Reduced CP Representation of Multilinear Models

Niklas Jöres[1][a], Christoph Kaufmann[1,2,4][b], Leona Schnelle[2][c], Carlos Cateriano Yáñez[1,2,3][d], Georg Pangalos[1][e] and Gerwald Lichtenberg[2][f]

[1]*Application Center for Integration of Local Energy Systems, Fraunhofer IWES, Hamburg, Germany*

[2]*Faculty of Life Science, HAW Hamburg, Germany*

[3]*Universitat Politècnica de València, Instituto Universitario de Automática e Informática Industrial, València, Spain*

[4]*Centre d'Innovació Tecnològica en Convertidors Estàtics i Accionaments (CITCEA),*
Departament d'Enginyeria Elèctrica, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

Keywords: Multilinear Systems, Tensor Decomposition.

Abstract: Large and highly complex systems can be found in various application areas. Modeling these systems requires appropriate representation of the underlying phenomena. Furthermore, due to the large dimensions efficient simulation and low memory requirements are needed for such models. Multilinear modeling is a promising approach to address these challenges. In this paper, we introduce a reduced canonical polyadic (CP) representation for implicit time-invariant multilinear (iMTI) models. This representation is capable of storing large models with very low memory requirements. This is particularly useful for efficient analyses of large systems with numerous inputs and states.

1 INTRODUCTION

Modeling and simulation of complex systems is an active field of research. Currently, e.g., in the field of modeling energy systems co-simulation methodology is used as one approach to address the high complexity, while maintaining a realistic representation (López et al., 2019; Farrokhseresht et al., 2021; Wiens et al., 2021; Vogt et al., 2018). However, modeling of such large and complex systems while capturing the relevant dynamics results in large computational resources and simulation times with the existing modeling approaches. Therefore, more computational power, more efficient algorithms and new modeling strategies are required (F. Milano et al., 2018).

Focusing on modeling strategies, a possibility is to rethink the fundamental question: Which class of models has the potential to cover all relevant nonlinear dynamics and at the same time enables efficient simulations as well as analysis and design algorithms? For some large scale complex application domains with similar modeling problems, recent research shows that multilinearity and tensor decomposition methods could lead to breakthroughs (Verstraete et al., 2008).

In recent years the multilinear modeling framework have been introduced first in an explicit form by (Pangalos et al., 2013) and then in the more general implicit form (Lichtenberg et al., 2022). The advantage of multilinear models is, that some nonlinear phenomena can be modeled while still maintaining an efficient and structured representation. Application examples range from heating systems (Pangalos et al., 2013) over chemical reactions (Kruppa et al., 2014) to energy systems (Lichtenberg et al., 2022). In addition, efficient simulation is possible, when using decompositions. However, the multilinear model is still an approximation and therefore, not as exact as the nonlinear model. In addition, the tools for multilinear modeling are not yet standard and further development is required.

Regarding controller synthesis, approaches to deal with multilinear models in the application domain of heating systems are given in (Pangalos, 2016; Kruppa, 2018). The heating sector has the advantage

[a] https://orcid.org/0000-0003-2471-3892
[b] https://orcid.org/0000-0002-0666-1104
[c] https://orcid.org/0000-0002-2600-8110
[d] https://orcid.org/0000-0001-5261-2568
[e] https://orcid.org/0000-0001-5094-8033
[f] https://orcid.org/0000-0001-6032-0733

Jöres, N., Kaufmann, C., Schnelle, L., Yáñez, C., Pangalos, G. and Lichtenberg, G.
Reduced CP Representation of Multilinear Models.
DOI: 10.5220/0011273100003274
In *Proceedings of the 12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2022)*, pages 252-259
ISBN: 978-989-758-578-4; ISSN: 2184-2841

of relatively slow processes, which corresponds to large time constants. This is in contrast to the electrical energy system, where the need to efficient simulation is therefore even bigger. In this contribution, we focus on improving the performance and memory requirements of multilinear time-invariant (MTI) models - as a prerequisite to meet real-time constraints for enabling model-based control of large scale systems.

First an introduction into multilinear functions is given. This is followed by the explanation of explicit multilinear time-invariant (eMTI) models, which includes decomposition, normalization and linearization of eMTI models. The extension of eMTI is then given with implicit multilinear time-invariant (iMTI) models. The last chapter shows how an iMTI model can be represented with very low memory requirements. A conclusion and further research is stated in the end.

2 MULTILINEAR FUNCTIONS

In this section different formats for multilinear functions are discussed as basis for the following modeling concepts. A multilinear function can be described by the factored polynomial

$$f(\mathbf{x})=\sum_{k=1}^{r}\prod_{i=1}^{n}(f_{1_{i,k}}+f_{x_{i,k}}x_i), \tag{1}$$

with a total number of $2rn$ parameters, where $f_{1_{i,k}}\in\mathbb{R}$ and $f_{x_{i,k}}\in\mathbb{R}\backslash\{0\}$ for all $i=1,2...n$ and $k=1,2...r$, and x_i are the variables of the function and k describes the number of the addend.

Remark: The example

$$(4+2x_1)(3+x_2)=(2+x_1)(6+2x_2),$$

shows, that the representation (1) is not unique.

Each factor of the polynomial (1) can be visualized in a 2-D Cartesian coordinate system, where the two parameters f_1 and f_{x_i} create a vector with one component in direction of the 1-axis and one component in direction of the x_i-axis as shown in Figure 1 with an example vector $(f_1\ f_x)^T$.

Next, we will use normalization to get unique representations, as the following example does for different norms,

$$1\left(1+\frac{1}{2}x\right)=\frac{1}{2}(2+x)=2\left(\frac{1}{2}+\frac{1}{4}x\right).$$

If we normalize this vector with the euclidean 2-norm condition to the length of one, the new vector points to the intersection S_2 of the original vector and the unit circle in Figure 1. The intercepts of the 2-norm normalized vector are then given by $\sin\alpha$ and

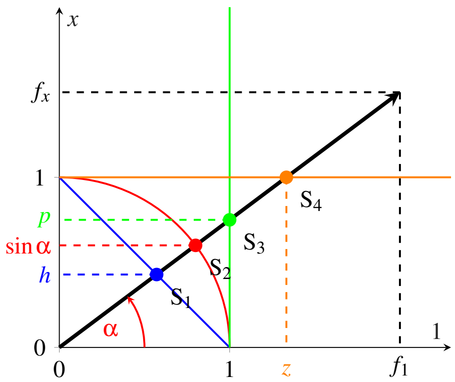


Figure 1: Normalized factor representation.

$\cos\alpha$ and the 2-norm normalized polynomial

$$f(\mathbf{x})=\sum_{k=1}^{r}\lambda_{\alpha,k}\left(\prod_{i=1}^{n}(\cos\alpha_{i,k}+\sin\alpha_{i,k}x_i)\right), \tag{2}$$

has with

$$\alpha_{i,k}=\operatorname{atan2}\left(f_{x_{i,k}},f_{1i,k}\right), \tag{3}$$

$$\lambda_{\alpha,k}=\prod_{i=1}^{n}\sqrt{f_{1i,k}^2+f_{x_{i,k}}^2}, \tag{4}$$

a much smaller number $r(n+1)$ of parameters as (1). The atan2 describes the four-quadrant inverse tangent which gives the angle between the first and the second argument in radians.

Similarly, the polynomial can be transformed in a 1-norm representation as shown by the intersection S_1 with the blue line as

$$f(\mathbf{x})=\sum_{k=1}^{r}\lambda_{h,k}\left(\prod_{i=1}^{n}(1-h_{i,k}+h_{i,k}x_i)\right), \tag{5}$$

where the new parameters are

$$h_{i,k}=\frac{f_{x_{i,k}}}{f_{1_{i,k}}+f_{x_{i,k}}}\quad\text{and}\quad\lambda_{h,k}=\prod_{i=1}^{n}\left(f_{1_{i,k}}+f_{x_{i,k}}\right). \tag{6}$$

By fixing the x-axis coordinate to one, the intersection S_4 lead to a polynomial

$$f(\mathbf{x})=\sum_{k=1}^{r}\lambda_{z,k}\left(\prod_{i=1}^{n}(z_{i,k}+x_i)\right), \tag{7}$$

with the parameters

$$z_{i,k}=\frac{f_{1i,k}}{f_{x_{i,k}}}\quad\text{and}\quad\lambda_{z,k}=\prod_{i=1}^{n}f_{x_{i,k}}, \tag{8}$$

and the zeros $x_{i,k}=-z_{i,k}$ of each factor of the polynomial.

The last intersection point S_3 with the green line in Figure 1 fixes the 1-axis coordinate to one, leading to a representation

$$f(\mathbf{x})=\sum_{k=1}^{r}\lambda_k\left(\prod_{i=1}^{n}(1+p_{i,k}x_i)\right), \tag{9}$$

called 'sparse' in the following and having the parameters

$$p_{i,k}{=}\frac{f_{x_{i,k}}}{f_{1_{i,k}}} \quad \text{and} \quad \lambda_k = \prod_{i=1}^{n} f_{1_{i,k}} f_{x_{j,k}}, \tag{10}$$

for all $f_{1_{j,k}} \neq 0$. In case of $f_{1_{j,k}} = 0$, the factor vector is pointing on the x_j-axis and the parameter $p_{j,k} \to \infty$. Like in time-constant formulation of transfer functions, the polynomial can then be represented by

$$f(\mathbf{x}) = \sum_{k=1}^{r} \lambda_k \prod_{j} x_j \prod_{i} (1 + p_{i,k} x_i) \ , \tag{11}$$

and the factors $\lambda_k = \prod_i f_{1_{i,k}} f_{x_{i,k}} \prod_j f_{x_{j,k}}$ adjusted.

The sparse representation offers advantages for modeling large systems and is mainly used in the next sections in this paper, which are outlined by Figure 2 and linked to the related examples and equations in the sequel.

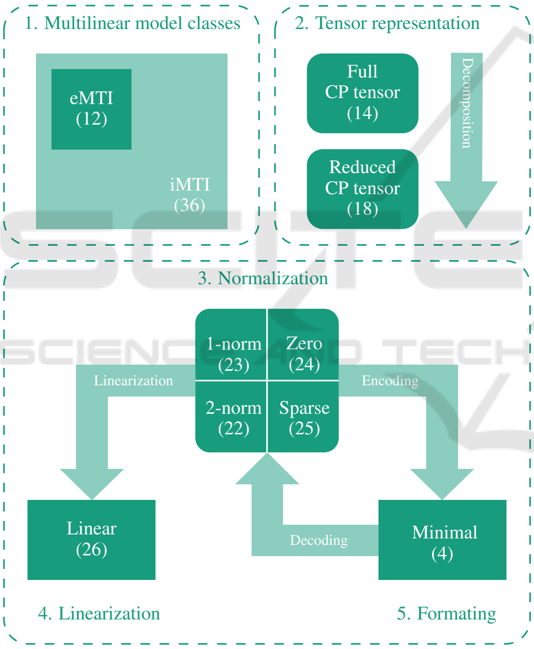


Figure 2: Multilinear modeling steps.

3 EXPLICIT MULTILINEAR MODELS

An eMTI state-space model in continuous time can be expressed by using the contracted tensor product, (Pangalos, 2016)

$$\dot{\mathbf{x}} = \langle \mathsf{F} \,|\, \mathsf{M}(\mathbf{x}, \mathbf{u}) \, \rangle \ , \tag{12}$$

with the parameter tensor $\mathsf{F} \in \mathbb{R}^{\overbrace{2 \times \cdots \times 2}^{n+m} \times n}$, where n declares the number of states, and m is the number of inputs. For brevity, we consider no extra output equations, but the general approach could be easily extended to this case if needed (Pangalos, 2016). The next example will be a running one throughout this paper.

Example 3.1. *Consider a second-order eMTI model with two states x_1, x_2 and one input u*

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0.4u + 0.2x_2 + 0.08ux_1 + 0.06x_1x_2 \\ 0.6 + 0.24x_1 + 0.94x_2 + 0.296x_1x_2 \end{pmatrix} . \tag{13}$$

The full parameter tensor $\mathsf{F} \in \mathbb{R}^{2\times 2\times 2\times 2}$ has the following nonzero elements

$$\begin{array}{rrrr} F(1,1,2,1) = & 0.4, & F(1,1,1,2) = & 0.6, \\ F(1,2,1,1) = & 0.2, & F(2,1,1,2) = & 0.24, \\ F(2,1,2,1) = & 0.08, & F(1,2,1,2) = & 0.94, \\ F(2,2,1,1) = & 0.06, & F(2,2,1,2) = & 0.296, \end{array} \tag{14}$$

where the indices are ordered from x_1, x_2 over u to ϕ, which specifies the corresponding row in (13).

3.1 Decomposition

Because of the similarity with the factorization discussed in Section 2, we focus on canonical polyadic (CP)-decompostions of the parameter tensor F as one of the methods, e.g., given in (Kolda and Bader, 2009). The decomposed tensor of rank r is given by factor matrices $\mathbf{F}_i \in \mathbb{R}^{2\times r}$ for all states and inputs with the common index $i = 1, \ldots n+m$ together with the matrix $\mathbf{F}_\phi \in \mathbb{R}^{n\times r}$ distributing factors over state derivatives and jointly represented by

$$\mathsf{F} = \left[\mathbf{F}_{x_1}, \ldots, \mathbf{F}_{x_n}, \mathbf{F}_{u_1}, \ldots, \mathbf{F}_{u_m}, \mathbf{F}_\phi\right] . \tag{15a}$$

As the monomial tensor has a rank-1 decomposition

$$\mathsf{M}(\mathbf{x}, \mathbf{u}) = \left[\begin{pmatrix} 1 \\ x_1 \end{pmatrix}, \cdots, \begin{pmatrix} 1 \\ x_n \end{pmatrix}, \begin{pmatrix} 1 \\ u_1 \end{pmatrix}, \cdots, \begin{pmatrix} 1 \\ u_m \end{pmatrix} \right], \tag{15b}$$

the model (12) can also be represented by the CP-factors, (Pangalos, 2016),

$$\begin{aligned} \dot{\mathbf{x}} = \mathbf{F}_\phi \Bigg(\left(\mathbf{F}_{x_1}^{\mathrm{T}} \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \right) \circledast \cdots \circledast \left(\mathbf{F}_{x_n}^{\mathrm{T}} \begin{pmatrix} 1 \\ x_n \end{pmatrix} \right) \circledast \\ \circledast \left(\mathbf{F}_{u_1}^{\mathrm{T}} \begin{pmatrix} 1 \\ u_1 \end{pmatrix} \right) \circledast \cdots \circledast \left(\mathbf{F}_{u_m}^{\mathrm{T}} \begin{pmatrix} 1 \\ u_m \end{pmatrix} \right) \Bigg), \end{aligned} \tag{16}$$

where $\circledast$ stands for the Hadamard (element-wise) product. Any internal structure of the model can be exploited to find a more compact CP representation as the full tensor, which is motivated by the running example next.

Example 3.2. *The full representation can be rewritten as CP model of rank 8 directly from the 8 addends in* (13)*. Due to the duplication of variable combinations x_2 and x_1x_2, the model can be reduced to rank 6 with factor matrices*

$$\mathbf{F}_{x_1} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{17a}$$

$$\mathbf{F}_{x_2} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}, \tag{17b}$$

$$\mathbf{F}_{u} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{17c}$$

$$\mathbf{F}_{\phi} = \begin{pmatrix} 0.4 & 0.2 & 0.08 & 0.06 & 0 & 0 \\ 0 & 0.94 & 0 & 0.296 & 0.6 & 0.24 \end{pmatrix}. \tag{17d}$$

The question of whether the parameter tensor F could be further reduced, i.e. represented by a lower number of factors is related to the non-trivial problem of tensor rank determination (Kolda and Bader, 2009). For the running example, the answer is yes.

Example 3.3. *The tensor F of the running example is representable as rank 4 shown by the factor matrices*

$$\mathbf{F}_{x_1} = \begin{pmatrix} 0.4 & 0.2 & 0.6 & 0.4 \\ 0.08 & 0.06 & 0.24 & 0.08 \end{pmatrix}, \tag{18a}$$

$$\mathbf{F}_{x_2} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0.9 & 1 \end{pmatrix}, \tag{18b}$$

$$\mathbf{F}_{u} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{18c}$$

$$\mathbf{F}_{\phi} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{18d}$$

Tensor decomposition algorithms find best rank-r factorizations of the original tensor by optimization (Kolda and Bader, 2009), which can be used for large models to compute numerical approximation of predefined sizes. Additionally, the factor matrices can be normalized for further reduction, which is discussed next.

3.2 Normalization

The representation of the multilinear model by its CP factors is not unique. To overcome this, normalization as introduced in Section 2 is used for the next definition.

Definition 3.1. *A CP decomposed eMTI model*

$$\dot{\mathbf{x}} = \left\langle [\mathbf{F}_1, \dots, \mathbf{F}_{n+m}, \mathbf{F}_\phi] \,|\, \mathsf{M}(\mathbf{x}, \mathbf{u} \right\rangle, \tag{19}$$

is called l-normalized if all $k = 1, \dots, r$ columns

$$||\mathbf{F}_i(:,k)||_l = 1 \quad , \tag{20}$$

of its factor matrices $\mathbf{F}_i$ with $i = 1, \dots, (n+m)$, i.e. except the last factor $\mathbf{F}_\phi$ have an l-norm of one.

Remark: For Absolute-value and Euclidean norm, (20) is

$$\|\mathbf{F}_i(:,k)\|_1 = \sum_{j=1}^{2} |\mathbf{F}_i(j,k)| = |\mathbf{F}_i(1,k)| + |\mathbf{F}_i(2,k)| = 1,$$

$$\|\mathbf{F}_i(:,k)\|_2 = \sqrt{\sum_{j=1}^{2} \mathbf{F}_i(j,k)^2} = \sqrt{\mathbf{F}_i(1,k)^2 + \mathbf{F}_i(2,k)^2} = 1,$$

and in Figure 1 an example visual summary is provided, which considers the intersection points S_1 and S_2 for all $i = 1, \dots, n+m$ and $k = 1, \dots, r$, whereas the last factor matrix $\mathbf{F}_\phi \in \mathbb{R}^{n \times r}$ holds the scaling factors λ_k.

It follows from (16), that the right side of a CP-decomposed eMTI model (12) contains polynomials like (1). Thus, the computation of the state derivatives of an eMTI model is also possible with normalized factored polynomials (2) to (7) in 1-norm, 2-norm, sparse or zero representation, which can be derived from the corresponding normalized parameter tensors of (19).

Moreover, each factor matrix $\mathbf{F}_i \in \mathbb{R}^{2 \times r}$ of the parameter tensor F of an l-normalized eMTI model (except the last $\mathbf{F}_\phi$) can be represented by a single parameter vector $\widetilde{\mathbf{F}}_i \in \mathbb{R}^r$, from which both elements can be reconstructed by the norm condition (20). The non-normalized last factor matrix $\mathbf{F}_\phi$ will contain the parameters λ_k of the corresponding form, which can be interpreted as the 'lengths' of the vectors in this norm.

Example 3.4. *The example* (18) *is rank 4 and has a CP parameter tensor from* (15a)*,*

$$\mathsf{F} = \left[\mathbf{F}_{x_1}, \mathbf{F}_{x_2}, \mathbf{F}_u, \mathbf{F}_\phi\right] = \left[\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_\phi\right], \tag{21}$$

where we use sequential indexing for simplification. Normalizing to 2-norm by applying (2) *leads to parameter vectors containing angles $\alpha_{i,k}$ for $i = 1, 2, 3$ and $k = 1, \dots, 4$:*

$$\widetilde{\mathbf{F}}_1^{\alpha} = \begin{pmatrix} 0.2 & 0.29 & 0.38 & 0.2 \end{pmatrix}, \tag{22a}$$

$$\widetilde{\mathbf{F}}_2^{\alpha} = \begin{pmatrix} 0 & \pi/2 & 0.73 & \pi/2 \end{pmatrix}, \tag{22b}$$

$$\widetilde{\mathbf{F}}_3^{\alpha} = \begin{pmatrix} \pi/2 & 0 & 0 & 0 \end{pmatrix}, \tag{22c}$$

$$\widetilde{\mathbf{F}}_\phi^{\alpha} = \begin{pmatrix} 0.41 & 0.21 & 0 & 0 \\ 0 & 0 & 0.87 & 0.41 \end{pmatrix}. \tag{22d}$$

The same model can also be represented in 1-norm (5) *as*

$$\widetilde{\mathbf{F}}_1^{h} = \begin{pmatrix} 0.167 & 0.23 & 0.29 & 0.167 \end{pmatrix}, \tag{23a}$$

$$\widetilde{\mathbf{F}}_2^{h} = \begin{pmatrix} 0 & 1 & 0.47 & 1 \end{pmatrix}, \tag{23b}$$

$$\widetilde{\mathbf{F}}_3^{h} = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \tag{23c}$$

$$\widetilde{\mathbf{F}}_\phi^{h} = \begin{pmatrix} 0.48 & 0.26 & 0 & 0 \\ 0 & 0 & 1.6 & 0.48 \end{pmatrix}. \tag{23d}$$

The parameter vector build from (7) *holds the zeros*

$$\widetilde{\mathbf{F}}_1^z = \begin{pmatrix} 5 & 3.33 & 2.5 & 5 \end{pmatrix}, \tag{24a}$$

$$\widetilde{\mathbf{F}}_2^z = \begin{pmatrix} \infty & 0 & 1.11 & 0 \end{pmatrix}, \tag{24b}$$

$$\widetilde{\mathbf{F}}_3^z = \begin{pmatrix} 0 & \infty & \infty & \infty \end{pmatrix}, \tag{24c}$$

$$\widetilde{\mathbf{F}}_\phi^z = \begin{pmatrix} 0.08 & 0.06 & 0 & 0 \\ 0 & 0 & 0.22 & 0.08 \end{pmatrix}, \tag{24d}$$

and finally the 'sparse' form (9) *results in*

$$\widetilde{\mathbf{F}}_1 = \begin{pmatrix} 0.2 & 0.3 & 0.4 & 0.2 \end{pmatrix}, \tag{25a}$$

$$\widetilde{\mathbf{F}}_2 = \begin{pmatrix} 0 & \infty & 0.9 & \infty \end{pmatrix}, \tag{25b}$$

$$\widetilde{\mathbf{F}}_3 = \begin{pmatrix} \infty & 0 & 0 & 0 \end{pmatrix}, \tag{25c}$$

$$\widetilde{\mathbf{F}}_\phi = \begin{pmatrix} 0.4 & 0.2 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 \end{pmatrix}, \tag{25d}$$

with a slight abuse of notation.

3.3 Linearization

Linearizing eMTI models is useful to access, e.g., the large library of linear stability analysis tools, such as, eigenvalue analysis, or determining the participation factor in modal analysis. Therefore, an eMTI model needs to be approximated around an operating point $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ to achieve a linear time-invariant state-space model

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{B}(\mathbf{u} - \bar{\mathbf{u}}), \tag{26}$$

with $\mathbf{A} \in \mathbb{R}^{n \times n}$ as system matrix, and $\mathbf{B} \in \mathbb{R}^{n \times m}$ as input matrix. For simplicity, the output equations are not shown. To obtain a linear approximation of (12), it is well known that partial derivatives have to be calculated w.r.t. each state and input.

For eMTI models, the matrices of the linearized state-space model (26) can be represented as contracted tensor products column-wise as given in the following, (Kruppa, 2018),

$$\mathbf{A}(:, j) = \left\langle \mathsf{F}_{x_j} \,|\, \mathsf{M}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \right\rangle, \tag{27}$$

$$\mathbf{B}(:, j) = \left\langle \mathsf{F}_{u_j} \,|\, \mathsf{M}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \right\rangle, \tag{28}$$

where the partial derivative tensor F_{x_j} of the tensor F over a variable x_j is calculated by the j-mode tensor matrix product

$$\mathsf{F}_{x_j} = \mathsf{F} \times_j \mathbf{\Theta}, \tag{29}$$

with the operator matrix $\mathbf{\Theta} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, see (Kruppa and Lichtenberg, 2018).

These differentiation operations could also be directly done in CP decomposed tensor form leading to

$$\mathsf{F}_{x_j} = \left[\bar{\mathbf{F}}_1, \ldots, \bar{\mathbf{F}}_{n+m}\right]. \tag{30}$$

To compute this CP representation, all factor matrices remain the same, except $\mathbf{F}_j$, i.e.

$$\bar{\mathbf{F}}_i = \begin{cases} \mathbf{\Theta}\mathbf{F}_i & \text{if } i = j \\ \mathbf{F}_i & \text{otherwise} \end{cases}, \tag{31}$$

the factor matrix $\mathbf{F}_{x_j}$ corresponding to the state x_j, which needs to be multiplied with $\mathbf{\Theta}$. Similar equations hold for the partial derivative w.r.t. the inputs, for details see (Kruppa and Lichtenberg, 2018).

Example 3.5. *The procedure is demonstrated for the partial derivative of* F *over* x_2*. Using, e.g., the 1-norm representation as in* (23b)*, first the CP form can be constructed, and then the new factor matrix*

$$\bar{\mathbf{F}}_2 = \mathsf{F} \times_2 \mathbf{\Theta} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0.53 & 0 \\ 0 & 1 & 0.47 & 1 \end{pmatrix} \tag{32a}$$

$$= \begin{pmatrix} 0 & 1 & 0.47 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{32b}$$

for the CP decomposed partial derivative tensor

$$\mathsf{F}_{x_2} = \left[\mathbf{F}_{x_1}, \bar{\mathbf{F}}_2, \mathbf{F}_u, \mathbf{F}_\Phi\right]. \tag{32c}$$

Remark: The resulting tensor F_{x_2} can be further reduced to rank 3: because the first column of $\bar{\mathbf{F}}_2$ only contains zeros, the first columns of all factors will be multiplied by zero and thus, can be removed.

Considering (32b), it is evident that partial derivative of (23b) could be obtained more efficiently without the matrix multiplication by $\mathbf{\Theta}$.

This leads to an elegant way to derive the linearized model directly in tensor form extending the description already shown in (27) to (Kruppa, 2018)

$$\mathbf{A}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \langle \mathsf{A} \,|\, \mathsf{M}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \rangle \in \mathbb{R}^{n \times n}, \tag{33}$$

$$\mathbf{B}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \langle \mathsf{B} \,|\, \mathsf{M}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \rangle \in \mathbb{R}^{n \times m}, \tag{34}$$

with tensors $\mathsf{A} \in \mathbb{R}^{\overbrace{2 \times \ldots \times 2}^{n+m} \times n \times n}$ and $\mathsf{B} \in \mathbb{R}^{\overbrace{2 \times \ldots \times 2}^{n+m} \times n \times m}$. Therefore, the linearization is done by a simple evaluation of the contracted product. The dimensions of the tensor B follow the same structure as A. The first $\overbrace{2 \times \ldots \times 2}^{n+m}$-dimension matches the dimension of the transition tensor F. The last two dimensions are the same dimensions as of the matrices of the linear state-space model. Therefore, the column entries of the Jacobian equivalent to the columns of the system matrix are now the fibers of the tensor, e.g., for A (Kruppa, 2018)

$$\mathbf{a}(\mathbf{i}_u, \mathbf{i}_x, :, j) = \mathbf{j}_{\mathbf{f}}(\mathbf{i}_u, \mathbf{i}_x, :, j) = \mathbf{f}_{x_j}(\mathbf{i}_u, \mathbf{i}_x, :), \tag{35}$$

where $j = 1, \ldots, n$, and the matching dimensions of the transition tensor are indicated by the index vector $\mathbf{i}_x \in \mathbb{R}^n$, and $\mathbf{i}_x \in \mathbb{R}^m$ (Kruppa, 2018) and the remaining fibers $\mathbf{b}$ follow a similar structure.

Example 3.6. *Continuing with the linearization using* (32c)*, the second columns of the system matrix is computed using* (27)

$$\mathbf{A}(:,2) = \langle \mathsf{F}_{x_2} \,|\, \mathsf{M}(\bar{x}_1,\bar{x}_2,\bar{u}) \,\rangle, \\ = \begin{pmatrix} 0.2\bar{u} + 0.06\bar{x}_2 \\ 0.94\bar{x}_2 + 0.296 \end{pmatrix},$$

which is also done for the first columns of **A***, and* **B***. This finally results in the matrices*

$$\mathbf{A} = \begin{pmatrix} 0.08\bar{u} + 0.06\bar{x}_2 & 0.2 + 0.06\bar{x}_1 \\ 0.296\bar{x}_2 + 0.24 & 0.94 + 0.296\bar{x}_1 \end{pmatrix}, \\ \mathbf{B} = \begin{pmatrix} 0.4 + 0.08\bar{x}_1 \\ 0 \end{pmatrix},$$

of the linearized model (13).

Remark: The reader could easily verify all steps of the running example also from its sparse representation (25)

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0.2(1{+}0.3x_1)x_2{+}0.4(1{+}0.2x_1)u \\ 0.6(1{+}0.4x_1)(1{+}0.9x_2) + 0.4(1{+}0.2x_1)x_2 \end{pmatrix}.$$

The next section extends the representation to so-called implicit multilinear models (iMTI), which recently have been shown to be advantageous over eMTI models because of their closedness w.r.t to standard compositions.

4 IMPLICIT MULTILINEAR MODELS

The development of the equations for iMTI models can be performed in analogy to Section 3. An iMTI model represented in CP form without outputs reads

$$\langle \mathsf{H} \,|\, \mathsf{M}(\dot{\mathbf{x}},\mathbf{x},\mathbf{u}) \,\rangle = \mathbf{0}\,, \tag{36}$$

with parameter tensor $\mathsf{H} \in \mathbb{R}^{\overbrace{2\times\ldots\times 2}^{2n+m}\times e}$, where e is the number of equations defining the iMTI model and M is constructed analogously to (15b).

Remark: Be aware that (36) in general is a system of differential-algebraic equations (DAEs) which demand other methods and tools than ODEs.

To include thresholds and Boolean expressions the inequality constraint

$$\langle \mathsf{L} \,|\, \mathsf{M}(\dot{\mathbf{x}},\mathbf{x},\mathbf{u}) \,\rangle \geq \mathbf{0}\,, \tag{37}$$

with $\mathsf{L} \in \mathbb{R}^{\overbrace{2\times\ldots\times 2}^{2n+m+p}\times N}$, where N is the number of inequalities is introduced in (Lichtenberg et al., 2022). To represent an iMTI model, all parameters can be gathered in the tensor L, which are very large for, e.g., energy system models. But the reduced CP form

$$\mathsf{L} = \left[\mathbf{L}_{\dot{x}_1},\ldots,\mathbf{L}_{\dot{x}_n},\mathbf{L}_{x_1},\ldots,\mathbf{L}_{x_n},\mathbf{L}_{u_1},\ldots,\mathbf{L}_{u_m},\mathbf{L}_{\phi}\right], \tag{38}$$

might have - depending on the system structure - a comparable low rank leading to a much smaller number of parameters in its factor matrices, especially when given in a normalized form. Analogous to (15a), the model can be given by

$$\mathbf{L}_{\phi}\left(\mathbf{L}_{\dot{x}_1}^{\mathrm{T}}\begin{pmatrix}1\\ \dot{x}_1\end{pmatrix}\right) \circledast \cdots \circledast \left(\mathbf{L}_{u_m}^{\mathrm{T}}\begin{pmatrix}1\\ u_m\end{pmatrix}\right) \geq \mathbf{0}\,. \tag{39}$$

The factor matrix $\mathbf{L}_{\phi} \in \mathbb{R}^{N\times r}$ has the row dimension N, i.e. the number of inequality constraints and a column dimension of r, i.e. the tensor rank. All other factor matrices $\mathbf{L}_i \in \mathbb{R}^{2\times r}$ have a row dimension of 2 and can be normalized as discussed.

The computation of the left side of the iMTI model from (36) is possible in the factored polynomial form with (5) to (7) if it is transformed in 1-norm, 2-norm, sparse or zero representation in analogy to the eMTI model in Section 3.2.

The variable vector

$$(\dot{\mathbf{x}},\mathbf{x},\mathbf{u}) = (\dot{x}_1,..\dot{x}_n,x_1\ldots x_n,u_1\ldots,u_m)^T, \tag{40}$$

consists of the state derivatives, states and inputs of the iMTI system. Because of the dimension of the parameter Tensor H in CP decomposition the polynomial has $2n+m$ factors in the products and r addends. A detailed description to convert any iMTI in a 1-norm normalized iMTI can be found in (Lichtenberg et al., 2022). It can be shown that an easy transformation of the eMTI to an iMTI model is achieved by bringing the derivatives to the other side. This will be used for the running example next.

Example 4.1. *For the example, a sparse representation* (25) *of the iMTI model is given by*

$$\widetilde{\mathbf{H}}_{\dot{x}_1} = \begin{pmatrix} \infty & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{41a}$$

$$\widetilde{\mathbf{H}}_{\dot{x}_2} = \begin{pmatrix} 0 & 0 & 0 & \infty & 0 & 0 \end{pmatrix}, \tag{41b}$$

$$\widetilde{\mathbf{H}}_{x_1} = \begin{pmatrix} 0 & 0.3 & 0.2 & 0 & 0.4 & 0.2 \end{pmatrix}, \tag{41c}$$

$$\widetilde{\mathbf{H}}_{x_2} = \begin{pmatrix} 0 & \infty & 0 & 0 & 0.9 & \infty \end{pmatrix}, \tag{41d}$$

$$\widetilde{\mathbf{H}}_{u} = \begin{pmatrix} 0 & 0 & \infty & 0 & 0 & 0 \end{pmatrix}, \tag{41e}$$

$$\widetilde{\mathbf{H}}_{\phi} = \begin{pmatrix} -1 & 0.2 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0.6 & 0.4 \end{pmatrix}. \tag{41f}$$

With this method, the rank of the model increases in comparison to the eMTI model by the number of states $n = 2$ from $r = 4$ to $r = 6$.

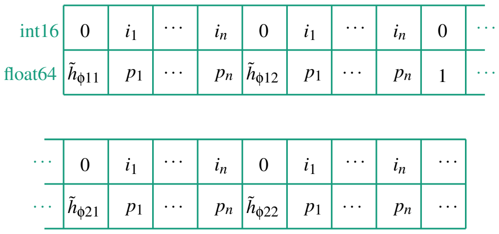
5 MODELING FORMATS

In Section 3.2 it was shown, that with the normalization procedure the memory requirements of the model can be reduced by almost 50%. However, this representation still contains zeros. For very large systems this will increase the size significantly. By making use of the sparsity this can be further reduced.

The iMTI model from (42) is given in the sparse representation. These equations are developed by using (36) with the factor matrices in (41). The parameter tensor in (41) contains 42 values. However, only the non-zero parameters from the parameter tensor H must be stored, which reduces the size of the system, c.f. 5.1. The sparse normalization format outperforms the others because a parameter $p_i = 0$ leads to the corresponding factor $(1 + p_i x_i) = 1$ which is the neural element of multiplication. This implies that the i-th variable has no influence.

The model parameters are saved in a binary file. In principle, this can also be done in human-readable formats such as ASCII. However, these formats have larger disk footprints as binary formats. In addition, reading and writing of binary files is faster compared to the ASCII format.

The binary file must have a clear structure such that it can be read-in and written correctly. A structured format with integers and floating-point numbers can be saved as shown in Figure 3. Here, the format is shown as two rows containing 16-bit integers in the first row and 64-bit floating-point numbers in the second row. However, the two rows are chosen for better illustration. In the raw binary format the integers and floating-point numbers are placed alternately. The indices corresponding to the specific states, inputs or outputs are stored as positive integers. The derivatives are specified with the same indices of the relating states, but negated. The actual factors for the states, derivatives, inputs and outputs are stored as floating-point numbers. An illustration of this is given in example 5.1.



int16	0	i_1	$\cdots$	i_n	0	i_1	$\cdots$	i_n	0	$\cdots$
float64	$\tilde{h}_{\phi 11}$	p_1	$\cdots$	p_n	$\tilde{h}_{\phi 12}$	p_1	$\cdots$	p_n	1	$\cdots$

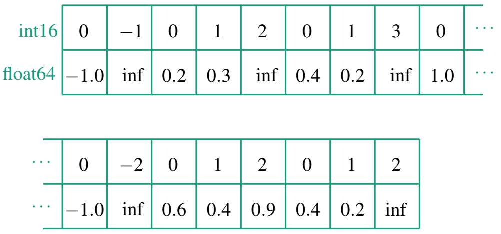
$\cdots$	0	i_1	$\cdots$	i_n	0	i_1	$\cdots$	i_n	$\cdots$
$\cdots$	$\tilde{h}_{\phi 21}$	p_1	$\cdots$	p_n	$\tilde{h}_{\phi 22}$	p_1	$\cdots$	p_n	$\cdots$

Figure 3: Structure of the general minimal representation.

Example 5.1. *The second-order multilinear model from the running example can be represented in sparse implicit CP format as*

$$\begin{pmatrix} -\dot{x}_1+0.2(1+0.3x_1)x_2+0.4(1+0.2x_1)u \\ -\dot{x}_2+0.6(1+0.4x_1)(1+0.9x_2)+0.4(1+0.2x_1)x_2 \end{pmatrix} = \mathbf{0}. \tag{42}$$

The corresponding minimal memory format will then be



int16	0	−1	0	1	2	0	1	3	0	$\cdots$
float64	−1.0	inf	0.2	0.3	inf	0.4	0.2	inf	1.0	$\cdots$

$\cdots$	0	−2	0	1	2	0	1	2
$\cdots$	−1.0	inf	0.6	0.4	0.9	0.4	0.2	inf

Figure 4: Structure of minimal representation for the example.

This results in a file of 170 bytes. One zero in the integer row indicates that a new addend starts. The number of addends is known by the rank of the system. Two following zeros indicate that a new equation of the system starts.

The iMTI model class gives the ability to compose models easily by appending new system equations. This is also directly possible in reduced CP representation if the indices are disjoint or shifted correctly. Thus, iMTI models allow reduced representation for large scale systems, which enable efficient composition, simulation and linearization techniques.

6 CONCLUSION

In this paper, a reduced normalized CP tensor representation for implicit multilinear (iMTI) models was presented. By applying tensor decomposition and normalization methods the memory requirements for the model can be significantly reduced. This is particularly relevant for modeling and simulation of large-scale systems in broad operational ranges, e.g., energy systems. Linearization methods have been adapted to normalized eMTI models providing access to methods of linear systems theory, such as local stability analysis. Future work will focus on efficient simulation and linearization methods as well as tool development for iMTI models.

ACKNOWLEDGMENT

This work was partly supported by the project SONDE of the Federal Ministry of Education and Research, Germany (Grant-No.: 13FH144PA8) and partly supported by the Free and Hanseatic City of Hamburg.

REFERENCES

F. Milano, F. Dörfler, G. Hug, D. J. Hill, and G. Verbič

(2018). Foundations and challenges of low-inertia systems (invited paper). In *2018 Power Systems Computation Conference (PSCC)*, pages 1–25.

Farrokhseresht, N., van der Meer, A. A., Rueda Torres, J., and van der Meijden, M. A. M. M. (2021). Mosaik and fmi-based co-simulation applied to transient stability analysis of grid-forming converter modulated wind power plants. *Applied Sciences*, 11(5):2410.

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*. PhD thesis, HafenCity Universität Hamburg.

Kruppa, K. and Lichtenberg, G. (2018). Feedback Linearization of Multilinear Time-invariant Systems using Tensor Decomposition Methods. In de Rango, F., Ören, T., and Obaidat, M. S., editors, *SIMULTECH 2018*, pages 232–243. SCITEPRESS - Science and Technology Publications Lda.

Kruppa, K., Pangalos, G., and Lichtenberg, G. (2014). Multilinear approximation of nonlinear state space models. *IFAC Proceedings Volumes*, 47(3):9474–9479.

Lichtenberg, G., Pangalos, G., Yáñez, C. C., Luxa, A., Jöres, N., Schnelle, L., and Kaufmann, C. (2022). Implicit multilinear modeling: An introduction with application to energy systems. *at - Automatisierungstechnik*, 70(1):13–30.

López, C. D., Cvetković, M., van der Meer, A., and Palensky, P. (2019). Co-simulation of intelligent power systems. In *Intelligent Integrated Energy Systems*, pages 99–119. Springer, Cham.

Pangalos, G. (2016). *Model-based controller design methods for heating systems*. Ph.d. dissertation, Technische Universität Hamburg-Harburg.

Pangalos, G., Eichler, A., and Lichtenberg, G. (2013). Tensor systems : multilinear modeling and applications. In *SIMULTECH 2013 - Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Application*. SciTePress.

Verstraete, F., Murg, V., and Cirac, J. I. (2008). Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224.

Vogt, M., Marten, F., and Braun, M. (2018). A survey and statistical analysis of smart grid co-simulations. *Applied Energy*, 222:67–78.

Wiens, M., Frahm, S., Thomas, P., and Kahn, S. (2021). Holistic simulation of wind turbines with fully aeroelastic and electrical model. *Forschung im Ingenieurwesen*, 85(2):417–424.