MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

An Extended Luenberger Observer for HVAC Application using FMI

Bortoff, S.A.; Laughman, C.R.

TR2018-205 March 15, 2019

Abstract

In this paper we show how a Functional Mockup Unit (FMU) may be used for the realization of an Extended Luenberger Observer (ELO), which may be considered the deterministic version of an Extended Kalman Filter (EKF). The ELO has advantages over an EKF in some situations, such as lower computational burden and improved convergence. Nonlinear observers, such as those that make use of changes of coordinates to linearize, or approximately linearize the estimate error, are continuoustime dynamical systems that use so-called output injection to modify the dynamics of a model. Output injection provides a similar feedback effect as the correction step of an EKF. However, nonlinear output injection is a slightly FMU different use case because the ELO is a continuous time object. It is realized by feedback around a modelsharing type of continuous time FMU, in contrast with the algorithmic realization of a discrete-time EKF, which uses the co-simulation form of FMU. We illustrate the design and realization of an ELO for a building HVAC example, in which we estimate unmeasured heat flows and unmeasured boundary conditions for use in a building "digital twin." We also make some remarks about model reduction and the challenges in realizing a conventional EKF for these types of models.

International Modelica Conference

An Extended Luenberger Observer for HVAC Application using FMI

Scott A. Bortoff[1] Christopher R. Laughman[1]

[1]Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, {bortoff, laughman}@merl.com

Abstract

In this paper we show how a Functional Mockup Unit (FMU) may be used for the realization of an Extended Luenberger Observer (ELO), which may be considered the deterministic version of an Extended Kalman Filter (EKF). The ELO has advantages over an EKF in some situations, such as lower computational burden and improved convergence. Nonlinear observers, such as those that make use of changes of coordinates to linearize, or approximately linearize the estimate error, are continuous-time dynamical systems that use so-called output injection to modify the dynamics of a model. Output injection provides a similar feedback effect as the correction step of an EKF. However, nonlinear output injection is a slightly FMU different use case because the ELO is a continuous time object. It is realized by feedback around a model-sharing type of continuous time FMU, in contrast with the algorithmic realization of a discrete-time EKF, which uses the co-simulation form of FMU. We illustrate the design and realization of an ELO for a building HVAC example, in which we estimate unmeasured heat flows and unmeasured boundary conditions for use in a building "digital twin." We also make some remarks about model reduction and the challenges in realizing a conventional EKF for these types of models.

Keywords: Estimation, Buildings, HVAC, FMI, FMU

1 Introduction

State estimation is one of the important use cases for the Functional Mockup Interface (FMI). For example, states of a nonlinear continuous-time model can be estimated from discrete-time measurements of the input and output of a plant using a continuous-discrete Extended Kalman Filter (EKF), realized using the co-simulation form of a Functional Mockup Unit (FMU) of the plant (Brembeck et al., 2014, 2011). Fundamentally, the EKF, and its various extensions estimate the state in a two-step process. In the prediction step, the EKF computes the predicted state estimate using a discretized plant model. Then in the correction step, the covariance and gain are computed as a function of the predicted state estimate, and the predicted state estimate state is corrected. The discrete-time prediction model is then initialized using the corrected state, and the process is repeated. Importantly, the two steps are coupled in a causal manner: The prediction step at time $(k+1)$ depends only upon the correction step at time k, and the correction step at time k depends only on the prediction step at time k. This fact allows an FMU to be used in an algorithm to estimate the state in the prediction step, since it can be initialized using the corrected state estimate from the previous correction step.

An *observer* is an alternative technology for estimation of the plant states and parameters. An observer is a deterministic, continuous-time dynamical system that takes as input the measured input and measured output of the plant, and produces as its output an estimate of the state of the plant. It is similar to the Kalman filter, but based on deterministic assumptions and mathematics. Fundamentally, the concept of *output injection* is used to stabilize the observer error dynamics, which govern the difference between the estimated state and the plant state. Output Injection means that a signal is injected (added) to the derivative of the observer state vector as stabilizing feedback. Because of this, it is the continuous-time dynamics of the plant *with* output injection that needs to be simulated. There are not separate prediction and correction steps.

In this paper we show how an instantiation of a model-exchange type of FMU can be used with the Dymola tool to realize output injection, enabling design and implementation of linear and nonlinear state observers and specifically the Extended Luenberger Observer (ELO). Our specific interest is to estimate unmeasured performance variables of a building and HVAC system as a part of a building "digital twin." Toward this end we have considered several alternative methods to estimate the performance variables, including various flavors of the EKF. However, these may prove too computationally burdensome for our application because the number of states can be large (hundreds), the number of measurements can be large (tens to hundreds), and the EKF can be computationally challenging because of the covariance update, although there are many techniques such as model reduction and square root filtering that are available to improve its computational efficiency. More importantly, an EKF can fail to converge, or in some cases, cause the model to fail at run time, at least for our building HVAC applications. Convergence failures are caused by some of the characteristics of the model that we consider in this paper, which are not unusual for this field of application. The model is stiff (with time constants ranging from milliseconds to several weeks — eight orders of magnitude), and is numerically ill-conditioned (with states varying 8-9 orders of magnitude because of the choice of units). Thus the Jacobian may not accurately predict the state over the fixed and usually large EKF sample time, causing it to diverge. Moreover, the model itself contains state constraints, such as a non-negative limit on mass concentrations, which can be violated at run time because of the EKF correction step, causing a run-time error.

On the other hand, the ELO is relatively simple and light-weight computationally. In its simplest form, it uses a constant feedback gain matrix that is computed at design time from the steady-state solution of a Ricatti equation, and therefore avoids the real-time covariance update and computation of the system Jacobian that is necessary for the EKF. Further, it may offer improved stability and performance advantages over the EKF (and similar filters) for certain applications because it makes use of implicit variable-step solvers for the continuous-time model.

This paper is organized as follows. In Section 2, we review the basics of the Extended Luenberger Observer. In Section 3, we construct an ELO for a case-study building and HVAC system and show some simulation results. We show how the FMU is used to allow for the output injection. Finally in Section 4 we conclude by making some observations on potential improvements of FMI to better enable realization of estimators of different types.

2 Background

Following (Zeitz, 1987), consider the nonlinear system

$$\dot{x} = f(x,u,d) \tag{1a}$$

$$y = h(x) \tag{1b}$$

$$z = g(x) \tag{1c}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, assumed measured, $d \in \mathbb{R}^q$ is a disturbance measurement, assumed measured, $y \in \mathbb{R}^r$ is the measured output, and $z \in \mathbb{R}^p$ is the performance output, assumed unmeasured. Our objective is to estimate the performance output z. The Extended Luenberger Observer is the system

$$\dot{\hat{x}} = f(\hat{x},u,d) + K(y-\hat{y}) \tag{2a}$$

$$\hat{y} = h(\hat{x}) \tag{2b}$$

$$\hat{z} = g(\hat{x}) \tag{2c}$$

where $\hat{x} \in \mathbb{R}^n$ is the state estimate, $\hat{z} \in \mathbb{R}^q$ is the performance output estimate, and K is the observer gain. System (2) is a copy of the original system, with the vector $K(y-\hat{y})$, which is called *output injection*, added to the state equations.

The state estimate error $\tilde{x} = x - \hat{x}$ is then governed by the system

$$\dot{\tilde{x}} = f(x,u,d) - f(\hat{x},u,d) - K(y-\hat{y}) \tag{3a}$$

$$\tilde{y} = h(x) - h(\hat{x}) \tag{3b}$$

$$\tilde{z} = g(x) - g(\hat{x}). \tag{3c}$$

We linearize (3) about an equilibrium $\bar{x}$ in a neighborhood of x, defining

$$F = \frac{\partial f}{\partial x}|_{x=\bar{x}}\ , \quad H = \frac{\partial h}{\partial x}|_{x=\bar{x}}\ , \quad \text{and} \quad G = \frac{\partial g}{\partial x}|_{x=\bar{x}}\ , \tag{4}$$

so that the linearized error dynamics, neglecting higher-order terms, are

$$\dot{\tilde{x}} = (F-KH)\tilde{x} \tag{5a}$$

$$\tilde{y} = H\tilde{x} \tag{5b}$$

$$\tilde{z} = G\tilde{x} \tag{5c}$$

There exists an observer gain K to make the origin of (5a) locally exponentially stable if the pair (F,H) is detectable.

There are many methods for the design of the observer gain K e.g. (Luenberger, 1971; Chen, 1984; Friedland, 1986). In fact, more generally we can consider nonlinear changes of state coordinates $z = \Phi(x,u,d)$, nonlinear changes of the output coordinates $\xi = \Gamma(y)$, and nonlinear output injection $K(y)$ as in (Krener and Isidori, 1983; Krenner and Respondek, 1985; Hou and Pugh, 1999). Research on methods for computing these remains an active area of research e,g, (Boutat et al., 2009; Tami et al., 2013). Here we will simply linearize the system (1) about an equilibrium and compute the gain K that minimizes the quadratic cost

$$J = \min \int_0^\infty \tilde{z}^T Q\tilde{z} + \tilde{y}^T R\tilde{y}\, d\tau \tag{6}$$

by solving the steady-state Algebraic Riccati Equation

$$0 = AP + PA^T - PH^T R^{-1} H^T P + \Phi^T Q\Phi, \tag{7}$$

from which the observer gain is $K = (R^{-1}HP)^T$.

3 Building "Digital Twin" Case Study

In this section we design an ELO to estimate unmeasured performance outputs in a commercial building HVAC system. The primary purpose of the observer is to estimate heat flows through the walls, ceiling and floor, and also to estimate the unmeasured heat loads, denoted q, in the occupied space. These estimates can be used to better understand building performance and improve human comfort and energy efficiency.

The building, diagrammed in Figure 1, is the top floor of a medium-sized commercial office building, with open floor plan for office work. We model the floor as a single room with four outside walls, a floor and a ceiling. Above the ceiling is a small plenum space that separates the ceiling from the roof. The walls are made up of between one and four layers of building materials. Windows are on the South and West facing facades. The air conditioning system is a chilled water plant, with fan coils for cooling. Outside air ventilation is provided by a constant speed ventilation fan, and the outside air passes through an Energy Recovery Ventilation Unit (ERV) for pre-cooling in

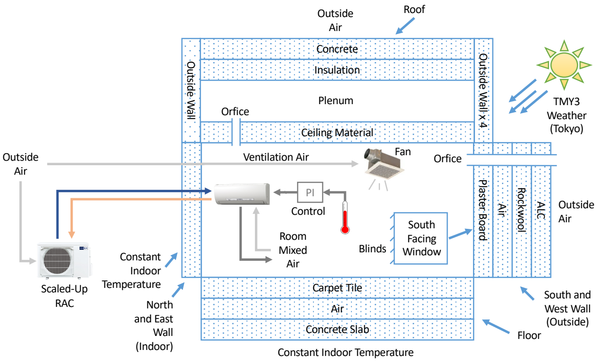


Figure 1. Building with plenum.

the summer season, but is otherwise not treated. For purposes of design, we assume there are three measurements available on a one minute sampling interval: The room temperature T_r, the plenum temperature T_p, and the return water temperature T_w. We also assume that the weather variables are measured hourly. These include the outside air temperature, humidity, wind speed, direct and indirect solar radiation in visible and infra red radiation, cloud conditions, and the atmospheric pressure. The room temperature T_r is compared to a reference set-point, and the error is fed back through Proportional-Integral (PI) feedback to actuate the valve in the fan coil.

The system is modeled using the Modelica buildings library (Wetter et al., 2014) as two rooms: one representing the working space, and the second representing the plenum, as shown in Figure 2. The outside walls have four layers, and the windows are double-paned glass. Orfices are put between the plenum and room to represent airflow between them, although its velocity is very close to zero nominally. A cooling coil is connected to a variable speed chilled water pump to provide variable capacity cooling. An Energy Recovery Ventilator (ERV) is included to precool the outside ventilation air, which is provided at a fixed rate. All of the model components are taken from the Modelica buildings library. Typical Meteorological Year (TMY) weather for Tokyo is used in all simulations. The complete model has 85 states, three measured outputs, one input (the water pump speed), and eleven disturbance inputs corresponding to the eleven weather variables used in the building library. A PID controller from the Modelica Standard Library is added to the model later for feedback to regulate the room temperature to a desired set-point.

We now step through the design and implementation steps, beginning with model augmentation, which is done in order to estimate unmeasured model inputs, then model linearization, order reduction, feedback gain design, and FMU realization.

3.1 Model Augmentation

After constructing the nominal model, it must be modified for use as an estimator. Normally the heat load q is considered an *input* to the model. (Actually, there are three different types of heat load: Radiative, Sensible and Latent. Here we assume all of the heat load is sensible.) However, in order to *estimate* q from the available measured outputs, we augment the model to include q as a state. We assume that the heat load is constant, and then add the equation

$$\dot{q} = 0 \tag{8}$$

to the Modelica model. This is done by adding an integrator to the model as the heat load, with its input set to zero. This will allow us to estimate the heat load with zero steady-state error if it is constant, and a small tracking error if it is time-varying.

Mathematically, the building and HVAC model is

$$\dot{x} = f(x,u,d,q) \tag{9a}$$
$$\dot{q} = 0 \tag{9b}$$
$$y = h(x) \tag{9c}$$
$$z = g(x) \tag{9d}$$

where z is the heat flow through the surfaces of interest (floor, walls, ceiling, and window), y is the three measurements, x is the 85-dimensional state vector, d represents the measured weather inputs into the model, and u is the water valve control input. The model used for estimator design does not include the PI feedback controller, which is added later for simulations.

3.2 Linearization

We then simulate model for approximately one million seconds (about 1 week). This is necessary because the slowest observable mode in the model has a time constant of approximately eight hours, which comes from the concrete building materials in the walls. For the linearization, we zero the radiative effects of the weather, and assume the outdoor temperature and humidity are constants representing typical weather in the summer. This is not ideal, since the radiative effects are dominant. However, it is effective for this particular application. The linearization is

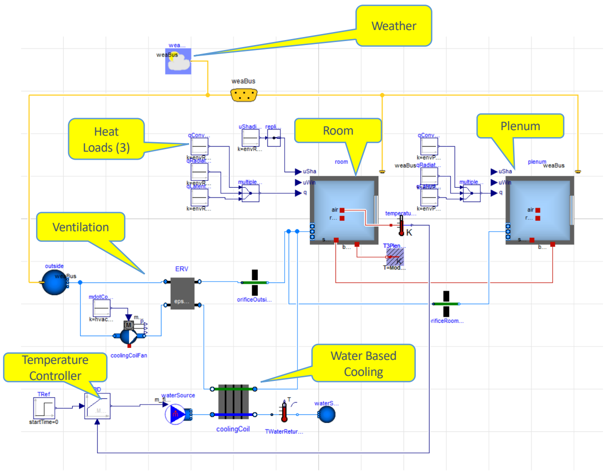


Figure 2. Modelica model.

represented as

$$\dot{x} = Ax + Bu \tag{10a}$$

$$y = Cx \tag{10b}$$

3.3 Observer Gain Design

We design the observer gain $K \in \mathbb{R}^{86\times 3}$ as outlined in the previous section, with a penalty $Q \in \mathbb{R}^{86\times 68}$ on the estimated states, and $R \in \mathbb{R}^{3\times 3}$ penalizing the measurements. For simplicity, these are set to be diagonal matrices. However, we find that a solution to the Riccati equation (7) for the linearized model and any such values of Q and R does not exist! We must analyze the linearized model (10), and then modify and reduce it in order to properly design the feedback gain K.

Computing the spectrum of A, we find a total of three states have eigenvalues at exactly zero, one state has an eigenvalue at almost zero, but corresponding to a time constant of several *months*, and the remainder have real negative parts with time constants ranging from 12 ms to 7 hours, as expected. (It may surprise the reader to see such fast modes in a model of an HVAC system. These are due to heat flow in the metal heat exchanger.) One of the three zero eigenvalues corresponds to the integrator, which can be verified by computing the left eigenvalues of A and showing that the integrator state corresponds exactly with the corresponding left eigenvector. (This means that the integrator state is affected by none of the other states, but it does affect other states, and is, in fact, observable.) The other two states with exactly zero eigenvalue correspond to "physical" states that are introduced into the orfice equations in the model, which can be seen by inspecting the following code taken from the Modelica buildings library.

```
Real mExc(quantity="Mass", final unit="kg")
    "Air mass exchanged (for purpose of
        error control only)";
initial equation
  mExc=0;
equation
  if forceErrorControlOnFlow then
    der(mExc) = port_a.m_flow;
  else
    der(mExc) = 0;
  end if;
```

We see that the state `mExc` is introduced for error control, and has its derivative set to zero if `forceErrorControlOnFlow=false`. This state has no effect on a simulation, but it is included in the linearization. Inspection of the corresponding rows of B and C verify that this state is neither controllable nor observable, and is obviously not stable. Its presence in the model therefore causes the Riccati equation solver to fail. We therefore symbolically remove the two states `mExc`, corresponding to the two orfices in our model, from the linearization by removing the corresponding rows and columns. Note that this is not a numerical calculation. Then in the estimator, we simply initialize these states at zero and they are effectively ignored.

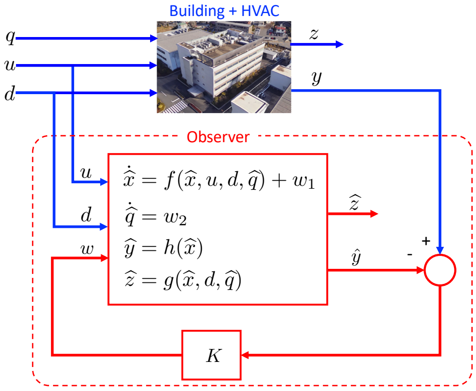


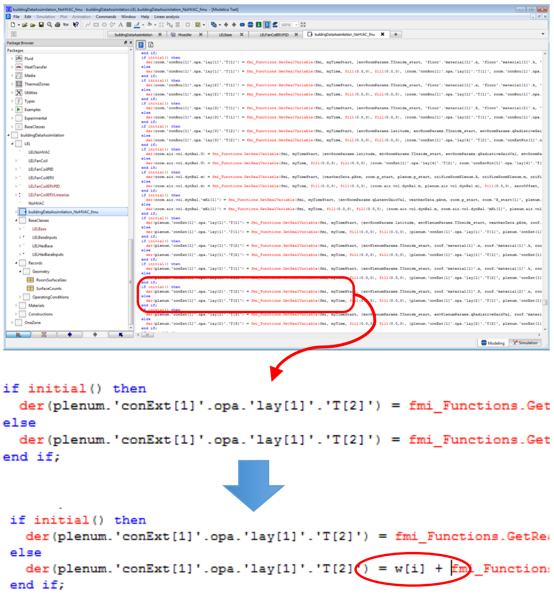
Figure 3. Observer block diagram.

The other eigenvalue near zero has an eigenvector that is nearly aligned with the potential energy state of the plenum air. However it is not an exact alignment, so we cannot say that the physical state is exactly this slow state. Its presence in the model causes the Riccati solver to fail for some values of Q and R. We therefore remove it from the linear model by modal decomposition, resulting in an 83-dimensional reduced model, which is detectable from our three measurements (because it is exponentially stable). This reduced model is used to design a reduced-order feedback gain K_r, and the full order gain is computed by using a value of zero for the three states that were removed and expanding back to the original 86-dimensional system.

3.4 FMU Realization

A block diagram of the observer is shown in Figure 3. This shows the structure of the inputs and outputs to the observer. It takes as input the control input u, the measured disturbances d, and the output injection vector w, which is the feedback signal $K(y-\hat{y})$. The output injection vector w is added to the dynamic equations. This diagram shows the augmented state to include the unmeasured heat loads q.

An FMU makes realization of the observer possible, because it is essentially a DLL for the right-hand side of the ordinary differential equation, and once loaded into a tool like Dymola, can be manipulated to allow for the output injection. Figure 4 shows the Modelica model that adds the output injection vector w to the right-hand side of the differential equation that is defined by the FMU. Essentially we declare the real input vector w and add each component to the lines that define the `der( · )`. We have created Python scripts to automate the process of editing the Modelica file. We then instantiate the modified FMU, wrap the feedback gain around it, and declare inputs and outputs to drive the new model with data. Note that the order of the states in the linearization is often different than the order of states in the FMU. So as a practical matter,



```
if initial() then
  der(plenum.'conExt[1]'.opa.'lay[1]'.'T[2]') = fmi_Functions.Get
else
  der(plenum.'conExt[1]'.opa.'lay[1]'.'T[2]') = fmi_Functions.Get
end if;
```

```
if initial() then
  der(plenum.'conExt[1]'.opa.'lay[1]'.'T[2]') = fmi_Functions.GetRe
else
  der(plenum.'conExt[1]'.opa.'lay[1]'.'T[2]') = w[i] + fmi_Function
end if;
```

Figure 4. Modification of FMI in Dymola.

we typically re-order the states of the linearization so that it corresponds to that in the FMU.

3.5 Simulation Results

To test the observer, we first simulate it using data generated from the original model. For both systems, we design a PI feedback controller to regulate the room temperature. We then simulate the data-generating model for Tokyo weather during the last week of June. We drive this model with an “actual" heat load as an input, assumed to be zero until 8:00am when the workday starts and it ramps up continuously to 4kW over one hour. (Of course, the observer *estimates* this value.). We sample the weather hourly, and the three temperature measurements on a one minute clock, which is the typical sampling rate for these applications. We then apply this data to the modified FMU, which also includes the same feedback controller.

Some of the results are shown in Figure 5 and 6. In Figure 5 we see that the ambient, plenum and water return temperatures have good information content, while the regulated room temperature remains relatively constant and therefore provides little information to the observer. The plot also shows the estimated heat flows. The flow through the ceiling is dominant, while that through the south and west walls is relatively small. Heat flow through the west wall is larger in the early evening, due to solar radiation. The heat flow through the ceiling peaks about six hours after the solar radiation peak, because of the large amount of heat storage in the concrete above the plenum. The plot at bottom shows the estimated and “actual” heat load. The observer is able to estimate the heat load with little lag, and with zero steady-state error as expected. Figure 6 shows a close-up of the estimated and actual heat loads. The observer is able to estimate the heat load with some small lag and zero steady-state accuracy when the actual load achieves its constant value at 9:00am.

4 Conclusions

In this work we have used FMU to realize an Extended Luenberger Observer for a building HVAC application. The approach is an alternative to an Extended Kalman Filter, and may offer some advantage in some applications, such as improved convergence and reduced computational complexity. The observer is constructed by augmenting the model dynamics to allow for estimation of boundary conditions, which is the heat load input to the model, linearizing, reducing and designing a feedback gain to stabilize the observer error dynamics, and then realizing the feedback using output injection by modifying the FMU. Some initial simulation results are provided as a simple proof of concept.

There are several extensions to this work and we expect to publish alternative formulations and experimental validation in the future. The most obvious is to compare the performance to an Extended Kalman Filter and its variants. The design of the EKF is made possible by features of FMI that allow for computation of the system Jacobian, starting and time stepping of the model, and setting of the model initial conditions which is done in the correction step.

To date we have experienced quite a few challenges with the EKF for this application. First, we find that the correction step, which modifies the state, can push the model outside its domain of validity. Often the states are corrected in a manner that causes a state to violate one of its limits. Mass fractions of water are particularly troublesome. Although we might consider using dry air models, the performance of the HVAC system is strongly affected by humidity, and neglecting this physics is not desirable. Is it possible to derive Modelica models that extend regions of validity, into perhaps non-physical domains? Modelers should think about this possibility, since the models themselves are useful for things beyond forward time-domain simulations. Of course, it may be possible to modify the EKF itself, preventing the correction step from violating constraints. Indeed, a key reason to consider Moving Horizon Estimators is that the constraints in the model may be enforced.

A second difficulty we have experienced with the EKF is divergence, which may be caused by the stiffness and poor conditioning of the model itself. We find that often the very slow states can be perturbed in the correction step, causing very slow convergence or simply poor performance. It may be possible to avoid some of this by projection or resetting some of the states, although some of the states of interest, e.g. some heat flows, depend on the slow dynamics in the model. On the other hand, the ELO seems more robust. This may be because it is using the implicit variable-step DASSL solver.

We remark that a more thorough analysis of the slow modes in these models is necessary. Often their presence in a linearized model can cause conventional Hankel-norm model truncation to fail. This is because these modes are very slow, with eigenvalues very close to zero. The Hankel-norm truncation begins by computing a spectral decomposition, and only removes those modes with sufficiently small Hankel singular value, and that are sufficiently stable i.e., have a sufficiently negative eigenvalue. Such a truncation will keep these slow modes in the model, even if they are very weakly controllable and observable. Therefore, they must be removed from the linearization before the Hankel-norm truncation is done. Although these modes can apparently be removed in a spectral decomposition of the linearization at design time, there is no guarantee that the resulting reduced order model will result in a correct estimator or controller design, and the modes are still present in the simulation model. There are open questions such as how these should be initialized in an estimator. The precise cause of these slow modes needs further investigation.

References

D. Boutat, A. Benali, H. Hammouri, and K. Busawon. New algorithm for observer error linearization with a diffeomorphism on the outputs. *Automatica*, 45(10):2187–2193, 2009.

Jonathan Brembeck, Martin Otter, and Dirk Zimmer. Nonlinear observers based on the functional mockup interface with applications to electric vehicles. In *Proceedings of the 8th Modelica Conference*, pages 474–483, 2011.

Jonathan Brembeck, Andreas Pfeiffer, Michael Fleps-Dezasse, Martin Otter, Karl Wernersson, and Hilding Elmqvist. Nonlinear state estimation with an extended FMI 2.0 co-simulation interface. In *Proceedings of the 10th International Modelica Conference*, pages 53–62, 2014.

Chi-Tsong Chen. *Linear System Theory and Design*. Holt, Rinehart and Winston, 1984.

Bernard Friedland. *Control System Design: An Introduction to State-Space Methods*. McGraw-Hill, 1986.

M. Hou and A. Pugh. Observer with linear error dynamics for nonlinear and multi-output systems. *Systems & Control Letters*, 37(1):1–9, 1999.

A. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 3(1):47–52, 1983.

A. Krenner and W. Respondek. Nonlinear observers with linearizable error dynamics. *SIAM Journal on Control and Optimization*, 23(2):197–216, 1985.

D. Luenberger. An introduction to observers. *IEEE Transactions of Automatic Control*, 16(6):596–602, 1971.

Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley, 2005.

R. Tami, D. Boutat, and G. Zheng. Extended output depending normal form. *Automatica*, 49(7):2192–2198, 2013.

Michael Wetter, Wangda Zuo, Thierry S. Nouidui, and Xiufeng Pang. Modelica buildings library. *Journal of Building Performance Simulation*, 7(4):253–270, 2014.

M. Zeitz. The extended luenberger observer for nonlinear systems. *Systems & Control Letters*, 9(2), 1987.

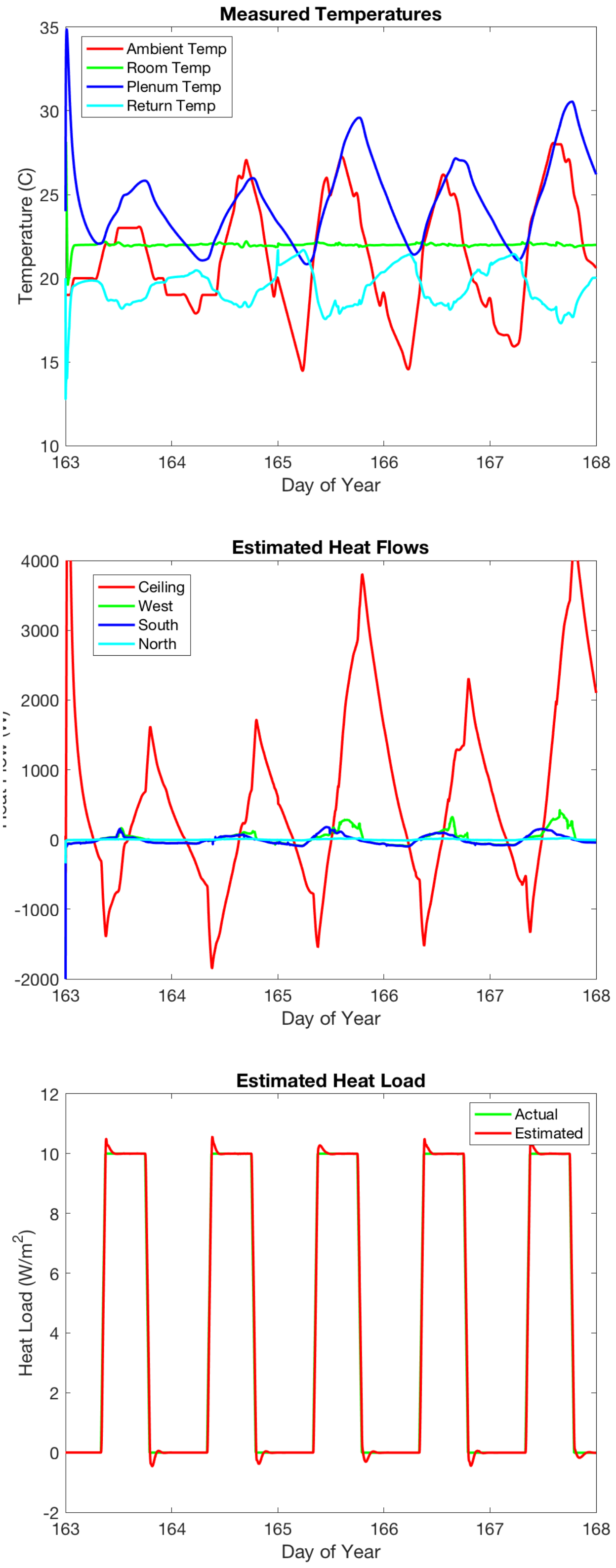


Figure 5. Simulation Results.

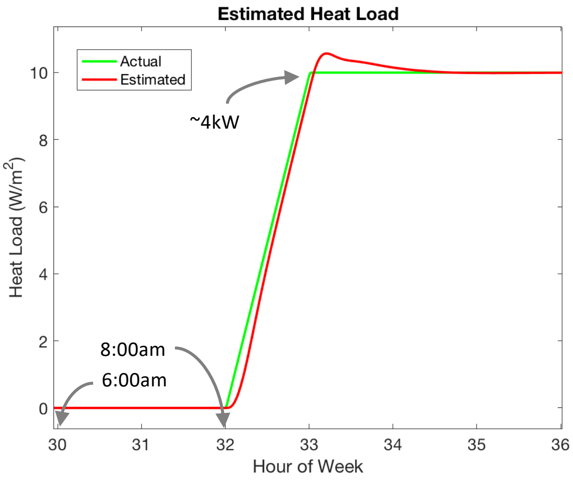


Figure 6. Close-up of the heat load estimation.