2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

An approach to parameter identification for Boolean-structured multilinear time-invariant models*

Marah Engels[1], Gerwald Lichtenberg[2] and Steffi Knorn[3]

Abstract— **In this paper, a parameter identification method for multilinear time-invariant grey-box models which are structured by Boolean functions is presented. Applying binary indexing to network graphs enables an efficient extraction of structural system information which can be transformed into continuous Zhegalkin polynomials. Their parameters are identified with a nonnegative alternating least squares (ALS)-algorithm. A running example demonstrates the application of this method and the algorithm is briefly analysed.**

I. INTRODUCTION

When modeling and simulating complex real-world systems, an efficient representation of the model parameter space is of growing significance. This importance is particularly evident in various practical scenarios, such as the application of model-based controllers that require real-time computational capabilities. Parameter spaces with a high number of signals and parameters often contradict the demand for efficient computations. For this reason, the development of efficient modeling strategies, that at the same time preserve important system properties as, e.g., it's basic connectivity structure, is a current area of research.

Certain subgroups of real-world systems, such as, e.g., industrial systems [1], manufacturing processes [2], and power systems [3], can be characterized as discrete event systems (DES). This paper deals with a new approach to the representation of DES. Effective approaches for the representation of discrete event systems, binary systems, graphs, and automata can be found by applying methods from the Boolean differential calculus, which was first introduced in [4]. A powerful tool to perform calculations of the Boolean differential calculus is "XBOOLE", see [5]. A link between Boolean and real-valued functions is provided by Zhegalkin polynomials [6], which have, e.g., been applied for the modeling of sequential systems in [7] and structural gene modeling in [8]. The use of Zhegalkin polynomials produces a subclass of multilinear functions.

Multilinear models, as initially documented in [9], offer a versatile framework for modeling complex systems with nonlinear dynamics. Simultaneously, they provide an organized and structured representation, making them a valuable tool in multiple domains. This model class finds utility in a range of applications, including simulating heating systems (as observed in [9], [10]) and analyzing energy systems (as detailed in [11]). Further application fields of multilinear models include, e.g., black box parameter identification, see [12], or system identification with tensor-based methods, see [13]. However, despite the merits of multilinear models, parameter identification methods for these are still topics of current research.

In [14] a multilinear modeling approach that derives state equations preserving the connectivity structure of a model by applying binary indexes has been introduced. Due to binary encoding, that approach allows the representation of a huge number of outputs by a small number of states. This paper introduces an approach to parameterization and grey-box parameter identification of the states of such Boolean-structured multilinear time-invariant (MTI)-models. In Section II, fundamentals used within this paper from the field of Boolean differential calculus and MTI-models are outlined. Section III describes the derivation of a reduced MTI-model structure using a running example. Based on that, Section IV defines a parameterization of the model equation which preserves the structural information, before an alternating least squares (ALS)-algorithm is introduced to enable the identification of parameters from a given state trajectory. Conclusively, the ALS-algorithm is applied for parameter identification from a given state trajectory and evaluated concerning accuracy and runtime.

II. FUNDAMENTALS

A. Multilinear and Boolean functions

Definition 1: A multilinear function is defined as the inner product between a row vector of coefficients $\mathbf{f}^{\mathrm{T}}$ and a monomial vector $\mathbf{m}(\mathbf{x})$:

$$f(\mathbf{x}) = \mathbf{f}^{\mathrm{T}}\mathbf{m}(\mathbf{x}) \tag{1}$$

where

$$\mathbf{m}(\mathbf{x}) = \begin{pmatrix} 1 \\ x_n \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \in \mathbb{R}^{2^n}. \tag{2}$$

Here, $\otimes$ represents the Kronecker product, and $\mathbf{f}^{\mathrm{T}}$ denotes the vector $\mathbf{f}$ transposed.

Boolean functions, a subclass of multilinear functions, are defined on binary vectors from the set $\mathbb{B} = \{0, 1\}$, where 0 denotes "False" and 1 signifies "True." This notation is consistently used throughout the paper. A Boolean function can be represented by a truth vector $\mathbf{b} = (b_1, ..., b_{2n})^{\mathrm{T}} \in \mathbb{B}^{2^n}$. Zhegalkin polynomials [6] offer an alternative representation for Boolean functions, employing conjunctions (i.e. AND connections) of non-negated

*This work has been developed within the project NRL (Living Lab Northern Germany) which is partly funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

[1]Competence Center for Renewable Energies and Energy Efficiency, HAW Hamburg marah.engels@haw-hamburg.de
[2]Faculty of Life Sciences, HAW Hamburg gerwald.lichtenberg@haw-hamburg.de
[3]Technical University Berlin knorn@tu-berlin.de

variables connected by XOR operations. As expounded in [7], the formal structure of Zhegalkin polynomials corresponds to multilinear functions by replacing conjunctions with multiplication and XOR operations with summation over the integers modulo 2. Using a slight notation adjustment, we express Zhegalkin polynomials as the inner product of a truth vector $\mathbf{b}^{\mathrm{T}}$ and a vector $\mathbf{l}(\mathbf{x})$, which is built as a Kronecker product of vectors of negative and positive literals:

$$f(\mathbf{x}) = \mathbf{b}^{\mathrm{T}}\mathbf{l}(\mathbf{x}) \tag{3}$$

where

$$\mathbf{l}(\mathbf{x}) = \begin{pmatrix} 1-x_n \\ x_n \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1-x_1 \\ x_1 \end{pmatrix} \in \mathbb{R}^{2^n}. \tag{4}$$

The term "literal", in mathematical logic, refers to the separate occurrences of a variable x_i. The evaluation of a Zhegalkin polynomial for values $\mathbf{x} \in \{0,1\}^{2^n}$ yields the same result as the Boolean function represented by the truth vector $\mathbf{b}$, while at the same time allowing calculations with continuous values, as further detailed in [15].

B. Autonomous multilinear time-invariant systems

Discrete-time autonomous MTI models characterize systems where the states vector $\mathbf{x}$, and outputs vector $\mathbf{y}$, can be effectively described by multilinear functions $f_i(\mathbf{x})$ and $g_j(\mathbf{x})$. For a model with n states and p outputs this representation takes the form:

$$\mathbf{x}(k+1) = (f_1(\mathbf{x}(k)), f_2(\mathbf{x}(k)), \ldots, f_n(\mathbf{x}(k)))^{\mathrm{T}}, \tag{5}$$

$$\mathbf{y}(k) = (g_1(\mathbf{x}(k)), g_2(\mathbf{x}(k)), \ldots, g_p(\mathbf{x}(k)))^{\mathrm{T}}. \tag{6}$$

Combining the row vectors of coefficients of the multilinear functions (see (1)) in a transition matrix $\mathbf{F} \in \mathbb{R}^{n \times 2^n}$ and an output matrix $\mathbf{G} \in \mathbb{R}^{p \times 2^n}$ yields a matrix representation

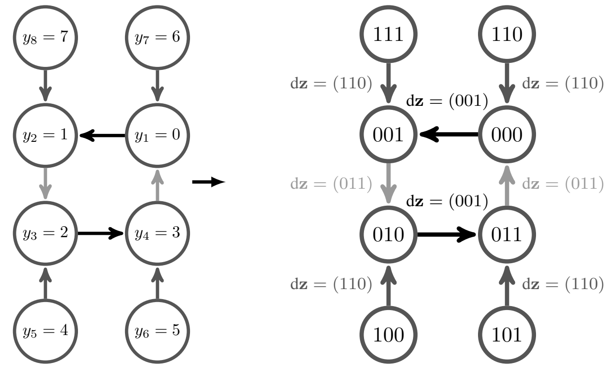
$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{m}(\mathbf{x}(k)) \tag{7}$$

$$\mathbf{y}(k) = \mathbf{G}\mathbf{m}(\mathbf{x}(k)). \tag{8}$$

In (7) and (8) it is evident that, due to the monomial vector $\mathbf{m}(\mathbf{x}(k))$, see (2), the used equation structure permits all multilinear combinations of the occurring variables which can be adjusted afterward by the factor matrices. For a comprehensive understanding of these models, a detailed exposition can be found in [10]. The modeling approach presented in this paper uses a subclass of MTI-models, as detailed in Section III.

C. Boolean differentials

For systems whose structure is representable by graphs, there exist various methods of representing the structural information. A promising approach to storing the structural information of a system which can be described by η nodes and ϵ edges is based on the field of Boolean algebra. The nodes represent distinct objects of the system, e.g. subsystems, while the edges indicate, whether there exists any type of interaction between the objects. The method uses binary indexing of the graph nodes of a model. A Boolean vector $\mathbf{z} = (z_1, \ldots, z_w)$ is used to reindex the graph nodes y_i with the number w of Boolean variables z_i depending on the number of nodes η by $w = \lceil \log_2 \eta \rceil$. Now the edges of



(a) Integer indices y_i (b) Binary indices z_i

Fig. 1: Binary indexing of graph nodes with $\mathbf{z} = (z_1, z_2, z_3)$ and $\mathrm{d}\mathbf{z} = (\mathrm{d}z_1, \mathrm{d}z_2, \mathrm{d}z_3)$

the graph can be described by a vector of Boolean differentials $\mathrm{d}\mathbf{z} = (\mathrm{d}z_1, \ldots, \mathrm{d}z_w)$. According to [5], this vector of Boolean differentials indicates the change in a Boolean space between the subsequent nodes $\mathbf{z}$ and $\mathbf{z}'$. A value of 1 denotes a change in the Boolean value of z_i, whereas a 0 represents no change in the value. Note, that the Boolean differential $\mathrm{d}\mathbf{z}$ itself does not contain information on the direction of the edge between two nodes. This information is obtained by representing the binary-valued nodes and edges of a graph in a table, as detailed in Subsection II-D.

Example 1: Binary reindexing for a directed deterministic graph with eight nodes is shown in Fig. 1. The nodes of the graph are represented by $w = 3$ Boolean variables. The edges can now be described by a vector of Boolean differentials $(\mathrm{d}z_1, \mathrm{d}z_2, \mathrm{d}z_3)$. The change of a Boolean variable z_i can be expressed using the logical XOR-Operator $\oplus$, which can also be represented by the identities of Boolean Algebra, as stated in [5]. Therefore, the relation between the Boolean indices of two consecutive nodes z_i and z_i' can be given by

$$z_i' = z_i \oplus \mathrm{d}z_i = (z_i \wedge \overline{\mathrm{d}z_i}) \vee (\overline{z_i} \wedge \mathrm{d}z_i), \tag{9}$$

for all $i = 1, 2, ..., w$ with $\wedge$ denoting the logical AND-operator, $\vee$ the OR-operator, and $\overline{z_i}$ expressing logical negation of the variable z_i.

D. Ternary vector lists

A table consisting of the information on the binary-valued nodes and edges of a graph is called a binary vector list (BVL), see [5]. Each row of a BVL contains a concatenation of a vector $\mathbf{z} \in \mathbb{B}^{1\times\eta}$ and a vector $\mathrm{d}\mathbf{z} \in \mathbb{B}^{1\times\eta}$. Therefore, the dimensions of a BVL containing the structural information of a graph with η nodes and ϵ edges are $2\eta \times \epsilon$. According to [5], the solution set of a graph, referring to occurring nodes and edges, can be described by a BVL. The rows of a BVL are hereafter referred to as vectors. This type of representation is applicable to directed, undirected, deterministic, and non-deterministic graphs. This paper is limited to investigations for directed, deterministic graphs. A BVL can serve as a starting point for an efficient representation of the structural information of a model graph. By

TABLE I: Representation of the nodes and edges of a graph as a binary and ternary vector list

(a) Boolean vector list

z_1	z_2	z_3	$\mathrm{d}z_1$	$\mathrm{d}z_2$	$\mathrm{d}z_3$
0	0	0	0	0	1
0	1	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	1
1	0	0	1	1	0
1	1	0	1	1	0
1	0	1	1	1	0
1	1	1	1	1	0

(b) Ternary vector list

z_1	z_2	z_3	$\mathrm{d}z_1$	$\mathrm{d}z_2$	$\mathrm{d}z_3$
0	—	0	0	0	1
0	—	1	0	1	1
1	—	—	1	1	0

replacing ”0/1-combinations” of BVL-vectors with a ”Don’t Care Operator” (—) also used in [5], a reduction can be achieved. The resulting ternary vector list (TVL) consists of a reduced number of rows, which will be referred to as vectors. Those vectors contain the three Boolean elements 0, 1, and —. A TVL still contains all information on the nodes and edges of the model graph but in fewer vectors.

Example 2: The BVL of the graph in Fig. 1b is given in Table Ia. Combining ”0/1-combinations” in the Boolean vectors results in the TVL of the graph given in Table Ib. See for example the vectors 1 and 2 in Table Ia, which only differ in z_2 being 0 or 1. Because the value of z_2 has no influence on the resulting $\mathrm{d}z_1$ and $\mathrm{d}z_2$ for vectors z_1 and z_2, it can be substituted by —. For the running example, the number of vectors has been reduced from 8 vectors in the BVL to only 3 vectors in the TVL.

An important property of a subclass of TVLs, that has been defined, e.g., in [5], is orthogonality. It is the foundation of many algorithms in the field of Boolean Algebra as well as the modeling approach presented in this paper.

Definition 2: In an orthogonal TVL (OTVL), no identical Boolean vectors exist, meaning each pair of vectors of the corresponding BVL differs in at least one position.

E. Zhegalkin Polynomials of Ternary Vector Lists

By using algebraic relaxations (see [7]), orthogonal TVLs can be written as Zhegalkin polynomials. For this, they are interpreted as a Boolean function in disjunctive form, where the elements in the individual OTVL-vectors are linked by AND-operations and the vectors by OR-operations. In mathematical terms, this is defined as the sum of all OTVL vectors for which the respective vector literals l_{ji} are multiplied, see [15]. The polynomial corresponding to the OTVL with elements v_{ji} is then given by

$$f(X) = \sum_{j=1}^{q} \left(\prod_{i=1}^{w} l_{ji} \right) \text{ with } l_{ji} = \begin{cases} x_i & \text{if } v_{ji} = 1 \\ 1 - x_i & \text{if } v_{ji} = 0 \\ 1 & \text{if } v_{ji} = — \end{cases} \tag{10}$$

with q denoting the number of OTVL vectors, w the number of Boolean variables, and state variables x_i. The label of the variable is derived from the heading of the TVL and refers to node indices as well as to Boolean differentials. Throughout this paper, variables x_i will represent the real-valued domain, while variables z_i denote the Boolean domain.

Example 3: The significance of orthogonality in TVLs will be demonstrated by this example. We have a TVL $\mathbf{T_1} = \begin{pmatrix} 0 & — \\ — & 1 \end{pmatrix}$ which is equivalent to the logic equation $\overline{z_1} \vee z_2 = 1$ (see Subsection II-F). Since both vectors of $\mathbf{T_1}$ contain the element $(0, 1)$, they are not orthogonal. The polynomial of the TVL is given by $(1 - x_1) + x_2$, which yields a value of 2 for $x_1 = 0; x_2 = 1$. Consequently, the non-orthogonal TVL does not fulfill the requirement of a Zhegalkin polynomial to stay in the set $\{0, 1\}$ when given values from that set as input. By fixing one ”Don’t care operator” to a value of 0, we orthogonalize the TVL. Hereby we obtain two possible orthogonal TVLs given by

$$\mathbf{T}_2 = \begin{pmatrix} 0 & — \\ 1 & 1 \end{pmatrix}; \quad \mathbf{T}_3 = \begin{pmatrix} 0 & 0 \\ — & 1 \end{pmatrix}.$$

The Zhegalkin polynomials corresponding to these OTVLs yield two different representations

$$(1 - x_1) + x_1 x_2 = (1 - x_1)(1 - x_2) + x_2. \tag{11}$$

of the same equation. For all combinations of values from the set $\{0, 1\}$, (11) is equivalent to the result of the given logic equation.

F. Solution of ternary vector lists for particular variables

An OTVL containing the structural information of a model represents a logic equation in disjunctive form, that reads as

$$F_s(\mathbf{z}) = \bigvee_{j=1}^{q} \left(\bigwedge_{i=1}^{2w} l_{ji} \right) = 1 \text{ with } l_{ji} = \begin{cases} z_i & \text{if } v_{ji} = 1 \\ \overline{z_i} & \text{if } v_{ji} = 0 \\ 1 & \text{if } v_{ji} = — \end{cases} \tag{12}$$

with the value of l_{ji} being assigned for $\mathbf{z} = (z_1, \ldots, z_w)$ and $\mathbf{dz} = (\mathrm{d}z_1, \ldots, \mathrm{d}z_w)$, accordingly. This logic equation can be solved for individual variables z_i or $\mathrm{d}z_i$. A detailed description of how to solve a logic equation with respect to certain variables can be found in [5].

Example 4: Following Definition 2, the TVL in Table Ib is orthogonal. The logic equation describing this OTVL is given by

$$\begin{aligned} F_s(z_1, z_2, z_3, \mathrm{d}z_1, \mathrm{d}z_2, \mathrm{d}z_3) &= \overline{z_1}\,\overline{z_3}\,\overline{\mathrm{d}z_1}\,\overline{\mathrm{d}z_2}\,\mathrm{d}z_3 \\ &\vee \overline{z_1} z_3 \overline{\mathrm{d}z_1}\,\mathrm{d}z_2\,\mathrm{d}z_3 \vee z_1\,\mathrm{d}z_1\,\mathrm{d}z_2 \overline{\mathrm{d}z_3}. \end{aligned} \tag{13}$$

Equation (13) is uniquely solvable with regard to the variables $\mathrm{d}z_1, \mathrm{d}z_2, \mathrm{d}z_3$. The concept of unique solvability of logic functions is detailed in [5]. The solution functions of the running example $\mathrm{d}z_1 = f_1(z_1, z_2, z_3)$, $\mathrm{d}z_2 = f_2(z_1, z_2, z_3)$, and $\mathrm{d}z_3 = f_3(z_1, z_2, z_3)$ can be determined as

$$\mathrm{d}z_1 = z_1; \ \ \mathrm{d}z_2 = z_1 \vee z_3; \ \ \mathrm{d}z_3 = \overline{z_1}. \tag{14}$$

III. Construction of a reduced MTI-model structure

Applying binary indexing and OTVLs can help to reduce the storage required for a model’s structural information. The number of vectors in the OTVL is defined as the rank q of the reduced model. In order to construct an MTI-model

TABLE II: OTVLs containing structural information for $\mathbf{z}'$

(a) z_1'

z_1	z_2	z_3

(b) z_2'

z_1	z_2	z_3
0	0	1
1	0	—
0	1	0

(c) z_3'

z_1	z_2	z_3
0	—	0
1	—	1

equation containing the structural information of a model, a representation for the state equation has to be found. From (9), for an edge $\mathbf{z} \to \mathbf{z}'$, the logical equivalent to a discrete-time state equation can be derived as

$$\mathbf{z}' = \mathbf{z} \oplus \mathrm{d}\mathbf{z} \tag{15}$$

with the XOR-operator being applied element-wise to the corresponding bits in the vectors and $\mathbf{z}', \mathbf{z}$ denoting two consecutive nodes of the graph. By this, a static, non time-varying structure is represented. The elements $\mathrm{d}z_i$ in the vector of Boolean differentials $\mathrm{d}\mathbf{z}$ result from the model structure. They can be obtained by solving the logic equation derived from the OTVL, which contains the structural information of the graph, for the variables $\mathrm{d}z_i$, as shown in Subsection II-F.

Example 5: Applying (15) to the running example and inserting the solutions in (14) for $\mathrm{d}z_i$ yields

$$\begin{pmatrix} z_1' \\ z_2' \\ z_3' \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \oplus \begin{pmatrix} \mathrm{d}z_1 \\ \mathrm{d}z_2 \\ \mathrm{d}z_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \overline{z_1}\,\overline{z_2} z_3 \vee z_1 \overline{z_2} \vee \overline{z_1} z_2 \overline{z_3} \\ \overline{z_1}\,\overline{z_3} \vee z_1 z_3 \end{pmatrix}. \tag{16}$$

The OTVL-representation of (16) is given in Table II.

By using the algebraic relaxations $z_i = x_i$ and $\overline{z_i} = 1 - x_i$ from [7], the logic equations defining $\mathbf{z}'$ can be written as Zhegalkin polynomials, which are valid for real numbers x_i. The result of this relaxation are state equations of a discrete-time MTI-model with w states, which contain the structural information of the model. In contrast to general MTI-models, see Subsection II-B, the multilinear equations derived from the model structure do not permit all combinations of occurring variables, but only a restricted subspace of those.

Example 6: Applying algebraic relaxations to (16) yields

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{pmatrix} 0 \\ (1-x_1(k))(1-x_2(k))x_3(k) + \\ (1-x_1(k))(1-x_3(k)) + \\ +0 \\ +x_1(k)(1-x_2(k)) + (1-x_1(k))x_2(k)(1-x_3(k)) \\ +x_1 x_3 \end{pmatrix} \tag{17}$$

for discrete time steps $k \geq 0$. Utilizing (10) on the OTVLs in Table II results in (17) as well due to the convertibility of logic functions, OTVLs, and Zhegalkin polynomials.

Calculating the state trajectory of (17) for an initial condition of $(1, 1, 1)$ produces the behavior represented in Fig. 2. The behavior of x_1, x_2, x_3 coincides with the sequence of binary node indices of the graph of the running example in Fig. 1b. From this, it can be seen that (17) contains the structural information of the running example.

Assuming the states x_i are observed, an output equation $y(k)$ is obtained by applying the Boolean literal vector (4). The construction of all possible combinations of negative and

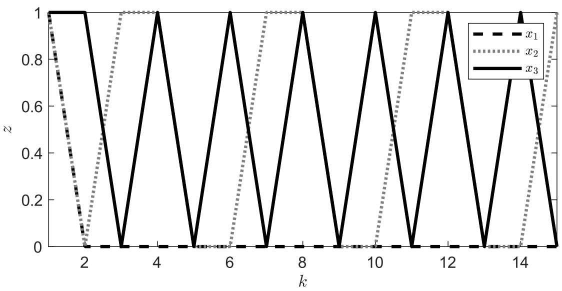


Fig. 2: State trajectory for initial states of $(1, 1, 1)$

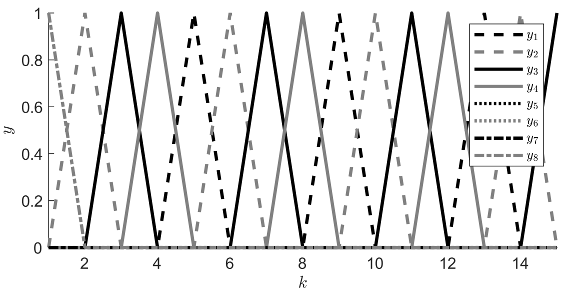


Fig. 3: Output trajectory for initial states of $(1, 1, 1)$

positive literals yields the output equation for the reduced MTI-model structure as

$$\mathbf{y}(k) = \begin{pmatrix} 1 - x_1(k) \\ x_1(k) \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 - x_w(k) \\ x_w(k) \end{pmatrix} \in \mathbb{R}^{2^w}. \tag{18}$$

It can be seen from (18), that a reduced MTI-model structure with w states x_i can represent 2^w model outputs in $\mathbf{y}$. The outputs $\mathbf{y}$ are always linearly independent due to the structure of the output function. Since the ratio $\frac{2^w}{w}$ between outputs and states increases with the number of model outputs, the proposed approach shows an enormous reduction potential.

Example 7: Inserting the state trajectories from Fig. 2 into (18) results in the output trajectories shown in Fig. 3. The behavior of the outputs y_i matches the sequence of integer-indexed graph nodes in Fig. 1a when starting at the node $y_8 = 7$. This indicates, that the output equation as well preserves the structural information of the model.

IV. Parameter identification for a reduced MTI-model structure

In the previous section the equation structure of a non-parameterized reduced MTI-model containing structural information of a system has been developed by applying (10). A representation of continuous-valued systems will, however, require the representation of continuous values. In [14] it has been investigated, that the use of parameterized Zhegalkin polynomials enables relaxation of the Boolean states and outputs to continuous-valued trajectories while preserving linear independence. This leads to the assumption, that parameter identification from given data is possible. By this, nonlinear system behavior can be approximated by a reduced MTI-model structure. This section will give an overview of a possible parameterization of the reduced MTI-model structure, followed by introducing an ALS-algorithm that enables parameter identification from given data.

A. Parameterization of the reduced MTI-model structure

To approximate nonlinear system behavior with the reduced MTI-model structure derived in Section III, the state equations of the model need to be parameterized. The parameterization is performed by introducing real-valued factors $\Phi \in \mathbb{R}_{\geq 0}^{w \times r_{\max}}$ and $\Lambda \in \mathbb{R}_{>0}^{r_{\max} \times w \times w}$ to (10), with $r_{\max}$ denoting the maximum number of rows over all OTVLs. This yields

$$x_s(k+1) = \sum_{j=1}^{r_s} \phi_{sj} \prod_{i=1}^{w} l_{jis}(k)$$

$$\text{with } l_{jis}(k) = \begin{cases} x_i(k) & \text{if } v_{jis} = 1 \\ 1 - \lambda_{jis} x_i(k) & \text{if } v_{jis} = 0 \\ 1 & \text{if } v_{jis} = — \end{cases}, \tag{19}$$

$$\text{and } \Lambda(j,i,s) = \begin{cases} 0 & \text{if } v_{jis} = 1 \\ \lambda_{jis} & \text{if } v_{jis} = 0 \\ 0 & \text{if } v_{jis} = — \end{cases}$$

where $r_s \leq r_{\max}$ refers to the number of rows in the OTVL corresponding to state equation x_s with $s \in \{1, \ldots, w\}$. The applied factorization relaxes the equations of state. The performed parameterization is the basis for a grey-box parameter identification with the structural model information contained in the state equations of the model. When setting all parameters ϕ_{sj} and λ_{jis} for $v_{jis} = 0$ in (19) to values of 1, the structural information given also by (10) is obtained. In order to preserve the structural information in the parameterized equation, the real-valued factors are bound to the non-negative domain. In contrast to the general MTI-model structure described in Subsection II-B, the developed reduced MTI-model structure does not allow all possible multilinear combinations of occurring variables. The allowed variables are limited by the structural constraints imposed by the derivation of the model equation from the system structure. This eventuates in smaller dimensions of the model parameter space for all cases that do not depend on all combinations of occurring variables.

B. Non-negative ALS algorithm

A grey-box parameter identification for the structured MTI-model equation derived in the previous chapters can be performed assuming a trajectory of measured states $\tilde{X}$ is known. The goal of the parameter identification is to find a minimum of the Frobenius norm

$$\min_{\Phi,\Lambda} ||\tilde{X} - X(\Phi,\Lambda)||_F \tag{20}$$

with

$$\tilde{X} = \begin{pmatrix} \tilde{x}_1(1) & \cdots & \tilde{x}_w(1) \\ \vdots & \ddots & \vdots \\ \tilde{x}_1(k_{\text{end}}) & \cdots & \tilde{x}_w(k_{\text{end}}) \end{pmatrix}; X = \begin{pmatrix} x_1(1) & \cdots & x_w(1) \\ \vdots & \ddots & \vdots \\ x_1(k_{\text{end}}) & \cdots & x_w(k_{\text{end}}) \end{pmatrix} \tag{21}$$

for the difference between measured states $\tilde{x}_s$ and simulated states x_s. The trajectory X of simulated states is obtained by (19). For the nonlinear identification problem in (20) we assume, that identifiability is given, if (20) is solvable. In order to find parameters in Φ and Λ to achieve the minimum in (20), a non-negative ALS approach is developed. The used approach is depicted in Algorithm 1. As a first step, Φ and Λ are initialized randomly. The sparsity pattern for Λ is defined by the positions of elements $v_{jis} = 0$ in the OTVLs containing the structural model information. While the cost of the identification is greater than a set limit, the actual ALS algorithm is performed. The algorithm iteratively updates Φ and Λ. By fixing all parameters except one, the minimization problem (20) is transformed into an easily solvable over-determined linear system of equations. This solution can be given as a fraction $\frac{B}{A}$, as shown in lines 6 and 12. The matrices A and B are then reshaped to column vectors $\mathbf{a}$ and $\mathbf{b}$ which are used to find the non-negative solution for a least squares problem $\min_x ||\mathbf{a}x - \mathbf{b}||_2^2$. The solution space for that problem is constrained because Φ and Λ are only defined for the positive orthant. To map this dependence, in lines 9 and 15 a non-negative solution for the least-squares-problem is calculated. After each sequence of iterations, a simulation of the model with updated parameter matrices is performed, to examine the cost of the identification. The cost is defined as $||\tilde{X} - X(\Phi,\Lambda)||_F$.

Algorithm 1 ALS algorithm for non-negative structured multilinear parameter identification

Input: model equations X_{Eq} based on (19), data $\tilde{X}$
Output: Φ, Λ

1: random initialization $\Phi \in \mathbb{R}_{\geq 0}^{w \times r_{\max}}$, $\Lambda \in \mathbb{R}_{>0}^{r_{\max} \times w \times w}$
2: *cost* $\leftarrow$ inf ▷ initialize cost
3: $k_{\max} \leftarrow$ rows($\tilde{X}$) ▷ number of discrete steps k
4: **while** $cost > costLimit$ **do**
5: **for** $\phi_{s,j}$ in Φ **do**
6: $\frac{B_\phi}{A_\phi} \leftarrow$ solution of (20) with $X = X_{\text{Eq}}$ for $\phi_{s,j}$
7: $\mathbf{a}_\phi \leftarrow$ reshape($A_\phi, [wk_{\max}, 1]$)
8: $\mathbf{b}_\phi \leftarrow$ reshape($B_\phi, [wk_{\max}, 1]$)
9: $\phi_{s,j} \leftarrow$ non-negative Lsq solution for $\min_{\phi_{s,j}} ||\mathbf{a}_\phi \phi_{s,j} - \mathbf{b}_\phi||_2^2$
10: **end for**
11: **for** $\lambda_{j,i,s}$ in Λ **do**
12: $\frac{B_\lambda}{A_\lambda} \leftarrow$ solution of (20) with $X = X_{\text{Eq}}$ for $\lambda_{j,i,s}$
13: $\mathbf{a}_\lambda \leftarrow$ reshape($A_\lambda, [wk_{\max}, 1]$)
14: $\mathbf{b}_\lambda \leftarrow$ reshape($B_\lambda, [wk_{\max}, 1]$)
15: $\lambda_{j,i,s} \leftarrow$ non-negative Lsq solution for $\min_{\lambda_{j,i,s}} ||\mathbf{a}_\lambda \lambda_{j,i,s} - \mathbf{b}_\lambda||_2^2$
16: **end for**
17: $X \leftarrow$ simulate($X_{\text{Eq}}, \Phi, \Lambda$)
18: *cost* $\leftarrow$ norm($\tilde{X} - X(\Phi,\Lambda)$, F)
19: **end while**

C. Evaluation of the non-negative ALS algorithm

In this section, the ALS algorithm developed in Subsection IV-B is utilized to identify parameters for a model with a structure equivalent to the running example. A non-parameterized model with a structure given by (17) and a noisy state trajectory with an SNR of 13 dB

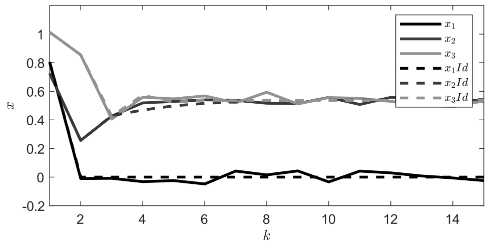


Fig. 4: Comparison of given and identified state trajectory

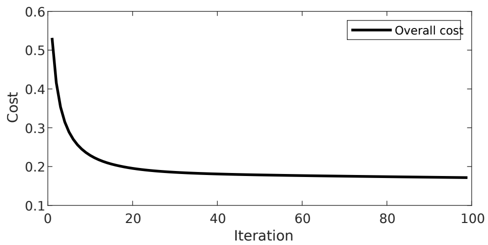


Fig. 5: Convergence behavior

for $k_{\text{end}} = 15$ steps are given as input to Algorithm 1. Afterward, a simulation of the identified model with the initial states of the given state trajectory is performed. The comparison between given and simulated trajectories is shown in Fig. 4 and the convergence behavior of the ALS-algorithm in Fig. 5. It is evident that the trajectories of the identified model are a good approximation of the given data. To evaluate the effectiveness of the ALS algorithm, the cost and runtime for identifying parameters to match the noisy state trajectory are compared between the ALS-algorithm and a nonlinear grey-box model identification performed by MATLABs “nlgreyest” toolbox with the “lsqnonlin”-solver. Termination conditions for the ALS algorithm are set to a cost limit of $5 \cdot 10^{-3}$ or a maximum of 100 iterations. For the identification with the “nlgreyest” toolbox the default settings with a function tolerance of $1 \cdot 10^{-5}$, a step tolerance of $1 \cdot 10^{-6}$, and a maximum of 20 iterations are used. Both algorithms are evaluated using an average of 500 executions, with the initial parameters varied randomly for each iteration. The results of the comparison can be taken from Table III. For the investigated example with the framework described in the previous paragraph, the nonnegative ALS algorithm achieves costs of a similar order of magnitude while at the same time achieving a much lower runtime.

V. Conclusion

An approach for grey-box parameter identification for the states of Boolean-structured multilinear models has been presented. For this, the modeling approach introduced in [14] has been extended by parameterizing the structural model equation in a way that allows parameter identification while preserving structural model information. Due to deriving the model equation from a binary indexed model structure by applying Zhegalkin polynomials, the model class is a restricted subspace in the class of multilinear models. This results in only a subclass of all possible multilinear variable combinations being permitted, which also reduces the number of model parameters to be determined.

TABLE III: Cost and computation time for nonlinear least-square and non-negative ALS

	Algorithm	
	nlgreyest (lsqnonlin)	nonnegative ALS
Cost	0.1792	0.1868
Time[s]	0.4770	0.0757

A possible use of grey-box system identification for Boolean-structured multilinear models is, for example, the modeling of energy network models for efficient controller design. In order to extend the proposed approach towards grey-box system identification, the states must be estimated from the measured inputs and outputs of a system. For this reason, further research will investigate methods of state identification as a preceding step before the parameter identification. Also, an extension of the proposed modeling approach towards models with a non-deterministic structure is necessary to enable the modeling of real-world systems as well.

References

[1] C. R. Paulista, T. A. Peixoto, and J. J. de Assis Rangel, “Modeling and discrete event simulation in industrial systems considering consumption and electrical energy generation,” *Journal of Cleaner Production*, 2019.

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Boston, MA: Springer US, 2008.

[3] M. Blas, G. Alvarez, and J. Sarli, “Improving the optimization of electric power systems through a discrete event based simulation model,” *Journal of Applied Research and Technology*, 2023.

[4] D. Bochmann and C. Posthoff, *Binäre dynamische Systeme*. Oldenburg Wissenschaftsverlag, 1981.

[5] C. Posthoff and B. Steinbach, *Logic Functions and Equations - Fundamentals and Applications using the XBOOLE-Monitor*, 3rd ed. Cham, Switzerland: Springer, 2022.

[6] I. I. Zhegalkin, “Die Arithmetisierung der symbolischen Logik,” *Mat. co*, 1928.

[7] D. Franke, *Sequentielle Systeme - Binäre und Fuzzy Automatisierung mit arithmetischen Polynomen*. Braunschweig; Wiesbaden: Vieweg, 1994.

[8] S. Faisal, G. Lichtenberg, and H. Werner, “Canalizing Zhegalkin Polynomials as Models for Gene Expression Time Series Data,” in *2006 IEEE International Conference on Engineering of Intelligent Systems*, 2006.

[9] G. Pangalos, A. Eichler, and G. Lichtenberg, *Hybrid Multilinear Modeling and Applications*. SpringerLink, 2014.

[10] K. Kruppa, “Multilinear Design of Decentralized Controller Networks for Building Automation Systems,” Ph.D. dissertation, Hafen City Universität Hamburg, 2018.

[11] G. Lichtenberg, G. Pangalos, C. C. Yáñez, A. Luxa, N. Jöres, L. Schnelle, and C. Kaufmann, “Implicit multilinear modeling,” *Automatisierungstechnik*, 2022.

[12] L. Schnelle, G. Lichtenberg, and C. Warnecke, “Using Low-rank Multilinear Parameter Identification for Anomaly Detection of Building Systems,” *IFAC Papers Online*, 2022.

[13] K. Batselier, C.-Y. Ko, A.-H. Phan, A. Cichocki, and N. Wong, “Multilinear state space system identification with matrix product operators,” *IFAC-PapersOnLine*, 2018, 18th IFAC Symposium on System Identification.

[14] M. Engels, G. Lichtenberg, and S. Knorn, “An approach to structured multilinear modeling with relaxed Boolean output functions,” in *22nd IFAC World Congress*, Yokohama, Japan, 2023.

[15] G. Lichtenberg and A. Eichler, “Multilinear Algebraic Boolean Modelling with Tensor Decomposition Techniques,” in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011.