true);
equation
  der(delta_T)= w;
  C*der(port.T) = port.Q_flow + C*w;
end Capacitor_stochastic_der_paper;
```

Figure 3: Stochastic Modelica model heat capacity

The process noise w is modelled as an input to this capacity model. The inputs ‘w’ of all states are optimization variables in the MHE problem. The equation shows that *C*w* physically represents an extra heat flux, not included in the original deterministic equation. It can also be seen that *delta_T* is a temperature deviation, which is the model error for the temperature (state). Since the problem is continuous in Modelica, the variable *w* can be determined in the MHE over the past N steps, to find the deviation *delta_T* of the state *T*. The Modelica model can be reconverted into a deterministic model for simulation purposes by setting the process noise to zero.

2.2 Optimal control problem

The optimal control problem is solved using a reduced order controller model. This controller model is identified based on monitoring data by use of a grey-box modeling approach. This approach starts from the Modelica library FastBuildings which defines potential low-order model candidates. For each potential model, parameter estimation is carried out, and the resulting models are compared using cross-validation, confidence intervals and other residual analyses. The best model is selected as controller model. This grey-box modeling approach is described in more detail in [1][6]. For the case investigated in this paper the time series consists of simulation data of the building zone temperature obtained by the emulator model, but this could as well be measured data. A third order resistance-capacitance (RC) model for the building zone fed by a heat pump with constant COP of 3.58 gives a good fit (rmse of 0.08 °C) to the time series generated by the emulator model. Figure 5 shows a visual representation of the selected RC model from the FastBuildings library. There are three heat capacitors which all have a different temperature (state): capEmb.T (heater), capWal.T (wall) en capZon.T (zone). The last state is an output as it is measured in the emulator model.

$$\min_{Q_{hea}} \quad \alpha * J_e + J_d$$

$$J_e = \int_T^T \frac{Q_{hea}}{COP}$$

$$J_d = \int_0^T (\beta * {\epsilon_h}^2 + {\epsilon_c}^2), \qquad with \begin{cases} \epsilon_h \geq T_z - T_{comf,max} \\ \epsilon_c \leq T_z - T_{comf,min} \end{cases}$$

$$\begin{aligned} & s.t. \\ & F(\dot{x}, x, u, d, p) = 0 \\ & Q_{hea} \leq Q_{HP,max} \\ & Q_{hea} \geq 0 \\ & T_z(T) \geq T_z(0) - 10 \\ & T_z(T) \leq T_z(0) + 10 \end{aligned}$$

Figure 4: multi-objective optimal control problem formulation.

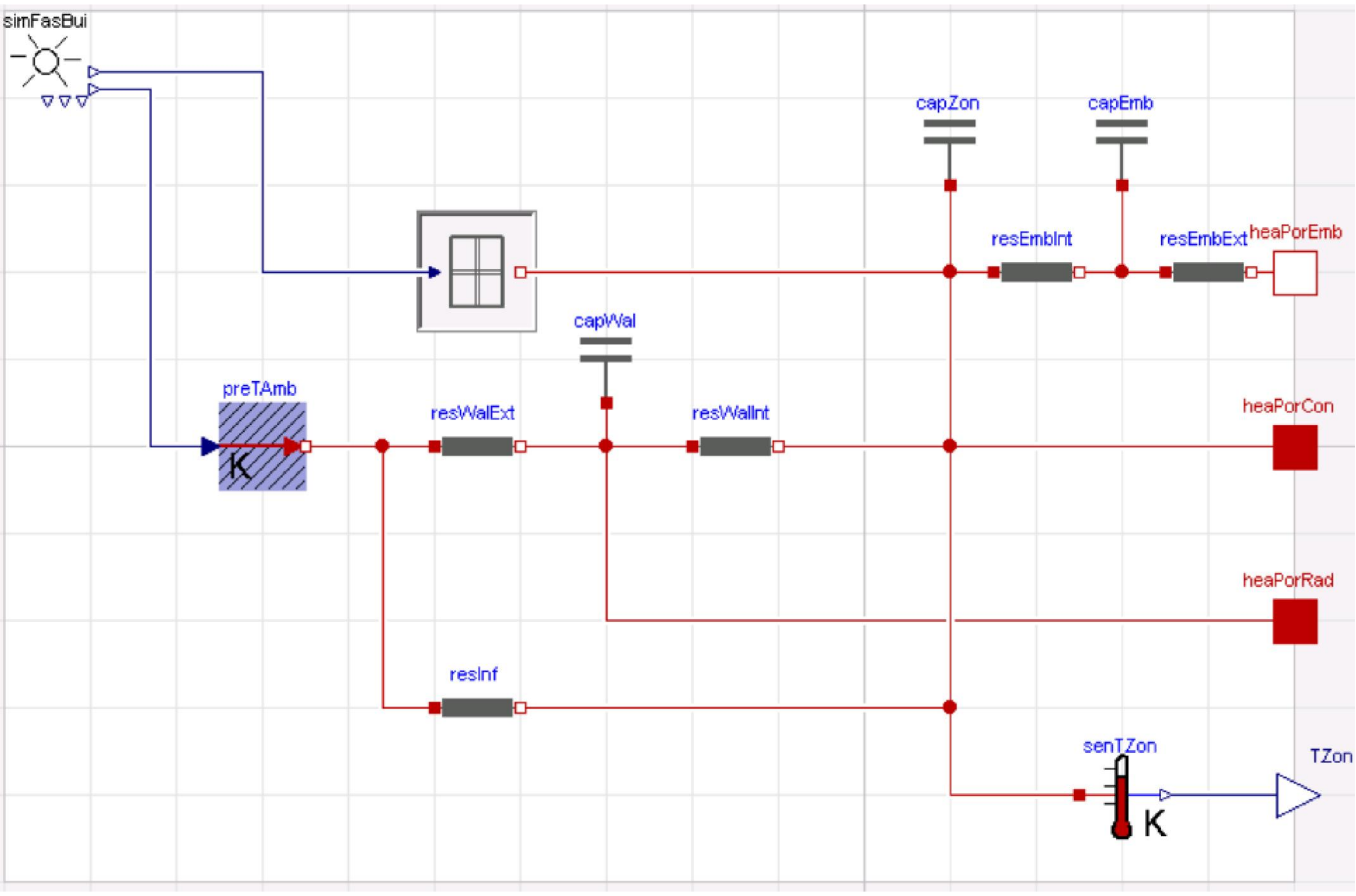


Figure 5: Visual representation of the third order greybox model in Modelica.

The optimal control problem is multi-objective. It minimizes both the energy use and the thermal discomfort and is based on a PhD regarding MPC in buildings [7]. The problem formulation is shown in Figure 4. The control input u (here: u=Qhea) determines the heat from the heating system to the zone to control the zone temperature T_z. The first term in the objective function J_e is the total electricity used by the heat pump, while the second term in the objective function J_d is the weighted sum of thermal discomfort (overheating and undercooling).

This multi-objective approach treats the discomfort boundaries *Tcomf,min/max* as soft, asymmetric (for β different from 1) constraints through the slack variables ϵ. The external inputs (/disturbances) d are ambient temperature, solar irradiance, internal gains and the electricity price. The latter is kept constant. For the disturbances, perfect predictions are used. The optimal control input is applied to the emulator model and the building zone temperature is measured.

2.3 MPC

The MPC framework is written in Python, because JModelica is interfaced in Python. It is tested on an emulator model of a single zone residential building with floor heating emission system and a heat pump for heat production [4].

The future horizon over which the optimal control problem is repeatedly solved and thus over which it needs future predictions is chosen to be 2 days. It is called the 'prediction horizon'. The past horizon over which the state estimation is repeatedly solved and thus over which it needs past measurements is chosen to be 2 days. It is called the 'state estimation horizon'.

The future horizon over which the optimally determined control inputs are repeatedly applied to the emulator model can vary. It is called the 'open loop horizon'. Choosing a shorter open loop horizon might improve the control, however it increases the number of optimization problems. In the example shown in Figure 6 it is chosen to be 1 day. This is a long period for control, but it is reasonable as we use perfect predictions. The figure also illustrates the horizons and the working principle by showing the zone temperature state.

First the state estimation (MHE) problem is solved to determine the initial state for the optimal control problem (OCP). The optimal control input determined in the OCP is then fed to the emulator model by translating the heat into a supply water temperature setpoint. The emulator is simulated with this heat input and the temperature of the building zone is measured. The measurement is the input to a new MHE problem.

This is shown in Figure 7, at March 2, 00:00h the states are updated. The temperatures in the controller model are discontinuously changed at every new iteration. The figure also illustrates the horizons and the working principle by showing the zone temperature state.

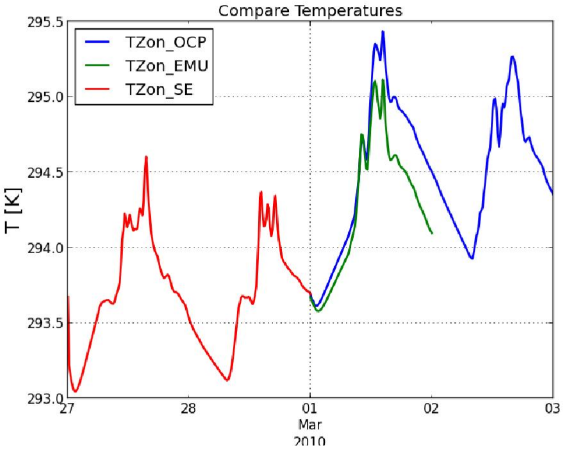


Figure 6: MPC framework, result for zone temperatures for 1 MPC iteration.

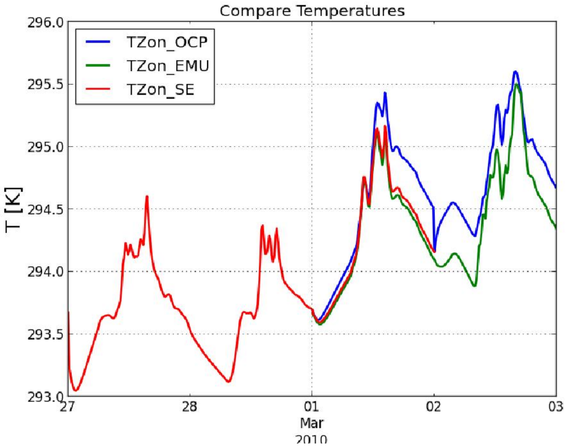


Figure 7: MPC framework, result for zone temperatures for 2 MPC iterations.

3 Results

The MPC is tested over a period of 10 days, with an open loop length of 6 hours. This means there will be four state updates throughout a day. As explained using Figure 6 and Figure 7, the initial values of the controller model states x_0 for the MPC are determined by a first solution of the MHE. The initial values of the controller model states x_0 for the MHE can be estimated freely to best capability. In order to evaluate the performance, some variations were made in the MPC formulation and settings: with and without state estimation, larger open loop (control) horizon, variation of the weighting in MHE formulation.

First a case without state estimation is solved. From is clear that the controller model would benefit from measurement feedback. The controller model (OCP) predicts a too high temperature for the states. It stays near its optimal temperature profile of around 295 K (21°C) while the emulator model (EMU) remains at a lower temperature. The third order RC model is not capable of modelling the steady state heat loss from the building zone to the soil. This third order RC model seems not appropriate for model predictive control without state estimation. The situation improves with state updates as can be seen in Figure 9. The controller model is still at its 295K (21°C), but due to the feedback of the state estimations, there is a better coupling with the emulator model. The red curve, which is in the middle of the two, represents the MHE-model state of the zone temperature. I is very close to the the measurements as the weights are chosen to be R^{-1}= {10}, and Q^{-1}= I^{3x3}, the identity matrix. This means we assume a higher covariance on the model states than on the measurements (R<Q). Remember that the inverse of the covariance determines the weighting and not the covariance matrix itself. If we change the values of these covariance matrices, we get a different result, which is visible in Figure 10.

The corrections made by the state estimator are visible in the optimal control problem. At every new control time step, the optimal control problem is solved again starting from the new (in this case lower) initial conditions. The lower temperature is immediately compensated by a control input determined heatflux from the heat pump to the building to regain thermal comfort.

The temperature change in the emulator does not quite follow the OCP. Firstly, this is because of a steady state heat loss to the soil and secondly because of the slow thermal response of the floor heating. A control input of 3kW does not spread as fast to the building zone as it would in the heater model.

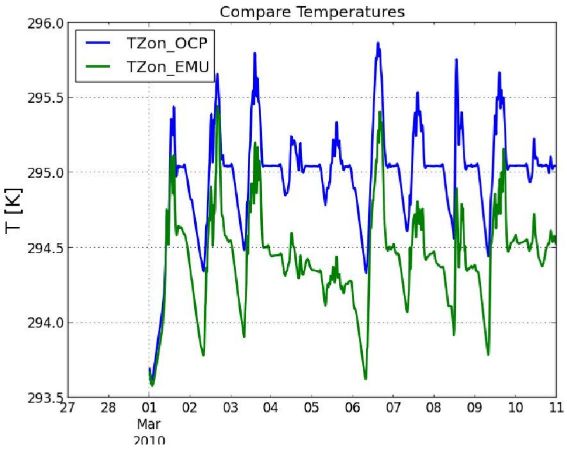


Figure 8: MPC over 10 days with no state updates.

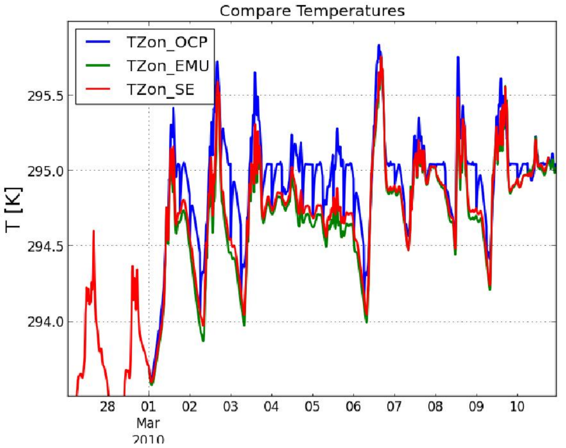


Figure 9: MPC over 10 days with 6 hourly state updates

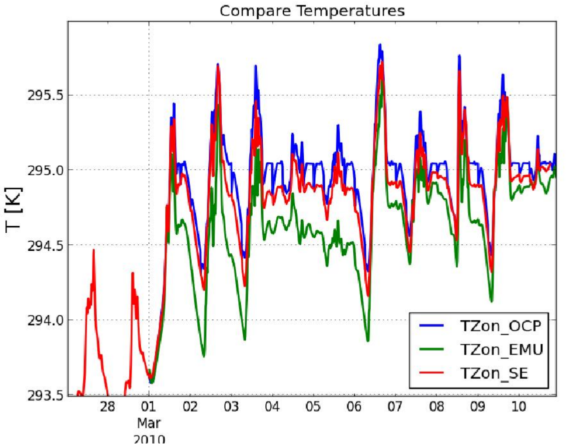


Figure 10: MPC over 10 days with 6 hourly state updates and a higher covariance for the measurements (less trustworthy)

These are two important controller model deficiencies. The first one is not so much a problem, as it can be overcome by the use of a state estimator. The second one can be misinterpreted by the state estimator as the heat to the emulator lags behind. It is therefore important to take into account the dynamics of the thermal system. If the controller time step is too small, the system might overheat as a reaction to the zone temperature measurements.

A comparison is made to MPC with a larger control time step of 1 hour update. As expected, an improved coupling pulls the emulator model building zone temperature towards the optimal values from the OCP.

In Figure 11 the coupling to the emulator model is not as would be expected. The building zone temperature of controller model is hourly updated to a higher value, and the emulator model does not seem able to catch up. One cause to this problem is found by examining the translation of the optimal control input to a physical temperature setpoint.

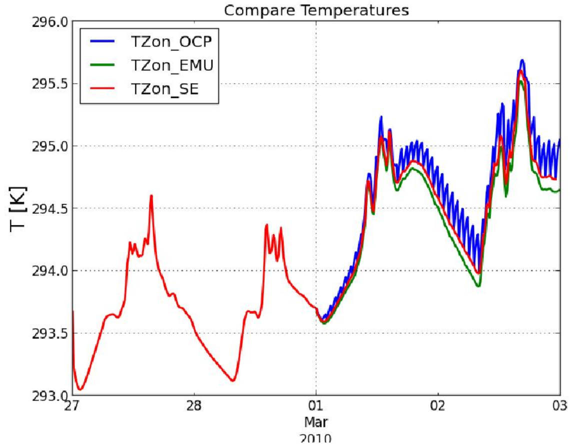


Figure 11: MPC over 1 day with 1 hourly state updates

Since the floor heating system supply temperature setpoint and the heat pump setpoint are calculated using a measurement of the outlet temperature of the floor heating, a problem always arises at startup. As the outlet temperature is low when the heating starts, the first heatflux through the floor heating is always lower than the optimal control heat flux from the OCP. This effect can be seen in Figure 12. This stresses the importance of good low level control. A PID controller could help overcome this discrepancy.

To conclude we study the result of the MHE optimization for temperature difference for the states. These temperature differences are the errors of the controller model compared to the observations. In parameter estimation processes, the model errors are studied to decide whether the estimated model is ‘good enough’. The decision criterium identifying a good model is whether the model error (or process noise) is white noise. The process noise produced by state estimation represents the same error of the model. This means that the controller models accuracy could be analyzed by looking at the error. The noise for the last MHE optimization problem is shown in Figure 13 for the three states.

The error is not white noise. This means there are phenomena which are not modelled. From the figure, it can be seen that the heater temperature has a steady state error with a periodic variation. This error might arise from a heat loss to the ground, which is not modelled in the controller model. The third order RC model cannot represent this loss through a model. A different RC model should thus be selected for parameter estimation. This would render a new better controller model, which will improved predictions allow better use of the optimal control horizon to improve energy savings.

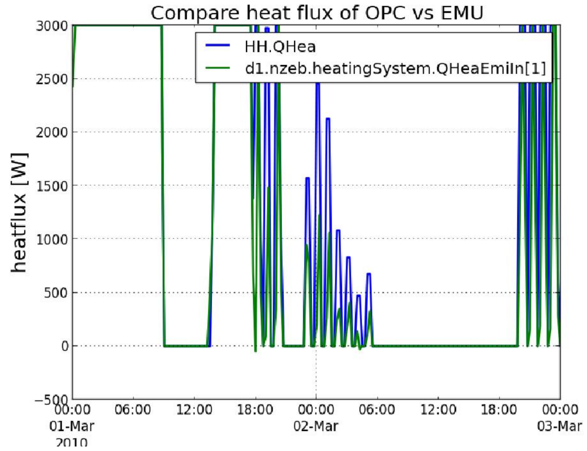


Figure 12: Optimal (HH.QHea) vs real (d1.nzeb…) heat flux

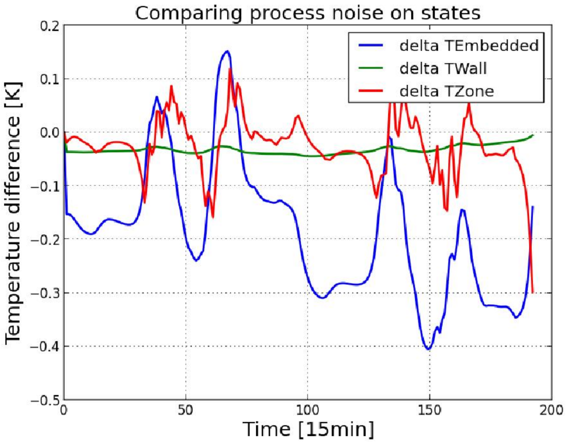


Figure 13: Process noise on states

It is also worth noting that the model error (temperature difference) in Figure 13 changes rapidly near the end. This is the case for most of the state updates of the MPC analyses with state estimation that were studied in this work. The process error on capEmb (state) is always high, but decreases rapidly in the end. The zone capZon state undergoes the inverse transition. This can be seen on Figure 14, where this last part of the results of repeated MHE solutions are shown.

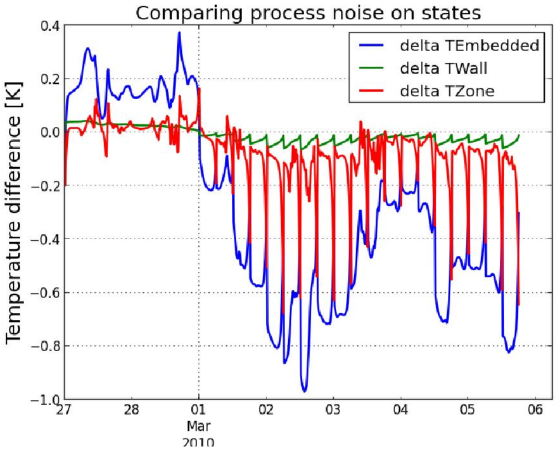


Figure 14: Process noise of the last steps in repeated solutions of the MHE problem.

This means that in the MHE model a high noise-heatflux is present towards the embedded state (*capEmb.T*) and a high negative noise-heatflux is present to the zone state (*TZon*) near the end of the interval. This seems to mean that the temperature of the Zone is kept high until the end, and is then lowered due to a non-physical heatflux. The question rises whether this behavior is to be avoided, although it seems to be mathematically correct. Since altering the weights R^{-1} and $Q^{-}1$ changes the behavior of the process noise for the states (and thus the timeseries in Figure 13 and Figure 14), the adoption of covariances for every states might mitigate this problem. Another solution might be to better initiate the MHE problem. This can be done by not disregarding the arrival cost, as is done now.

4 Conclusions

In this work, it is shown that stochastic models can be implemented in Modelica. This is an important step for optimal control framework using MPC. The MPC framework uses greybox models, which will produce an output which is different from the real building. Because not all the states in the controller model are measured states, the moving horizon estimator estimates the new initial states of the controller model. This feedback is shows in the fact that the measured state in the emulator model will be closer to the output state of the controller model. The feedback of the measurements in the emulator can be tuned by changing the weights in the objective function of the state estimator. These weights are mathematically the covariance matrices of the model states and the measurements, but since they are not always known, they can be fine-tuned to give state estimation results.

The results from the state estimation can be used to detect modeling errors or deficiencies in the controller model in the same way the residuals are checked in a parameter estimation problem. If the residuals are white noise, the controller model accurately models the system. In case the residuals are not white noise the controller model does not model the system very accurately. The state estimator can prevent the controller model from diverging from the real situation (observations).

5 Acknoledgements

Mats Vande Cavey wishes to acknowledge the EU FP7 project PerformancePlus (contract nb. 308991) for supporting his work on behalf of KU Leuven.

References

[1] R. De Coninck, F. Magnusson, J. Åkesson, L. Helsen, Grey-box Building Models for Model Order Reduction and Control, accepted for oral presentation at 10th International Modelica Conference, Lund, Sweden, 2014.

[2] J. Andersson, J. ° Akesson, F. Casella, and M. Diehl, “Integration of CasADi and JModelica.org,” in 8th International Modelica Conference, Dresden, Germany 2011.

[3] E. Haseltine, J. Rawlings, Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation, Industrial & Engineering Chemistry Research 2005 44 (8), pp. 2451-2460, 2005.

[4] R. De Coninck, R. Baetens, D. Saelens, A. Woyte, L. Helsen, a Rule-based demand-side management of domestic hot water production with heat pumps in zero energy neighbourhoods, Journal of Building Performance Simulation, 2013.

[5] Niels Rode Kristensen, Henrik Madsen, Sten Bay Jørgensen, Parameter estimation in stochastic grey-box models, Automatica, Volume 40, Issue 2, pp. 225-237, February 2004.

[6] R. De Coninck, L. Helsen, A tool chain for model based predictive control of buildings based on grey-box models, Intelligent Building Operations location:Boulder, Colorado, USA, 2013.

[7] C. Verelst, Model Predictive Control of Ground Coupled Heat Pump Systems in Office Buildings, Leuven, Belgium, Ph.D. dissertation, Department of Mechanical Engineering, 2012, p 73.