Tensor Systems
Multilinear Modeling and Applications

G. Pangalos[1], A. Eichler[1] and G. Lichtenberg[2]

[1]*Institute of Control Systems, Hamburg University of Technology, Eißendorfer Straße 40, Hamburg, Germany*
[2]*Faculty of Life Sciences, Hamburg University of Applied Sciences, Lohbrügger Kirchstraße 65, Hamburg, Germany*

Keywords: Multilinear Systems, Hybrid Systems, Tensor Decomposition, Heating Systems, Multi-Agent Systems.

Abstract: Tensor systems are a framework for modeling of multilinear hybrid systems with discrete and continuous valued signals. Two examples from building services engineering and multi-agent systems show applications of this framework. A tensor model of a heating system is derived and approximated by tensor decomposition methods first. Second, a tensor model of a multi-agent system with a structure already given in a decomposed form is reduced further by the same decomposition methods. The validity of these low rank approximations is shown for both examples.

1 INTRODUCTION

Multilinear models are a subclass of nonlinear models which is interesting for engineering applications because several systems behave multilinear, as shown by the examples. In most applications the processes are driven by some automation device with fixed sampling time. Thus, only discrete time models are investigated here.

The class of multilinear models obviously includes linear continuous valued models but moreover, all discrete valued models are inherently multilinear, see (Lichtenberg, 2010). As multilinearity is mathematically closely related to tensor calculus, there are tensor representations of multilinear systems. Of special interest are discrete time hybrid systems consisting of a multilinear continuous valued subsystem and an arbitrary discrete valued subsystem connected by a quantizer and an injector. These can be represented by so-called *Hybrid Tensor Systems* defined in (Lichtenberg, 2011). This modeling approach was first applied to the identification problem of discrete valued models from continuous data, (Lichtenberg and Eichler, 2011).

Over the last years simulation of building and heating systems has attracted a lot of interest, see e.g. (Nouidui et al., 2012) or (Wetter, 2006). In these approaches nonlinear models are derived and simulated, mostly for performance evaluation. Heating systems are application examples for hybrid tensor systems because of their physical structure, see (Pangalos and Lichtenberg, 2012). This paper shows in detail how to derive a multilinear one zone model of a large scale building using RC-models of the walls.

Another field of applications are distributed network systems like Multi-Agent Systems (MAS), which have received considerable attention over the past ten years, because of its broad variety of applications, see (Ren and Beard, 2007; Ren and Cao, 2011). To perform tasks like Decision-Making or Policy Formulation, hybrid MASs are required, (Srinivasan and Choy, 2010). These can be represented by the same class of hybrid tensor systems.

Numerical tensor calculus with special interest on decomposition methods is an active reseach field in applied mathematics, see e.g. (Hackbusch, 2012). Tensor models can be approximated by standard tensor decomposition techniques - similar to model reduction techniques for linear systems. Most of the latter techniques like PCA use singular value decompostion (SVD) of matrices as workhorse for the reduction process. For tensor systems, so-called *higher-order SVD* decomposition algorithms are appropriate reduction methods, see (Kolda and Bader, 2009).

The paper is organized as follows. After this introduction the class of hybrid tensor systems is described in section 2. The model of a complex heating system is developed, validated, represented by a hybrid tensor system, and finally decomposed in section 3. In section 4 multilinear modeling and transformation to a tensor system is shown for a hybrid MAS. Final conclusions are drawn in section 5.

2 TENSOR SYSTEMS

Tensor representations of multilinear systems are introduced in (Lichtenberg, 2011) and reviewed in this section. All linear systems can be represented by this class, which on the other hand is a subset of the class of polynomial systems.

First, a matrix format to describe multilinear systems is introduced. Next, it is shown how to represent these systems as tensor models. Finally, the notation is extended to hybrid systems which include continuous and discrete states and inputs.

2.1 Multilinear Matrix Models

With the help of the symbol $\otimes$ denoting the standard Kronecker product, the monomial vector introduced in the following Definition contains all possible multilinear terms.

Definition 2.1. *The* monomial vector *is defined as*

$$\mathbf{m}(\mathbf{x},\mathbf{u}) = \begin{pmatrix} 1 \\ u_m \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ u_1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ x_n \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ *is the state vector and* $\mathbf{u} \in \mathbb{R}^m$ *is the input vector.*

With the monomial vector, a multilinear state space model can be written in matrix form

$$\Phi(\mathbf{x}) = \mathbf{F}\,\mathbf{m}(\mathbf{x},\mathbf{u})\,, \tag{2}$$
$$\mathbf{y} = \mathbf{G}\,\mathbf{m}(\mathbf{x},\mathbf{u})\,, \tag{3}$$

where $\Phi(\mathbf{x})$ denotes the next state vector, $\mathbf{y} \in \mathbb{R}^l$ the output vector, $\mathbf{F} \in \mathbb{R}^{n \times 2^{(n+m)}}$ the transition matrix and $\mathbf{G} \in \mathbb{R}^{l \times 2^{(n+m)}}$ the output matrix.

Example 2.1. *Consider the state transition function of a multilinear second order model with one input and next state vector*

$$\begin{pmatrix} f_{11}+f_{12}x_1+f_{15}u+f_{16}ux_1 \\ f_{22}x_1+f_{23}x_2+f_{24}x_1x_2+f_{26}ux_1+f_{27}ux_2+f_{28}ux_1x_2 \end{pmatrix}.$$

The state space model (2) is given as

$$\begin{pmatrix} \Phi(x_1) \\ \Phi(x_2) \end{pmatrix} = $$
$$\begin{pmatrix} f_{11} & f_{12} & 0 & 0 & f_{15} & f_{16} & 0 & 0 \\ 0 & f_{22} & f_{23} & f_{24} & 0 & f_{26} & f_{27} & f_{28} \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1x_2 \\ u \\ ux_1 \\ ux_2 \\ ux_1x_2 \end{pmatrix}. \tag{4}$$

In the following, the output equation (3) will not be considered – for a description of multilinear input-output models see (Lichtenberg, 2011).

For multilinear models (2), linear state and input variable transformations

$$\tilde{x}_i = \frac{x_i - b_i}{a_i} \; \forall i = 1, \ldots, n \tag{5}$$

$$\tilde{u}_i = \frac{u_i - b_{n+i}}{a_{n+i}} \; \forall i = 1, \ldots, m \tag{6}$$

are computable in closed form. The transformed state transition matrix is given by

$$\widetilde{\mathbf{F}} = \operatorname*{diag}_{i=1,\ldots,n} \left(\frac{1}{a_i}\right) \left(\mathbf{FT} - \begin{pmatrix} \mathbf{b} & 0_{n \times 2^{(n+m)}-1} \end{pmatrix}\right) \tag{7}$$

where diag denotes the diagonal matrix with elements a_i,

$$\mathbf{T} = \begin{pmatrix} 1 & 0 \\ b_{n+m} & a_{n+m} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ b_1 & a_1 \end{pmatrix}$$

is a transformation matrix, $\mathbf{b}$ is a column vector containing the b_i's and $0_{n \times 2^{(n+m)}-1}$ denotes a zero matrix with dimension $(n \times 2^{(n+m)} - 1)$. Linear transformations allow a numerical preconditioning of models which is used in section 3.4.

2.2 Multilinear Tensor Models

To rewrite (2) in a compact tensor notation, following standard definitions from e.g. (Kolda and Bader, 2009) or (Cichocki et al., 2009) are introduced.

Definition 2.2. *A Tensor*

$$\mathsf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$$

of order n is an n-way array where elements $x_{i_1 i_2 \cdots i_n}$ *are indexed by* $i_j \in \{1, 2, \ldots I_j\}$ *for* $j = 1, \ldots, n$.

Example 2.2. *The elements of a tensor* $\mathsf{X} \in \mathbb{R}^{2 \times 3 \times 2}$ *can be illustrated and arranged in 3 dimensions as*

Definition 2.3. *A Kruskal tensor*

$$\mathsf{K} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n] \cdot \boldsymbol{\lambda} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_n}$$

is a tensor of dimension $(r_1, \ldots, r_n)$, *with elements given by the sums of the outer products of the column vectors of so-called* factor matrices $\mathbf{X}_i \in \mathbb{R}^{r_i \times r}$, *weighted by the elements of the so-called* weighting *or* parameter vector $\boldsymbol{\lambda}$. *An element of the multidimensional tensor* K *is given by*

$$K_{jk \ldots p} = \sum_{i=1}^{r} \lambda_i (\mathbf{X}_1)_{ji} (\mathbf{X}_2)_{ki} \ldots (\mathbf{X}_n)_{pi}$$

If no weighting vector is given, it is assumed to be a vector of ones, i.e. $\boldsymbol{\lambda} = \begin{pmatrix} 1 & 1 & \ldots & 1 \end{pmatrix}^T$.

Definition 2.4. *The* contracted product *of two tensors* $\mathsf{Y} \in \mathbb{R}^{I_1 \times \dots \times I_n}$ *and* $\mathsf{X} \in \mathbb{R}^{I_1 \times \dots \times I_n \times I_{n+1} \times \dots \times I_{n+m}}$ *is*

$$\langle \mathsf{X} \,|\, \mathsf{Y} \rangle (k_1, \dots, k_m) = \sum_{i_1} \cdots \sum_{i_n} x_{i_1,\dots,i_n,k_1,\dots,k_m} y_{i_1,\dots,i_n} , \tag{8}$$

is a tensor of dimension $I_{n+1} \times \dots \times I_{n+m}$.

Example 2.3. *Consider a tensor* $\mathsf{X} \in \mathbb{R}^{2 \times 2 \times 3}$ *and a tensor* $\mathsf{Y} \in \mathbb{R}^{2 \times 2}$. *To calculate the contracted product* $\mathsf{Z} = \langle \mathsf{X} \,|\, \mathsf{Y} \rangle \in \mathbb{R}^3$, *the matching dimensions have to be multiplied elementwise, as highlighted in red in the following illustrations.*

A simple notation for tensor spaces is introduced:

$$\mathbb{R}^{\times(n+m)2} := \mathbb{R}^{\overbrace{2 \times \dots \times 2}^{n+m}} .$$

The state transition function (2) of a multilinear state space model can then be rewritten with the monomial Kruskal tensor

$$\mathsf{M}(\mathbf{x}, \mathbf{u}) = \left[\begin{pmatrix} 1 \\ u_m \end{pmatrix}, \dots, \begin{pmatrix} 1 \\ u_1 \end{pmatrix}, \begin{pmatrix} 1 \\ x_n \end{pmatrix}, \dots, \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \right] .$$

in terms of a contracted tensor product as

$$\Phi(\mathbf{x}) = \langle \mathsf{F} \,|\, \mathsf{M}(\mathbf{x}, \mathbf{u}) \rangle , \tag{9}$$

where the transition tensor $\mathsf{F} \in \mathbb{R}^{\times(n+m)2 \times n}$ contains all parameters of the model while the structure is given by the monomial tensor $\mathsf{M}(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^{\times(n+m)2}$.

2.3 Decomposed Tensor Models

As contracted products could be efficiently computed in Kruskal form, and monomial tensors are easily

given in this form by their internal structure, it is worth looking for decompositions

$$\mathsf{F} = [\mathbf{F}_{u_m}, \dots, \mathbf{F}_{u_1}, \mathbf{F}_{x_n}, \dots, \mathbf{F}_{x_1}, \mathbf{F}_\Phi] \cdot \lambda_f \tag{10}$$

of the state transition tensor having r_x factors. This will be illustrated by the Example 2.1. Setting the matrix factors of (10) accordingly leads to

$$\mathbf{F}_{x_1} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} ,$$

$$\mathbf{F}_{x_2} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} ,$$

$$\mathbf{F}_{u} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} ,$$

$$\mathbf{F}_{\Phi} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} .$$

Together with the continuous valued parameter vector

$$\lambda_f = \begin{pmatrix} f_{11} & f_{12} & f_{15} & f_{16} & f_{22} & f_{23} & f_{24} & f_{26} & f_{27} & f_{28} \end{pmatrix}^T$$

they form the state transition tensor

$$\mathsf{F} = [\mathbf{F}_u, \mathbf{F}_{x_2}, \mathbf{F}_{x_1}, \mathbf{F}_\Phi] \cdot \lambda_f.$$

The contracted product can now be calculated by simple matrix operations (Lichtenberg, 2011)

$$\langle \mathsf{F} \,|\, \mathsf{M}(\mathbf{x}, \mathbf{u}) \rangle = \tag{11}$$

$$\mathbf{F}_\Phi^T \left(\lambda_f \circledast \left(\mathbf{F}_u^T \begin{pmatrix} 1 \\ u \end{pmatrix} \right) \circledast \left(\mathbf{F}_{x_2}^T \begin{pmatrix} 1 \\ x_2 \end{pmatrix} \right) \circledast \left(\mathbf{F}_{x_1}^T \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \right) \right) ,$$

where $\circledast$ denotes the (element-wise) Hadamard product. It can be checked that calculating the contracted product (9) leads to the transition equation (4).

At a first glance, for Example 2.1 this tensor notation does not seem to be more compact than the matrix notation. It is easy to verify that the system matrix has $n \cdot 2^{(m+n)} = 16$ elements, whereas the tensor representation takes $r_x (1 + 3n + 2m) = 90$ elements as the dimension of the $n + m$ factor matrices is $2 \times r_x$, $\mathbf{F}_\Phi$ has dimension $n \times r_x$ and the parameter vector has another r_x entries.

But looking at the model of the heating system that will be introduced in section 3 with $n = 12$ states, no inputs $m = 0$ and $r_x = 46$ factors, this would lead to $12 \cdot 2^{12} = 49152$ elements in matrix representation whereas only $46(1 + 3 \cdot 12) = 1702$ Kruskal tensor elements need to be stored.

For construction of the factor matrices and the parameter vector set the columns of the factor matrices to $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, if the summand contains the corresponding state and to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ otherwise. The product of states is weighted by a constant given in the parameter vector. The factor matrix $\mathbf{F}_\Phi$ indicates to which next state element the summand is added.

This can is shown exemplarily by the first summand of the Example. All first rows of the factor matrices are $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, i.e. the first summand of the first next state does not depend on any current state or input but is equal to the first entry f_{11} of the parameter vector.

With standard tensor decomposition algorithms like CP-ALS low rank approximations of the state transition tensor could be found, see (Kolda and Bader, 2009).

2.4 Boolean Tensor Models

In the following, Boolean variables will be indicated by underlining. Heading for tensor models of hybrid systems, the binary state $\underline{\mathbf{x}} \in \mathbb{B}^N$ and input $\underline{\mathbf{u}} \in \mathbb{B}^M$ are defined. The state transition function of a Boolean tensor state space model

$$\Phi(\underline{\mathbf{x}}) = \langle \underline{\mathsf{F}} \mid \underline{\mathsf{L}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \rangle , \qquad (12)$$

with a transition tensor $\underline{\mathsf{F}} \in \mathbb{B}^{\times (N+M)2 \times N}$ describes a binary dynamical system by a contracted product with a Boolean literal tensor $\underline{\mathsf{L}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in \mathbb{B}^{\times (N+M)2}$ defined as

$$\underline{\mathsf{L}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \left[\begin{pmatrix} \bar{\underline{u}}_n \\ \underline{u}_n \end{pmatrix} , \cdots , \begin{pmatrix} \bar{\underline{u}}_1 \\ \underline{u}_1 \end{pmatrix} , \begin{pmatrix} \bar{\underline{x}}_n \\ \underline{x}_n \end{pmatrix} , \cdots , \begin{pmatrix} \bar{\underline{x}}_1 \\ \underline{x}_1 \end{pmatrix} \right] , \qquad (13)$$

where the bar symbol is used to denote the negation of a variable.

If the values of the state and input variables are from a continuous domain but the transition tensor is still Boolean, the resulting next state is in general a continuous variable.

With the help of the symbol $\langle \cdot \mid \cdot \rangle^+$ defining a contracted real tensor product which converts any Boolean FALSE to a real zero and a Boolean TRUE to a real one, the state transition of an *algebraic Boolean tensor state space model*

$$\Phi(\mathbf{x}) = \langle \underline{\mathsf{F}} \mid \mathsf{L}(\mathbf{x}, \mathbf{u}) \rangle^+ , \qquad (14)$$

can be defined, where

$$\mathsf{L}(\mathbf{x}, \mathbf{u}) = \left[\begin{pmatrix} 1 - u_n \\ u_n \end{pmatrix} , \cdots , \begin{pmatrix} 1 - u_1 \\ u_1 \end{pmatrix} , \right.$$
$$\left. \begin{pmatrix} 1 - x_n \\ x_n \end{pmatrix} , \cdots , \begin{pmatrix} 1 - x_1 \\ x_1 \end{pmatrix} \right] . \qquad (15)$$

It has the property that - if all initial states and inputs are Boolean - the state and output trajectories only take values 0 and 1 and thus, show a behavior which includes the behavior of the binary system.

Transformation of (14) to monomial form is possible and given by

$$\Phi(\mathbf{x}) = \langle \mathsf{F} \mid \mathsf{M}(\mathbf{x}, \mathbf{u}) \rangle^+ , \qquad (16)$$

with the transition tensor $\mathsf{F} \in \mathbb{Z}^{\times (n+m) \times n}$ having integer elements which are easy computable from $\underline{\mathsf{F}}$.

2.5 Hybrid Tensor Models

Figure 1 shows a block diagram of the class of hybrid systems investigated further. The outputs are omitted, it is assumed that the states can be measured.
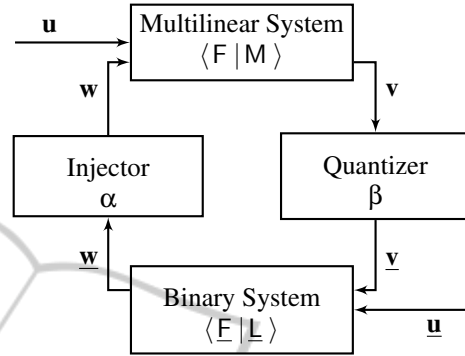

Figure 1: Block diagram of a multilinear hybrid system.

Without loss of generality, it is assumed that all discrete signals are encoded Boolean. The continuous valued subsystem and the discrete valued subsystem are connected via a quantizer and an injector given in the next Definitions, see (Lichtenberg, 2011), where $\mathbb{H}$ is a hybrid space. The hybrid state and input vectors

$$\underset{\sim}{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \underline{\mathbf{x}} \end{pmatrix} \in \mathbb{R}^n \times \mathbb{B}^N \qquad (17)$$

$$\underset{\sim}{\mathbf{u}} = \begin{pmatrix} \mathbf{u} \\ \underline{\mathbf{u}} \end{pmatrix} \in \mathbb{R}^m \times \mathbb{B}^M \qquad (18)$$

are given by appending the Boolean signal vectors to the continuous signal vectors.

Definition 2.5. *The* standard injector *is the function* $\alpha : \mathbb{H}^{I_1 \times \cdots \times I_N} \to \mathbb{R}^{I_1 \times \cdots \times I_N}$, *which is given for all elements with index vector* $\mathbf{i} \in \mathbb{N}^N$ *by*

$$(\alpha(\underset{\sim}{\mathbf{x}}))_{\mathbf{i}} = \begin{cases} 1 \in \mathbb{R} & \text{if } x_{\mathbf{i}} = TRUE , \\ 0 \in \mathbb{R} & \text{if } x_{\mathbf{i}} = FALSE , \\ x_{\mathbf{i}} & \text{if } x_{\mathbf{i}} \in \mathbb{R} . \end{cases} \qquad (19)$$

Definition 2.6. *The* standard quantizer *is the function* $\beta : \mathbb{R}^{I_1 \times \cdots \times I_N} \to \mathbb{B}^{I_1 \times \cdots \times I_N}$, *which is given for all elements with index vector* $\mathbf{i} \in \mathbb{N}^N$ *by*

$$(\beta(\underset{\sim}{\mathbf{x}}))_{\mathbf{i}} = \sigma(x_{\mathbf{i}} - \frac{1}{2}) , \qquad (20)$$

where the Heaviside *function is given as*

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0 , \\ 0 & \text{otherwise} . \end{cases} \qquad (21)$$

These definitions make the need of explicit equations for $\mathbf{w}, \underline{\mathbf{w}}, \mathbf{v}$ and $\underline{\mathbf{v}}$ obsolete, as they can be calculated with the help of the standard injector and

the standard quantizer which are also used for the next Definition - an extension of the contracted tensor product taking into account Boolean as well as continuous values.

Definition 2.7 (Hybrid Contracted Tensor Product). *The notation* $\langle \cdot \mid \cdot \rangle^{\boxplus}$ *for the* hybrid contracted tensor product *implies that all values are mapped to the correct domains given by the domain of the tensor on the left side of the contracted product*

$$\langle F \mid M \rangle^{\boxplus} = \langle F \mid \alpha(M(\beta(\mathbf{x}))) \rangle \qquad (22)$$

with standard quantizer and injector.

If the tensor on the left side has hybrid domains, quantization and injection are applied to compute the results corresponding to the domain on the left hand side.

By using this Definition, the state transition function of a hybrid tensor system can be written in a very compact way as

$$\Phi(\mathbf{x}) = \langle \underset{\sim}{F} \mid \underset{\sim}{M}(\mathbf{x}, \mathbf{u}) \rangle^{\boxplus} . \qquad (23)$$

This will be illustrated by Examples in the following sections.

3 HEATING SYSTEMS EXAMPLE

The heating system mentioned in the introduction will now be modeled as a hybrid tensor system. This is possible because of the inherit multilinear nature of heating systems.

The heating system consists of a boiler with burner, a consumer, where a building temperature can be measured and a pump, which is controlled (see Figure 2)
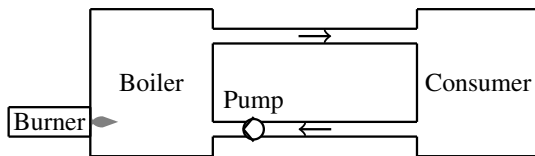

Figure 2: Heating system.

Signals being real or Boolean will be clear from the context, therefore no underlines or "undertildes" will be used in this section.

For each of the elements of the heating system, first the thermal balances are stated and then transformed. After a validation of the model the factor matrices and parameter vector are given such that the system can be represented as a tensor system.

3.1 Differential Equations

The continuous valued states of the system are the supply temperature T_s, the return temperature T_r and the buildings temperature T_b. Day oscillation and year oscillation of the ambient temperature are modeled by four continuous valued states $T_{a1} \ldots T_{a4}$, two for each oscillation. Two Boolean states X_l and X_u determine whether or not the buildings temperature is in the interval where the thermostatic valves are linear, in the lower or in the upper saturation. Two further Boolean states H_l and H_u determine a supply temperature above an upper limit or below a lower limit and a Boolean state U that states if the boiler is running or not. The state vector is given by $\mathbf{x} \in \mathbb{R}^7 \times \mathbb{B}^5$

$$\mathbf{x} = [T_s \; T_r \; T_b \; T_{a1} \; T_{a2} \; T_{a3} \; T_{a4} \; X_l \; X_u \; H_l \; H_u \; U]^T . \qquad (24)$$

Controlled Pump

The flow rate $\dot{V}$ is a consequence of the room temperature T_b and the reference temperature for the room T_{br}. The relationship is shown in Figure 3.
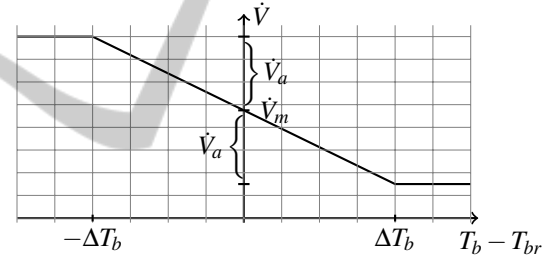

Figure 3: Flow rate over control error of the room temperature.

In this Figure $\dot{V}_m$ is the mean flow rate for $T_b = T_{br}$ and $\dot{V}_a$ is the amplitude from the mean flow rate at reference room temperature to its extrema. The extrema in the flow rate are reached at a room temperature that is at least ΔT_b different from the reference room temperature.

To state the flow rate $\dot{V}$ in terms of the discrete states and the room temperature, the discrete states

$$X_l = \begin{cases} 1 & \text{if} \quad T_b > T_{br} - \Delta T_b \\ 0 & \text{else} \end{cases} \qquad (25)$$

$$X_u = \begin{cases} 1 & \text{if} \quad T_b > T_{br} + \Delta T_b \\ 0 & \text{else} \end{cases} \qquad (26)$$

are introduced. Thus the flow rate $\dot{V}$ is given as

$$\begin{aligned} \dot{V} &= \dot{V}_m + \dot{V}_a + \left(\frac{T_{br} \dot{V}_a}{\Delta T_b} - \dot{V}_a \right) X_l \ldots \\ &\quad - \left(\frac{T_{br} \dot{V}_a}{\Delta T_b} + \dot{V}_a \right) X_u - \frac{\dot{V}_a}{\Delta T_b} T_b X_l \ldots \end{aligned}$$

$$+ \frac{\dot{V}_a}{\Delta T_b} T_b X_u \ . \tag{27}$$

Boiler

The thermal balance of the boiler is given as

$$\dot{T}_s V_b \rho c = T_r \dot{V} \rho c - T_s \dot{V} \rho c + P_{max} U \ . \tag{28}$$

Using the Euler forward method with sampling time t_s to discretize this differential equation and rearranging leads to

$$\begin{aligned}
T_s(k+1) &= T_s(k) - \frac{t_s}{V_b} T_s(k) \dot{V}(k) \ldots \tag{29} \\
&+ \frac{t_s}{V_b} T_r(k) \dot{V}(k) + \frac{t_s}{V_b \rho c} P_{max} U(k) \ .
\end{aligned}$$

Dropping the time index, denoting the next state by the operator Φ and substituting (27) for the flow rate gives

$$\begin{aligned}
\Phi(T_s) &= T_s \left(1 - \frac{t_s(\dot{V}_a + \dot{V}_m)}{V_b} \right) + T_r \frac{t_s(\dot{V}_a + \dot{V}_m)}{V_b} \ldots \\
&+ T_s X_l \frac{\dot{V}_a t_s (\Delta T_b - T_{br})}{\Delta T_b V_b} - T_r X_l \frac{\dot{V}_a t_s (\Delta T_b - T_{br})}{\Delta T_b V_b} \ldots \\
&+ T_b T_s X_l \frac{\dot{V}_a t_s}{\Delta T_b V_b} - T_b T_r X_l \frac{\dot{V}_a t_s}{\Delta T_b V_b} \ldots \tag{30} \\
&+ T_s X_u \frac{\dot{V}_a t_s (\Delta T_b + T_{br})}{\Delta T_b V_b} - T_r X_u \frac{\dot{V}_a t_s (\Delta T_b + T_{br})}{\Delta T_b V_b} \ldots \\
&- T_b T_s X_u \frac{\dot{V}_a t_s}{\Delta T_b V_b} + T_b T_r X_u \frac{\dot{V}_a t_s}{\Delta T_b V_b} + P_{max} U \frac{t_s}{V_b c \rho} .
\end{aligned}$$

Ambient Temperature

The ambient temperature is modeled as an oscillator with two frequencies. One for the temperature oscillation over the day and one for the oscillation of the year. The sinusoidal curves estimate the ambient temperature roughly.

The continuous time state space model can be given by

$$\begin{pmatrix} \dot{T}_{a1} \\ \dot{T}_{a2} \\ \dot{T}_{a3} \\ \dot{T}_{a4} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\omega_d^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_y^2 & 0 \end{pmatrix} \begin{pmatrix} T_{a1} \\ T_{a2} \\ T_{a3} \\ T_{a4} \end{pmatrix}$$

$$T_a = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} T_{a1} \\ T_{a2} \\ T_{a3} \\ T_{a4} \end{pmatrix} + T_{a,m} \ . \tag{31}$$

Note that a multilinear state space model allows a constant term, here $T_{a,m}$. Using the Euler forward method, the state equation in discrete time with sampling time t_s is given by

$$\Phi \left(\begin{pmatrix} T_{a1} \\ T_{a2} \\ T_{a3} \\ T_{a4} \end{pmatrix} \right) = \begin{pmatrix} 1 & t_s & 0 & 0 \\ -\omega_d^2 & 1 & 0 & 0 \\ 0 & 0 & 1 & t_s \\ 0 & 0 & -\omega_y^2 & 1 \end{pmatrix} \begin{pmatrix} T_{a1} \\ T_{a2} \\ T_{a3} \\ T_{a4} \end{pmatrix} . \tag{32}$$

The output equation of system (31) does not change.

Building

The temperature of the building T_b is introduced as the average temperature of all rooms. The temperature can be given as

$$k_b \dot{T}_b = k_{r,b} (T_r - T_b) - k_{b,a} (T_b - T_a) \ , \tag{33}$$

where k_b is the capacity of the building, $k_{r,b}$ the heat transfer coefficient from the radiators to the building and $k_{b,a}$ the heat transfer coefficient from the building to the outside. Again using the Euler forward method for temporal discretization and substituting the output equation of (31) gives

$$\begin{aligned}
\Phi(T_b) &= T_b \left(1 - \frac{t_s(k_{r,b} + k_{b,a})}{k_b} \right) + T_r \frac{t_s k_{r,b}}{k_b} \ldots \\
&+ T_{a1} \frac{t_s k_{b,a}}{k_b} + T_{a3} \frac{t_s k_{b,a}}{k_b} + \frac{t_s k_{b,a} T_{a,m}}{k_b} \ . \tag{34}
\end{aligned}$$

The parameters $k_b, k_{r,b}, k_{b,a}$ are estimated with real measurement data of the building. The validation of the model is described in the next section, where the numerical values of the constants are given in Table 3.

Consumer

The thermal balance of the consumer is given by

$$\dot{T}_r(t) V_c \rho c = T_s(t) \dot{V}(t) \rho c - T_r(t) \dot{V}(t) \rho c - \dot{Q}_d(t) \ . \tag{35}$$

The heat demand $\dot{Q}_d$ can be calculated in terms of the return temperature and the building temperature as

$$\dot{Q}_d(t) = (T_r(t) - T_b(t)) k_{r,b} \ . \tag{36}$$

Inserting (36) and (27) into (35) and discretizing the result gives

$$\begin{aligned}
\Phi(T_s) &= T_s \frac{t_s(\dot{V}_a + \dot{V}_m)}{V_c} \ldots \tag{37} \\
&+ T_r \left(1 - \frac{t_s(k_{r,b} + c\dot{V}_a \rho + c\dot{V}_m \rho)}{V_c c \rho} \right) \ldots \\
&+ T_b \frac{k_{r,b} t_s}{V_c c \rho} - T_s X_l \frac{\dot{V}_a t_s (\Delta T_b - T_{br})}{\Delta T_b V_c} \ldots \\
&+ T_r X_l \frac{\dot{V}_a t_s (\Delta T_b - T_{br})}{\Delta T_b V_c} - T_b T_s X_l \frac{\dot{V}_a t_s}{\Delta T_b V_c} \ldots \\
&+ T_b T_r X_l \frac{\dot{V}_a t_s}{\Delta T_b V_c} - T_s X_u \frac{\dot{V}_a t_s (\Delta T_b + T_{br})}{\Delta T_b V_c} \ldots \\
&+ T_r X_u \frac{\dot{V}_a t_s (\Delta T_b + T_{br})}{\Delta T_b V_c} + T_b T_s X_u \frac{\dot{V}_a t_s}{\Delta T_b V_c} \ldots \\
&- T_b T_r X_u \frac{\dot{V}_a t_s}{\Delta T_b V_c} \ ,
\end{aligned}$$

where the time dependencies are dropped.

Bang Bang Controller

The discrete states H_l and H_u are used to determine whether or not the supply temperature T_s is below a lower limit, here: $75°C$ or above an upper limit, here: $95°C$. The signal U steering the boiler is then computed by

$$\Phi(U) = \begin{cases} U & \text{if} & 75°C < T_s < 95°C \\ 1 & \text{if} & T_s \leq 75°C \\ 0 & \text{if} & T_s \geq 95°C \end{cases} \quad . \quad (38)$$

3.2 Validation of the Consumer

In this section the validation of the consumer is shown. For this purpose the discrete bang bang controller is replaced by a PI-controller as in the real building. The ambient temperature is modeled separately. For the validation, the ambient temperature is taken from measurement data. Not all states are displayed but just the supply temperature T_s, the return temperature T_r and the flow rate $\dot{V}$. Note that the flow rate is not a state of the system but depends on the buildings temperature T_b and the two Boolean states X_l and X_u. The system has two main time scales caused by the oscillating ambient temperature over the day and over the year.

The temperatures over a whole year, starting January 1st are given in Figure 4. The mismatch in the return temperature in the summer is probably caused by a later detected bypass of the boiler.
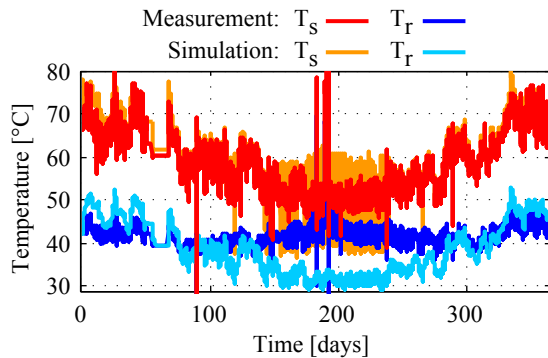

Figure 4: Supply and return temperatures over a year.

The flow rate over the year is given in Figure 5. This shows a quite good fit of the signals in a year scale. Now the second scale should be investigated. Therefore in Figure 6 the temperatures are given and in Figure 7 the flow rate is given for two days. As one can see also the day scale has a good fit. Still remaining for validation is the use of an oscillator for the ambient temperature. Figure 8 shows the ambient temperature from measurements and the results from the oscillator over a year in the upper plot and the
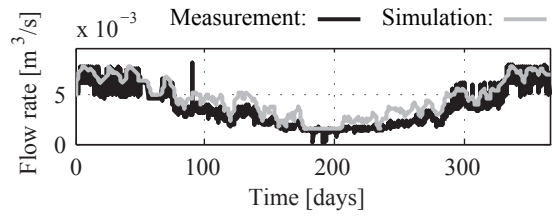
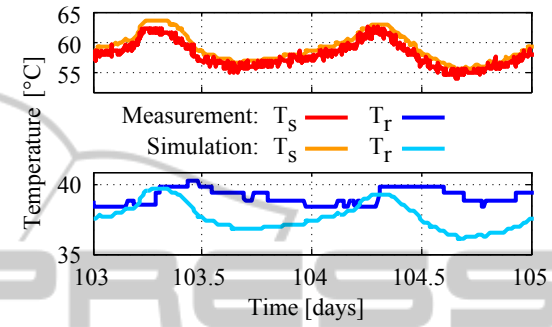

Figure 5: Flow rates over a year.

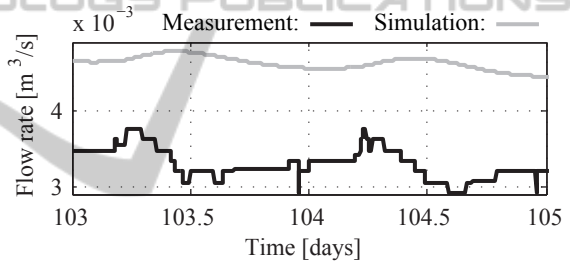Figure 6: Supply and return temperatures over two days.
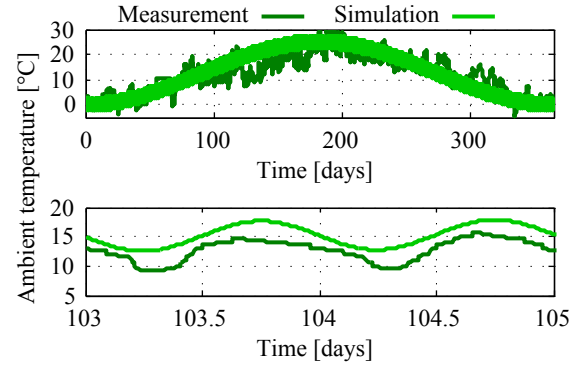

Figure 7: Flow rates over two days.

Figure 8: Measurement and simulation of the ambient temperature.

ambient temperatures over two days in the lower plot. The parameters for the signal generator are given in Table 1

For completeness Table 2 states the used material constants and Table 3 states all estimated parameters.

Table 1: Parameters of the ambient temperature signal generator.

Frequency	Symbol	Value
Day oscillation	ω_d	$-\left(\frac{2\pi}{86400}\right)^2 \frac{rad}{s}$
Year oscillation	ω_y	$-\left(\frac{2\pi}{31536000}\right)^2, \frac{rad}{s}$

Table 2: Material constants of water.

Density of water	ρ	$1000 \frac{Kg}{m^3}$
Specific heat capacity of water	c	$4182 \frac{Ws}{Kg\,K}$

Table 3: Parameters of the building model.

Parameter	Symbol	Value
Volume boiler	V_b	$1.05\,m^3$
Volume consumer	V_v	$5\,m^3$
Mean flow rate	$\dot{V}_m$	$4.75 \cdot 10^{-3} \frac{m^3}{s}$
Flow rate amplitude	$\dot{V}_a$	$3.25 \cdot 10^{-3} \frac{m^3}{s}$
Building reference temperature	T_{br}	$21\,°C$
Linear temperature amplitude of the thermostat	ΔT_b	$4\,°C$
Thermal capacity of the building	k_b	$10^{10} \frac{Ws}{K}$
Total heat transfer coefficient (radiator building)	$k_{r,b}$	$2.5 \cdot 10^4 \frac{W}{K}$
Total heat transfer coefficient (building ambiance)	$k_{b,a}$	$4.7338 \cdot 10^4 \frac{W}{K}$
Mean ambient temperature	$T_{a,m}$	$12.5\,°C$
Maximal power of the boiler	P_{max}	$1.1\,MW$

The initial state vector used is

$$\mathbf{x}(0) = \left(253 \quad 323 \quad 294 \quad 0 \quad -\frac{5\pi}{86400} \quad \cdots \right.$$
$$\left. -12.5 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \right)^T. \quad (39)$$

Note that the simulation was performed with SI-units, such that the temperatures are in Kelvin. For better legibility the temperatures in Table 3 were given in $°C$.

3.3 System Tensor

With the difference equations from the section 3.1 it is now straight forward to construct the system ten-

sor. Here just the first entries of the system tensors factor matrices, that correspond to the next state of the supply temperature T_s are given. The entries are constructed as shown in Example 2.1 The Factor matrices, in order of the states (24) are:

$$\mathbf{F}_{T_s} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & \ldots \\ 1 & 0 & 1 & 0 & 1 & \ldots \end{pmatrix}, \quad \mathbf{F}_{T_r} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & \ldots \\ 0 & 1 & 0 & 1 & 0 & \ldots \end{pmatrix}$$

$$\mathbf{F}_{T_b} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 1 & \ldots \end{pmatrix}, \quad \mathbf{F}_{T_{a1}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}$$

$$\mathbf{F}_{T_{a2}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}, \quad \mathbf{F}_{T_{a3}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}$$

$$\mathbf{F}_{T_{a4}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}, \quad \mathbf{F}_{X_l} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 1 & 1 & 1 & \ldots \end{pmatrix}$$

$$\mathbf{F}_{X_u} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}, \quad \mathbf{F}_{H_l} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}$$

$$\mathbf{F}_{H_u} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}, \quad \mathbf{F}_{U} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{pmatrix}$$

$$\mathbf{F}_{\Phi} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (40)$$

The matrix $\mathbf{F}_{\Phi}$ has dimension 12×46 to assign the 46 summands of the system to the corresponding next state. Note, that the factor matrices are not complete but just given for the first five summands of the first next state. Since all entries that were looked at correspond to the first next state, the first five entries of the first row of $\mathbf{F}_{\Phi}$ are 1. The corresponding part of the parameter vector is given by

$$\lambda_{\mathbf{f}} = \begin{bmatrix} 1 - \frac{t_s(\dot{V}_a+\dot{V}_m)}{V_b} \\ \frac{t_s(\dot{V}_a+\dot{V}_m)}{V_b} \\ \frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_b} \\ -\frac{\dot{V}_a t_s(\Delta T_b - T_{br})}{\Delta T_b V_b} \\ \frac{\dot{V}_a t_s}{\Delta T_b V_b} \\ \vdots \end{bmatrix}. \quad (41)$$

3.4 Tensor Decomposition

In this section a reduction of the tensor rank is performed using tensor decomposition techniques. With the factor matrices (40) and the parameter vector (41) defined in section 3 a state transition tensor

$$\mathsf{F} = \left[\mathbf{F}_{T_s}, \mathbf{F}_{T_r}, \mathbf{F}_{T_b}, \mathbf{F}_{T_{a1}}, \mathbf{F}_{T_{a2}}, \mathbf{F}_{T_{a3}}, \mathbf{F}_{T_{a4}}, \ldots \right. \quad (42)$$

$$\left. \mathbf{F}_{X_l}, \mathbf{F}_{X_u}, \mathbf{F}_{H_l}, \mathbf{F}_{H_u}, \mathbf{F}_{U}, \mathbf{F}_{\Phi} \right] \cdot \lambda_f \quad (43)$$

is constructed. The Tensor Toolbox (Bader and Kolda, 2012) and the command cp_als is used to reduce the rank of the tensor F, which was constructed

to have a tensor rank of 46. Using the Toolbox, the rank of the system tensor is reduced to 23, which has a fit of over 99%. The first five entries of the first two factor matrices are

$$\mathbf{F}_1 = \begin{pmatrix} -0.999 & 0.999 & 1.000 & 1.000 & 1.000 & \dots \\ 0.045 & -0.054 & 0.009 & 0.007 & 0.000 & \dots \end{pmatrix},$$

$$\mathbf{F}_2 = \begin{pmatrix} 0.988 & 0.984 & 0.999 & 1.000 & 1.000 & \dots \\ -0.152 & -0.179 & 0.040 & 0.031 & 0.000 & \dots \end{pmatrix}. \quad (44)$$

The corresponding entries in the parameter vector are

$$\lambda = \begin{pmatrix} 10.443 & 6.967 & 4.322 & 2.598 & 2.587 & \dots \end{pmatrix}^T. \quad (45)$$

The results of the simulation with the original and the decomposed state transition tensor are given in Figure 9. One can see, that the main system dynamics are captured with the decomposed system. The supply temperature drops to 75°C if 95°C is reached and rises again. Also the offset and the dynamics of the return temperature show similar behavior for both state transition functions.
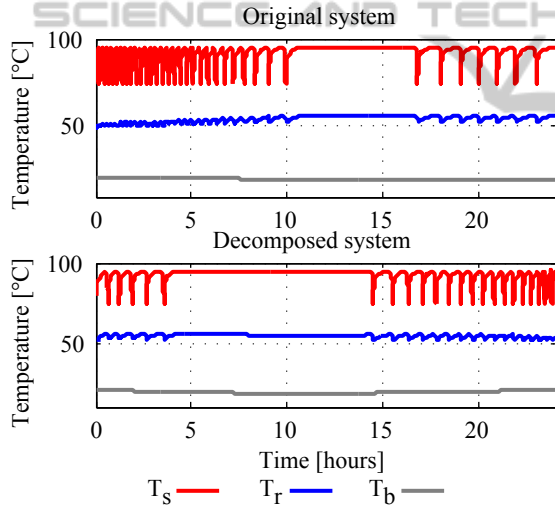

Figure 9: Original and decomposed system.

The value of the states has a big impact on the results. For this reason a state transformation (see section 2) was performed such that all states of the transformed system are in the interval [0 0.5]. Since we have a 12th order system a very small factor in front of the multiplication of all states (which was not used, i.e. 0 in the original system) can give a next state many orders of magnitude away from the original next state. This effect can be reduced significantly by normalizing the states to the interval [0 0.5]. To illustrate this issue assume, that $T_s = T_r = T_b = 300$. To calculate the next state of T_s the multiplication of all three temperatures is not used, i.e. $\Phi(T_s) = \dots + 0 \cdot T_s T_r T_b + \dots$.

Assume a CP decompostion is performed and the factor is now 0.001. This still gives a very good fit for the tensor. The next state of T_s however is $300^3 \cdot 0.001 = 27 \cdot 10^3$ different to that of the original system. In contrast to that consider a transformed system. Now the states are transformed. They are assumed to be in the interval [280 300] and therefore transformed with $\tilde{\mathbf{x}} = \frac{\mathbf{x}-280}{40}$ leading to $T_s = T_r = T_b = 0.5$. Again calculating the error of the next state gives $0.5^3 \cdot 0.001 = 125 \cdot 10^{-6}$ transforming this error back to the original coordinates gives an error of $5 \cdot 10^{-3}$.

4 MULTI-AGENT SYSTEMS EXAMPLE

A MAS consists of a group of agents (e.g. vehicles with their computing entities) that exchange information and share knowledge to solve a common task like in (Balaji and Srinivasan, 2010). The demand of autonomous MAS's with the ability to perform tasks as Decision-Making or Policy Formulation leads to hybrid MAS's. There are various different applications of hybrid MAS's and many different ways to describe them, e. g. (Srinivasan and Choy, 2010). How a hybrid MAS is modeled as tensor system is shown in the following.

Graphs are a natural way to describe the communication topology between agents of an MAS, see (Mesbahi and Egerstedt, 2010). A directed Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ consists of a set of N vertices, representing the single agents, and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, describing the communication topology. If agent i receives information from agent k, there is an edge $(k,i) \in \mathcal{E}$. The mainly used matrix representation of graphs for MAS is the Laplacian $\mathcal{L}$. The normalized Laplacian is defined as (Ren and Beard, 2007)

$$\mathcal{L}_{ik} := \begin{cases} 1 & \text{if } i = k \text{ and } |\mathbf{N}_i| \neq 0 \\ -\frac{1}{|\mathbf{N}_i|} & \text{if } k \in \mathbf{N}_i \\ 0 & \text{otherwise.} \end{cases} \quad (46)$$

Here $\mathbf{N}_i$ is the set of neighbors of agent i, that is defined as $\mathbf{N}_i = \{k|(k,i) \in \mathcal{E}\}$, and $|\mathbf{N}_i|$ is its cardinality.

Consider a MAS of N equal agents, where each agent $i = 1 \dots N$ is described by the switched system

$$\Phi(x_i) = ax_i + u_i \text{ if } \Phi^{-1}(x_i) \geq 0.5$$
$$\Phi(x_i) = bx_i + u_i \text{ if } \Phi^{-1}(x_i) < 0.5, \quad (47)$$

with $x_i, u_i \in \mathbb{R}$ being continuous signals and $\Phi^{-1}(x_i) = x_i(k-1)$ denoting the previous state. By introducing the discrete signal $\underline{z}_i = \beta(\Phi^{-1}(x_i))$,

being the quantized state variable, system (47) can be written as the multilinear system

$$\Phi(x_i) = (a-b)\underline{z}_i x_i + b x_i + u_i. \qquad (48)$$

This can be described as a tensor system for one agent i by

$$\Phi(x_i) = \left\langle \, \mathsf{F}^i \mid \underline{\mathsf{M}}(x,\underline{z},u) \, \right\rangle = \left\langle \, \left[\mathbf{F}_u^i, \mathbf{F}_z^i, \mathbf{F}_x^i \right] \mid \underline{\mathsf{M}}(x,\underline{z},u) \, \right\rangle \qquad (49)$$

with $\mathbf{F}_u^i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{F}_z^i = \begin{pmatrix} b & 1 \\ a-b & 0 \end{pmatrix}, \mathbf{F}_x^i = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$

The control input is determined by the communication topology as $\mathbf{u} = -\mathcal{L}\mathbf{x}$ like simple consensus algorithms propose (Mesbahi and Egerstedt, 2010) with $\mathbf{x} = (x_1 \ldots x_N)^T$, $\mathbf{u}$ and $\underline{z}$ defined respectively.

The discrete system is easily described by

$$\Phi(\underline{z}) = \beta(\mathbf{x}). \qquad (50)$$

The continuous part can be described as a continuous tensor system of the form $\Phi(\mathbf{x}) = \left\langle \, \mathsf{F} \mid \underline{\mathsf{M}}(\mathbf{x},\underline{z}) \, \right\rangle^+$. It is obvious that the continuous system only contains multilinear terms of the form $x_i \underline{z}_i$ introduced by the system equations of the single agents in (48), whereas the control input is linear in $\mathbf{x}$. Thus the standard monomial tensor of dimension $\mathbb{R}^{\times 2N2}$ is very sparse. A special formation monomial tensor is introduced here as $\underline{\mathsf{M}}^\mathsf{F}(\mathbf{x},\underline{z}) \in \mathbb{H}^{2 \times 2 \times N}$

$$\underline{\mathsf{M}}^\mathsf{F}(\mathbf{x},\underline{z}) = \left[\begin{pmatrix} 1 & \cdots & 1 \\ \underline{z}_1 & & \underline{z}_N \end{pmatrix}, \begin{pmatrix} 1 & \cdots & 1 \\ x_1 & & x_N \end{pmatrix}, \mathbf{I}_N \right]. \qquad (51)$$

This is a tensor of rank N in contrast to the standard monomial tensor of rank 1.

With (51) a state transition function of a hybrid tensor system for the overall system, with state vector $\underline{\mathbf{x}} = (\mathbf{x},\underline{z})^T$ is defined as

$$\Phi(\underline{\mathbf{x}}) = \left\langle \, \underline{\mathsf{F}}^\mathsf{F} \mid \underline{\mathsf{M}}^\mathsf{F}(\mathbf{x},\underline{z}) \, \right\rangle^{\boxplus} \qquad (52)$$

with $\underline{\mathsf{F}}^\mathsf{F} = \left[\mathbf{F}_z^F, \mathbf{F}_x^F, \mathbf{F}_\mathbf{I}^F, \mathbf{F}_\Phi^F \right] \in \mathbb{R}^{2 \times 2 \times N \times 2N}$. The state transition tensor F^F has the factor matrices with $2N$ factors

$$\mathbf{F}_\Phi^F = \begin{pmatrix} \mathbf{I}_N & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{pmatrix},$$
$$\mathbf{F}_\mathbf{I}^F = \begin{pmatrix} \mathbf{I}_N & -\mathcal{L} & \mathbf{I}_N \end{pmatrix},$$
$$\mathbf{F}_x^F = \begin{pmatrix} \mathbf{1}_{3N}^T \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix},$$
$$\mathbf{F}_z^F = \begin{pmatrix} \mathbf{1}_N^T \begin{pmatrix} b \\ a-b \end{pmatrix} & \mathbf{1}_{2N}^T \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix}. \qquad (53)$$

The factor matrix $\mathbf{F}_\Phi^F$ specifies the state, $\underline{\mathbf{x}} = (\mathbf{x},\underline{z})^T$. First the continuous state vector $\mathbf{x}$ is considered. Due

to the identity matrices in $\mathbf{F}_\Phi^F$ and $\mathbf{F}_\mathbf{I}^F$, the first N columns of the factor matrices describes the internal dynamics of the agents. This can be easily seen by comparing the first N columns of $\mathbf{F}_x^F$ and $\mathbf{F}_z^F$ with $\mathbf{F}_x^i$ and $\mathbf{F}_z^i$ in (49). The Laplacian in $\mathbf{F}_\mathbf{I}^F$ describes the communication between the different agents. The fact, that the communication is linear in the states x^i, explains the second N vectors in $\mathbf{F}_x^F$ and $\mathbf{F}_z^F$ since

$$\left\langle \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right], \left[\begin{pmatrix} 1 \\ \underline{z}_i \end{pmatrix}, \begin{pmatrix} 1 \\ x_i \end{pmatrix} \right] \right\rangle = x_i.$$

The discrete signal $\underline{z}$ is simply calculated by $\Phi(\underline{z}) = \beta(\mathbf{x})$, which explains the choice of the last N columns in $\mathbf{F}_x^F$ and $\mathbf{F}_z^F$.

Next, a hybrid tensor system with $N = 100$ agents is modeled. The considered connected graph is directed and contains 473 random edges. The internal dynamics of the single agents is described by (48) with $a = 0.5$ and $b = 1$. The number of factors of the exact system tensor is $300 = 3N$. CP decompositions of the system tensor from rank 1 to 299 have been calculated with cp_als.
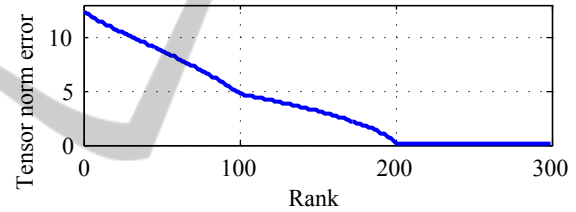

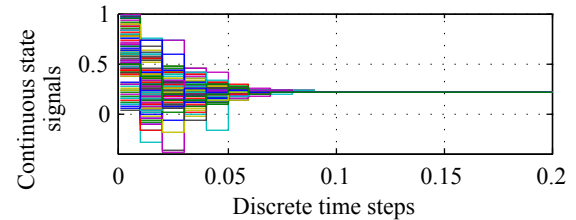Figure 10: Frobenius norm of error.

Figure 11: Continuous valued state trajectories for rank 200 approximation.

The Frobenius error between the exact and approximated hybrid tensor model is shown for tensor approximations of different rank in Figure 10. A tensor of rank $200 = 2N$ describes the system dynamics almost exactly. The Frobenius error for graphs of every tested different size shows qualitatively the same shape. Thus it can be supposed that the system is in fact not a tensor system of rank $3N$ but of $2N$. In Figure 11, continuous state trajectories of the rank 200 approximation are shown. The maximal mean squared error between the original and approximated trajectories for a rank 200 approximation is
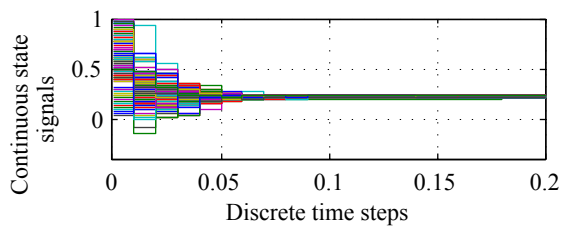
Figure 12: Continuous valued state trajectories for rank 150 approximation.

with $< 10^{-17}$ negligible and thus approximation and original system match exactly. For comparison the rank 150 approximation is shown on Figure 12. Here the same consensus value of approximately 0.2 is reached with the same convergence shape. However, the states does not exactly converge to the real consensus value but reach an envelope around it, which gets small for higher order approximations. For approximations with smaller order instability of the convergence process may occur.

5 CONCLUSIONS

Hybrid tensor systems are an adequate framework for modeling discrete time multilinear systems with continuous and discrete valued states and inputs. A hybrid tensor model of a complex heating system is derived as a real-world example. All parameters are stored in a state transition tensor which in a second step is reduced to a low rank Kruskal tensor using standard decomposition techniques. As simulations show, this decomposed tensor is still capable to capture the main dynamics of the heating system.

As a second application example from a quite different application domain, a hybrid tensor model of a Multi-Agent System (MAS) is built. Structural contraints are imposed in an easy way, a reduced rank state transition tensor of the system is again computed by tensor decomposition algorithms and simulations are carried out based on the decomposed model. The reduced rank model converges to the same final values as the original one. Moreover, the original N-agents system can be modeled exactly by a hybrid tensor system with a small rank of $2N$.

Further research will be done to investigate the stability behaviour of hybrid tensor systems. Another focus will be the derivation of tensor representations for nonlinear normal forms - as well for general nonlinear as for multilinear systems. Tensor decomposition techniques will play an important role in these fields and extensions of the continuous algorithms to hybrid spaces would be essential tools for analysis and design of multilinear hybrid systems.

REFERENCES

Bader, B. and Kolda, T. (2012). MATLAB Tensor Toolbox Version 2.5. Available online.

Balaji, P. and Srinivasan, D. (2010). *An Introduction to Multi-Agent Systems*, volume 310 of *Studies in Computational Intelligence*, chapter 1, pages 1–27. Springer.

Cichocki, A., Zdunek, R., Phan, A., and Amari, S. (2009). *Nonnegative Matrix and Tensor Factorizations*. Wiley, Chichester.

Hackbusch, W. (2012). *Tensor Spaces and Numerical Tensor Calculus*, volume 42 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg.

Kolda, T. and Bader, B. (2009). Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500.

Lichtenberg, G. (2010). Tensor Representation of Boolean Functions and Zhegalkin Polynomials. In *International Workshop on Tensor Decompositions*.

Lichtenberg, G. (2011). *Hybrid Tensor Systems*. Habilitation, Hamburg University of Technology.

Lichtenberg, G. and Eichler, A. (2011). Multilinear Algebraic Boolean Modelling with Tensor Decompositions Techniques. In *18th IFAC World Congress*, page TuC01.2.

Mesbahi, M. and Egerstedt, M. (2010). *Graph Theoretic Methods for Multiagent Networks*. Princeton University Press.

Nouidui, T.S., Zuo, K.P.W., and Wetter, M. (2012). Validation and Application of the Room Model of the Modelica Buildings Library. *9th International Modelica Conference*, pages 727–736.

Pangalos, G. and Lichtenberg, G. (2012). Approach to Boolean Controller Design by Algebraic Relaxation for Heating Systems. In *4th IFAC Conference on Analysis and Design of Hybrid Systems*.

Ren, W. and Beard, R. (2007). *Distributed Consensus in Multi-Vehicle Cooperative Control - Theory and Applications*. Communications and Control Engineering. Springer Publishing Company, Incorporated.

Ren, W. and Cao, Y. (2011). *Distributed Coordination of Multi-agent Networks*. Springer-Verlag London Limited.

Srinivasan, D. and Choy, M. (2010). *Hybrid Multi-Agent Systems*, volume 310 of *Studies in Computational Intelligence*, chapter 2, pages 29–42. Springer.

Wetter, M. (2006). Multizone Building Model for Thermal Building Simulation in Modelica. *5th International Modelica Conference*, pages 517–526.