How to Vary the Input Space of a T–S Fuzzy Model: A TP Model Transformation-Based Approach

Péter Baranyi, *Member, IEEE*

***Abstract*—The motivation behind 15 years of continuous development within the topic of the tensor product (TP) model transformation is that the greater the number of parameters or components of the Takagi–Sugeno (T–S) fuzzy model one can manipulate, the larger complexity reduction or control optimization one can achieve. This article proposes a radically new type of extension to the TP model transformation. While earlier variants of the TP model transformation focused on how the antecedent—consequent fuzzy set system of a given T–S fuzzy model could be varied, this article, in contrast, focuses on how the number of inputs to a given T–S fuzzy model can be manipulated. The proposed extension is capable of changing the number of inputs or transforming the nonlinearity between the fuzzy rules and the input dimensions. These new features considerably increase the modeling power of the TP model transformation, allowing for further complexity reduction and more powerful control optimization to be achieved. This article provides two examples to show how the proposed extension can be used in a routine-like fashion.**

***Index Terms*—PDC control design, TS fuzzy model, TP model transformation.**

I. Introduction

THE MAIN motivation behind 15 years continuous development within the topic of the tensor product (TP) model transformation has been to modify as many parameters or components of Takagi–Sugeno (T–S) fuzzy models as possible, in order to achieve greater complexity reduction and better control optimization. The TP model transformation is capable of deriving the absolute minimal number of fuzzy rules by modifying the shape of the antecedent fuzzy sets and the related consequent sets or the elements of the vertex systems. At the same time, many of the linear matrix inequality (LMI) based control design techniques are highly sensitive to such kinds of modifications. Thus, finding the proper fuzzy rules may lead to considerably better control performance.

Previous extensions of the TP model transformation focus on the internal parameters and components of the T–S fuzzy models, such as antecedents and consequents. In contrast, this article focuses on external parameters such as the number and nonlinearity of the inputs that lead to a further reduction in the number of fuzzy rules and lets us further manipulate the T–S fuzzy model for control optimization.

Manuscript received September 11, 2020; revised November 9, 2020; accepted November 12, 2020. Date of publication November 19, 2020; date of current version February 3, 2022. This work was supported by the "Thematic Excellence Program – Institutional Excellence Subprogram – Digital Industrial Technologies Research at Széchenyi István University TKP2020-IKA-10" Project.

The author is with the Research Center of Vehicle Industry, Szechenyi Istvan University, 9026 Gyor, Hungary (e-mail: prof.peter.baranyi@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TFUZZ.2020.3038488.

Digital Object Identifier 10.1109/TFUZZ.2020.3038488

This article proposes an extension to the TP model transformation capable of transforming a given T–S fuzzy model to an alternative T–S fuzzy model with the following benefits.

1) The transformed T–S fuzzy model can have a different number of inputs.
2) Inputs of the transformed T–S fuzzy model can be specified as a function of the original inputs.

This article shows that we can further decrease the rank of the dimensions if nonlinear inputs or additional inputs are defined. This automatically decreases the possible minimal number of antecedents. At the same time, it defines a conceptually new convex hull of the vertexes to which the LMI design techniques are typically sensitive, hence, it leads to further dimensions in optimization. Thus, in pursuing better control performance we can modify the antecedent-consequent pairs, namely the shape of the convex hull of the vertexes, using the previous variants of the TP model transformation. However, additionally, we can even modify the number and nonlinearity of the inputs using the currently proposed extension of the TP model transformation, which leads to the modification of the dimensions of the parameter vector of the convex hull as well, thereby projecting the manipulation of the convex hull defined by the vertexes to a new space.

A. Preliminaries

The very first idea of the higher order singular value decomposition (HOSVD) based handling of T–S fuzzy models was invented by Yam [1]. A further variant was proposed in [2]. This idea motivated the development of the TP model transformation and still gives the core of its different variants.

Assume a given T–S fuzzy model with the following symbolic notation:

$$\mathbf{y} = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{c}, \mathbf{r}) \tag{1}$$

where $\mathbf{x} \in \Omega^x \subset \mathbb{R}^N$ and $\mathbf{y} \in \Omega^y \subset \mathbb{R}^O$ represent the inputs and outputs, $\mathbf{a}$ and $\mathbf{c}$ represent the antecedents and the consequents, further, $\mathbf{r}$ represents the fuzzy rules.

1) *Approximation - complexity tradeoff:* The first variant of the TP model transformation was proposed as a complexity reduction technique of T–S fuzzy models [2], [3]. It transforms (1) to

$$\forall \mathbf{x} : f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{c}, \mathbf{r}) \approx_{\varepsilon} f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}^r, \mathbf{c}^r, \mathbf{r}^r) \tag{2}$$

where $\mathbf{a}^r$, $\mathbf{c}^r$, $\mathbf{r}^r$ represent the new antecedent sets, consequents, and fuzzy rules, respectively. This method was developed to find the minimal number of rules at $\varepsilon = 0$. In this regard, the HOSVD-based canonical form of T–S fuzzy models was published in [4]. In order to have further complexity reduction, it was capable of performing a tradeoff between the number of rules and approximation accuracy while ε was under control. Further investigations about the approximation properties of the transformation can be found in [5] and [6]. The TP model transformation was soon formalized and positioned to transform given functions $f(\mathbf{x})$ to T–S fuzzy model representation, so that the transformation determined $\mathbf{a}$, $\mathbf{c}$, and $\mathbf{r}$ as

$$\forall \mathbf{x} : f(\mathbf{x}) = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{c}, \mathbf{r}). \tag{3}$$

Computationally relaxed variants were investigated in [7] and [8].

2) *Varying the components of the T–S fuzzy model:* Further extensions of the TP model transformation were published to yield T–S fuzzy models having antecedent fuzzy sets with specific characteristics

$$\forall \mathbf{x} : f(\mathbf{x}) = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}^{co}, \mathbf{c}^{co}, \mathbf{r}^{co}). \tag{4}$$

Here, “co” means convex. The resulting antecedent fuzzy sets are expressed as a Ruspini partition (convex combination of the consequent sets). The first methods capable of deriving antecedent fuzzy sets with special characteristics were published in [1] and [2]. Various further extensions were developed, where the size of the convex hull defined by the resulting consequent vertices were also considered [6], [9]–[11]. As a consequence, the TP model transformation was capable of deriving an infinite number of different proper T–S fuzzy models which define exactly the same mapping between the inputs and outputs

$$\forall \mathbf{x} : f(\mathbf{x}) = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}_k, \mathbf{c}_k, \mathbf{r}_k) \tag{5}$$

where $k = 1 \ldots \infty$.

3) *Generalized TP model transformation:* The multi-TP model transformation, published in [12] and [13], is capable of even transforming a set of functions that may have a different number of outputs to a common antecedent and rule system

$$\forall \mathbf{x}, i : f_i(\mathbf{x}) = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{c}_i, \mathbf{r}) \tag{6}$$

where $i = 1 \ldots I$. The pseudo TP model transformation [12], [13] was developed to transform a given function or a T–S fuzzy model to a predefined antecedent fuzzy set system $\mathbf{b}$, such that the consequent sets are determined accordingly

$$\forall \mathbf{x} : f(\mathbf{x}) = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{b}, \mathbf{c}, \mathbf{r}). \tag{7}$$

The multi- and pseudo-TP model transformation were combined with various practical extensions and introduced as the generalized TP model transformation in [12] and [13]. The next extension proposed in [14] was executable on a set of functions having a different number of inputs

$$\forall i, \mathbf{x}_i : f_i(\mathbf{x}_i) = f^{\mathrm{TS}}(\mathbf{x}_i, \mathbf{a}, \mathbf{c}_i, \mathbf{r}) \tag{8}$$

where input vectors $\mathbf{x}_i$ can have a different number of dimensions as $\mathbf{x}_i \in \mathbb{R}^{N_i}$.

4) *TP model transformation in control design:* The early variants of the TP model transformation were already utilized in control design theories [15]. The latest extension of the TP model transformation in [16] focuses on state-space dynamic model structures and aims to extract both the quasi linear parameter varying (qLPV) structure [17], [18] and the T–S fuzzy model of the parameter varying system matrix. Thus, it starts with the dynamic model

$$\dot{\mathbf{x}}(t) = f_x(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)) \tag{9}$$

$$\mathbf{y}(t) = f_y(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)) \tag{10}$$

where $\mathbf{x}(t)$, $\mathbf{y}(t)$, $\mathbf{u}(t)$, and $\mathbf{p}(t)$ are the state, output, input, and parameter vectors, respectively, and ends up at

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \tag{11}$$

with T–S fuzzy model

$$\forall \mathbf{p}(t) \in \Omega : \mathbf{S}(\mathbf{p}(t)) = f^{\mathrm{TS}}(\mathbf{p}(t), \mathbf{a}, \mathbf{c}, \mathbf{r}). \tag{12}$$

Note that, this final extension holds all the advantageous properties of the generalized TP model transformation.

One of the most widely applied control design concept, the parallel distributed compensation design technique [17], starts with the T–S fuzzy model

$$f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{c}, \mathbf{r}) \tag{13}$$

then, derives feedback to each consequent system as shown in a symbolic form here

$$\mathrm{Design}(\mathbf{c}) \rightarrow \mathbf{f}. \tag{14}$$

Finally, the feedback gains $\mathbf{f}$ are substituted into the T–S fuzzy controller as

$$-f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{f}, \mathbf{r})\mathbf{x} \tag{15}$$

where the antecedent fuzzy sets and the rules are inherited form the T–S fuzzy model (13).

As discussed previously, we can have infinite number of variants of the T–S fuzzy model by varying the antecedent fuzzy sets. Obviously, the consequent sets will change accordingly, such that the following equality is kept:

$$\forall \mathbf{x}, k : f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}, \mathbf{c}, \mathbf{r}) = f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}_k, \mathbf{c}_k, \mathbf{r}_k) \tag{16}$$

which leads to an infinite number of controllers to the same model

$$\mathrm{Design}(\mathbf{c}_k) \rightarrow \mathbf{f}_k \tag{17}$$

$$\mathbf{u} = -f^{\mathrm{TS}}(\mathbf{x}, \mathbf{a}_k, \mathbf{f}_k, \mathbf{r}_k)\mathbf{x}. \tag{18}$$

Thus, in the end, we have an infinite number of different solutions by varying the antecedents fuzzy sets of the T–S fuzzy model we started from. This demonstrates that the design technique may be very sensitive to the shape of the antecedent fuzzy sets. This is in full accordance with the conclusion drawn from the sum of squares (SOS) based control design theories introduced by Tanaka *et al.* [19], [20].

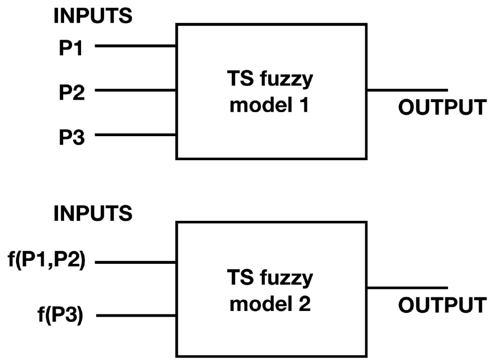

Fig. 1. Input space variations of the same model.

This optimization capability behind the TP model transformation was utilized in a number of publications solving engineering tasks. A summary is given in [6]. [21]–[23] revisit this optimization possibility and show in a comprehensive and detailed analysis that transforming the shape of the antecedents via the TP model transformation strongly influences the entire design process, and a suboptimal choice of antecedents can lead to a bad solution, while a suitable selection of antecedents can lead to a very good solution under the same design process. These results are summarized in [12]. Finally, it was concluded that it is necessary to manipulate the shape of the antecedent fuzzy sets when pursuing the best control solution.

B. Novel Contribution of this Article

The abovementioned points highlight the fact that both complexity reduction and control design theories motivate the further development of the TP model transformation. Previous variants of the TP model transformation focus on how to manipulate the antecedent-consequent fuzzy set system. This article, in contrast, focuses on how to manipulate the input space, and extends the TP model transformation with a further capability to transform the given function or T–S fuzzy model to a different combination of inputs. Of special interest is the case when the number of inputs can be decreased or increased to achieve considerable advantages in complexity reduction and control design. This article picks up the key idea discussed in the Section V and example 4 of [16]. That key idea was discussed in the context of dynamic models from which the LPV structure is extracted. This article develops this idea further and revises the entire TP model transformation to incorporate this feature at a generic level, such that it can be executed in a wider spectrum of problems. Fig. 1 shows an example of an alternative T–S fuzzy model in which the number of inputs is changed and the inputs are functions (combinations) of the original inputs. It is important to emphasize that both T–S fuzzy models represent the same mapping between inputs and outputs.

This article shows that the proposed transformation can remove the nonlinearity from the T–S fuzzy models, by defining new inputs, which are functions of the original ones. This leads to a considerable simplification of the given T–S fuzzy model. For example, a well-selected input space can even linearize a T–S fuzzy model or, on the other hand, it can even embed various nonlinearities depending on what is required. A removal of nonlinearity can decrease the rank of the T–S fuzzy model (see the HOSVD-based canonical form of T–S fuzzy models) that helps perform additional tradeoffs between the number of inputs, complexity, and accuracy.

The examples in this article consider complex qLPV models, such that the T–S fuzzy model represents the parameter varying system matrix and the inputs of the T–S fuzzy model are the parameters. This article shows that selecting an alternative parameter space leads to a completely different T–S fuzzy model (representing the same dynamics) and, hence, leads to a different design and controller.

C. Related Literature

Most recently, a very powerful Nested TP model transformation was developed in [24] to derive multilevel TP model structures. Very effective convex hull manipulation methods were incorporated into the TP model transformation in [11]. Computational analyses and improvements to the original formulation were proposed in [7] and [8]. TP model transformation based novel control approaches and applications were published in [25]–[32]. New design theories were also introduced in sliding mode control in [27], [33], and [34]. A method to transform time delayed systems to nontime delayed models, where the time delay is an external parameter was proposed in [35]. For further key applications readers are referred to [36]–[69]. Most recent results are published in [69]–[80].

D. Structure of this Article

The rest of this article is organized as follows. Section II defines the notation and the basic concepts of the TP model transformation. Section III presents the new extension of the TP model transformation. Section IV gives detailed examples to help readers to apply the proposed method. Section V presents an additional example. Finally, Section VI concludes this article.

II. Notations and definitions

This section provides the notations and definitions used in this article.

1) Indices: $i, j, \ldots$ the upper bounds of the indices are denoted by the uppercase letter, e.g., $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$ or $i_n = 1, 2, \ldots, I_n$, where $n = 1, 2, \ldots N$.
2) Scalar: $a \in \mathbb{R}$.
3) Vector: $\mathbf{a} \in \mathbb{R}^I$ contains elements $a_i \in \mathbb{R}$.
4) Matrix: $\mathbf{A} \in \mathbb{R}^{I\times J}$ contains elements $a_{i,j} \in \mathbb{R}$.
5) Tensor: $\mathcal{A} \in \mathbb{R}^{I\times J\times K\times\cdots}$ has elements $a_{i,j,k,\cdots} \in \mathbb{R}$.
6) $\mathbb{R}^{I^N}$ is brief notation of $\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. For instance $\mathbb{R}^{I^N\times O^K}$ denotes $\mathbb{R}^{I_1\times I_2\times\cdots\times I_N\times O_1\times O_2\times\cdots\times O_K}$.
7) Interval: $\omega \subset \mathbb{R}$ is bounded as $\omega = [\omega_{\min}, \omega_{\max}]$.
8) Space: $\Omega \subset \mathbb{R}^N$ is an N dimensional bounded space as $\Omega = \omega_1 \times \omega_2 \times \cdots \times \omega_N$.
9) Ω^p is the space of vector $\mathbf{p} \in \Omega^p \subset \mathbb{R}^N$.

Definition 1: Hyper rectangular grid $G(\Omega)$.

$G(\Omega)$ denotes a hyper rectangular grid defined over space $\Omega \subset \mathbb{R}^N$, where G_n is the number of grids defined on each interval ω_n of the space as

$$[g_{n,1} < g_{n,2} < \cdots < g_{n,G_n}] \quad (19)$$

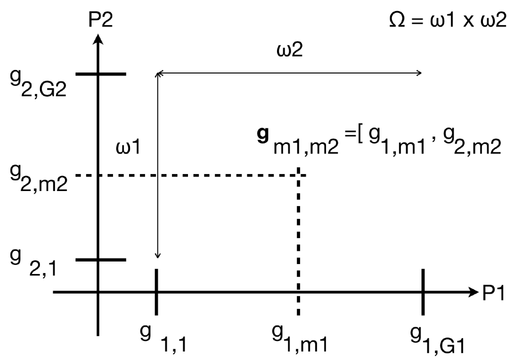

Fig. 2. Defining a grid.

where $\forall n : g_{n,1} = \omega_{n,\min}, g_{n,G_n} = \omega_{n,\max}$. Here $\omega_{n,\min}$ denotes the minimum value of ω_n on dimension n, while $\omega_{n,\max}$ denotes the maximum value. The size or density of the grid is denoted by $G^N = G_1 \times G_2 \cdots \times G_N$. See Fig. 2.

Definition 2: Grid vector $\mathbf{g}$.

Grid vectors $\mathbf{g}$ define the coordinates of the points of the hyper rectangular grid $G(\Omega)$ as $\mathbf{g}_{m_1,m_2,\ldots,m_N} = [g_{1,m_1}\ g_{2,m_2}\ \ldots\ g_{N,m_N}] \in \mathbb{R}^N$, $m_n = 1, 2, \ldots, G_n$.

Definition 3: Grid tensor $\mathcal{G}$.

Grid tensor $\mathcal{G} \in \mathbb{R}^{G^N \times N}$, stores the grid vectors $\mathbf{g}_{m_1,m_2,\ldots,m_N}$. Namely each entry of the tensor determines the corresponding coordinates of the hyper rectangular grid.

Definition 4: Grid operation $f(*\mathcal{G})$.

Assume a given function $\mathcal{Y} = f(\mathbf{p}) \in \mathbb{R}^{O^K}$, $\mathbf{p} \in \Omega^p$ and grid tensor $\mathcal{G} \in \mathbb{R}^{G^N \times N}$ of grid $G(\Omega^p)$ with grid density G^N. The grid operation defines

$$\mathcal{B} = f(*\mathcal{G}) \tag{20}$$

where tensor $\mathcal{B} \in \mathbb{R}^{G^N \times O^K}$ contains elements

$$\mathcal{Y}_{m_1,m_2,\ldots,m_N} = f(\mathbf{g}_{m_1,m_2,\ldots,m_N}) \tag{21}$$

where $\mathbf{g}_{m_1,m_2,\ldots,m_N}$, are the elements of grid tensor $\mathcal{G}$.

Definition 5: Grid model $\mathcal{F}^G$.

The grid model $\mathcal{F}^G$ of function $f(\mathbf{p})$, $\mathbf{p} \in \Omega^p$, is a tensor resulted by the grid operation

$$\mathcal{F}^{\mathcal{G}} = f(*\mathcal{G}) \tag{22}$$

where $\mathcal{G}$ is the grid tensor of grid $G(\Omega^p)$.

Definition 6: TP structure.

Any tensor $\mathcal{S} \in \mathbb{R}^{M^N \times O^K}$ can be given as a tensor product such as

$$\mathcal{S} = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \mathbf{U}_n \tag{23}$$

where $\mathcal{B} \in \mathbb{R}^{I^N \times O^K}$ termed as core tensor and $\mathbf{U}_n \in \mathbb{R}^{M_n \times I_n}$ are the weighting matrices. The column vectors $\mathbf{u}_{n,i_n}$ of matrix $\mathbf{U}_n$ are termed as weighting vectors

$$\mathcal{S} = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \left[\mathbf{u}_{n,1}\ \mathbf{u}_{n,2}\ \ldots\ \mathbf{u}_{n,I_n}\right]. \tag{24}$$

If the tensor product in (23) is derived by HOSVD then it is termed as HOSVD-based TP structure [4].

Definition 7: TP model.

The TP model is a continuous variant of the TP structure. Here, instead of weighting vectors we have weighting functions as

$$\mathcal{S}(\mathbf{p}) = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \left[w_{n,1}(p_n)\ w_{n,2}(p_n)\ \ldots\ w_{n,I_n}(p_n)\right] \tag{25}$$

that is

$$\mathcal{S}(\mathbf{p}) = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \mathbf{w}_n(p_n) \tag{26}$$

where the vector of the weighting function is $\mathbf{w}_n(p_n) = [w_{n,1}(p_n)\ w_{n,2}(p_n)\ \ldots\ w_{n,I_n}(p_n)] \in \mathbb{R}^{I_n}$ and $\mathcal{S}(\mathbf{p}) \in \mathbb{R}^{O^K}$, $\mathbf{p} \in \mathbb{R}^N$ and core tensor $\mathcal{B} \in \mathbb{R}^{I^N \times O^K}$ contains the vertexes $\mathcal{S}_{i_1,i_2,\ldots,i_N} \in \mathbb{R}^{O^K}$.

Remark 1: T–S fuzzy model versus TP model

This article focuses on the widely applied type of the T–S fuzzy models whose transfer function is equivalent with the TP model. The transfer function of the typical T–S fuzzy model belongs to the class of tensor product functions, hence, TP models see [1], [2], [6], [12], [17]. Consider the following fuzzy rule structure:

$$\textbf{IF } \mathbf{A_{1,i_1}} \textbf{ AND} \ldots \textbf{AND } \mathbf{A_{N,i_N}} \textbf{ THEN } \mathbf{B_{i_1,\ldots,i_N}}. \tag{27}$$

If the observations are singleton fuzzy sets with the element of p_n, and the consequent $B_{i_1,\ldots,i_N}$ are also singleton fuzzy sets having elements $b_{i_1,\ldots,i_N}$, and we apply product-sum-gravity inference then the transfer function of the T–S fuzzy model takes the form of

$$\mathcal{S}(\mathbf{p}) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n) b_{i_1,i_2,\ldots,i_N} \tag{28}$$

where functions $w_{n,i_n}(p_n)$ are the membership functions of the antecedent sets A_{n,i_n} and the elements of the consequents are $b_{i_1,i_2,\ldots,i_N}$. In case of state space system modeling the consequents are system matrices $\mathbf{B}_{i_1,i_2,\ldots,i_N} \in \mathbb{R}^{O_1 \times O_2}$, which also termed as vertex systems. In case of higher structures, where for instance the consequents are polynomials, the consequents can be a tensors of the coefficient parameters such as $\mathcal{B}_{i_1,i_2,\ldots,i_N} \in \mathbb{R}^{O^K}$. This transfer function in (28) belongs to a class of TP models and it takes a more compact form using tensor operations as

$$\mathcal{S}(\mathbf{p}) = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \mathbf{w}_n(p_n). \tag{29}$$

Here, core tensor $\mathcal{B}$ stores the vertices $\mathcal{B}_{i_1,i_2,\ldots,i_N}$ and the weighting functions in $\mathbf{w}_n(p_n)$ represent the membership functions of the antecedent fuzzy sets. When the TP model represents a T–S fuzzy models then all weighting functions $\forall n, i, p : w_{n,i}(p) \in [0, 1]$. Further discussion about how the TP model is utilized as the transfer function of the most spread variants of the T–S fuzzy model are on [6, p. 18], in [12, Ch. 1.4], and [17, Ch. 2.1] and also be referred to [1], [2].

III. New Extension of the TP Model Transformation

This section introduces the proposed extension of the TP model transformation. Assume a given function

$$\mathcal{S}(\mathbf{p}) \in \mathbb{R}^{O^K},\ \ \mathbf{p} \in \Omega^p \subset \mathbb{R}^N. \tag{30}$$

Executing the previous variants of the TP model transformation results in

$$\forall \mathbf{p} : \mathcal{S}(\mathbf{p}) = \mathcal{S} \mathop{\boxtimes}_{n=1}^{N} \mathbf{w}_n(p_n). \tag{31}$$

Here, we have the opportunity to vary the number and shape of the antecedent fuzzy sets, hence, the consequents.

The proposed extension goes further and is capable of transforming to an alternative input space $\mathbf{b} \in \Omega^b \subset \mathbb{R}^M$, where the relation between $\mathbf{p}$ and $\mathbf{b}$ is defined. Thus, the expected result is

$$\forall \mathbf{p} : \mathcal{S}(\mathbf{p}) = \mathcal{T}(\mathbf{b}) = \mathcal{T} \mathop{\boxtimes}_{m=1}^{M} \mathbf{v}_m(b_m) \tag{32}$$

where the input space $\mathbf{p} \in \Omega^p$ is replaced by $\mathbf{b} \in \Omega^b$.

1) *Step 1:* Determination of the grid model $\mathcal{T}^{\mathcal{G}}$.
The goal of the first step is to determine grid model $\mathcal{T}^{\mathcal{G}}$ over grid $G(\Omega^b)$. The challenge here is that $\mathcal{T}(\mathbf{b})$ is unknown, but $\mathcal{S}(\mathbf{p})$ can be calculated for any $\mathbf{p} \in \Omega^p$. Further difficulty is that the inner formulas of the elements of $\mathcal{S}(\mathbf{p})$ may not be given.
This step has two different ways depending on whether $\mathbf{p} = f(\mathbf{b})$ or $b_n = f(p_n)$ is known or at least can be calculated over the given grid.

a) $\mathbf{p} = f(\mathbf{b})$
Define grid $G(\Omega^b)$, with density G^M that leads to grid tensor $\mathcal{G}^b \in \mathbb{R}^{G^M \times M}$. Then, transform the grid tensor to the original parameter space as

$$\mathcal{G}^p = f(*\mathcal{G}^b) \tag{33}$$

where $\mathcal{G}^p \in \mathbb{R}^{G^N \times N}$. Then, the grid model is calculated as

$$\mathcal{T}^{\mathcal{G}} = \mathcal{S}(*\mathcal{G}^p) = \mathcal{S}(*f(*\mathcal{G}^b)) \tag{34}$$

where $\mathcal{T}^{\mathcal{G}} \in \mathbb{R}^{G^M \times O^K}$. Then, we assume that we have the grid model $\mathcal{T}^{\mathcal{G}}$ of the unknown $\mathcal{T}(\mathbf{b}) \in \mathbb{R}^{O^K}$, thus, we know the left side of

$$\mathcal{T}^{\mathcal{G}} = \mathcal{T}(*\mathcal{G}^b). \tag{35}$$

b) $b_n(t) = f(p_n(t))$
Here, the size of the vector $\mathbf{b}(t)$ and $\mathbf{p}(t)$ are equal, as $M = N$. Define hyper rectangular grid $G(\Omega^p)$ as

$$[p_{n,1} < p_{n,2} < \cdots < p_{n,G_n}]. \tag{36}$$

Then, define the grid $G(\Omega^b)$ as

$$[b_{n,1} \le b_{n,2} \le \cdots \le b_{n,G_n}] \tag{37}$$

where $b_{n,i} = f(p_{n,j})$ are arranged into increasing order and $i, j = 1, \ldots, G_n$. This defines grid tensor G^b. Then, rearrange (36) as

$$\left[p_{n,1}\ p_{n,2}\ \ldots\ p_{n,G_n}\right] \tag{38}$$

where the ordering satisfy $b_{n,i} = f(p_{n,i})$. Then, grid tensor $\mathcal{G}^p \in \mathbb{R}^{G^N \times N}$ is constructed from the grid vectors defined in (38) based on Definition 2

$$\mathbf{g}_{m_1,m_2,\ldots,m_N} = \left[p_{1,m_1}\ p_{2,m_2}\ \ldots\ p_{N,m_N}\right] \tag{39}$$

where $m_n = 1, 2, \ldots, G_n$. Then, the grid model is calculated as

$$\mathcal{T}^{\mathcal{G}} = \mathbf{S}(*\mathcal{G}^p). \tag{40}$$

Then, we assume that we have the grid model $\mathcal{T}^{\mathcal{G}}$ of the unknown $\mathcal{T}(\mathbf{b}(t))$, thus, we know the left side of

$$\mathcal{T}^{\mathcal{G}} = \mathcal{T}(*\mathcal{G}^b) \tag{41}$$

where $\mathcal{T}^{\mathcal{G}} \in \mathbb{R}^{G^N \times O^K}$. Since $M = N$, then $\mathcal{T}^{\mathcal{G}} \in \mathbb{R}^{G^M \times O^K}$. The grid tensor $\mathcal{G}^b \in \mathbb{R}^{G^N \times N}$ is defined in the same way from (37).

2) *Step 2:* Extraction of the TP structure
This step has two key steps

a) Complexity tradeoff
Executing HOSVD on $\mathcal{T}^{\mathcal{G}}$ while discarding all zero singular values results in

$$\mathcal{T}^{\mathcal{G}} = \mathcal{T} \mathop{\boxtimes}_{m=1}^{M} \mathbf{U}_m \tag{42}$$

where the minimal number of vertices is obtained. Thus, we have core tensor $\mathcal{T} \in \mathbb{R}^{R^M \times O^K}$, where R_m ($R^M = R_1 \times R_2 \times \cdots \times R_M$) equals the rank of dimension m of $\mathcal{T}^{\mathcal{G}}$. The size of $\mathbf{U}_m$ is $G_m \times R_m$. Approximation tradeoffs can also be performed if necessary by discarding nonzero singular values [6].

b) Vary the convexity of the structure.
An important benefit of this step is that we can rather easily tune the convex hull defined by vertexes by transforming matrices $\mathbf{U}_m$ to $\mathbf{H}_m$ satisfying specific conditions, see, e.g., the nonnegativeness, sum normalization (SN), normalized, close to normal (CNO), inverse normalization, and various further transformations in [6] and [9]. The transformation may change the number of columns to I_n (the SN transformation may add one more column), therefore, let the size of $\mathbf{H}_m$ be $G_m \times I_m$. Then, we arrive at

$$\mathcal{T}^{co} = \mathcal{T}^{\mathcal{G}} \mathop{\boxtimes}_{m=1}^{M} (\mathbf{H}_m)^{+} \tag{43}$$

where $\mathcal{T}^{co} \in \mathbb{R}^{I^M \times O^K}$. Thus,

$$\mathcal{T}^{\mathcal{G}} = \mathcal{T}^{co} \mathop{\boxtimes}_{m=1}^{M} \mathbf{H}_m \tag{44}$$

where superscript "co" indicates that the vertexes form the desired type of convex hull.

3) Determination of the weighting functions.

Column vectors $\mathbf{h}_{m,i}$ of matrices $\mathbf{H}_m$ determine the weighting functions $v_{m,i}(b_m)$ over the grid $G(\Omega^b)$, where the grid is defined, per dimension, from grid tensor $\mathcal{G}^b$ as $[b_{m,1}\ b_{m,2}\ \ldots\ b_{m,G_m}]$. Thus, the elements of vector $\mathbf{v}_m(b_m)$ over the grid are $v_{m,i}(b_{m,j}) = h_{m,i,j}$, where $h_{m,i,j}$ is the jth element of vector $\mathbf{h}_{m,i}$, and $j = 1, 2, \ldots, G_m$. There are two ways to reconstruct the continuous weighting functions. The simplest way is to connect the elements of one column by straight lines to derive piecewise linear approximations of the weighting functions (see Fig. 3). This is known as the bilinear TP model [12]. If the grid is dense enough, then the bilinear approximation leads to a good quality model. To increase the grid density without executing the TP models transformation, the enriched TP model transformation was published in [12]. The other way is to determine the exact weighting functions from the columns vectors of $\mathbf{H}_m$ see [6].

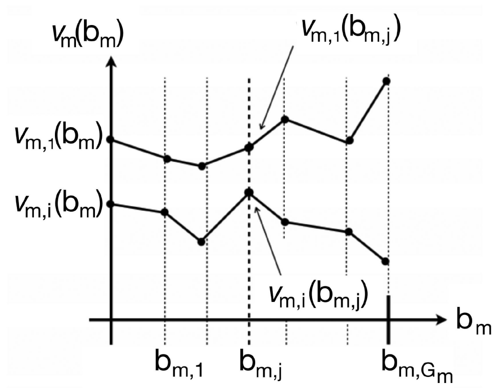

Fig. 3. Piecewise linear weighting functions.

IV. Example 1

The previous variants of the TP model transformation and its effectiveness in real word engineering control design were extensively investigated on the 2DoF and 3DoF model of the aeroelastic wing section [21]–[23], [68], [81], [82]. Therefore, in this section, we continue this series of examples. For the detailed description of this complex physical model, readers are referred to [68], [81], [82].

A. Model of the Aeroelastic Wing Section

The state-space model of the two-dimensional aeroelastic wing section has state vector $\mathbf{x} \in \mathbb{R}^4$ as

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} h(t) \\ \alpha(t) \\ \dot{h}(t) \\ \dot{\alpha}(t) \end{bmatrix} \tag{45}$$

where $x_1(t)$ is the plunging displacement and $x_2(t)$ is the pitching displacement. The state-space model has the form of

$$\dot{\mathbf{x}}(t) = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \tag{46}$$

where parameter vector $\mathbf{p}(t) \in \mathbb{R}^2$ has elements

$$\mathbf{p}(t) = \begin{bmatrix} U(t) \\ x_2(t) \end{bmatrix}. \tag{47}$$

Here, free stream velocity $U(t)$ is an external parameter. The entries of the system matrix are

$$\mathbf{S}(\mathbf{p}(t)) = \left[\mathbf{S}_1(\mathbf{p}(t))\ \mathbf{S}_2(\mathbf{p}(t))\right] \tag{48}$$

where

$$\mathbf{S}_1(\mathbf{p}(t)) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -k_1 & -k_2U^2(t) - p(k_\alpha(x_2(t))) \\ -k_3 & -k_4U^2(t) - q(k_\alpha(x_2(t))) \end{bmatrix} \tag{49}$$

$$\mathbf{S}_2(\mathbf{p}(t)) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_1(U(t)) & -c_2(U(t)) & g_3U^2(t) \\ -c_3(U(t)) & -c_4(U(t)) & g_4U^2(t) \end{bmatrix} \tag{50}$$

where

$$d = m(I_\alpha - mx_\alpha^2 b^2), \quad z = \frac{1}{2} - a, \quad k_1 = \frac{I_\alpha k_h}{d} \tag{51}$$

$$k_2 = \frac{I_\alpha \rho b c_{l_\alpha} + m x_\alpha b^3 \rho c_{m_\alpha}}{d}, \quad k_3 = \frac{-m x_\alpha b k_h}{d} \tag{52}$$

$$k_4 = \frac{-m x_\alpha b^2 \rho c_{l_\alpha} - m \rho b^2 c_{m_\alpha}}{d} \tag{53}$$

$$p(k_\alpha(x_2(t))) = \frac{-m x_\alpha b}{d} k_\alpha(x_2(t)) \tag{54}$$

$$q(k_\alpha(x_2(t))) = \frac{m}{d} k_\alpha(x_2(t)) \tag{55}$$

$$k_\alpha(x_2(t)) = 2.82(1 - 22.1x_2(t) + 1315.5x_2^2(t) + 8580x_2^3(t) + 17289.7x_2^4(t)) \tag{56}$$

$$c_1(U(t)) = (I_\alpha(c_h + \rho U(t) b c_{l_\alpha}) \tag{57}$$

$$+ m x_\alpha \rho U(t) c_{m_\alpha})/d \tag{58}$$

$$c_2(U(t)) = (I_\alpha \rho U(t) b^2 c_{l_\alpha}(1/2 - a) - m x_\alpha b c_\alpha \tag{59}$$

$$+ m x_\alpha \rho U(t) b^4 c_{m_\alpha}(1/2 - a))/d \tag{60}$$

$$c_3(U) = (-m x_\alpha b c_h - m x_\alpha \rho U(t) b^2 c_{l_\alpha} \tag{61}$$

$$- m \rho U(t) b^2 c_{m_\alpha})/d \tag{62}$$

$$c_4(U) = (m c_\alpha - m x_\alpha \rho U(t) b^3 c_{l_\alpha}(1/2 - a) \tag{63}$$

$$- m \rho U(t) b^3 c_{m_\alpha}(1/2 - a))/d \tag{64}$$

$$g_3 = (-I_\alpha \rho b c_{l_\beta} - m x_\alpha b^3 \rho c_{m_\beta})/d \tag{65}$$

$$g_4 = (m x_\alpha b^2 \rho c_{l_\beta} + m \rho b^2 c_{m_\beta})/d. \tag{66}$$

Here, $a = -0.673$, $b = 0.135$, $k_h = 2844.4$, $c_h = 27.43$, $c_\alpha = 0.036$, $\rho = 1.225$, $c_{l_\alpha} = 6.28$, $c_{l_\beta} = 3.358$, $c_{m_\alpha} = (0.5 + a) * c_{l_\alpha}$, $m = 12.387$, $c_{m_\beta} = -0.635$, $x_\alpha = -0.3533 - a$, and $I_\alpha = 0.065$. In order to study the proposed method, a simple MATLAB code of the key steps is given here. The TPtool MATLAB toolbox is needed to execute the presented codes. This toolbox can be downloaded, for instance, on the Wikipedia site on the topic of "TP model transformation in control theory".

First of all, define a function

```
function S=aero1(U,x2)
```

that returns the systems matrix **S** in (48) for a given value of U and x_2. Then, define the TP model transformation resulting in a CNO type antecedent fuzzy system as

```
  function [B,Ucno]=TPmodel(S)
[Sc,U,sv]=hosvd(S,1e-9)
    dim=ndims(S)-2
    for i=1:dim
     Ucno{i}=genhull(U{i},'cno')
     Up{i}=pinv(Ucno{i})
    end
    B=tprods(S,Up).
```

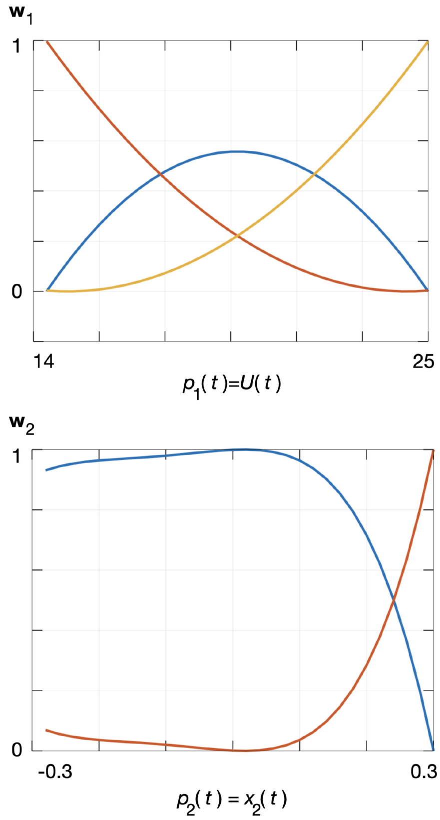

Fig. 4. Membership functions of T–S fuzzy model 1.

B. T–S Fuzzy Model 1

Assume that the inner formulas of the system matrix $\mathbf{S}(\mathbf{p}(t))$ are unknown and $\mathbf{p}(t) \in \Omega^p$. Further assume that the system matrix can be discretized over parameter space $\Omega^p = [14, 25] \times [-0.3, 0.3] \subset \mathbb{R}^2$. This wind speed interval is crucial since the limit cycle oscillation and the chaotic behavior occurs here [68], [81], [82]. Let the discretization density be $G^2 : 137 \times 137$. Executing the TP model transformation results in

$$\mathbf{S}(\mathbf{p}(t)) = \mathcal{S} \mathop{\boxtimes}_{n=1}^{2} \mathbf{w}_n(p_n(t)) \tag{67}$$

where $\mathbf{w}_1(p_1(t)) \in \mathbb{R}^3$ and $\mathbf{w}_2(p_2(t)) \in \mathbb{R}^2$. The membership functions are given in Fig. 4. The same functions are generated in papers [68], [81], [82]. The number of the vertices are $3 \times 2 = 6$. The MATLAB code is

```
U(:)=linspace(14,25,137)
x2(:)=linspace(-0.3,0.3,137)
for m1=1:137 for m2=1:137
  S(m1,m2,:,:)=aero1(U(m1),x2(m2))
end end
[B,Uc]=TPmodel(S)
plot(U(:),Uc{1}); plot(x2(:),Uc{2}).
```

C. T–S Fuzzy Model 2

Assume again that we do not know the inner formulas of the parameter varying system matrix. However, we know that $x_2(t)$ influences the system matrix as given in (56). Applying the proposed method we can transform the parameter space to

$$\mathbf{p}(t) = \begin{bmatrix} U(t) \\ k_\alpha(x_2(t)) \end{bmatrix} \in \Omega^p. \tag{68}$$

Let us emphasize here again that only $\mathbf{S}(U(t), x_2(t))$ is available for discretization, since the new functions of the system $\mathbf{T}(\mathbf{p}(t))$ are unknown. The transformation results in

$$\mathbf{S}(U(t), x_2(t)) = \mathbf{T}(\mathbf{p}(t)) = \mathcal{T} \mathop{\boxtimes}_{n=1}^{2} \mathbf{w}_n(p_n(t)) \tag{69}$$

where $\mathbf{w}_1(p_1(t)) \in \mathbb{R}^3$ and $\mathbf{w}_2(p_2(t)) \in \mathbb{R}^2$. The membership functions of p_1 is the same as in case of T–S fuzzy model 1. The membership functions are depicted on Fig. 5. The number of the vertices are $3 \times 2 = 6$, like in the case of TP model 1. However a very important difference is that the nonlinear behavior of the weighting functions is removed form the second dimension. Thus, the dimension of x_2 is transformed. The modified key points in the MATLAB code are

```
for i=1:137 x2b(i)=kalp(x2(i))
```

where `kalp(.)` is the function of $k_\alpha(x_2(t))$, see (56). The rest of the code is

```
[x2b(:),I]=sort(x2b(:))
for m1=1:137 for m2=1:137
  mp2=I(m2)
  S(m1,m2,:,:)=aero1(U(m1),x2(mp2))
```

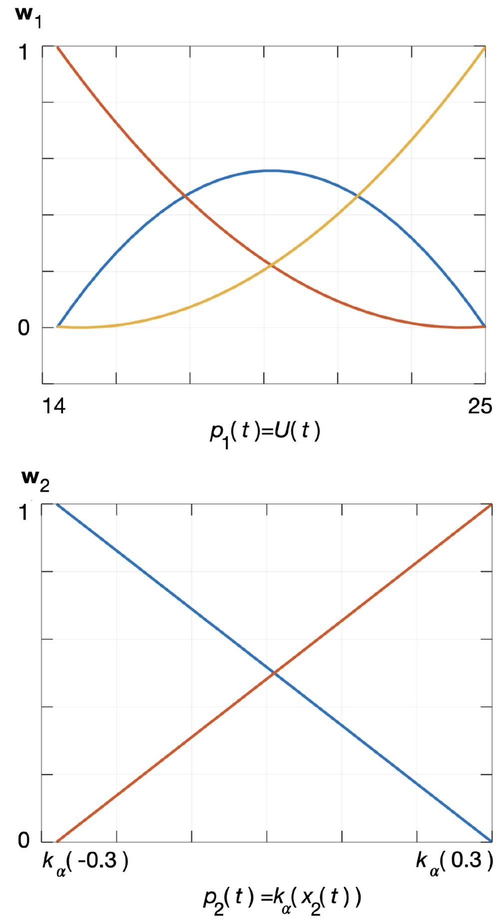

Fig. 5. Membership functions of T–S fuzzy model 2.

```
end end
[B,Uc]=TPmodel(S)
plot(U(:),Uc{1}); plot(x2b(:),Uc{2}).
```

D. T–S Fuzzy Model 3

Assume that we still do not know the inner formulas of the system matrix, but we have the entries of parameter $U^2(t)$, see (49) and (50). Therefore, we may have the chance to decrease the rank of the T–S fuzzy model representation on the related dimensions by introducing a new parameter $U^2(t)$. Thus, we have the new parameter space as

$$\mathbf{p}(t) = \begin{bmatrix} U(t) \\ U^2(t) \\ x_2(t) \end{bmatrix} \in \Omega^p. \quad (70)$$

Although $U(t)$ and $U^2(t)$ do not really define two dimensions as they are directly dependent, the rank of these dimensions can be decreased or even linearised, which would simplify the system description and may have additional benefits in further design, see the SOS design theories by Tanaka *et al.* [19], [20]. Executing the proposed method we arrive at

$$\forall \mathbf{p}(t) : \mathbf{S}(U(t), x_2(t)) = \mathbf{T}(\mathbf{p}(t)) = \mathcal{T} \boxtimes_{n=1}^{3} \mathbf{w}_n(p_n(t)) \quad (71)$$

where $\forall n : \mathbf{w}_n(p_n(t)) \in \mathbb{R}^2$. The resulting membership functions are depicted on Fig. 6. We can conclude that we have one more dimension, but rather simple weighting function systems on the first two dimensions. The number of the vertices are $2 \times 2 \times 2 = 8$.

E. T–S Fuzzy Model 4

This section combines results from the abovementioned two. Here, we transform to parameter space

$$\mathbf{p}(t) = \begin{bmatrix} U(t) \\ U^2(t) \\ k_\alpha(x_2(t)) \end{bmatrix} \in \Omega^p. \quad (72)$$

The transformation results in

$$\mathbf{S}(U(t), x_2(t)) = \mathbf{T}(\mathbf{p}(t)) = \mathcal{T} \boxtimes_{n=1}^{3} \mathbf{w}_n(p_n(t)) \quad (73)$$

where $\forall n : \mathbf{w}_n(p_n(t)) \in \mathbb{R}^2$. The number of vertices is $2 \times 2 \times 2 = 8$. We now have two extra vertices compared to T–S fuzzy model 1, however, we obtain the simplest T–S fuzzy model is obtained, as shown in Fig. 7.

V. Example 2

In this section, we focus on the translational oscillator with rotational actuator (TORA) system, which was a key example upon which the first variants of the TP model transformation were tested [6]. Those investigations focused primarily on the shape of the antecedents, hence, the convex hull defined by the vertexes, and the number of fuzzy rules. In the present investigation, we focus on the number of inputs as well. In this example, we use the same grid density and derive CNO type fuzzy sets as in Example 1.

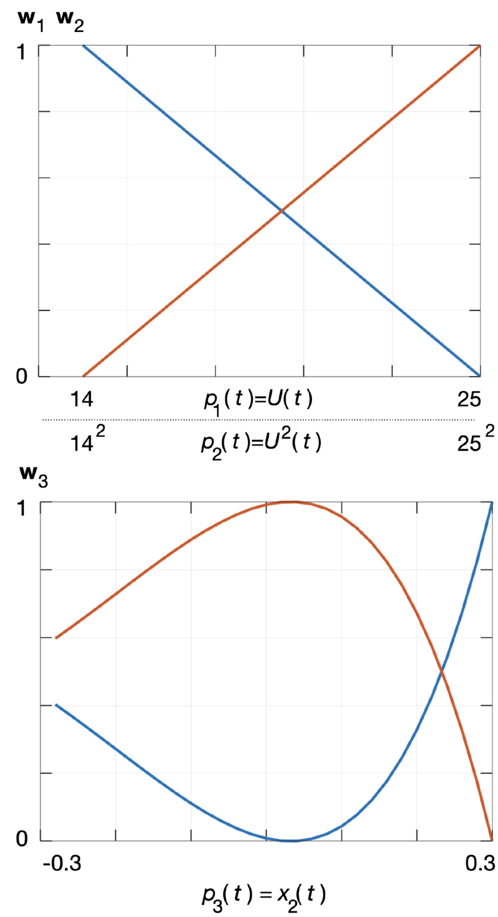

Fig. 6. Membership functions of T–S fuzzy model 3.

A. qLPV Model of the TORA

Assume the following qLPV model of the TORA system:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \quad (74)$$

where $\mathbf{x}(t), \mathbf{u}(t)$, and $\mathbf{y}(t)$ are the state, input, and output vectors, respectively. The system matrix $\mathbf{S}(\mathbf{p}(t))$ takes the form of

$$\begin{bmatrix} 0 & 1\ 0 & 0 & 0 \\ \frac{-1}{f(x_3(t))} & 0\ 0 & \frac{\rho x_4(t)\sin(x_3(t))}{f(x_3(t))} & \frac{-\rho\cos(x_3(t))}{f(x_3(t))} \\ 0 & 0\ 0 & 1 & 0 \\ \frac{\rho\cos(x_3(t))}{f(x_3(t))} & 0\ 0 & \frac{-\rho x_4(t)\sin(x_3(t))}{f(x_3(t))} & \frac{1}{f(x_3(t))} \end{bmatrix} \quad (75)$$

where

$$f(x_3(t)) = 1 - \rho^2\cos^2(x_3(t)) \quad (76)$$

and $p_1(t) = x_3(t)$ amd $p_2(t) = x_4(t)$.

B. T–S Fuzzy Model 1

This section simply executes the previous version of the TP model transformation on (75). Let $p_1(t) = x_3(t)$ and $p_2(t) = x_4(t)$. The space of the transformation is $\Omega = [-45°, 45°] \times [-45°, 45°]$. The number of the resulting rules is $5 \times 2 = 10$. The membership functions are depicted on Fig. 8. The resulting

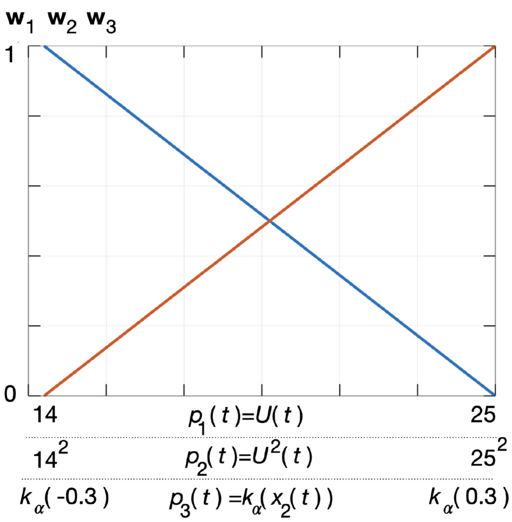

Fig. 7. Membership functions of T–S fuzzy model 4.

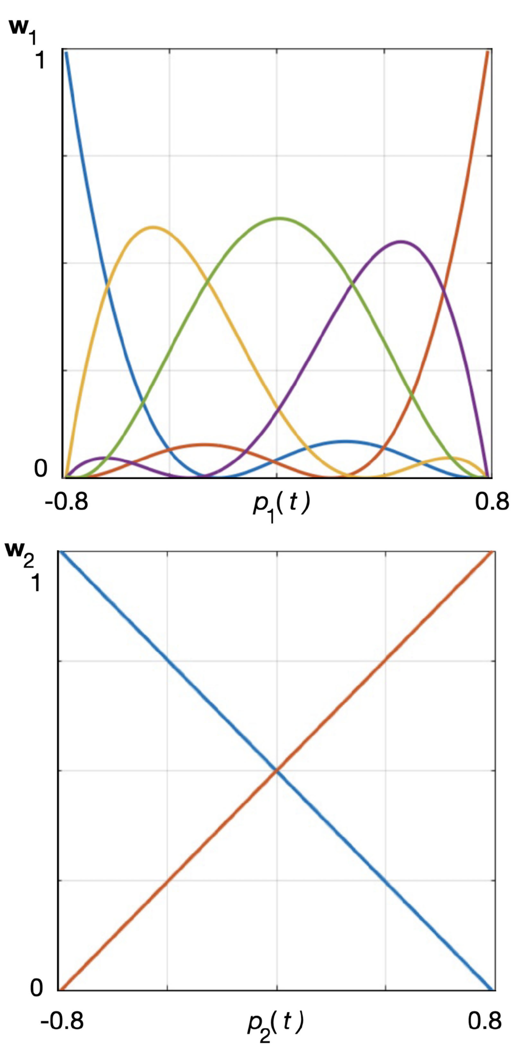

Fig. 8. Membership functions of T–S fuzzy model 1.

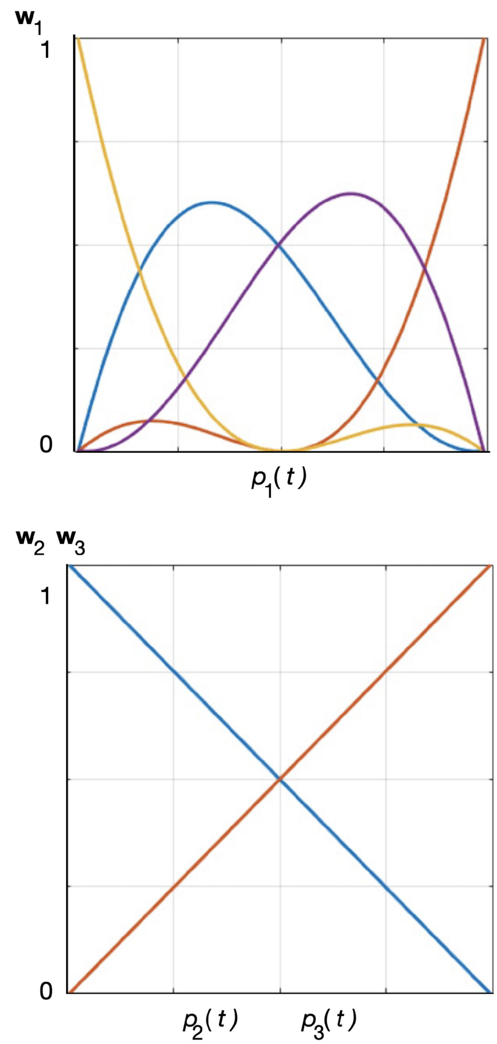

Fig. 9. Membership functions of T–S fuzzy model 2.

membership functions are the same as in the publication [6]. The problem here is that the membership function system of the first dimension is complex, making the manipulation of the convex hull defined by the vertexes challenging [6].

C. T–S Fuzzy Model 2

In order to decrease the nonlinear complexity of the model let us define the parameter space as $p_1(t) = x_3(t)$, $p_2(t) = x_4(t)$, $p_3(t) = \frac{1}{f(x_3(t))}$, where a new dimension is introduced. Let $\Omega = [-45°, 45°] \times [-45°, 45°] \times [1, 1.05]$. Executing the proposed TP model transformation, we obtain $4 \times 2 \times 2 = 16$ fuzzy rules. The resulting membership functions are depicted on Fig. 9. The rank, hence, the number of, and nonlinear complexity of the membership functions on the first dimension is considerably decreased. Thus, the convex hull manipulation leads to a more simple task and a more fine tuning of the convex hull can be performed. The price we have to pay is that a new dimension of the parameter space is introduced, which increases the total number of rules.

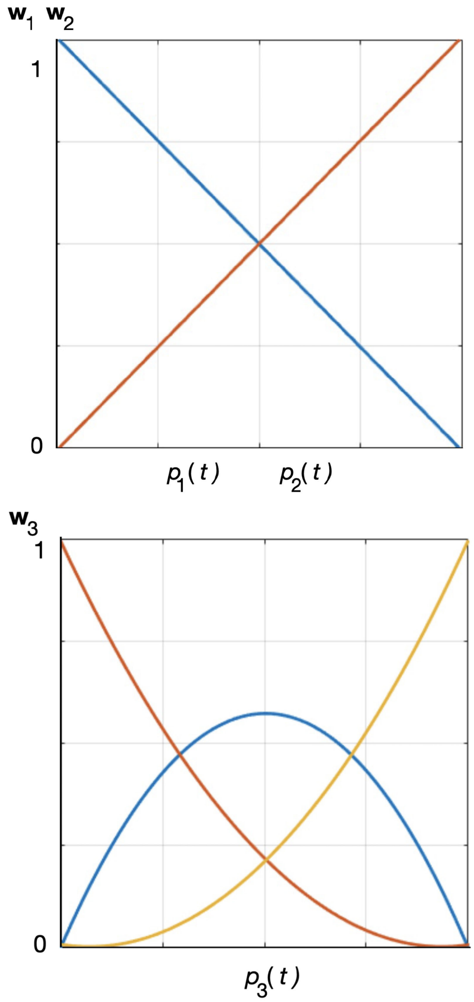

Fig. 10. Membership functions of T–S fuzzy model 3.

D. T–S Fuzzy Model 3

In order to further decrease the complexity of the first dimension, let us define the parameter space as $p_1(t) = \sin(x_3(t))$, $p_2(t) = x_4(t)$, and $p_3(t) = \cos(x_3(t))$. Thus, $\Omega = [-0.8, 0.8] \times [-45°, 45°] \times [0, 0.8]$. The resulting number of fuzzy rules is $2 \times 3 \times 2 = 12$. The membership functions are given on Fig. 10. We minimized the rank on the first dimension. However, the selected parameter on the third dimension introduces three membership functions. In contrast to T–S fuzzy model 2, we have achieved a reduction since the maximal number of the membership functions on any dimension is three. Even the simplest variant of the CNO transformations can be executed since a triangular fitting is needed only in the case of three membership functions. In conclusion, we have a considerably less complex model.

E. T–S Fuzzy Model 4

Let us define the parameter space as $p_1(t) = x_4(t)\sin(x_3(t))$, $p_2(t) = \frac{1}{f(x_3(t))}$ and $p_3(t) = \rho\cos(x_3(t))$, and $\Omega = [0, 0.6] \times [1, 1.05] \times [0.1, 0.2]$. Executing the proposed TP model transformation, we obtain only eight fuzzy rules, which is even less than in the case of T–S fuzzy model 1 (that was the minimum derived by the previous TP model transformation). The membership functions are given on Fig. 11. This is the simplest variant (both in number of rules and nonlinear complexity of the antecedents) in contrast to the T–S fuzzy models 1–3. Again, model 1 was the simplest possible variant using the previous TP model transformation. In case of further control design, the simplest LMI systems can be used since the vertexes are assigned to intervals only.

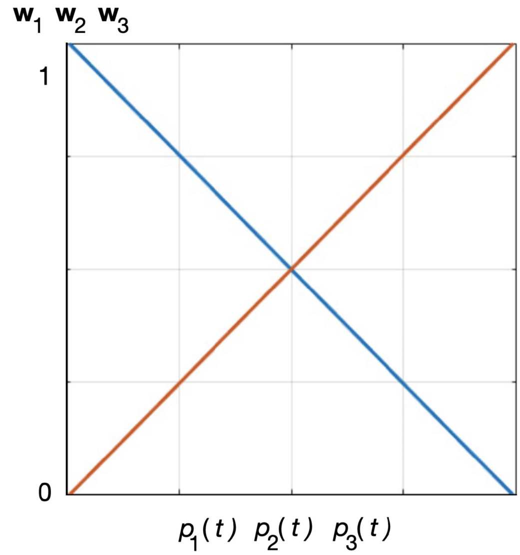

Fig. 11. Membership functions of T–S fuzzy model 4 and 5.

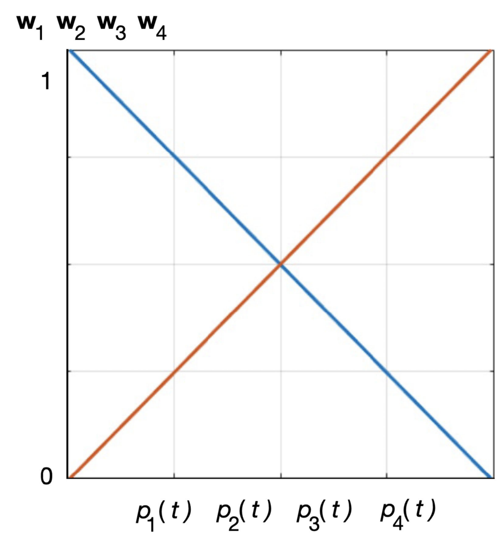

Fig. 12. Membership functions of T–S fuzzy model 6.

F. T–S Fuzzy Model 5

Let us define the parameter space as $p_1(t) = \frac{x_4(t)\sin(x_3(t))}{f(x_3(t))}$, $p_2(t) = \frac{1}{f(x_3(t))}$, and $p_3(t) = \cos(x_3(t))$. and $\Omega = [0, 0.6] \times [1, 1.05] \times [0.7, 1]$. Executing the proposed method the resulting number of fuzzy rules is eight again. The membership functions are given on Fig. 11. As a matter of fact, the consequents are different from T–S fuzzy model 4, hence, this will leads to alternative control solution.

G. T–S Fuzzy Model 6

Let us define the parameter space as $p_1(t) = \frac{x_4(t)\sin(x_3)(t)}{f(x_3(t))}$, $p_2(t) = \frac{1}{f(x_3(t))}$, $p_3(t) = \frac{\rho\cos(x_3(t))}{f(x_3(t))}$ and $p_4(t) = \frac{\rho^2 x_4(t)\cos(x_3(t))\sin(x_3(t))}{f(x_3(t))}$, and $\Omega = [0, 0.6] \times [1, 1.05, 0.1, 0.2] \times [0.7, 1] \times [0, 0.02]$. Executing the proposed TP model transformation the resulting number of fuzzy rules is 16. The membership functions are given on Fig. 12.

VI. Conclusion

This article showed that the proposed radically new extension of the TP model transformation is capable of transforming a given T–S fuzzy model to an alternative input space in which the intervals, number of inputs, and their nonlinear gains can be varied. Thus, the extended version can manipulate all components and parameters in the resulting T–S fuzzy model representation of a given function. The resulting T–S fuzzy model provided further opportunities for the complexity relaxation of the antecedents, decreasing the number of rules. Furthermore, convex hull manipulation can be performed for control design optimization. The proposed extension retains all the beneficial features of the previous TP model transformation. These new features considerably increased the capability of the TP model transformation to achieve flexible performance, and brought new aspects into the theories of complexity reduction and optimal control.

References

[1] Y. Yam, "Fuzzy approximation via grid point sampling and singular value decomposition," *IEEE Trans. Syst., Man, Cybern.*, vol. 27, no. 6, pp. 933–951, Dec. 1997.

[2] Y. Yam, P. Baranyi, and C. Yamg, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 120–132, Apr. 1999.

[3] P. Baranyi and Y. Yam, "Fuzzy rule base reduction," in *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications.*, D. Ruan and E. E. Kerre, Eds. Norwell, MA, USA: Kluwer, 2000, ch. 7, pp. 135–160.

[4] L. Szeidl and P. Várlaki, "HOSVD based canonical form for polytopic models of dynamic systems," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 13, no. 1, pp. 52–60, 2009.

[5] D. Tikk, P. Baranyi, and R. Patton, "Approximation properties of TP model forms and its consequences to TPDC design framework," *Asian J. Control*, vol. 9, no. 3, pp. 221–231, Oct. 2008.

[6] P. Baranyi, Y. Yam, and P. Várlaki, *Tensor Product Model Transformation in Polytpic Model Based Control* (ser. Automation and Control Engineering). Boca Raton, FL, USA: CRC Press, Mar. 2017.

[7] S. Nagy, Z. Petres, P. Baranyi, and H. Hashimoto, "Computational relaxed TP model transformation: Restricting the computation to subspaces of the dynamic model," *Asian J. Control*, vol. 11, no. 5, pp. 461–475, Sep. 2009.

[8] J. Cui, K. Zhang, and T. Ma, "An efficient algorithm for the tensor product model transformation," *Int. J. Control, Automat. Syst.*, vol. 14, pp. 1205–1212, Oct. 2016.

[9] P. Várkonyi, D. Tikk, P. Korondi, and P. Baranyi, "A new algorithm for RNO-INO type tensor product model representation," in *Proc. 9th IEEE Int. Conf. Intell. Eng. Syst.*, 2005, pp. 263–266.

[10] J. Kuti, P. Galambos, and P. Baranyi, "Minimal volume simplex (MVS) polytopic model generation and manipulation methodology for TP model transformation," *Asian J. Control*, vol. 19, no. 1, pp. 289–301, Aug. 2017.

[11] X. Liu, Y. Yu, Z. Li, H. H. C. Lu, and T. Fernando, "An efficient algorithm for optimally reshaping the TP model transformation," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 64, no. 10, pp. 1187–1191, Oct. 2017.

[12] P. Baranyi, *TP-Model Transformation Based Control Design Frameworks* (ser. Control Engineering). New York, NY, USA: Springer, Jul. 2016.

[13] P. Baranyi, "The generalized TP model transformation for T'S fuzzy model manipulation and generalized stability verification," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 934–948, Aug. 2014.

[14] P. Baranyi, "Extension of the multi-TP model transformation to functions with different numbers of variables," *Complexity*, vol. 2018, 2018, Art. no. 8546976, doi: 10.1155/2018/8546976.

[15] P. Baranyi, "TP model transformation as a way to LMI-based controller design," *IEEE Trans. Ind. Electron.*, vol. 51, no. 2, pp. 387–400, Apr. 2004.

[16] P. Baranyi, "Extracting LPV and qLPV structures from state-space functions: A TP model transformation based framework," *IEEE Trans. Fuzzy Syst.,* vol. 28, no. 3, pp. 499–509, Mar. 2020.

[17] K. Tanaka and H. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. Hoboken, NJ, USA: Wiley, 2001.

[18] C. Scherer and S. Weiland, "Linear matrix inequalities in control," Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands, 2000. [Online]. Available: http://www.cs.ele.tue.nl/sweiland/lmi.htm

[19] K. Tanaka, H. Yoshida, H. Ohtake, and H. O. Wang, "A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 911–922, Aug. 2009.

[20] K. Tanaka, H. Ohtake, T. Seo, M. Tanaka, and H. O. Wang, "Polynomial fuzzy observer designs: A sum-of-squares approach," *IEEE Trans. Syst., Man, Cybern., B*, vol. 42, no. 5, pp. 1330–1342, Oct. 2012.

[21] A. Szollosi and P. Baranyi, "Influence of the Tensor Product model representation of qLPV models on the feasibility of Linear Matrix Inequality," *Asian J. Control*, vol. 18, no. 4, pp. 1328–1342, Dec. 2015.

[22] A. Szollosi and P. Baranyi, "Improved control performance of the 3-DoF aeroelastic wing section: A TP model based 2D parametric control performance optimization," *Asian J. Control*, vol. 19, no. 2, pp. 450–466, Dec. 2016.

[23] A. Szollosi and P. Baranyi, "Influence of the Tensor Product Model Representation of qLPV Models on the Feasibility of Linear Matrix Inequality Based Stability Analysis," *Asian J. Control*, vol. 10, no. 1, pp. 531–547, Jul. 2017.

[24] Y. Yu, Z. Li, X. Liu, K. Hirota, X. Chen, and H. Iu, "A nested tensor product model transformation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 1, pp. 1–15, Jan. 2019.

[25] X. Liu, X. Xin, Z. Li, and Z. Chen, "Near optimal control based on the tensor-product technique," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 64, no. 5, pp. 560–564, May 2017.

[26] X. Liu, Y. Yu, Z. Li, and H. H. C. Lu, "Polytopic H_∞ filter design and relaxation for nonlinear systems via tensor product technique," *Signal Process., Elsevier*, vol. 127, pp. 191–205, Oct. 2016.

[27] G. Zhao, D. Wang, and Z. Song, "A novel tensor product model transformation-based adaptive variable universe of discourse controller," *J. Franklin Inst., Elsevier*, vol. 353, no. 17, Nov. 2016, pp. 4471–4499.

[28] T. Jiang and D. Lin, "Tensor product model-based gain scheduling of a missile autopilot," *Trans. Japan Soc. Aeronaut. Space Sci.*, vol. 59, no. 3, pp. 142–149, 2016.

[29] A. Hajiloo and W. F. Xie, "The stochastic robust model predictive control of shimmy vibration in aircraft landing gears," *Asian J. Control*, vol. 17, no. 2, pp. 476–485, Mar. 2015.

[30] V. C. S. Campos, F. O. Souza, L. A. B. Torres, and R. M. Palhares, "New stability conditions based on piecewise fuzzy lyapunov functions and tensor product transformations," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 784–760, Aug. 2013.

[31] S. Kuntanapreeda, "Tensor product model transformation based control and synchronization of a class of fractional-order chaotic systems," *Asian J. Control*, vol. 17, no. 2, pp. 371–280, Mar. 2015.

[32] R.-E. Precup, E. M. Petriu, M.-B. Radac, S. Preitl, L.-O. Fedorovici, and C.-A. Dragos, "Cascade control system-based cost effective combination of tensor product model transformation and fuzzy control," *Asian J. Control*, vol. 17, no. 2, pp. 381–391, 2015.

[33] G. Zhao, H. Li, and Z. Song, "Tensor product model transformation based decoupled terminal sliding mode control," *Int. J. Syst. Sci., Taylor Frances*, vol. 47, no. 8, pp. 1791–1803, Aug. 2014.

[34] P. Korondi, "Tensor product model transformation-based sliding surface design," *Acta Polytechnica Hungarica*, vol. 3, no. 4, pp. 23–35, 2006.

[35] P. Galambos and P. Baranyi, "TP-tau model transformation: A systematic modelling framework to handle internal time delays in control systems," *Special Issue TP Model Transformation-Based Control Syst. Des. Asian J. Control*, vol. 17, no. 2, pp. 486–496, Jan. 2015.

[36] T. Wang, W. Xie, G. Liu, and Y. Zhao, "Quasi-min-max model predictive control for image-based visual servoing with tensor product model transformation," *Asian J. Control*, vol. 17, no. 2, pp. 402–416, 2015.

[37] J. Kuti, P. Galambos, and Á. Miklós, "Output feedback control of a dual-excenter vibration actuator via qLPV model and TP model transformation," *Asian J. Control*, vol. 17, no. 2, pp. 432–442, 2015.
[38] J. Pan and L. Lu, "TP model transformation via sequentially truncated higher-order singular value decomposition," *Asian J. Control*, vol. 17, no. 2, pp. 467–475, 2015.
[39] J. Matuško, Š. Ileš, F. Kolonić, and V. Lešić, "Control of 3D tower crane based on tensor product model transformation with neural friction compensation," *Asian J. Control*, vol. 17, no. 2, pp. 443–458, 2015.
[40] S. Campos, V. Costa, L. A. B. Tôrres, and R. M. Palhares, "Revisiting the TP model transformation: Interpolation and rule reduction," *Asian J. Control*, vol. 17, no. 2, pp. 392–401, 2015.
[41] G. Zhao, K. Sun, and H. Li, "Tensor product model transformation based adaptive integral-sliding mode controller: Equivalent control method," *Sci. World J.*, vol. 2013, 2013.
[42] S. Chumalee and J. F. Whidborne, "Gain-scheduled H_∞ control for tensor product type polytopic plants," *Asian J. Control*, vol. 17, no. 2, pp. 417–431, 2015.
[43] B. Takarics and Y. Yam, "Robust grid point-based control design for LPV systems via unified TP transformation," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2015, pp. 2626–2631.
[44] Z. He, M. Yin, and Y.-P. Lu, "Tensor product model-based control of morphing aircraft in transition process," in *Proc. Institution Mech. Eng., G, J. Aerosp. Eng.*, vol. 230, no. 2, pp. 378–391, 2016.
[45] J. Chen, R. Li, and C. Cao, "Convex polytopic modeling for flexible joints industrial robot using TP-model transformation," in *Proc. IEEE Int. Conf. Inf. Automat.*, 2014, pp. 1046–1050.
[46] R.-E. Precup, C.-A. Dragos, S. Preitl, M.-B. Radac, and E. M. Petriu, "Novel tensor product models for automatic transmission system control," *IEEE Syst. J.*, vol. 6, no. 3, pp. 488–498, Sep. 2012.
[47] K. K. Wu and Y. Yam, "Control stability of TP model transformation design via probabilistic error bound of plant model," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2013, pp. 1259–1264.
[48] R.-E. Precup, L.-T. Dioanca, E. M. Petriu, M.-B. Rădac, S. Preitl, and C.-A. Dragoş, "Tensor product-based real-time control of the liquid levels in a three tank system," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2010, pp. 768–773.
[49] F. Kolonić, A. Poljugan, and I. Petrović, "Tensor product model transformation-based controller design for gantry crane control system–an application approach," *Acta Polytechnica Hungarica*, vol. 3, no. 4, pp. 95–112, 2006.
[50] P. Grof and Y. Yam, "Furuta pendulum–a tensor product model-based design approach case study," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2015, pp. 2620–2625.
[51] C. Ariño and A. Sala, "Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure," *Eng. Appl. Artif. Intell.*, vol. 20, no. 8, pp. 1036–1046, 2007.
[52] F. Yang, Z. Chen, X. Liu, and B. Liu, "A new constant gain kalman filter based on TP model transformation," in *Proc. Chinese Intell. Automat. Conf.*, Springer, 2013, pp. 305–312.
[53] Š. Ileš, J. Matuško, and F. Kolonić, "Tensor product transformation based speed control of permanent magnet synchronous motor drives," in *Proc. 17th Int. Conf. Elect. Drives Power Electron. (5th Joint Slovak-Croatian Conf.)*, 2011, pp. 323–328.
[54] Š. Ileš, J. Matuško, and F. Kolonić, "TP transformation based control of rotary pendulum," in *Proc. 34th IEEE Int. Conv.*, 2011, pp. 833–839.
[55] R. Precup, C. Dragoş, S. Preitl, M. Rădac, and E. Petriu, "Tensor product models for automotive applications," in *Proc. 14th Int. Conf. Syst. Theory Control.*, 2010.
[56] P. Korondi, "Sector sliding mode design based on tensor product model transformation," in *Proc. 11th IEEE Int. Conf. Itell. Eng. Syst.*, 2007, pp. 253–258.
[57] Y. Kunii, B. Solvang, G. Sziebig, and P. Korondi, "Tensor product transformation based friction model," in *Proc. 11th IEEE Int. Conf. Intell. Eng. Syst.*, 2007, pp. 259–264.
[58] P. Korondi, "Sliding sector design based on tensor product model transformation," *Trans. Autom. Control Comput. Sci.*, vol. 51, no. 1, pp. 101–108, 2006.
[59] P. Korondi, P. Bartal, and F. Kolonić, "Friction model based on tensor product transformation," in *Proc. 7th Int. Symp. Hung. Res. Comput. Intell.*, 2006, pp. 27–31.
[60] Š. Ileš, F. Kolonić, and J. Matuško, "Linear matrix inequalities based H_∞ control of gantry crane using tensor product transformation," in *Proc. 18th Int. Conf. Process Control*, 2011.
[61] A. Rövid, L. Szeidl, and P. Várlaki, "On tensor-product model based representation of neural networks," in *Proc. 15th IEEE Int. Conf. Itell. Eng. Syst.*, 2011, pp. 69–72.
[62] A. M. F. Pereire, L. M. S. Vianna, N. A. Keles, and V. C. S. Campos, "Tensor product model transformation simplification of takagi-sugeno control and estimation laws – an application to a thermoelectric controlled chamber," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 13–29, 2018.
[63] V. C. S. Campos, L. M. S. Vianna, and M. F. Braga, "A tensor product model transformation approach to the discretization of uncertain linear systems," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 31–43, 2018.
[64] Y. Kan, Z. He, and J. Zhao, "Tensor product model-based control design with relaxed stability conditions for perching maneuvers," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 45–61, 2018.
[65] H. Gong *et al.*, "Tensor product model-based control for spacecraft with fuel slosh dynamics," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 63–80, 2018.
[66] H. Du, J. Yan, and Y. Fan, "A state and input constrained control method for air-breathing hypersonic vehicles," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 81–99, 2018.
[67] L. Kovács and G. Eigner, "Tensor product model transformation-based parallel distributed control of tumor growth," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 101–123, 2018.
[68] B. Takarics and P. Baranyi, "Tensor-product-model-based control of a three degrees- of-freedom aeroelastic model," *J. Guid., Control, Dyn.*, vol. 36, no. 5, pp. 1527–1533, 2013.
[69] B. Takarics, A. Szollosi, and B. Vanek, "Tensor product type polytopic LPV modeling of aeroelastic aircraft," in *Proc. IEEE Aerosp. Conf.*, 2018, pp. 1–10.
[70] Y. Kan, Z. He, and J. Zhao, "Tensor product model-based control design with relaxed stability conditions for perching maneuvers," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 45–61, 2018.
[71] V. C. de Silva Campos, L. M. S. Vianna, and M. F. Braga, "A tensor product model transformation approach to the discretization of uncertain linear systems," *Acta Polytechnica Hungarica*, vol. 15, no. 3, 2018, pp. 31–43.
[72] L. Kovacs and G. Eigner, "Tensor product model transformation-based parallel distributed control of tumor growth," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 1–22, 2018.
[73] A. M. F. Pereira, L. M. S. Vianna, N. A. Keles, and V. C. da Silva Campos, "Tensor product model transformation simplification of takagi-sugeno control and estimation laws'an application to a thermoelectric controlled chamber," *Acta Polytechnica Hungarica*, vol. 15, no. 3, 2018, pp. 13–29.
[74] J. Kuti and P. Galambos, "Affine tensor product model transformation," *Complexity*, vol. 2018, 2018.
[75] D. A. Eigner, G. Drexler, and L. Kovàcs, "Tumor growth control by tp-lpv-lmi based controller," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2018, pp. 2564–2569.
[76] H. Gong *et al.*, "Tensor product model-based control for space-craft with fuel slosh dynamics," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 64–80, 2018.
[77] H. Bouzaouache, "Tensor product-based model transformation and optimal controller design for high order nonlinear singularly perturbed systems," *Asian J. Control*, vol. 22, no. 1, pp. 486–499, Jan. 2019.
[78] T. Wang, Y. Bi, T. Hou, B. Liu, and J. Cui, "An efficient tp model transformation algorithm for robust visual servoing in the presence of uncertain data," *Acta Polytechnica Hungarica*, vol. 17, no. 6, pp. 155–169, 2020.
[79] V. C. d. S. Campos, M. F. Braga, and L. Frezzatto, "Analytical upper bound for the error on the discretization of uncertain linear systems by using the tensor product model transformation," *Acta Polytechnica Hungarica*, vol. 17, no. 6, pp. 61–74, 2020.
[80] A. Boonyaprapasorn, S. Kuntanapreeda, T. Sangpet, P. S. Ngiamsunthorn, and E. Pengwang, "Biological pest control based on tensor product transformation method," *Acta Polytechnica Hungarica*, vol. 17, no. 6, pp. 25–40, 2020.
[81] P. Baranyi and B. Takarics, "Aeroelastic wing section control via relaxed tensor product model transformation framework," *J. Guid., Control Dyn.*, vol. 37, no. 5, pp. 1671–1678, 2014.
[82] P. Baranyi, "Tensor product model based control of two dimensional aeroelastic system," *J. Guid., Control Dyn.*, vol. 29, no. 2, pp. 391–400, 2006.