Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Simulation with Qualitative Models in Reduced Tensor Representations

Thorsten Müller-Eping * **Gerwald Lichtenberg** **

* *Fraunhofer Institute for Solar Energy Systems - ISE, 79110 Freiburg, Germany (e-mail: thorsten.mueller-eping@ise.fraunhofer.de)*
** *Hamburg University of Applied Sciences, 21033 Hamburg, Germany (e-mail: gerwald.lichtenberg@haw-hamburg.de)*

Abstract: The paper proposes simulation algorithms for tensor representations of qualitative models based on stochastic automata. We show that storing the transition probabilities of the automaton in tensor formats will help to break the *curse of dimensionality*, i.e. to overcome the storage complexity problem of the automaton that occurs due to the exponential growth in the quantity of automata transitions when the number of input, state and output signals of the underlying discrete-time system rises. In addition, we present the application of a modern tensor optimization method for the completion of qualitative models identified by data-driven black-box approaches and thus suffering from the problem of unobserved sets of training data.

Keywords: Qualitative Simulation, Quantized Systems, Stochastic Automata, Model Reduction, Tensor Decomposition, Tensor Completion

1. INTRODUCTION

Qualitative models deal with a symbolic representation of the input, state and output signals of a underlying discrete-time system. They have been proposed decades ago and can be used for observation, fault diagnosis and simulation (Lichtenberg, 1998; Lichtenberg and Lunze, 1997; Schröder, 2003). However, the storage complexity of the underlying stochastic automaton increases rapidly with the number of system signals to be considered by the qualitative model. Thus, reducing the storage complexity of the stochastic automaton is the main challenge for making qualitative models applicable to large discrete-time systems.

In previous work we presented an approach to reduce the storage amount of qualitative models by storing the transition probabilities of the automaton in a tensor, which is then reduced by a factorization method known as *canonical polyadic (CP) tensor decomposition* (Kolda and Bader, 2009). For the resulting so-called *CP tensor representation*, which is usually an approximation of the qualitative model, we have provided algorithms for qualitative observation and fault detection that exploit the reduced model structure and therefore can be efficiently implemented (Müller et al., 2015; Müller-Eping et al., 2017). Here we focus on *qualitative simulation* with regard to applications like supervisory control, predictive maintenance or the investigation of qualitative properties of stochastic systems behavior. That is, we first introduce a new description of qualitative models based on an exact *direct* CP tensor representation. Furthermore, we show how a modern mathematical method, referred to as *tensor completion* (Acar et al., 2011) can be used to overcome the problem of incomplete qualitative models identified by data-driven black-box approaches using historical measurement data of the underlying process.

The structure of the paper is as follows: Section 2 recalls the basics of qualitative models and the principles of qualitative simulation. In Section 3, the definitions and notations of tensors are given while Section 4 introduces the simulation with qualitative models in tensor representation. Section 5 presents the direct CP representation of qualitative models. How qualitative models can be further reduced and completed via CP tensor decomposition is investigated in Section 6. The paper concludes with an application example in Section 7.

2. QUALITATIVE MODELS

In this section we introduce the qualitative models and how they can be used for qualitative simulation.

2.1 Quantized System

In Fig. 1, the process with the input vector $\mathbf{u}(k) \in \mathbb{R}^m$, the state vector $\mathbf{x}(k) \in \mathbb{R}^n$ and the output vector $\mathbf{y}(k) \in \mathbb{R}^q$ is a discrete-time, continuous-variable system

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \ , \tag{1}$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \ , \tag{2}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \ , \tag{3}$$

where the behavior of the process is described by the vector functions $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$.

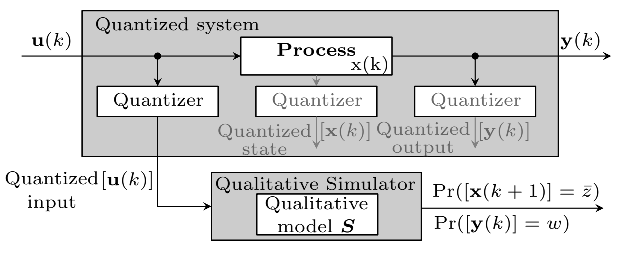

Fig. 1. Qualitative simulation of a quantized system

For an arbitrary time horizon T and a given input sequence

$$\mathbf{U}(0...T) = (\mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(T)) \ , \tag{4}$$

the process generates a unique state and output sequence

$$\mathbf{X}(0...T) = (\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(T)) \ , \tag{5}$$

$$\mathbf{Y}(0...T) = (\mathbf{y}(0), \mathbf{x}(1), \ldots, \mathbf{y}(T)) \ . \tag{6}$$

The *quantized system* consists of the process as well as the *quantizers* and shows the *qualitative* behavior of the process, which will be described by a qualitative model (Lichtenberg, 1998). The quantizers transform the real-valued input $\mathbf{u}(k)$, state $\mathbf{x}(k)$ and output $\mathbf{y}(k)$ vectors into a scalar input $[\mathbf{u}(k)]=v\in\mathcal{U}$, state $[\mathbf{x}(k)]=z\in\mathcal{X}$ and output $[\mathbf{y}(k)]=w\in\mathcal{Y}$, where $\mathcal{U}:=\{1,\ldots,M\}$, $\mathcal{X}:=\{1,\ldots,N\}$ and $\mathcal{Y}:=\{1,\ldots,Q\}$ are finite sets, called the qualitative input, state and output space (Lichtenberg and Lunze, 1997). The integer values v,z and w which the quantized input $[\mathbf{u}(k)]$, state $[\mathbf{x}(k)]$ and output $[\mathbf{y}(k)]$ can take, are called *qualitative input, state* and *output*. The realization of the quantization procedure is introduced by a rectangular partitioning of the quantitative input $\mathbb{R}^m$, state $\mathbb{R}^n$ and output $\mathbb{R}^q$ space, where each space is separated into disjoint sets $\mathcal{Q}_u(v)$, $\mathcal{Q}_y(w)$ and $\mathcal{Q}_x(z)$, such that e.g. for the state space

$$\forall i \neq j : \mathcal{Q}_x(i) \cap \mathcal{Q}_x(j) = \emptyset \quad i,j \in \mathcal{X} \ , \tag{7}$$

$$\bigcup_{i\in\mathcal{X}} \mathcal{Q}_x(i) = \mathbb{R}^n \ , \tag{8}$$

hold (Lichtenberg and Lunze, 1997). Due to the partitioning, the qualitative inputs v, states z and outputs w are given by the relations

$$[\mathbf{u}(k)] = v \iff \mathbf{u}(k) \in \mathcal{Q}_u(v) \ , \tag{9}$$

$$[\mathbf{x}(k)] = z \iff \mathbf{x}(k) \in \mathcal{Q}_x(z) \ , \tag{10}$$

$$[\mathbf{y}(k)] = w \iff \mathbf{y}(k) \in \mathcal{Q}_y(w) \ . \tag{11}$$

That is e.g., the qualitative value $z \in \mathcal{X}$ of the quantized state $[\mathbf{x}(k)]$ is given by the number of the partition $\mathcal{Q}_x(z)$ to which the vector $\mathbf{x}(k)$ belongs. Due to the quantization (9)–(11), the quantitative sequences of the inputs, states and outputs (4)–(6) are transformed into the qualitative input, state and output trajectories

$$[\mathbf{U}(0...T)] = ([\mathbf{u}(0)] = v(0), \ldots, [\mathbf{u}(T)] = v(T)) \ , \tag{12}$$

$$[\mathbf{X}(0...T)] = ([\mathbf{x}(0)] = z(0), \ldots, [\mathbf{x}(T)] = z(T)) \ , \tag{13}$$

$$[\mathbf{Y}(0...T)] = ([\mathbf{y}(0)] = w(0), \ldots, [\mathbf{y}(T)] = w(T)) \ . \tag{14}$$

In Fig. 2 the separation of a two dimensional output space into six partitions $\mathcal{Q}_y(1),\ldots,\mathcal{Q}_y(6)$ is exemplarily shown.

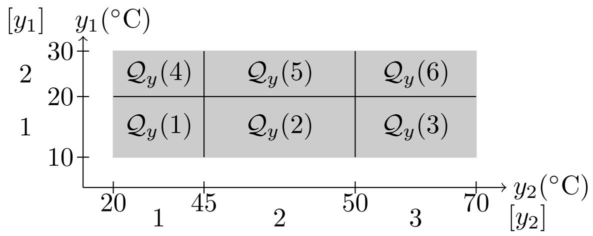

Fig. 2. Partitioning of a 2-dimensional output space $\mathbb{R}^2$

2.2 Stochastic Automata as Qualitative Model

Assume a discrete stochastic process with random inputs $V(k)$, states $Z(k)$ and outputs $W(k)$. That is, for a given random input $V(k)=v \in \mathcal{V}:=\{1,..,M\}$ the stochastic process passes some state $Z(k)=z \in \mathcal{Z}:=\{1,..,N\}$ and produces an output $W(k)=w \in \mathcal{W}:=\{1,..,Q\}$. The behavior of such processes, i.e. how the states and outputs behave for a given input, can be described by a stochastic automaton which is defined next.

Definition 2.1. (Stochastic Automaton (Schröder, 2003)). The 5-tuple

$$\boldsymbol{S}^* = (\mathcal{V}, \mathcal{Z}, \mathcal{W}, \mathcal{L}^*, \mathbf{p}_0^*) \tag{15}$$

consisting of the set of inputs $\mathcal{V}$, the set of states $\mathcal{Z}$ and the set of outputs $\mathcal{W}$, as well as the behavior relation $\mathcal{L}$ and the initial state distribution $\mathbf{p}_0$, is called *stochastic automaton* (SA). The *behavior relation* of the SA

$$\mathcal{L}^*(\bar{z}, w \,|\, z, v) = \Pr\left(\begin{matrix} Z(k+1) = \bar{z}, \\ W(k) = w \end{matrix} \left| \begin{matrix} Z(k) = z, \\ V(k) = v \end{matrix} \right. \right) \tag{16}$$

denotes the conditional probability that the SA changes its state from z to the successor state $\bar{z}$ while receiving the input v and giving the output w, where

$$\forall z \in \mathcal{Z}, v \in \mathcal{V} : \sum_{\bar{z}=1}^{N} \sum_{w=1}^{Q} \mathcal{L}(\bar{z}, w \,|\, z, v) = 1 \tag{17}$$

holds. Note that the one-step transition probabilities (16) are time-invariant, what makes the SA (15) a *homogeneous Markov chain* of first order (Blanke et al., 2006). The vector $\mathbf{p}_0^* \in [0,1]^N$ describes the initial state distribution, i.e. its components are given by the probabilities $\Pr(Z(0) = z), z \in \mathcal{Z}$.

Definition 2.2. (Qualitative Model (Lichtenberg, 1998)). For using a SA as *qualitative model*, the sets of the inputs $\mathcal{V}$, states $\mathcal{Z}$ and outputs $\mathcal{W}$ in (15) are replaced by the sets of qualitative inputs $\mathcal{U}$, states $\mathcal{X}$ and outputs $\mathcal{Y}$, such that the SA is given by

$$\boldsymbol{S} = (\mathcal{U}, \mathcal{X}, \mathcal{Y}, \mathcal{L}, \mathbf{p}_0) \ . \tag{18}$$

Now, each automaton input $v \in \mathcal{U}$, state $z \in \mathcal{X}$ and output $w \in \mathcal{Y}$ is associated with a quantized input $[\mathbf{u}(k)]$, state $[\mathbf{z}(k)]$ and output $[\mathbf{y}(k)]$ and the behavior relation is defined by

$$\mathcal{L}(\bar{z}, w \,|\, z, v) = \Pr\left(\begin{matrix} [\mathbf{x}(k+1)] = \bar{z}, \\ [\mathbf{y}(k)] = w \end{matrix} \left| \begin{matrix} [\mathbf{x}(k)] = z, \\ [\mathbf{u}(k)] = v \end{matrix} \right. \right) . \tag{19}$$

The initial distribution of the qualitative states is given by the vector $\mathbf{p}_0$ with components $\Pr([\mathbf{x}(0)] = z), z\in\mathcal{X}$. If there is no knowledge about this probability distribution available, a uniform distribution is used at time $k = 0$:

$$\Pr([\mathbf{x}(0)] = z) = 1/N \ , \quad \forall z \in \mathcal{X} := \{1,\ldots,N\} \ . \tag{20}$$

Note that the qualitative model will not be a precise model of the quantized system, because the underlying SA (18) possesses the Markov property but the quantized system usually does not (Blanke et al., 2006).

2.3 Model Identification

In Schröder (2003) different identification methods for qualitative models are given. Here we use a black box method referred to as *stochastic qualitative identification*, which is based on determining the conditional probabilities (19) by the use of historical measurement data of the underlying process (Lichtenberg, 1998). Therefore, the elements of the measured quantized input, output and state sequences (12) − (14) are sorted in tuples for $k = 0, 1, \ldots, T$

$$[\Theta(k)] = ([\mathbf{x}(k+1)], [\mathbf{y}(k)], [\mathbf{x}(k)], [\mathbf{u}(k)]) \ .$$

Then, the relative frequencies

$$\hat{h}(\bar{z}, w, z, v) = \frac{\#\ \text{n-tuples}\,[\Theta(k)] = (\bar{z}, w, z, v)}{T} \ ,$$

can be determined which are approximations of the joint probabilities

$$\hat{h}(\bar{z}, w, z, v) \approx \Pr\left(\begin{matrix} [\mathbf{x}(k+1)] = \bar{z}, \\ [\mathbf{y}(k)] = w \end{matrix} \left| \begin{matrix} [\mathbf{x}(k)] = z, \\ [\mathbf{u}(k)] = v \end{matrix} \right. \right) .$$

After that, an approximation of the boundary distributions

$$\hat{h}(z, v) = \sum_{\bar{z}=1}^{N} \sum_{w=1}^{Q} \hat{h}(\bar{z}, w, z, v)$$

can be calculated, what finally leads to the conditional probabilities

$$\mathcal{L}(\bar{z}, w|z, v) = \begin{cases} \frac{\hat{h}(\bar{z},w,z,v)}{\hat{h}(z,v)} & \text{if } \hat{h}(z,v) \neq 0 \ , \\ 0 & \text{if } \hat{h}(z,v) = 0 \ . \end{cases} \quad (21)$$

Note that the qualitative model identified by this procedure may be *structurally incomplete*, i.e. the condition (17) may not be fulfilled. A solution to this issue is investigated in Section 6.

2.4 Qualitative Simulation

The procedure of qualitative simulation is straightly related to the simulation of stochastic automata (see Blanke et al. (2006)). Here we introduce the mathematical background directly in the sense of qualitative modeling. In a first step, the *transition* and *output relation*, which are boundary distributions of the conditional probability (19), have to be determined.

Definition 2.3. (Transition Relation (Schröder, 2003)). The *transition relation*

$$\mathcal{F}(\bar{z}\,|\,z, v) = \sum_{w=1}^{Q} \mathcal{L}(\bar{z}, w\,|\,z, v) \quad (22)$$

yields the conditional probability, that the SA moves from state z to the successor state $\bar{z}$ while receiving the input v.

Definition 2.4. (Output Relation, (Lichtenberg, 1998)). The *output relation*

$$\mathcal{G}(w\,|\,z, v) = \sum_{\bar{z}=1}^{N} \mathcal{L}(\bar{z}, w\,|\,z, v) \quad (23)$$

represents the conditional probability, that the SA gives the output w if being in state z and receiving the input v.

The aim of qualitative simulation is the calculation of the probabilities $\Pr([\mathbf{x}(k+1)] = \bar{z})$ and $\Pr([\mathbf{y}(k)] = w)$ for each of the qualitative states $\bar{z} \in \mathcal{X}$ and qualitative outputs $w \in \mathcal{Y}$ for a certain distribution of qualitative inputs $\Pr([\mathbf{u}(k)]=v), v \in \mathcal{U}$. This is realized as follows.

Definition 2.5. (Qualitative Simulation (Lunze, 1998)). For a given probability distribution $\Pr([\mathbf{u}(k)] = v), v \in \mathcal{U}$ of the inputs, the probabilities of the qualitative states

$$\begin{aligned} &\Pr([\mathbf{x}(k+1)] = \bar{z}) \quad (24)\\ &= \sum_{z=1}^{N} \sum_{v=1}^{M} \mathcal{F}(\bar{z}\,|\,z, v) \Pr([\mathbf{x}(k)] = z) \Pr([\mathbf{u}(k)] = v) \ , \end{aligned}$$

and qualitative outputs

$$\begin{aligned} &\Pr([\mathbf{y}(k)] = w) \quad (25)\\ &= \sum_{z=1}^{N} \sum_{v=1}^{M} \mathcal{G}(w\,|\,z, v) \Pr([\mathbf{x}(k)] = z) \Pr([\mathbf{u}(k)] = v) \ . \end{aligned}$$

can be obtained for all $z \in \mathcal{X}$ and $w \in \mathcal{Y}$. At time $k = 0$ the recursive equations (24), (25) are initialized with the probability distribution of the qualitative states (20). If the inputs are exactly known and given by a unique input sequence like (12), then the probability of this unique input at time k is one, i.e.

$$\Pr([\mathbf{u}(k)] = v(k)) = 1 \ . \quad (26)$$

In this case, the summation over v in (24) and (25) vanishes and the inputs can be directly indexed by $v(k)$:

$$\Pr([\mathbf{x}(k+1)] = \bar{z}) = \sum_{z=1}^{N} \mathcal{F}(\bar{z}\,|\,z, v(k)) \Pr([\mathbf{x}(k)] = z) \ , \quad (27)$$

$$\Pr([\mathbf{y}(k)] = w) = \sum_{z=1}^{N} \mathcal{G}(w\,|\,z, v(k)) \Pr([\mathbf{x}(k)] = z) \ . \quad (28)$$

Remark 2.1. Note that strictly spoken, the state and output probabilities are conditional probabilities. That is, calculating the state and output probability distributions (24), (25) or (27), (28) for times $k = 0, 1, \ldots, T$, the state and output probabilities at time T depend on the whole sequence of input distributions or unique inputs, that have been occurred between $k = 0$ and $k = T$.

2.5 Storage Amount of Qualitative Models

The storage amount of qualitative models rises rapidly with the number of qualitative inputs M, states N and outputs Q. That is, the conditional probabilities of the behavior relation $\mathcal{L}(\bar{z}, w\,|\,z, v)$ have to be stored for all combinations of inputs $v \in \mathcal{U}$, outputs $w \in \mathcal{Y}$, states and successor states $z, \bar{z} \in \mathcal{Z}$, what leads to a number of values to be stored of

$$\alpha = MN^2Q \ . \quad (29)$$

In the following, we introduce a model reduction method, which is based on a *tensor representation* of the conditional probabilities $\mathcal{L}(\bar{z}, w\,|\,z, v)$, $\mathcal{F}(\bar{z}\,|\,z, v)$ and $\mathcal{G}(w\,|\,z, v)$.

3. TENSOR CALCULUS

In this section we give the basics of tensors and tensor operations needed for the simulation with qualitative models in tensor representation. The standard definitions used within this paper can be found in Cichocki et al. (2009); Kolda and Bader (2009); Lee and Cichocki (2018).

3.1 Basic Definitions and Notations

Fig. 3 depicts a tensor of order three which is defined in the following.

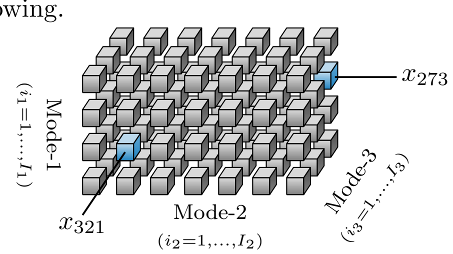

Fig. 3. $3rd$ order tensor $\mathsf{X} \in \mathbb{R}^{4\times7\times3}$ (Cichocki et al., 2009)

Definition 3.1. (Tensor (Cichocki et al., 2009)). A Nth order *tensor*

$$\mathsf{X} \in \mathbb{R}^{I_1\times\cdots\times I_N} \quad (30)$$

is a N-way array with elements $x(i_1, \ldots, i_N)$ indexed by $i_k \in \{1, \ldots, I_k\}$ for tensor modes $k = 1, \ldots, N$.

Table 1. Notations

$w \in \mathbb{R}\,, \mathbf{x} \in \mathbb{R}^I$	Scalar, Vector
$\mathbf{Y} \in \mathbb{R}^{I\times J}\,, \mathsf{Z} \in \mathbb{R}^{I_1\times\cdots\times I_N}$	Matrix, Tensor of order N
$z(i_1, \ldots, i_N) \in \mathbb{R}$	$(i_1, \ldots, i_N)$th element of Z
$\mathbf{z}(:, i_2, \ldots, i_N) \in \mathbb{R}^{I_1}$	Mode-1 tensor fiber of Z, a vector
$\mathbf{Z}(:, :, i_3, \ldots, i_N) \in \mathbb{R}^{I_1\times I_2}$	Tensor slice of Z, a (sub)matrix

As Tab. 1 shows, the elements of a tensor are indexed by comma separated values in brackets. Thus, $x(i_1, i_2, i_3)$ denotes the (i_1, i_2, i_3)th element of a third order tensor $\mathsf{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$. In Figures, we also use standard subscript indices like $x_{i_1i_2i_3} = x(i_1, i_2, i_3)$. In the tensor-context, matrices are interpreted as tensors of order two, vectors are first order tensors and scalars are tensors of order zero (Kolda and Bader, 2009). Tensors of order N can be transformed into lower order tensors, e.g. into matrices or vectors and vice-versa. In this paper, only the so-called *vectorization*, where a Nth order tensor is flattened into a vector, is needed. To define the vectorization, we use a *multi-index* notation.

Definition 3.2. (Multi-index (Lee and Cichocki, 2018)). The big-endian *multi-index*

$$\overline{i_1 \cdots i_N} = i_N + (i_{N-1} - 1)J_N + \cdot\cdot + (i_1 - 1)I_2 \cdot\cdot I_N \quad (31)$$

merges indices $i_k \in \{1, \ldots, I_k\}$ for $k = 1, \ldots, N$ into a single integer $\overline{i_1 \cdots i_N} \in \mathbb{N}$.

Definition 3.3. (Vectorization (Lee and Cichocki, 2018)). The *vectorization* of a tensor $\mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined as

$$\mathbf{z} = \mathsf{vec}(\mathsf{X}) \in \mathbb{R}^{I_1 \cdots I_N} , \quad (32)$$

where the components of the vector $\mathbf{z}$ are given by

$$z(\overline{i_1 \cdots i_N}) = x(i_1, \ldots, i_N) . \quad (33)$$

The vectorization of a $3rd$ order tensor is shown in Fig. 4.

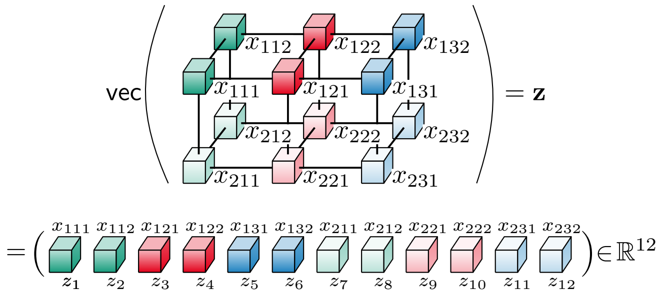

Fig. 4. Vectorization of a $3rd$ order tensor $\mathsf{X} \in \mathbb{R}^{2\times3\times2}$

3.2 Tensor Products

In this subsection we define basic mathematical products of linear and multi-linear algebra.

Definition 3.4. (Kronecker Product). The *Kronecker product* of N vectors $\mathbf{x}_n \in \mathbb{R}^{I_n}$, $n = 1, \ldots, N$ is defined as a vector

$$\mathbf{z} = \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_N \in \mathbb{R}^{I_1 \cdots I_N} , \quad (34)$$

with components

$$z(\overline{i_1 \cdots i_N}) = x_1(i_1) \cdots x_N(i_N) . \quad (35)$$

Definition 3.5. (Outer Product (Kolda and Bader, 2009)). The *outer product* of N vectors $\mathbf{x}_n \in \mathbb{R}^{I_n}$, $n = 1, .., N$ gives a tensor of order N

$$\mathsf{Z} = \mathbf{x}_1 \circ \cdots \circ \mathbf{x}_N \in \mathbb{R}^{I_1 \times \cdots \times I_N} , \quad (36)$$

with elements defined by

$$z(i_1, \ldots, i_N) = x_1(i_1) \cdots x_N(i_N) . \quad (37)$$

The outer product yields a tensor of rank one, what is denoted by $\mathsf{rank}(\mathsf{Z}) = 1$. Fig. 5 depicts the outer product of three vectors.

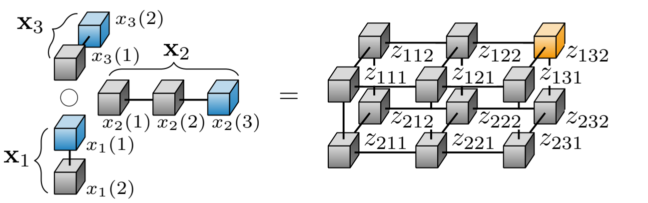

Fig. 5. The outer product of three vectors $\mathbf{x}_1 \in \mathbb{R}^2, \mathbf{x}_2 \in \mathbb{R}^3$ and $\mathbf{x}_3 \in \mathbb{R}^2$ gives a $3rd$ order rank-1 tensor $\mathsf{Z} \in \mathbb{R}^{2\times3\times2}$

Definition 3.6. (k-Mode Tensor Vector Product (Kolda and Bader, 2009)). The *k-mode product* of a Nth order tensor $\mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and a vector $\mathbf{y} \in \mathbb{R}^{I_k}, k = 1, \ldots, N$ gives a tensor of order $N - 1$

$$\mathsf{Z} = \mathsf{X} \ \bar{\times}_k \ \mathbf{y} \in \mathbb{R}^{I_1 \times \cdots \times I_{k-1} \times I_{k+1} \times \cdots \times I_N} , \quad (38)$$

with elements defined by

$$z(i_1, .., i_{k-1}, i_{k+1}, .., i_N) = \sum_{i_k=1}^{I_N} x(i_1, .., i_N)\, y(i_k) . \quad (39)$$

Fig. 6 shows a 2-mode tensor vector product.

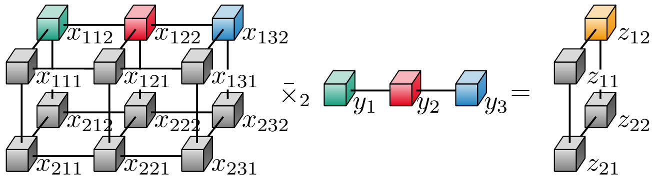

Fig. 6. 2-mode tensor vector product of a $3rd$ order tensor $\mathsf{X} \in \mathbb{R}^{2\times3\times2}$ and a $1st$ order tensor $\mathbf{y} \in \mathbb{R}^3$. Summing up the products of the same-colored elements gives the element z_{12} of the $2nd$ order tensor $\mathbf{Z} \in \mathbb{R}^{2\times2}$

Definition 3.7. (Contracted Tensor Product (Cichocki et al., 2009)). The contraction along the last K modes of a tensor $\mathsf{X} \in \mathbb{R}^{I \times J_1 \times \cdots \times J_K}$ and each mode of a tensor $\mathsf{Y} \in \mathbb{R}^{J_1 \times \cdots \times J_K}$ is defined as a tensor of order one, i.e. as a vector

$$\mathbf{z} = \langle \mathsf{X} \,|\, \mathsf{Y} \rangle \in \mathbb{R}^{I} , \quad (40)$$

with components

$$z(i) = \sum_{j_1=1}^{J_1} \cdots \sum_{j_K=1}^{J_K} x(i, j_1, \ldots, j_K)\, y(j_1, \ldots, j_K) . \quad (41)$$

Note that the contracted product can be defined for an arbitrary number and combination of same-sized modes. An example of a tensor contraction is depicted in Fig. 7.

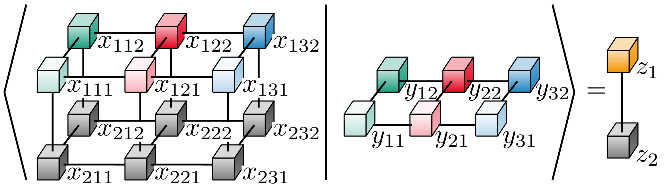

Fig. 7. Contraction of a $3rd$ order tensor $\mathsf{X} \in \mathbb{R}^{2\times3\times2}$ with a tensor of order two $\mathbf{Y} \in \mathbb{R}^{3\times2}$. Summing up the products of the same-colored elements gives the component z_1 of the resulting $1st$ order tensor $\mathbf{z} \in \mathbb{R}^2$

4. SIMULATION IN TENSOR REPRESENTATION

In this section we introduce the *tensor representation* of qualitative models and how they can be used for qualitative simulation. In a first step, the conditional probabilities described by the behavior relation (19), the transition relation (22) and the output relation (23) are set up as the elements of corresponding tensors.

4.1 Behavior, Transition & Output Tensor Representation

Definition 4.1. (Behavior Tensor (Müller-Eping et al., 2017)). The conditional probabilities (19) can be easily stored in a $4th$ order *behavior tensor*

$$\mathsf{L} \in [0, 1]^{N \times Q \times N \times M} , \quad (42)$$

with elements defined by

$$l(\bar{z}, w, z, v) = \mathcal{L}(\bar{z}, w|z, v) \quad \forall\, \bar{z}, z \in \mathcal{Z}, w \in \mathcal{W}, v \in \mathcal{V} . \quad (43)$$

Definition 4.2. (Transition Tensor). The conditional probabilities (22) can be represented a $3rd$ order *transition tensor*

$$\mathsf{F} \in [0, 1]^{N \times N \times M} , \quad (44)$$

with elements given by

$$f(\bar{z}, z, v) = \mathcal{F}(\bar{z} \,|\, z, v) \quad \forall\, \bar{z}, z \in \mathcal{Z}, v \in \mathcal{V} . \quad (45)$$

Definition 4.3. (Output Tensor). The output tensor

$$\mathsf{G} \in [0, 1]^{Q \times N \times M} \quad (46)$$

is of order three, with elements defined by the conditional probabilities (23), such that

$$g(w, z, v) = \mathcal{G}(w \,|\, z, v) \quad \forall\, w \in \mathcal{W}, z \in \mathcal{Z}, v \in \mathcal{V} . \quad (47)$$

4.2 Simulation in Tensor Representation

First, we introduce a vectorial representation of the inputs

$$\mathbf{p}_v(k) \in [0,1]^M \text{ with } p_v(k,v) := \Pr([\mathbf{u}(k)] = v)\ , \quad (48)$$

outputs

$$\mathbf{p}_w(k) \in [0,1]^Q \text{ with } p_w(k,w) := \Pr([\mathbf{y}(k)] = w) \quad (49)$$

and states

$$\mathbf{p}_z(k) \in [0,1]^N \text{ with } p_z(k,z) := \Pr([\mathbf{x}(k)] = z)\ , \quad (50)$$

where $p_v(k,v), v \in \mathcal{U}$, $p_w(k,w), w \in \mathcal{Y}$ and $p_z(k,z), z \in \mathcal{X}$ denote the components of these vectors. With this setting, we introduce the simulation algorithm.

Lemma 4.1. (Simulation in Tensor Representation). For a given input probability distribution $\mathbf{p}_v(k) \in [0,1]^M$, the probabilities of the qualitative states

$$\mathbf{p}_z(k+1) = \Big\langle \mathsf{F} \,\Big|\, \big(\mathbf{p}_z(k) \circ \mathbf{p}_v(k)\big) \Big\rangle \in [0,1]^N \quad (51)$$

and qualitative outputs

$$\mathbf{p}_w(k) = \Big\langle \mathsf{G} \,\Big|\, \big(\mathbf{p}_z(k) \circ \mathbf{p}_v(k)\big) \Big\rangle \in [0,1]^Q \quad (52)$$

are given by tensor contraction.

Proof *4.1.* The element-wise notations of (51) and (52) are given by

$$p_z(k+1,\bar{z}) = \sum_{z=1}^{N} \sum_{v=1}^{M} f(\bar{z},z,v)\, p_z(k,z)\, p_v(k,v)\ , \quad (53)$$

$$p_w(k,w) = \sum_{z=1}^{N} \sum_{v=1}^{M} g(w,z,v)\, p_z(k,z)\, p_v(k,v)\ , \quad (54)$$

what is equal to equation (24) and (25), resp.

Corollary 4.1. For a unique input sequence (12), equation (26) holds. That means, only the $v(k)$-th component of the input vector (48) has the value one, while all other components are zero. For such an unique input sequence, the summation over the index v in the element-wise notations (53) and (54) vanishes and the tensor elements $f(\bar{z},z,v)$ and $g(w,z,v)$ can be directly indexed by this unique input given at time k:

$$p_z(k+1,\bar{z}) = \sum_{z=1}^{N} f(\bar{z},z,v(k))\, p_z(k,z)\ , \quad (55)$$

$$p_w(k,w) = \sum_{z=1}^{N} g(w,z,v(k))\, p_z(k,z)\ . \quad (56)$$

That is, for unique input sequences equation (51) and (52) yield the same result as equation (27) and (28), resp.

Note that storing the conditional probabilities in a tensor as specified in the Definitions 4.1–4.3 and using the tensor-based simulation in Lemma 4.1 alone does not result in any storage reduction, but it paves the way for it.

5. SIMULATION IN CP TENSOR REPRESENTATION

In this section, qualitative models are represented in a so-called *CP tensor*, which already leads to a reduction in the number of values to be stored and, in addition, allows the use of modern mathematical methods for further model reduction. Again, we first need some basic definitions.

Definition 5.1. (CP Tensor Representation (Kolda and Bader, 2009)). The *CP tensor representation* is defined as the expression of a tensor $\mathsf{X} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ as a sum of rank-1 tensors

$$\mathsf{X} = [\![\boldsymbol{\lambda}_{\mathsf{X}}; \mathbf{U}_1, \ldots, \mathbf{U}_N]\!] \quad (57)$$

$$\equiv \sum_{r=1}^{R_{\mathsf{X}}} \lambda_{\mathsf{X}}(r)\, \mathbf{u}_1(:,r) \circ \cdots \circ \mathbf{u}_N(:,r) \in \mathbb{R}^{J_1 \times \cdots \times J_N}\ ,$$

which are given by the outer products of the column vectors $\mathbf{u}_n(:,r) \in \mathbb{R}^{J_n}$ of the factor matrices $\mathbf{U}_n \in \mathbb{R}^{J_n \times R_{\mathsf{X}}}$ for $n = 1,\ldots,N$ and weighted by the elements $\lambda_{\mathsf{X}}(r)$ of a vector $\boldsymbol{\lambda}_{\mathsf{X}} \in \mathbb{R}^{R_{\mathsf{X}}}$. Element-wise notation of (57) gives

$$x(j_1,\ldots,j_N) = \sum_{r=1}^{R_{\mathsf{X}}} \lambda_{\mathsf{X}}(r)\, u_1(j_1,r) \cdots u_N(j_N,r)\ . \quad (58)$$

The term $[\![\cdot]\!]$ appearing in (57) is either called *kruskal tensor* or *CP tensor* (Kolda and Bader, 2009). Fig. 8 depicts a third order CP tensor.

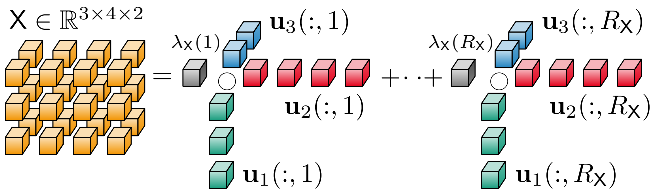

Fig. 8. Third order tensor $\mathsf{X} \in \mathbb{R}^{3\times4\times2}$ constructed by a sum of R_{X} rank one tensors, each of which given by three vectors of corresponding length

The variable R is called the *CP rank* of a tensor and must not be confused with the tensor rank. The rank of a tensor $\mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, denoted by $\mathsf{rank}(\mathsf{X})$, is defined as the smallest number R_{X} of rank one tensors in (57), needed to represent X exactly. While the storage complexity of a Nth order tensor $\mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is given by $\mathcal{O}(I^N)$, for a CP tensor $[\![\boldsymbol{\lambda}; \mathbf{U_1}, \ldots, \mathbf{U_N}]\!] = \mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ it reduces to $\mathcal{O}(NIR_{\mathsf{X}})$, where $I = \mathsf{max}\{I_1,\ldots,I_N\}$ holds for a worst case estimate (Cichocki, 2014). This shows, that it is desirable to find a *direct CP tensor representation* of the behavior tensor (42), the transition tensor (44) and the output tensor (46), without generating their full representations. How this can be realized is explained next.

5.1 Direct CP Representations of the Behavior, Transition and Output Tensor

Definition 5.2. (CP Behavior Tensor). The conditional probabilities (21) for which $\mathcal{L}(\bar{z},w\,|\,z,v) \geq 0$ holds true, can be sorted and defined as components

$$\lambda_{\mathsf{L}}(i) = \mathcal{L}(\bar{z},w\,|\,z,v)_i \text{ with } \lambda_{\mathsf{L}}(1) \geq \cdots \geq \lambda_{\mathsf{L}}(R_{\mathsf{L}}) \quad (59)$$

of the weighting vector $\boldsymbol{\lambda}_{\mathsf{L}} \in [0,1]^{R_{\mathsf{L}}}$ of a CP tensor

$$\mathsf{L} = [\![\boldsymbol{\lambda}_{\mathsf{L}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4]\!] \in [0,1]^{N\times Q\times N\times M} \quad (60)$$

with factor matrices $\mathbf{A}_1 \in \{0,1\}^{N\times R_{\mathsf{L}}}$, $\mathbf{A}_2 \in \{0,1\}^{Q\times R_{\mathsf{L}}}$, $\mathbf{A}_3 \in \{0,1\}^{N\times R_{\mathsf{L}}}$, $\mathbf{A}_4 \in \{0,1\}^{M\times R_{\mathsf{L}}}$ whose column vectors are constructed by the qualitative values $\bar{z}, w, z, v$ in (59) for all $i = 1,2,\ldots,R_{\mathsf{L}}$:

$$\mathbf{a}_1(:,i) = (a_1(1)\cdots a_1(N))^T \text{ with } a_1(j) = \begin{cases} 1 \text{ for } j = \bar{z} \\ 0 \text{ else,} \end{cases}$$

$$\mathbf{a}_2(:,i) = (a_2(1)\cdots a_2(Q))^T \text{ with } a_2(j) = \begin{cases} 1 \text{ for } j = w \\ 0 \text{ else,} \end{cases}$$

$$\mathbf{a}_3(:,i) = (a_3(1)\cdots u_3(N))^T \text{ with } a_3(j) = \begin{cases} 1 \text{ for } j = z \\ 0 \text{ else,} \end{cases}$$

$$\mathbf{a}_4(:,i) = (a_4(1)\cdots a_4(M))^T \text{ with } a_4(j) = \begin{cases} 1 \text{ for } j = v \\ 0 \text{ else.} \end{cases}$$

The storage amount α_{cp} of the CP behavior tensor (60) is given by

$$\alpha_{cp} = R_{\mathsf{L}}(M + 2N + Q + 1)\ . \quad (61)$$

Note, that this amount of values to be stored is lower than in the full tensor representation in (42), as long as

$$R_{\mathsf{L}} < (MN^2Q/M+2N+Q+1) \quad (62)$$

is true. Even if (60) is an exact representation of the behavior tensor (42), it is in general not a *minimal* representation. That means, the rank R_{L} of the CP representation in (60) is usually higher than the rank of the full tensor $\mathsf{rank}(\mathsf{L})$. Finding a minimal and *exact* CP representation with $R_{\mathsf{L}} = \mathsf{rank}(\mathsf{L})$ is a NP-hard problem (Håstad, 1990). In practice, so-called low-rank or best rank-R *approximations*, for which $R_{\mathsf{L}} < \mathsf{rank}(\mathsf{L})$ holds, are of special interest. This is introduced in Section 6.

Based on the CP behavior tensor (60), the CP representation of the transition and output tensor have to be derived for qualitative simulation. Therefore, the k-mode tensor vector product from Definition 3.6 is needed, but has to be modified such that it can be applied to a CP tensor.

Proposition 5.1. (k-Mode CP Tensor Vector Product (Bader and Kolda, 2008)). The k-mode product of a CP tensor

$$\mathsf{X} = [\![\boldsymbol{\lambda}_{\mathsf{X}}; \mathbf{U}_1, \ldots, \mathbf{U}_N]\!] \in \mathbb{R}^{I_1 \times \cdots \times I_N} \tag{63}$$

with factor matrices $\mathbf{U}_n \in \mathbb{R}^{I_n \times R_{\mathsf{X}}}, n = 1, \ldots N$, weighting vector $\boldsymbol{\lambda}_{\mathsf{X}} \in \mathbb{R}^{R_{\mathsf{X}}}$ and a vector $\mathbf{y} \in \mathbb{R}^{I_n}$, is defined as a CP tensor

$$\begin{aligned} \mathsf{Z} &= [\![\boldsymbol{\lambda}_{\mathsf{Z}}; \mathbf{U}_1, \ldots, \mathbf{U}_{n-1}, \mathbf{U}_{n+1}, \ldots, \mathbf{U}_N]\!] , \\ &\text{with } \boldsymbol{\lambda}_{\mathsf{Z}} = (\mathbf{U}_n^T \mathbf{y}) \circledast \boldsymbol{\lambda}_{\mathsf{X}} \in \mathbb{R}^{R_{\mathsf{X}}} , \end{aligned} \tag{64}$$

which is of size $I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N$. The CP rank of the product tensor Z is given by $R_{\mathsf{Z}} = R_{\mathsf{X}}$.

Proof *5.1.* Element-wise notation of Equation (64) and rearranging gives

$$\begin{aligned} z &= \sum_{r=1}^{R_{\mathsf{X}}} \lambda_{\mathsf{Z}}(r)\, u_1(i_1, r) \cdots u_{n-1}(i_{n-1}, r)\, u_{n+1}(i_{n+1}, r) \\ &\quad \cdots u_N(i_N, r) \;\text{ with }\; \lambda_{\mathsf{Z}}(r) = \sum_{i_n=1}^{J_n} u_n(i_n, r)\, y(i_n)\, \lambda_{\mathsf{X}}(r) \\ &= \sum_{i_n=1}^{I_n} \sum_{r=1}^{R_{\mathsf{X}}} \lambda_{\mathsf{X}}(r)\, u_1(i_1, r) \cdots u_{n-1}(i_{n-1}, r)\, u_n(i_n, r) \\ &\quad \cdot u_{n+1}(i_{n+1}, r) \cdots u_N(i_N, r)\, y(i_n) \\ &= \sum_{i_n=1}^{I_n} \underbrace{\sum_{r=1}^{R_{\mathsf{X}}} \lambda_{\mathsf{X}}(r)\, u_1(i_1, r) \cdots u_N(i_N, r)}_{= x(i_1, \ldots, i_N)}\, y(i_n) \\ &= \sum_{i_n=1}^{I_n} x(i_1, \ldots, i_N)\, y(i_n) . \end{aligned} \tag{65}$$

Equation (65) represents exactly the element-wise notation of the k-mode tensor vector product in Definition 3.6. □

Based on Proposition 5.1, the CP transition and output tensor can be defined.

Definition 5.3. (CP Transition Tensor). Based on the factor matrices of the CP behavior tensor (60), the CP transition tensor is given by the use of the k-mode CP tensor vector product in Proposition 5.1:

$$\begin{aligned} \mathsf{F} &= [\![\boldsymbol{\lambda}_{\mathsf{F}}; \mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_4]\!] \in [0,1]^{N \times N \times M} \\ &\text{with } \boldsymbol{\lambda}_{\mathsf{F}} = (\mathbf{A}_2^T\, \mathbf{1}) \circledast \boldsymbol{\lambda}_{\mathsf{L}} \in [0,1]^{R_{\mathsf{L}}} \text{ and } \mathbf{1} \in \{1\}^{R_{\mathsf{L}}} . \end{aligned} \tag{66}$$

Definition 5.4. (CP Output Tensor). Based on the factor matrices of the CP behavior tensor (60), the CP output tensor is given by the use of the k-mode CP tensor vector product from Proposition 5.1

$$\begin{aligned} \mathsf{G} &= [\![\boldsymbol{\lambda}_{\mathsf{G}}; \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4]\!] \in [0,1]^{Q \times N \times M} \\ &\text{with } \boldsymbol{\lambda}_{\mathsf{G}} = (\mathbf{A}_1^T\, \mathbf{1}) \circledast \boldsymbol{\lambda}_{\mathsf{L}} \in [0,1]^{R_{\mathsf{L}}} \text{ and } \mathbf{1} \in \{1\}^{R_{\mathsf{L}}} . \end{aligned} \tag{67}$$

5.2 Simulation in CP Tensor Representation

In this section the qualitative simulation based on the CP representation of the transition (66) and output tensor (67) is introduced. Therefore, we first need to modify the contracted tensor product in Definition 3.7 for making it applicable to CP tensors. In Kruppa (2018), the contraction of two CP tensors is described and proved, but not for the combination of tensor modes needed here. Thus, we build up on this previous work but introduce another definition and proof as follows.

Proposition 5.2. (Contracted CP Tensor Product). The contraction along the last K modes of a CP tensor

$$\mathsf{X} = [\![\boldsymbol{\lambda}_{\mathsf{X}}; \mathbf{S}, \mathbf{U}_1, \ldots, \mathbf{U}_K]\!] \in \mathbb{R}^{I \times J_1 \times \cdots \times J_K}$$

with rank R_{X} and all modes of a CP tensor

$$\mathsf{Y} = [\![\boldsymbol{\lambda}_{\mathsf{Y}}; \mathbf{G}_1, \ldots, \mathbf{G}_K]\!] \in \mathbb{R}^{J_1 \times \cdots \times J_K}$$

with rank R_{Y}, results in a first order CP tensor

$$\begin{aligned} &\Big\langle [\![\boldsymbol{\lambda}_{\mathsf{X}}; \mathbf{S}, \mathbf{U}_1, \ldots, \mathbf{U}_K]\!] \Big| [\![\boldsymbol{\lambda}_{\mathsf{Y}}; \mathbf{G}_1, \ldots, \mathbf{G}_K]\!] \Big\rangle \\ &= [\![\boldsymbol{\lambda}_{\mathbf{z}}; \mathbf{E}]\!] = \mathbf{z} \in \mathbb{R}^I , \end{aligned} \tag{68}$$

where the weighting vector $\boldsymbol{\lambda}_{\mathbf{z}} \in \mathbb{R}^{R_{\mathsf{X}} R_{\mathsf{Y}}}$ and the factor matrix $\mathbf{E} \in \mathbb{R}^{I \times R_{\mathsf{X}} R_{\mathsf{Y}}}$ are defined by

$$\boldsymbol{\lambda}_{\mathbf{z}} = \mathsf{vec}\big((\mathbf{U}_1^T \mathbf{G}_1) \circledast \cdots \circledast (\mathbf{U}_K^T \mathbf{G}_K)\big) \circledast (\boldsymbol{\lambda}_{\mathsf{X}} \otimes \boldsymbol{\lambda}_{\mathsf{Y}}) , \tag{69}$$

$$\mathbf{E} = \mathbf{S} \otimes \mathbf{1}^T, \text{ with } \mathbf{1} \in \{1\}^{R_{\mathsf{Y}}} . \tag{70}$$

Proof *5.2.* The element-wise notation of the first order tensor $\mathbf{z} \in \mathbb{R}^I$ in (68) is given by

$$z(i) = \sum_{d=1}^{R_{\mathsf{X}} R_{\mathsf{Y}}} \lambda_{\mathbf{z}}(d)\, e(i, d) , \quad i \in \{1, \ldots, I\}$$

and can be written equivalently as

$$z(i) = \sum_{h=1}^{R_{\mathsf{X}}} \sum_{t=1}^{R_{\mathsf{Y}}} \lambda_{\mathbf{z}}(\overline{ht})\, e(i, \overline{ht}) , \tag{71}$$

where the multi-index notation $\overline{ht} \in \{1, \ldots, R_{\mathsf{X}} R_{\mathsf{Y}}\}$ for indices $h = 1, \ldots, R_{\mathsf{X}}$ and $t = 1, \ldots, R_{\mathsf{Y}}$ is used. Following this, the element-wise notation of the weighting vector $\boldsymbol{\lambda}_{\mathbf{z}} \in \mathbb{R}^{R_{\mathsf{X}} R_{\mathsf{Y}}}$ in (69) is as follows

$$\begin{aligned} \lambda_{\mathbf{z}}(\overline{ht}) = \lambda_{\mathsf{X}}(h) \lambda_{\mathsf{Y}}(t) \sum_{j_1=1}^{J_1} \cdots \sum_{j_K=1}^{J_K} u_1(j_1, h)\, g_1(j_1, t) \\ \cdots u_K(j_K, h)\, g_K(j_K, t) . \end{aligned} \tag{72}$$

The elements of the factor matrix $\mathbf{E} \in \mathbb{R}^{I \times R_{\mathsf{X}} R_{\mathsf{Y}}}$ in equation (70) are calculated by

$$e(i, \overline{ht}) = s(i, h)\, 1(t) = s(i, h) . \tag{73}$$

Inserting (72) and (73) into (71) and rearranging gives

$$\begin{aligned} z(i) &= \sum_{h=1}^{R_{\mathsf{X}}} \sum_{t=1}^{R_{\mathsf{Y}}} \lambda_{\mathsf{X}}(h)\, \lambda_{\mathsf{Y}}(t) \sum_{j_1=1}^{J_1} \cdots \sum_{j_K=1}^{J_K} u_1(j_1, h)\, g_1(j_1, t) \\ &\quad \cdots u_K(j_K, h)\, g_K(j_K, t)\, s(i, h) \\ &= \sum_{j_1=1}^{J_1} \cdots \sum_{j_K=1}^{J_K} \underbrace{\sum_{h=1}^{R_{\mathsf{X}}} \lambda_{\mathsf{X}}(h)\, s(i, h)\, u_1(j_1, h) \cdots u_K(j_K, h)}_{= x(i, j_1, \ldots, j_K)} \\ &\quad \cdot \underbrace{\sum_{t=1}^{R_{\mathsf{Y}}} \lambda_{\mathsf{Y}}(t)\, g_1(j_1, t) \cdots g_K(j_K, t)}_{= y(j_1, \ldots, j_K)} \\ &= \sum_{j_1=1}^{J_1} \cdots \sum_{j_K=1}^{J_K} x(i, j_1, \ldots, j_K)\, y(j_1, \ldots, j_K) . \end{aligned} \tag{74}$$

Equation (74) represents exactly the element-wise notation of the contracted tensor product in Definition 3.7. □

With Proposition 5.2 the simulation algorithm for qualitative models in CP tensor representation is given.

Lemma 5.1. (Simulation in CP Tensor Representation). For a given input probability distribution $\mathbf{p}_v(k) \in [0,1]^M$, the probabilities of the qualitative states

$$\mathbf{p}_z(k+1) = \Big\langle [\![\boldsymbol{\lambda}_{\mathsf{F}}; \mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_4]\!] \Big| [\![\mathbf{p}_z(k), \mathbf{p}_v(k)]\!] \Big\rangle \tag{75}$$

and qualitative outputs

$$\mathbf{p}_w(k) = \left\langle [\![\boldsymbol{\lambda}_{\mathsf{G}}; \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4]\!] \,\middle|\, [\![\mathbf{p}_z(k), \mathbf{p}_v(k)]\!] \right\rangle \quad (76)$$

can be determined. Because of Proposition 5.2, Lemma 5.1 gives the same result as Lemma 4.1 or Corollary 4.1, if a unique input sequence (12) is given.

6. MODEL REDUCTION AND COMPLETION

In this section, we introduce the mathematical background of the *CP tensor decomposition and completion* and show how these methods can be used for model reduction and to solve the problem of structurally incomplete models. The practical implementation of the following is realized by using the MATLAB® Tensor Toolbox (Bader et al., 2019).

6.1 CP Tensor Decomposition and Completion

The term CP *decomposition* commonly refers to the *factorization* of a tensor $\mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ into a CP tensor

$$\hat{\mathsf{X}} = [\![\boldsymbol{\lambda}_{\hat{\mathsf{X}}}; \mathbf{U}_1, \ldots, \mathbf{U}_N]\!] \in \mathbb{R}^{I_1 \times \cdots \times I_N} \quad (77)$$

with fixed rank $R_{\hat{\mathsf{X}}}$, which is in practice usually an approximation of the original tensor, such that $\hat{\mathsf{X}} \approx \mathsf{X}$ (Kolda and Bader, 2009). In equation (77), $\boldsymbol{\lambda}_{\hat{\mathsf{X}}} \in \mathbb{R}^{R_{\hat{\mathsf{X}}}}$ again denotes the weighting vector and the factor matrices are given by $\mathbf{U}_k \in \mathbb{R}^{I_k \times R_{\hat{\mathsf{X}}}}$, $k = 1, \ldots, N$. Here, the CP rank $R_{\hat{\mathsf{X}}}$ is referred to as *rank of the decomposition.* In the computation of the decomposition (77), $R_{\hat{\mathsf{X}}}$ is an adjustable number that can be used to control the storage amount (61) and the accuracy of the approximation. The higher $R_{\hat{\mathsf{X}}}$ is chosen, the higher the accuracy and the more values have to be stored. *Tensor completion* algorithms are based on tensor decomposition and represent a strong method for recovering missing data (Acar et al., 2011). Decomposing and recovering incomplete data of a tensor $\mathsf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is realized by solving an optimization problem (Vervliet et al., 2014)

$$\min_{\boldsymbol{\lambda}_{\hat{\mathsf{X}}}, \mathbf{U}_1, \ldots, \mathbf{U}_N} \frac{1}{2} \left\| \mathsf{C} \circledast \left(\mathsf{X} - [\![\boldsymbol{\lambda}_{\hat{\mathsf{X}}}; \mathbf{U}_1, \ldots, \mathbf{U}_N]\!] \right) \right\|^2 , \quad (78)$$

where missing values in X are indicated by a same-sized binary tensor $\mathsf{C} \in \{0,1\}^{I_1 \times \cdots \times I_N}$ with elements

$$c(i_1, \ldots, i_N) = \begin{cases} 0 & \text{if } x(i_1, \ldots, i_N) \text{ is missing}, \\ 1 & \text{if } x(i_1, \ldots, i_N) \text{ is present}. \end{cases} \quad (79)$$

Note, that in general the elements of the factor matrices $\mathbf{U}_k \in \mathbb{R}^{I_k \times R_{\hat{\mathsf{X}}}}$ are randomly chosen for the initialization of the optimization. Hence, each decomposition and completion of a tensor may lead to different results, even for the same decomposition rank $R_{\hat{\mathsf{X}}}$.

Also CP tensors can be further reduced and completed. That is, a CP tensor $\mathsf{X} = [\![\boldsymbol{\lambda}_{\mathsf{X}}; \mathbf{F}_1, .., \mathbf{F}_N]\!] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with CP rank R_{X} is approximated by a completed CP tensor $\hat{\mathsf{X}} = [\![\boldsymbol{\lambda}_{\hat{\mathsf{X}}}; \mathbf{U}_1, \ldots, \mathbf{U}_N]\!] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with usually lower CP rank $R_{\hat{\mathsf{X}}} < R_{\mathsf{X}}$, such that

$$\mathsf{X} = [\![\boldsymbol{\lambda}_{\mathsf{X}}; \mathbf{F}_1, \ldots, \mathbf{F}_N]\!] \approx [\![\boldsymbol{\lambda}_{\hat{\mathsf{X}}}; \mathbf{U}_1, \ldots, \mathbf{U}_N]\!] = \hat{\mathsf{X}}. \quad (80)$$

In the MATLAB® Tensor Toolbox, the binary tensor C can be a *sparse* tensor which enables an efficient computation.

6.2 Decomposition and Completion of Qualitative Models

Using tensor completion to overcome the problem of structurally incomplete qualitative models means to set up a binary tensor $\mathsf{C} \in \{0,1\}^{N \times Q \times N \times M}$, preferably in sparse format, indicating the missing probabilities $\mathcal{L}(\bar{z}, w \,|\, z, v)$ in the CP tensor $\mathsf{L} = [\![\boldsymbol{\lambda}_{\mathsf{L}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4]\!] \in [0,1]^{N \times Q \times N \times M}$ in (60). The information about the probabilities which are treated as missing values is given by condition (17). Hence, the binary tensor C can be constructed as follows: For each combination of qualitative states $z \in \mathcal{X}$ and inputs $v \in \mathcal{U}$, check the condition

$$\textstyle\sum_{\bar{z}=1}^{N} \sum_{w=1}^{Q} \mathcal{L}(\bar{z}, w \,|\, z, v) = 0 \quad (81)$$

and construct the binary tensor $\mathsf{C} \in \{0,1\}^{N \times Q \times N \times M}$ with elements

$$c(\bar{z}, w, z, v) = \begin{cases} 0 & \forall \bar{z} \in \mathcal{X}, w \in \mathcal{Y} \text{ if (81) is true}, \\ 1 & \forall \bar{z} \in \mathcal{X}, w \in \mathcal{Y} \text{ if (81) is false}. \end{cases} \quad (82)$$

Now, decomposing and completing the CP behavior tensor of the qualitative model $\mathsf{L} = [\![\boldsymbol{\lambda}_{\mathsf{L}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4]\!]$ in (60) with non-negative constraints on the elements of the factor matrices and an appropriate decomposition rank $R_{\hat{\mathsf{L}}} < R_{\mathsf{L}}$, gives the reduced CP behavior tensor

$$\hat{\mathsf{L}} = [\![\boldsymbol{\lambda}_{\hat{\mathsf{L}}}; \mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4]\!] \in \mathbb{R}_{\geq 0}^{N \times Q \times N \times M}, \quad (83)$$

with $\mathbf{B}_1 \in \mathbb{R}_{\geq 0}^{N \times R_{\mathsf{L}}}$, $\mathbf{B}_2 \in \mathbb{R}_{\geq 0}^{Q \times R_{\mathsf{L}}}$, $\mathbf{B}_3 \in \mathbb{R}_{\geq 0}^{N \times R_{\mathsf{L}}}$, $\mathbf{B}_4 \in \mathbb{R}_{\geq 0}^{M \times R_{\mathsf{L}}}$ and $\boldsymbol{\lambda}_{\hat{\mathsf{L}}} \in \mathbb{R}_{\geq 0}^{R_{\hat{\mathsf{L}}}}$. For qualitative simulation the reduced CP transition and output tensor

$$\hat{\mathsf{F}} = [\![\boldsymbol{\lambda}_{\hat{\mathsf{F}}}; \mathbf{B}_1, \mathbf{B}_3, \mathbf{B}_4]\!] \in \mathbb{R}_{\geq 0}^{N \times N \times M}, \quad (84)$$

$$\hat{\mathsf{G}} = [\![\boldsymbol{\lambda}_{\hat{\mathsf{G}}}; \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4]\!] \in \mathbb{R}_{\geq 0}^{Q \times N \times M}, \quad (85)$$

are derived by the use of equation (66) and (67). Due to the fact, that the CP tensors (83), (84) and (85) are approximations of the exact CP tensors, i.e. their elements lie in $\mathbb{R}_{\geq 0}$, the probability vectors of the states and outputs in Lemma 5.1 have to be normalized to a length of one after each time step k and will also represent an approximation.

7. APPLICATION EXAMPLE

The system investigated is a room of a building heated by a radiator as shown on the left-hand side of Fig. 9.

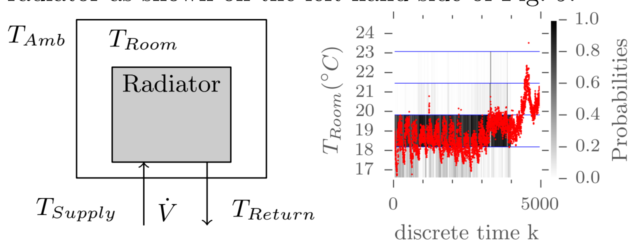

Fig. 9. Room scheme (left) and a section of the qualitative trajectory of the state variable T_{Room} (right)

Model identification: The parameters of the qualitative model are set up as follows: As inputs, the water supply temperature T_{Supply}, the water volume flow of the radiator $\dot{V}$ and the ambient temperature T_{Amb} are used. The states are given by the water return temperature of the radiator T_{Return} and the room temperature T_{Room}. The room temperature T_{Room} is also used as output. This setting corresponds to a typical *consumer model*, where the radiator represents a heat sink which transfers heat to the room (see Pangalos (2016)). The qualitative model of the system is generated via stochastic qualitative identification (Section 2.3) and the use of measurement data containing the months Dec. 2015, Jan. 2016 and July 2016 for covering the heating and non-heating period of the building. Thereby each input is quantized into three intervals, while the two states and the output are separated into five intervals. Because we have three inputs ($m = 3$), two states ($n = 2$) and one output ($q = 1$), this gives $M = 27$ qualitative inputs, $N = 25$ qualitative states and $Q = 5$ qualitative outputs. Using Definition 4.1, this will lead to a full behavior tensor $\mathsf{L} \in [0,1]^{25 \times 5 \times 25 \times 27}$ with $\alpha = 84\,375$ values to be stored. However, the model identification only leads to 333 different combinations of qualitative values $\bar{z}, w, z, v$ for which the conditional probabilities $\mathcal{L}(\bar{z}, w|z, v)$ are non-zero. Storing only these non-zero probabilities and using the direct CP behavior representation given in Definition 5.2 with $R_{\mathsf{L}} = 333$, this gives an exact CP tensor $\mathsf{L} = [\![\boldsymbol{\lambda}_{\mathsf{L}}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4]\!]$ with $\mathbf{A}_1 \in \{0,1\}^{25\times 333}$, $\mathbf{A}_2 \in \{0,1\}^{5\times 333}$, $\mathbf{A}_3 \in \{0,1\}^{25\times 333}$, $\mathbf{A}_4 \in \{0,1\}^{27\times 333}$ and $\boldsymbol{\lambda}_{\mathsf{L}} \in [0,1]^{333}$. Obviously condition (62) is fulfilled, because the storage amount of the exact CP representation is given by $\alpha_{cp} = 27\,639$ values.

Model application: For demonstrating the application of qualitative simulation, we use the measurement data of the input variables of the whole year 2016 to simulate the return T_{Return} and room temperature T_{Room} by the application of Lemma 5.1. The qualitative state trajectory of the room temperature T_{Room} is shown on the right-hand side of Fig. 9. The red dots are the measurements of the room temperature and the different gray shades in the background represent the probabilities of the five qualitative states, where a dark color indicates a high probability. The boundaries of the qualitative intervals are given by the blue lines and one discrete time step k equals 40 minutes. That is, the plot shows a time range of around 5.5 month (Jan. until mid-May 2016). As can be seen, at time $k = 4000$ (mid-March) there are no non-zero probabilities given by the qualitative simulation algorithm. This is because the model is *structurally incomplete* due to the set of training data consisting of only two month of data from the winter period and one month from summer. This problem will now be solved via tensor completion (78).

Model completion: Checking equation (81) turns out that there are 573 combinations of qualitative states z and inputs v for which the condition is true, what leads to a number of $71\,625$ values treated as missing data in the binary tensor C in (82). This corresponds to a percentage share of 85% of unobserved data. For solving the optimization problem (78) a decomposition rank of $R_{\hat{\mathsf{L}}} = 100$ is chosen, what results in a CP tensor (83) consisting of only 8800 values. The simulation results generated by the use of (84), (85) and Lemma 5.1 are shown in Fig. 10. As can be seen, the whole year 2016 could successfully simulated with plausible results, although the qualitative model was significantly incomplete and reduced by a factor of 3.1 in comparison to the exact CP representation.

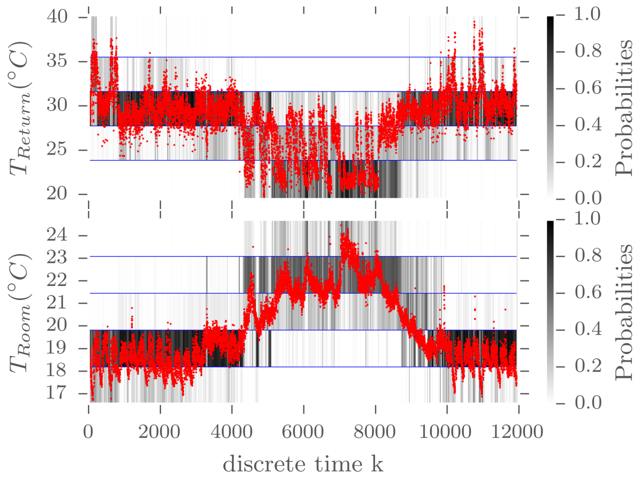

Fig. 10. Qualitative state trajectories T_{Return} and T_{Room}

8. CONCLUSION

As the paper shows, qualitative simulation models can be successfully completed and reduced by modern tensor decomposition algorithms. The provided simulation algorithms for the CP representation of qualitative models enable an efficient calculation of the state and and output probability distributions. This gives hope for the realization of qualitative supervisory control and predictive maintenance applications in large discrete time systems.

REFERENCES

Acar, E., Kolda, T.G., Dunlavy, D.M., and Morup, M. (2011). Scalable Tensor Factorizations for Incomplete Data. *Chemometrics and Intelligent Laboratory Systems*, 106(1), 41–56.

Bader, B.W. and Kolda, T.G. (2008). Efficient MATLAB Computations with Sparse and Factored Tensors. *SIAM Journal on Scientific Computing*, 30(1), 205–231.

Bader, B.W., Kolda, T.G., et al. (2019). Matlab tensor toolbox version 3.1. Available online. URL `https://www.tensortoolbox.org`.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki (2006). *Diagnosis and Fault-Tolerant Control*. Springer.

Cichocki, A. (2014). Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions. *arXiv:1403.2048 [cs]*.

Cichocki, A., Zdunek, R., Phan, H.A., and Amari, S.I. (2009). *NonnegativeMatrix and Tensor Factorizations*. John Wiley & Sons, Ltd.

Håstad, J. (1990). Tensor rank is NP-complete. *Journal of Algorithms*, 11(4), 644–654.

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

Kruppa, K. (2018). *Multilinear Design of Decentralized Controller Networks for Building Automation Systems*. Ph.D. thesis, HafenCity Universität Hamburg.

Lee, N. and Cichocki, A. (2018). Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, 29(3), 921–960.

Lichtenberg, G. (1998). *Theorie und Anwendung der qualitativen Modellierung zeitdiskreter dynamischer Systeme durch nichtdeterministische Automaten*. VDI Verlag.

Lichtenberg, G. and Lunze, J. (1997). Observation of qualitative states by means of a qualitative model. *International Journal of Control*, 66(6), 885–903.

Lunze, J. (1998). Qualitative Modellierung dynamischer Systeme durch stochastische Automaten. *at - Automatisierungstechnik*, 46(6), 271–283.

Müller, T., Kruppa, K., Lichtenberg, G., and Réhault, N. (2015). Fault Detection with Qualitative Models reduced by Tensor Decomposition methods. *IFAC-PapersOnLine*, 48(21), 416–421.

Müller-Eping, T., Lichtenberg, G., and Vogelmann, V. (2017). Fault Detection Algorithms based on Decomposed Tensor Representations for Qualitative Models. *IFAC-PapersOnLine*, 50(1), 5622–5629.

Pangalos, G. (2016). *Model-Based Controller Design Methods for Heating Systems*. Ph.D. thesis, Technische Universität Hamburg-Harburg.

Schröder, J. (2003). *Modelling, State Observation and Diagnosis of Quantised Systems*. Springer.

Vervliet, N., Debals, O., Sorber, L., and Lathauwer, L.D. (2014). Breaking the Curse of Dimensionality Using Decompositions of Incomplete Tensors. *IEEE Signal Processing Magazine*, 31(5), 71–79.