WILEY | Hindawi

Research Article

Extension of the Multi-TP Model Transformation to Functions with Different Numbers of Variables

Péter Baranyi (ORCID)

Széchenyi István University, Győr, Hungary

Correspondence should be addressed to Péter Baranyi; prof.peter.baranyi@gmail.com

The tensor product (TP) model transformation defines and numerically reconstructs the Higher-Order Singular Value Decomposition (HOSVD) of functions. It plays the same role with respect to functions as HOSVD does for tensors (and SVD for matrices). The need for certain advantageous features, such as rank/complexity reduction, trade-offs between complexity and accuracy, and a manipulation power representative of the TP form, has motivated novel concepts in TS fuzzy model based modelling and control. The latest extensions of the TP model transformation, called the multi- and generalised TP model transformations, are applicable to a set functions where the dimensionality of the outputs of the functions may differ, but there is a strict limitation on the dimensionality of their inputs, which must be the same. The paper proposes an extended version that is applicable to a set of functions where both the input and output dimensionalities of the functions may differ. This makes it possible to transform complete multicomponent systems to TS fuzzy models along with the above-mentioned advantages.

1. Introduction

The appearance of the Singular Value Decomposition (SVD) was one of the largest breakthroughs in matrix algebra [1]. Its applicability was extended to tensors in the form of the Higher-Order SVD [2] around 2000. Recently, a further extension of the SVD and HOSVD concept, known as the tensor product (TP) Model Transformation, was proposed for functions in control theory [3]. A comprehensive overview is given in [4]. Various extensions of the TP model transformation such as the bilinear-, pseudo-, multi-, and generalised TP model transformation, as well as the concept of HOSVD canonical form of TS fuzzy or TP models, were proposed in [4–7], with a special focus on TS fuzzy models in [8]. The approximation power of the TP model transformation applied to TS fuzzy models is investigated in [9].

The above-mentioned extensions and variations of the TP model transformation were primarily applied to fuzzy model complexity reduction [10, 11] and in the widely used TS fuzzy model based PDC (Parallel Distributed Compensation) control theories [12–14]. But also, in general, it has been applied to polytopic model, TP/TS fuzzy model, and LMI (Linear Matrix Inequality [15]) based control theories. The most important features of the TP model transformation

are guaranteed by the key transformation step whereby a numerically reconstructed HOSVD structure is determined. Key features of the transformation are as follows:

(i) It is executable on models given by equations or soft computing based representations, such as fuzzy rules or neural networks or other black-box models. The only requirement is that the model must provide an output for each input (at least on a discrete scale, see Section 4, Step 1).

(ii) It will find the minimal complexity, namely, the minimal number of rules of the TS fuzzy model. If further complexity reduction is required, it provides one of the best trade-offs between the number of rules and approximation error.

(iii) It works like a principle component analysis, in that it determines the order of the components/fuzzy rules according to their importance.

(iv) It is capable of deriving the antecedent fuzzy sets according to various constraints. For instance, it can be used to define different convex hulls, a capability which has recently been shown to play an important role in control theory.

(v) It is capable of transforming the given model to predefined antecedent fuzzy sets (pseudo-TP model transformation)

(vi) It is capable of transforming a set of models simultaneously, while common antecedent fuzzy sets are derived for all models.

Based on the above, various theories and applications have emerged using the TP model transformation. Further computational improvements were proposed in [16, 17]. It has been proved in [5, 18–20] that LMI based control design theories are very sensitive for convex hulls defined by consequents (vertices) of TS fuzzy models. Thus, the convex hull manipulation capability of the TP model transformation is an important and necessary step in LMI based control design. Very effective convex hull manipulation methods were incorporated into the TP model transformation in [21–23]. Further useful control approaches and applications were published in the field of control theory [24–41]. Many powerful approaches are published on the field of sliding mode control in [29, 42, 43]. In physiological control the usability of TP model transformation has been demonstrated as well [44–49]. Various further theories and applications are studied in [50–87].

One of the key advantages of the TP model transformation is that is capable of finding the minimal complexity of all components of the system and guarantees the same antecedent system for all components. This is a very typical requirement in design or stability verification methodologies, that is, the model, controller, and observer need to have the same antecedent system, hence, convex representation. Therefore, the simultaneous manipulation of the components with the multi-TP model transformation or the generalised TP model transformation (that combines all variants of the TP model transformation) yields further possibilities for control performance optimisation [18–20].

Despite the above advantages, a crucial limitation of the generalised TP model transformation is that it can only be applied to a set of systems which have the same number of inputs. For instance, consider four different systems given with different representations, as shown in Figure 1. S1 is a fuzzy logic model; S2 is neural network; S3 is given by an equation; and S4 is a black-box model. All of these models have the same inputs but may have different sized output tensors. The multi-TP model transformation is capable of simultaneously transforming all systems to TP or TS fuzzy model form, such that the same antecedent sets are defined on the inputs. The generalised TP model transformation can also transform to predefined antecedent fuzzy sets.

A further generalisation proposed in this paper can be applied to systems like in the example given in Figure 2. Here each system may be given by different representations (like in the above case) but may also have different numbers of inputs. The transformation can simultaneously convert all of the systems to TS fuzzy model form, such that the antecedent fuzzy sets will either be the same or assume a predefined structure. From all other perspectives, the proposed TP model transformation inherits all of the advantageous features of the previous TP-based approaches.
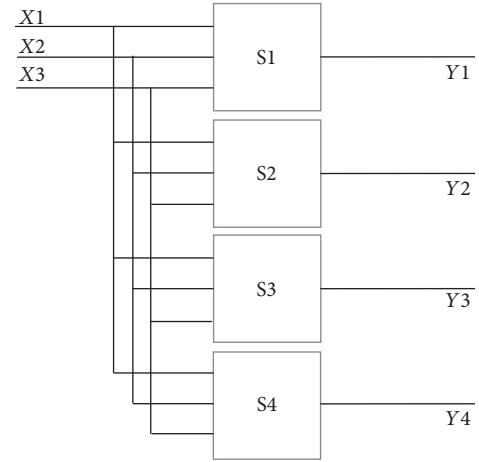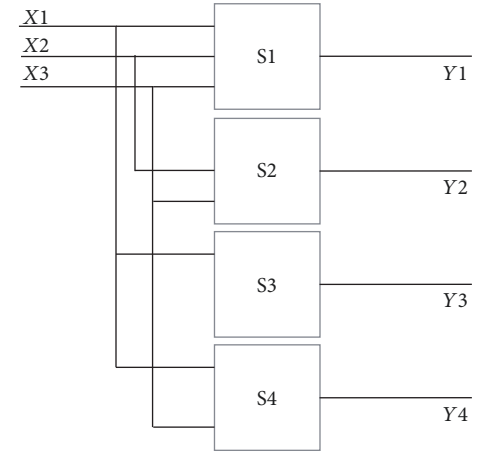

Figure 1: Multi TP model transformation.

Figure 2: Proposed extension of the TP model transformation.

Recently proposed SOS-type (Sum-of-sqares) TS fuzzy LPV models are also widely applied in fuzzy control theories [88, 89]. The further extension of the TP model transformation to such systems is highly welcome in future works.

2. Notation and Concepts

2.1. Notation. The following notations are used in the paper:

(i) Scalar: a is scalar.

(ii) Vector: $\mathbf{a}$ contains elements a_i.

(iii) Matrix: $\mathbf{A}$ contains elements $a_{i,j}$.

(iv) Tensor: $\mathcal{A}$ contains elements $a_{i,j,k,...}$.

(v) Set: A : $\{a, b, c, \ldots\}$, for example, $a \in$ A.

(vi) Index i: the upper bounds of the indices are denoted by the uppercase letter, for example, I.

(vii) Index $i \in$ I denotes that index i takes the elements of set I $\subseteq \{1, 2, \ldots, I\} \subset \mathbb{N}$, respectively. I : $\{1, 2, \ldots, I\}$ is understood as per default.

(viii) Interval: $\omega = [\omega_{\min}, \omega_{\max}]$.

(ix) Space: $\Omega : \omega_1 \times \omega_2 \times \cdots \times \omega_N$ is an N dimensional hypercube.

(x) $\mathbf{x} \in \Omega$ expresses the fact that vector $\mathbf{x}$ is within the space Ω. The dimensions of $\mathbf{x}$ and Ω are the same.

(xi) $\sqsubset$ denotes a dimensionality reduced subset in general as follows:

    (a) In the case of spaces: $\Theta \sqsubset \Omega$ states that Θ is a hypercube with the same sized intervals as Ω, but has a smaller number of dimensions.

    (b) In the case of vectors: $\mathbf{a} \sqsubset \mathbf{b}$, where $\mathbf{a} \in \Theta \subset \mathbb{R}^N$ and $\mathbf{b} \in \Omega \subset \mathbb{R}^M$ means that $N < M$ and $\Theta \sqsubset \Omega$.

    (c) In the case of tensors: $\mathscr{A} \sqsubset \mathscr{B}$ means for instance that $\mathscr{A}$ is obtained by deleting complete dimensions from tensor $\mathscr{B}$.

(xii) Grid: $G : \mathbf{g}_1 \times \mathbf{g}_2 \times \cdots \times \mathbf{g}_N$ is a rectangular hyper grid (tensor), where $\mathbf{g}_n = \begin{bmatrix} g_{n,1} < g_{n,2} < \cdots < g_{n,M_n} \end{bmatrix}$ defines the locations of the M_n different grid points in increasing order.

(xiii) Pair (Ω, G): space $\Omega \in \mathbb{R}^N$ and grid G are in a pair, meaning that $\forall n \in N : g_{n,1} = \omega_{n,\min}$ and $g_{n,M_n} = \omega_{n,\max}$.

(xiv) Discretised function $\mathscr{F}^{D(\Omega,G)}$ of $f(\mathbf{x})$ denotes the sampling of $f(\mathbf{x})$ over pair (Ω, G). Thus, it is a tensor with the size of $M_1 \times M_2 \times \cdots \times M_N$ and entries:

$$f_{m_1,m_2,\ldots,m_N} = f\left(\begin{bmatrix} g_{1,m_1} & g_{2,m_2} & \cdots & g_{N,m_N} \end{bmatrix}\right). \quad (1)$$

(xv) $\mathcal{S} \underset{N}{\boxtimes} \mathbf{U}_n$ is the tensor product (TP); for details, refer to [4, 5, 8]. A slight difference in notation here is that N under the tensor product operation $\boxtimes$ is only a set numbers $n \in N$ to which the product should be applied.

(xvi) $f(\mathbf{x}) = \mathcal{S} \underset{N}{\boxtimes} \mathbf{w}_n(x_n)$ represents the TP function, $\mathbf{x} \in \mathbb{R}^N$, where $\mathbf{w}_n(x_n) = \begin{bmatrix} w_{n,1}(x_n) & w_{n,2}(x_n) & \cdots & w_{n,I_n}(x_n) \end{bmatrix}$ is called the weighting function system.

(xvii) Types of the weighting functions are as follows:

    (a) SN: sum normalised
    (b) NN: nonnegativeness
    (c) NO: normalised
    (d) CNO: close to normalised
    (e) RNO: relaxed normalised
    (f) INO: inverse normalised
    (g) IRNO: inverse relaxed normalised.

For further details, refer to [4, 5].

3. The Proposed TP Model Transformation

Assume that a set of functions is given as $\mathscr{Y}_l = f_l(\mathbf{x}_l)$, $\mathbf{x}_l \sqsubseteq \mathbf{x} \in \Omega$; thus $\mathbf{x}_l \in \Omega_l \sqsubseteq \Omega \subset \mathbb{R}^N$, $\mathbf{x}_l \in \mathbb{R}^{N_l}$. The output tensor $\mathscr{Y}_l$ of each function $f_l(\mathbf{x}_l)$ may differ in the number of dimensions and its size as $\mathscr{Y}_l \in \mathbb{R}^{O_{1,l} \times O_{2,l} \times \cdots \times O_{K_l,l}}$, where K_l denotes the number of dimensions of the output and O_k denotes the number of elements in dimension k.

The goal of the TP model transformation is to transform $\forall l : f_l(\mathbf{x}_l)$ into TP function form as

$$\mathscr{Y}_l = f_l(\mathbf{x}_l) = \mathcal{S}_l \underset{N_l}{\boxtimes} \mathbf{w}_{l,n}(x_n) \quad (2)$$

under the following constraints given on the weighting functions.

(i) Unified Constraints for $\forall l : f_l(\mathbf{x}_l)$. All resulting TP functions will have the same weighting function system on each dimension defined by the set $V \subseteq N$ (obviously, if the function has that input dimension):

    (a) Weighting function systems $\mathbf{w}_a(x_a)$, $a \in A \subseteq V$, are predefined.

    (b) Weighting function systems $\mathbf{w}_b(x_b)$, $b \in B \subseteq V$ will be derived by the transformation; only their types are predefined (i.e., SN, NN, NO, CNO, RNO, INO, and IRNO). Further the number of the weighting functions are minimised.

(ii) Different Constraints for Each $f_l(\mathbf{x}_l)$. The resulting TP functions have different weighting functions on dimensions $Z \subseteq N$:

    (a) Weighting function systems $\mathbf{w}_{l,c}(x_c)$, $c \in C \subseteq Z$ are predefined of each $f_l(\mathbf{x}_l)$.

    (b) The types (i.e., SN, NN, NO, CNO, RNL, INO, and IRNO) of the weighting function systems $\mathbf{w}_{l,d}(\mathbf{x}_d)$ are predefined for dimensions $d \in D \subseteq Z$ of each $f_l(\mathbf{x}_l)$.

Thus, (2) can be given as follows:

$$\begin{aligned} \mathscr{Y}_l &= f_l(\mathbf{x}_l) \\ &= \mathcal{S}_l \underset{A}{\boxtimes} \mathbf{w}_a(x_a) \underset{B}{\boxtimes} \mathbf{w}_b(x_b) \underset{C}{\boxtimes} \mathbf{w}_{l,c}(x_c) \underset{D}{\boxtimes} \mathbf{w}_{l,d}(x_d). \end{aligned} \quad (3)$$

4. The Computation of the Proposed TP Model Transformation

Step 1 (discretisation).

    (i) Discretisation of all $f_l(\mathbf{x}_l)$ results in tensor $\mathscr{F}_l^{D(\Omega_l,G_l)}$, $(G_l \sqsubseteq G)$. The size of $\mathscr{F}_l^{D(\Omega_l,G_l)}$ in dimension $n \in N_l$ is M_n.

    (ii) Discretise the predefined weighting functions over the dimensions of Ω:

$$\begin{aligned} &\mathscr{W}_a^{D(\omega_a,\mathbf{g}_a)}, \\ &\mathscr{W}_{l,c}^{D(\omega_c,\mathbf{g}_c)}. \end{aligned} \quad (4)$$

Remark 1. This step is executed in the same way as in the case of the original TP model transformation; see [4, 5, 8].

Step 2 (defining TP structures). Execute the following steps in each dimension $n \in N$:

    (i) Lay out tensors $\mathscr{F}_l^{D(\Omega,G)}$ in dimensions n if vector $\mathbf{x}_l$ has the following dimension:

$$\mathbf{H}_l = \left(\mathscr{F}_l^{D(\Omega_l,G_l)}\right)_{(n)}. \quad (5)$$

(ii) If $n \in B$ then create

$$\mathbf{T}_n = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \cdots & \mathbf{H}_L \end{bmatrix}. \tag{6}$$

Execute SVD on $\mathbf{T}_n$ and SN, NN, NO, CNO and complexity trade-off by discarding singular values in the same way as in the original TP model transformation, which results in

$$\mathbf{T}_n = \mathbf{U}_n \mathbf{D}_n \mathbf{V}_n^T. \tag{7}$$

As a matter of fact, if nonzero singular values are discarded then it is only an approximation. Let

$$\mathscr{W}_n^{D(\omega_n, \mathbf{g}_n)} = \mathbf{U}_n. \tag{8}$$

(iii) If $n \in D$ then execute SVD on $\mathbf{H}_l$ as

$$\mathbf{H}_l = \mathbf{U}_l \mathbf{D}_l \mathbf{V}_l^T; \tag{9}$$

and, according to the conditions, execute SN, NN, NO, CNO, and complexity trade-off by discarding singular values in the same way as in the original TP model transformation:

$$\mathbf{H}_l = \mathbf{U}'_l \mathbf{D}'_l {\mathbf{V}'_l}^T. \tag{10}$$

Again, if nonzero singular values are discarded then it is only an approximation. Let

$$\mathscr{W}_{l,n}^{D(\omega_n, \mathbf{g}_n)} = \mathbf{U}'_l. \tag{11}$$

(iv) Finally,

$$\mathcal{S}_l = \mathscr{F}_l^{D(\Omega, G)} \underset{n \in A \cup B}{\boxtimes} \mathbf{K}_n \underset{n \in C \cup D}{\boxtimes} \mathbf{K}_{l,n}, \tag{12}$$

where

$$\begin{aligned} \mathbf{K}_n &= \left(\mathscr{W}_n^{D(\omega_n, \mathbf{g}_n)} \right)^+, \\ \mathbf{K}_{l,n} &= \left(\mathscr{W}_{l,n}^{D(\omega_n, \mathbf{g}_n)} \right)^+, \end{aligned} \tag{13}$$

where $(\cdot)^+$ denotes the pseudoinverse.

Step 3 (reconstruction of the weighting functions). This step is the same as in the multi-TP model transformation [4, 5, 8]. Having the result of the above steps, $\mathscr{F}_l^{D(\Omega_l, G_l)}$ and $\mathcal{S}_l$, we can recalculate the weighting functions at any point. We may calculate the first two steps over a grid, which is not too dense, but calculate the weighting function over a very dense grid (as suggested in [5]), and then construct piecewise linear functions. As a result we have $\mathbf{w}_v(x_v)$ and $\mathbf{w}_{l,z}(x_z)$.

Then we achieved the goal. We have the TP model form of all functions with the given constraints:

$$f_l(\mathbf{x}_l) = \mathcal{S}_l \underset{N_l}{\boxtimes} \mathbf{w}_{l,n}(x_n), \tag{14}$$

or

$$f_l(\mathbf{x}_l) \approx \mathcal{S}_l \underset{N_l}{\boxtimes} \mathbf{w}_{l,n}(x_n), \tag{15}$$

if a complexity trade-off is executed (nonzero singular values are discarded), where

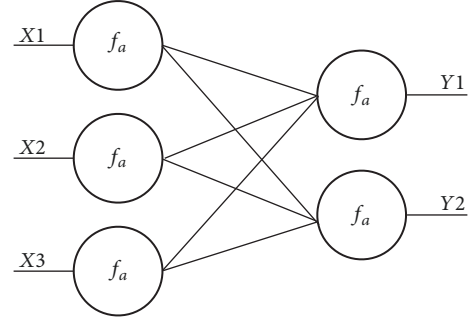$$\forall l : \mathbf{w}_{l,a}(\mathbf{x}_l) = \mathbf{w}_a(\mathbf{x}_l),$$

FIGURE 3: System 3: neural network.

$$\forall l : \mathbf{w}_{l,b}(\mathbf{x}_l) = \mathbf{w}_b(\mathbf{x}_l). \tag{16}$$

Or in other words,

$$f_l(\mathbf{x}_l) = \mathcal{S}_l \underset{A}{\boxtimes} \mathbf{w}_a(\mathbf{x}_a) \underset{B}{\boxtimes} \mathbf{w}_b(\mathbf{x}_b) \underset{C}{\boxtimes} \mathbf{w}_{l,c}(\mathbf{x}_l) \underset{D}{\boxtimes} \mathbf{w}_{l,d}(\mathbf{x}_l). \tag{17}$$

Remark 2. The convex hull manipulation and the complexity trade-off are done in the second step. Therefore the approximation accuracy is controlled here by the discarded nonzero singular values. However, the discarded nonzero singular values lead to approximation error. If the given weighting function system is not sufficient (i.e., the number of the weighting functions is less than the rank of that dimension) then we arrive at an approximation only. The use of the pseudoinverse guarantees, however, that it will be the best approximation.

5. Example

5.1. The System. Consider a multicomponent system with input vector $\mathbf{x} = [x_1 \ x_2 \ x_3]$, where $\mathbf{x} \in \Omega = \omega_1 \times \omega_2 \times \omega_3 \in \mathbb{R}^3$ and $\omega_n = [0, 1]$. The system has four subsystems, $l \in L : \{1, \dots, 4\}$ as shown in Figure 1.

System 3. In order to have a systematic notation, we denote the input vector of System 3 as $\mathbf{x}_1 = \mathbf{x}$ that is $\mathbf{x}_1 \in \Omega_1 = \Omega$. It is a neural network; see Figure 3:

$$\begin{aligned} y_1 &= f_a \left(\sum_{i=1}^{3} f_a(x_i) b_{1,i} \right), \\ y_2 &= f_a \left(\sum_{i=1}^{3} f_a(x_i) b_{2,i} \right), \end{aligned} \tag{18}$$

where $f_a(z)$ is the activation function (let it be a very simple one in the present case: $f_a(z) = z$) of the neurons and $b_{j,i}$ are the weights connecting the ith input neuron to the jth output neuron. Thus the output of the system is

$$\mathbf{y}_1 = f_1(\mathbf{x}_1) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \tag{19}$$

System 4. The input vector of System 4 is $\mathbf{x}_2 = [x_2 \ x_3] \in \Omega_2 = \omega_2 \times \omega_3$, where $\mathbf{x}_2 \sqsubset \mathbf{x}$ and $\Omega_2 \sqsubset \Omega$.

This system is given by formulas such as

$$\mathbf{Y}_2 = f_2\left(\mathbf{x}_2\right) = \begin{bmatrix} x_2^2 & 2x_2 \\ 3x_3^{-1} & 5 \end{bmatrix}. \tag{20}$$

System 5. The input vector of System 5 is $\mathbf{x}_3 = x_1 \in \Omega_3 = \omega_1$, where $\mathbf{x}_3 \sqsubset \mathbf{x}$ and $\Omega_3 \sqsubset \Omega$.

This is given by a fuzzy logic model. Assume that two rules are given ($i = 1, 2$):

IF A_i THEN B_i.

Further assume that the membership functions are in Ruspini partition:

$$\mu_{A_1}\left(x_3\right) = x_3;$$
$$\mu_{A_2}\left(x\right) = 1 - x_3; \tag{21}$$

and the consequent sets are singleton sets located at elements 5 and 6 of the output universe. It is a TS fuzzy model and, therefore, the transfer function (product-sum-gravity) of the model is

$$y_3 = f_3\left(\mathbf{x}_3\right) = 5x_3 + 6\left(1 - x_3\right). \tag{22}$$

System 6. The input vector of System 6 is $\mathbf{x}_4 = \begin{bmatrix} x_1 & x_3 \end{bmatrix} \in \Omega_4 = \omega_1 \times \omega_3$, where $\mathbf{x}_4 \sqsubset \mathbf{x}$ and $\Omega_4 \sqsubset \Omega$.

This is a black-box model that can provide y_4 for any input x_1, x_3.

$$y_4 = f_4\left(\mathbf{x}_4\right). \tag{23}$$

(In order to follow all computational steps of the example, let us reveal what is the output of the black-box $y_4 = 3x_1 + x_3$.)

5.2. Conditions of the TP Model Transformation. The goal of the example is to transform all the four systems to TS fuzzy representations (or TP model if the resulting weighting functions cannot be represented as antecedent fuzzy states), with the following conditions:

(i) All systems must have the same antecedent function system on the input interval of x_1. The antecedent functions must be in Ruspini partition, namely, in SN and NN type. In order to have a complexity minimised representation, a further requirement is that the number of antecedent functions must be minimal.

(ii) The same antecedent function system of variable x_2 is predefined for all systems:

$$\mathbf{w}_2^p\left(x_2\right) = \begin{bmatrix} w_{2,1}\left(x_2\right) & w_{2,2}\left(x_2\right) & w_{2,3}\left(x_2\right) \end{bmatrix}, \tag{24}$$

where "p" denotes "predefined," and

$$w_{2,1}\left(x_2\right) = 0.5x_2^2;$$
$$w_{2,2}\left(x_2\right) = 0.5\left(1 - x_2\right)^2 \tag{25}$$
$$w_{2,3}\left(x_2\right) = 1 - \left(w_{2,1}\left(x_2\right) + w_{2,2}\left(x_2\right)\right).$$

(iii) The only requirement for the weighting function system of the input x_3 of each system is that they must be the singular functions of the HOSVD canonical form (othonormed system ordered by the higher-order singular values). These functions are not representable as antecedent functions of fuzzy sets, since they may take negative values as well. Obviously they will not be the same for all systems.

5.3. Execution of the Proposed TP Model Transformation. It is worth emphasizing again that the previous methods for TP model representation cannot be applied in the present case, since the elements of the input vectors are different.

Step 1.

(i) Let us define grid G to $\Omega = [0, 1] \times [0, 1] \times [0, 1]$:

$$\mathbf{g}_1 = \mathbf{g}_2 = \mathbf{g}_3 = \begin{bmatrix} 0 & 0.01 & 0.02 & \cdots & 1 \end{bmatrix}. \tag{26}$$

Thus the number of points on the discretisation grid is $M_1 = M_2 = M_3 = 101$.

(ii) Let us discretise the systems over the rectangular grid defined by vectors $\mathbf{g}_n$, $n = 1, 2, 3$. The discretisation of System l results in $\mathscr{F}_l^{D(\Omega_l, G_l)}$, $l \in L$. In case of System 3,

$$\mathscr{F}_1^{D(\Omega_1, G_1)} \in \mathbb{R}^{M_1 \times M_2 \times M_3 \times 2}, \tag{27}$$

where the first three dimensions are assigned to the input variables and the last dimension is assigned to the output vector. The discretisation of System 4 yields

$$\mathscr{F}_2^{D(\Omega_2, G_2)} \in \mathbb{R}^{M_2 \times M_3 \times 2 \times 2}, \tag{28}$$

where the first two dimensions are assigned to the input variables x_2, x_3 and the last two dimensions are assigned to the output matrix. The discretisation of System 5 yields the following vector:

$$\mathscr{F}_3^{D(\Omega_3, G_3)} \in \mathbb{R}^{M_3}. \tag{29}$$

The discretisation of System 6 results in

$$\mathscr{F}_4^{D(\Omega_4, G_4)} \in \mathbb{R}^{M_1 \times M_3}, \tag{30}$$

where the first two dimensions are assigned to the input variables x_1, x_3 and the last dimension is assigned to the output vector.

Let us discretise the predefined weighting function as well:

$$\mathscr{W}_2^{D(\omega_2, \mathbf{g}_2)}$$
$$= \begin{bmatrix} w_{2,1}\left(g_{2,1}\right) & w_{2,2}\left(g_{2,1}\right) & w_{2,3}\left(g_{2,1}\right) \\ \vdots & \vdots & \vdots \\ w_{2,1}\left(g_{2,M_2}\right) & w_{2,2}\left(g_{2,M_2}\right) & w_{2,3}\left(g_{2,M_2}\right) \end{bmatrix}. \tag{31}$$

Step 2.

(i) *Dimension x_1*

Lay out tensors $\mathscr{F}_l^{D(\Omega_l, G_l)}$, $l \in \{1, 3, 4\}$ in the dimension assigned x_1:

$$\mathbf{H}_l = \left(\mathscr{F}_l^{D(\Omega_l, G_l)} \right)_{(1)}. \tag{32}$$

Create

$$\mathbf{K} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_3 & \mathbf{H}_4 \end{bmatrix}. \tag{33}$$

Execute SVD on $\mathbf{K}$ incorporating SN and NN condition [4] (only nonzero singular values are kept):

$$\mathbf{K} = \mathbf{U}_1 \mathbf{D}' {\mathbf{V}'}^T. \tag{34}$$

The result of this step to be used later is $\mathbf{U}_1$.

(ii) *Dimension x_2*

Let

$$\mathbf{U}_2 = \mathscr{W}_2^{D(\omega_2, \mathbf{g}_2)}. \tag{35}$$

(iii) *Dimension x_3*

Lay out tensors $\mathscr{F}_l^{D(\Omega_l, G_l)}$, $l \in \{1, 2, 4\}$ in the dimension assigned x_3:

$$\mathbf{H}_l = \left(\mathscr{F}_l^{D(\Omega_l, G_l)} \right)_{(3)}. \tag{36}$$

Then execute HOSVD on each $\mathbf{H}_l$, $l \in \{1, 2, 4\}$ (only the nonzero singular values are kept):

$$\mathbf{H}_1 = \mathbf{U}_{3,1} \mathbf{D}_1 \mathbf{V}_1^T.$$
$$\mathbf{H}_2 = \mathbf{U}_{3,2} \mathbf{D}_2 \mathbf{V}_2^T. \tag{37}$$
$$\mathbf{H}_4 = \mathbf{U}_{3,4} \mathbf{D}_4 \mathbf{V}_4^T.$$

The result of this step is $\mathbf{U}_{3,l}$.

(iv) *Reconstructing the core tensors*

$$\forall l : \mathscr{S}_l = \mathscr{F}_1^{D(\Omega_l, G_l)} \underset{\Omega_l}{\boxtimes} \left(\mathbf{U}_n \right)^+. \tag{38}$$

Step 3. Let

$$\mathscr{W}_1^{D(\omega_1, \mathbf{g}_1)} = \mathbf{U}_1,$$
$$\mathscr{W}_{3,l}^{D(\omega_3, \mathbf{g}_3)} = \mathbf{U}_{3,l}, \tag{39}$$

$l : \{1, 2, 4\}$.

Then having the discretised tensors and weighting functions of all systems we can numerically reconstruct the weighting functions [4, 5] as

$$\mathscr{W}_1^{D(\omega_1, \mathbf{g}_1)} = \mathbf{U}_1 \longrightarrow \mathbf{w}_1 (x_1),$$
$$\mathscr{W}_{3,l}^{D(\omega_3, \mathbf{g}_3)} = \mathbf{U}_{3,l} \longrightarrow \mathbf{w}_{3,l} (x_3), \tag{40}$$

$l : \{1, 2, 4\}$.

Thus, we have achieved our goal:

$$f_1 (x_1, x_2, x_3) = \mathscr{S}_1 \times_1 \mathbf{w}_1 (x_1) \times_2 \mathbf{w}_2^p (x_2) \times_3 \mathbf{w}_{1,3} (x_3);$$
$$f_2 (x_2, x_3) = \mathscr{S}_2 \times_2 \mathbf{w}_2^p (x_2) \times_3 \mathbf{w}_{2,3} (x_3);$$
$$f_3 (x_1) = \mathscr{S}_3 \times_1 \mathbf{w}_1 (x_1); \tag{41}$$
$$f_4 (x_1, x_3) = \mathscr{S}_4 \times_1 \mathbf{w}_1 (x_1) \times_3 \mathbf{w}_{4,3} (x_3).$$

6. Conclusion

The proposed TP model transformation can be executed on a set of models where the dimensionality of the inputs may differ. The proposed TP model transformation has all the advantages of the previous ones, including easy convex hull manipulation, complexity trade-offs, pseudo TP model transformation, and automatic and numerical execution.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993.

[2] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[3] P. Baranyi, "TP model transformation as a way to LMI-based controller design," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 387–400, 2004.

[4] P. Baranyi, Y. Yam, and P. Varlaki, *Tensor Product Model Transformation in Polytpic Model Based Control*, Automation and Control Engineering, CRC Press, Taylor & Frances Grouop, March 2017.

[5] P. Baranyi, *TP-Model Transformation-Based-Control Design Frameworks*, Control Engineering, Springer International Publishing, Chengdo, China, 2016.

[6] P. Baranyi, L. Szeidl, P. Várlaki, and Y. Yam, "Definition of the HOSVD based canonical form of polytopic dynamic models," in *Proceedings of the IEEE International Conference on Mechatronics, ICM '06*, pp. 660–665, Hungary, July 2006.

[7] P. Baranyi, L. Szeidl, and P. Várlaki, "Numerical reconstruction of the HOSVD based canonical form of polytopic dynamic models," in *Proceedings of the 10th International Conference on Intelligent Engineering Systems '06*, pp. 196–201, Agadir, Morocco, June 2006.

[8] P. Baranyi, "The generalized TP model transformation for T–S fuzzy model manipulation and generalized stability verification," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 934–948, 2014.

[9] D. Tikk, P. Baranyi, R. Patton, and J. Tar, "Approximation capability of TP model forms," *Australian Journal of Intelligent Information Processing Systems*, vol. 8, no. 3, pp. 155–163, 2004.

[10] P. Baranyi and Y. Yam, "Fuzzy Rule Base Reduction," in *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications*, D. Ruan and and E. E. Kerre, Eds., ch. 7, pp. 135–160, Kluwer, 2000.

[11] Y. Yam, P. Baranyi, and C.-T. Yang, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 120–132, 1999.

[12] K. Tanaka, T. Ikeda, and H. O. Wang, "Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, pp. 250–265, 1998.

[13] K. Tanaka and H. Wang, "Fuzzy regulators and fuzzy observers: a linear matrix inequality approach," in *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 1315–1320, San Diego, CA, USA, 2001.

[14] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets and Systems*, vol. 45, no. 2, pp. 135–156, 1992.

[15] C. Scherer and S. Weiland, "Linear Matrix Inequalities in Control," in *Lecture Notes*, Dutch Institute for Systems and Control, Delft, The Netherlands, 2000.

[16] J. Cui, K. Zhang, and T. Ma, "An efficient algorithm for the tensor product model transformation," *International Journal of Control, Automation, and Systems*, vol. 14, no. 5, pp. 1205–1212, 2016.

[17] S. Nagy, Z. Petres, P. Baranyi, and H. Hashimoto, "Computational relaxed TP model transformation by restriction of the computation to subspaces of the dynamic model," in *Proceedings of the ISCIII'07: 3rd International Symposium on Computational Intelligence and Intelligent Informatics*, pp. 99–104, mar, March 2007.

[18] A. Szollosi and P. Baranyi, "Influence of the tensor product model representation of qLPV models on the feasibility of linear matrix inequality," *Asian Journal of Control*, vol. 18, no. 4, pp. 1328–1342, 2016.

[19] 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 Journal of Control*, vol. 19, no. 2, pp. 450–466, 2017.

[20] 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 Journal of Control*, 2017.

[21] X. Liu, Y. Yu, Z. Li, H. H. C. Iu, and T. Fernando, "An Efficient Algorithm for Optimally Reshaping the TP Model Transformation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 10, pp. 1187–1191, 2017.

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

[23] P. Várkonyi, D. Tikk, P. Korondi, and P. Baranyi, "A new algorithm for RNO-INO type Tensor Product model representation," in *Proceedings of the INES'05: IEEE 9th International Conference on Intelligent Engineering Systems*, pp. 263–266, September 2005.

[24] X. Liu, X. Xin, Z. Li, and Z. Chen, "Near Optimal Control Based on the Tensor-Product Technique," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 5, pp. 560–564, 2017.

[25] X. Liu, Y. Yu, Z. Li, H. H. C. Iu, and T. Fernando, "A novel constant gain Kalman filter design for nonlinear systems," *Signal Processing*, vol. 135, pp. 158–167, 2017.

[26] X. Liu, Y. Yu, Z. Li, and H. H. C. Lu, "Polytopic h1 filter design and relaxation for nonlinear systems via tensor product technique," in *Signal Processing*, vol. 127, pp. 191–205, 2016.

[27] P. S. Saikrishna, R. Pasumarthy, and N. P. Bhatt, "Identification and Multivariable Gain-Scheduling Control for Cloud Computing Systems," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 792–807, 2017.

[28] Q. Zhang and X. Zhang, *Vapor Compression Cycle Control for Automotive Air Conditioning Systems with a Linear Parameter Varying Approach*, Cornell University Library, 2017.

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

[30] W. Qin, B. He, Q. Qin, and G. Liu, "Robust active controller of hypersonic vehicles in the presence of actuator constraints and input delays," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 10718–10723, Chengdu, China, July 2016.

[31] Q. Weiwei, H. Bing, L. Gang, and Z. Pengtao, "Robust model predictive tracking control of hypersonic vehicles in the presence of actuator constraints and input delays," *Journal of The Franklin Institute*, vol. 353, no. 17, pp. 4351–4367, 2016.

[32] T. Wang and B. Liu, "Different polytopic decomposition for visual servoing system with LMI-based Predictive Control," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 10320–10324, China, July 2016.

[33] T. Wang and W. Zhang, "The visual-based robust model predictive control for two-DOF video tracking system," in *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, pp. 3743–3747, China, May 2016.

[34] T. Jiang and D. Lin, "Tensor product model-based gain scheduling of a missile autopilot," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 59, no. 3, pp. 142–149, 2016.

[35] A. Hajiloo, *Robust and multi-objective model predictive control design for nonlinear systems [Ph.D. thesis]*, Concordia University, 2016.

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

[37] Y. Yu, Z. Li, J. Li, L. Li, and X. Liu, "H∞ polytopic filter design for nonlinear systems," in *Proceedings of the 35th Chinese Control Conference (CCC '16)*, pp. 620–625, Chengdu, China, July 2016.

[38] P. Bo, Y. Lingyu, and Y. Xiaoke, "LPV-MRAC Method for Aircraft with Structural Damage," in *Proceedings of the 35th Chinese Control Conference (CCC)*, pp. 3262–3267, Chengdu, China, July 2016.

[39] 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 Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 748–760, 2013.

[40] S. Kuntanapreeda, "Tensor product model transformation based control and synchronization of a class of fractional-order chaotic systems," *Asian Journal of Control*, vol. 17, no. 2, pp. 371–380, 2015.

[41] 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 Journal of Control*, vol. 17, no. 2, pp. 381–391, 2015.

[42] G. Zhao, H. Li, and Z. Song, "Tensor product model transformation based decoupled terminal sliding mode control," *International Journal of Systems Science*, vol. 47, no. 8, pp. 1791–1803, 2016.

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

[44] G. Eigner, I. Böjthe, P. Pausits, and L. Kovács, "Investigation of the TP modeling possibilities of the Hovorka T1DM model," in *Proceedings of the 15th IEEE International Symposium on Applied Machine Intelligence and Informatics, SAMI 2017*, pp. 259–264, Slovakia, January 2017.

[45] G. Eigner, P. Pausits, and L. Kovacs, "Control of T1DM via tensor product-based framework," in *Proceedings of the 17th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2016*, pp. 55–60, Hungary, November 2016.

[46] G. Eigner, I. Rudas, A. Szakal, and L. Kovacs, "Tensor product based modeling of tumor growth," in *Proceedings of the 2017 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 900–905, Banff, AB, October 2017.

[47] L. Kovacs and G. Eigner, Usability of the tensor product based modeling in the modeling of diabetes mellitus, manuscript in preparation, 2016.

[48] G. Eigner, I. J. Rudas, and L. Kovacs, "Investigation of the TP-based modeling possibility of a nonlinear ICU diabetes model," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 3405–3410, Hungary, October 2016.

[49] L. Kovacs and G. Eigner, "Convex polytopic modeling of diabetes mellitus: A Tensor Product based approach," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 3393–3398, Hungary, October 2016.

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

[51] J. Kuti, P. Galambos, and Á. Miklós, "Output feedback control of a dual-excenter vibration actuator via qLPV model and TP model transformation," *Asian Journal of Control*, vol. 17, no. 2, pp. 432–442, 2015.

[52] J. Pan and L. Lu, "TP model transformation via sequentially truncated higher-order singular value decomposition," *Asian Journal of Control*, vol. 17, no. 2, pp. 467–475, 2015.

[53] J. Matuško, Š. Ileš, F. Kolonić, and V. Leši, "Control of 3D tower crane based on tensor product model transformation with neural friction compensation," *Asian Journal of Control*, vol. 17, no. 2, pp. 443–458, 2015.

[54] V. c. Campos, L. A. Tôrres, and R. M. Palhares, "Revisiting the TP model transformation: interpolation and rule reduction," *Asian Journal of Control*, vol. 17, no. 2, pp. 392–401, 2015.

[55] G. Zhao, K. Sun, and H. Li, "Tensor product model transformation based adaptive integral-sliding mode controller: Equivalent control method," *The Scientific World Journal*, vol. 2013, Article ID 726963, 2013.

[56] S. Chumalee and J. F. Whidborne, "Gain-Scheduled H∞ Control for Tensor Product Type Polytopic Plants," *Asian Journal of Control*, vol. 17, no. 2, pp. 417–431, 2015.

[57] G. Zhao, S. Huang, Y. Zhang, T. Zhang, and Y. Zhang, "Tensor product model transformation based fractional decoupled sliding-mode control for cart-pole system with time-varying sliding surfaces and Dahl friction model," in *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, pp. 3582–3589, China, May 2017.

[58] J. Klespitz, I. J. Rudas, and L. Kovacs, "LMI-Based Feedback Regulator Design via TP Transformation for Fluid Volume Control in Blood Purification Therapies," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 2615–2619, Hong Kong, October 2015.

[59] G. Zhao, C. Zhao, and D. Wang, "Tensor product model transformation based integral sliding mode control with reinforcement learning strategy," in *Proceedings of the 33rd Chinese Control Conference, CCC 2014*, pp. 77–82, China, July 2014.

[60] S. Huang, G. Zhao, Y. Yuan, Q. Ren, and Y. Liu, "Intelligent tensor product mode transformation-based three-sliding-surface sliding mode controller design," in *Proceedings of the 34th Chinese Control Conference, CCC 2015*, pp. 3258–3263, China, July 2015.

[61] Á. Takács, T. Haidegger, P. Galambos, J. Kuti, and I. J. Rudas, "Nonlinear soft tissue mechanics based on polytopic Tensor Product modeling," in *Proceedings of the IEEE 14th International Symposium on Applied Machine Intelligence and Informatics, SAMI '16*, pp. 211–215, Slovakia, January 2016.

[62] B. Takarics and Y. Yam, "Robust Grid Point-Based Control Design for LPV Systems via Unified TP Transformation," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC '15*, pp. 2626–2631, Hong Kong, October 2015.

[63] Z. He, M. Yin, and Y.-P. Lu, "Tensor product model-based control of morphing aircraft in transition process," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 2, pp. 378–391, 2016.

[64] J. Chen, R. Li, and C. Cao, "Convex polytopic modeling for flexible joints industrial robot using TP-model transformation," in *Proceedings of the IEEE International Conference on Information and Automation, ICIA '14*, pp. 1046–1050, Hailar, China, July 2014.

[65] 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 Systems Journal*, vol. 6, no. 3, pp. 488–498, 2012.

[66] K. K. Wu and Y. Yam, "Control stability of TP model transformation design via probabilistic error bound of plant model," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC '13*, pp. 1259–1264, UK, October 2013.

[67] 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 *Proceedings of the 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2010*, pp. 768–773, Canada, July 2010.

[68] K. Széll, A. Czmerk, and P. Korondi, "Automatika : časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije," *Identifikacija modela trenja s histerezom korištenjem tenzorskog produkta*, vol. 55, no. 4, pp. 463–473, 2015.

[69] 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.

[70] F. Kolonie and A. Poljugan, "Experimental control design by TP model transformation," in *Proceedings of the 2006 IEEE International Conference on Mechatronics, ICM*, pp. 666–671, Hungary, July 2006.

[71] P. Grof and Y. Yam, "Furuta Pendulum-A Tensor Product Model-Based Design Approach Case Study," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 2620–2625, Hong Kong, October 2015.

[72] Z. Petres, Polytopic decomposition of linear parameter-varying models by tensor-product model transformation, 2006.

[73] P. Galambos, *TP Model Transformation Based Control Design for Time-Delay Systems: Application in Telemanipulation*, 2013.

[74] C. Ariño and A. Sala, "Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1036–1046, 2007.

[75] P. Gróf, B. Takarics, Z. Petres, and P. Korondi, "Tensor product model transformation based control of a pneumatic cylinder," in *Proceedings of the 8th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, vol. 2008.

[76] F. Yang, Z. Chen, X. Liu, and B. Liu, "A new constant gain Kalman filter based on TP model transformation," in *Proceedings of the Chinese Intelligent Automation Conference*, vol. 254, pp. 305–312, Springer, 2013.

[77] Š. Ileš, J. Matuško, and F. Kolonić, "Tensor product transformation based speed control of permanent magnet synchronous motor drives," in *in 17th international conference on electrical drives and power electronics, EDPE '11 (5th Joint Slovak-Croatian Conference)*, 2011.

[78] S. Ileš, J. Matuško, and F. Kolonić, "TP transformation based control of rotary pendulum," in *Proceedings of the 34th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO '11*, pp. 833–839, IEEE, Opatija, Croatia, May 2011.

[79] R. Precup, C. Dragos, S. Preitl, M. Radac, and E. Petriu, "Tensor product models for automotive applications," in *Proceedings of the 14th International Conference on System Theory and Control (ICSTC '10)*, 2010.

[80] P. Korondi, "Sector sliding mode design based on tensor product model transformation," in *Proceedings of the INES 2007 - 11th International Conference on Intelligent Engineering Systems*, pp. 253–258, Hungary, July 2007.

[81] Y. Kunii, B. Solvang, G. Sziebig, and P. Korondi, "Tensor product transformation based friction model," in *Proceedings of the INES 2007 - 11th International Conference on Intelligent Engineering Systems*, pp. 259–264, Hungary, July 2007.

[82] P. Korondi, "Sliding sector design based on tensor product model transformation," *Transaction on Automatic Control and Computer Science*, vol. 51, no. 1, pp. 101–108, 2006.

[83] P. Korondi, P. Bartal, and F. Koloni, "Friction model based on tensor product transformation," in *Proceedings of the in 7th International Symposium of Hungarian Researchers on Computational Intelligence*, 2006.

[84] G. Hancke and A. Szeghegyi, "nonlinear control via tp model transformation: The tora system example," in *Proceedings of the Symposium on Applied Machine Intelligence (SAMI, '04)*, pp. 333–340, Herlany, Slovakia, 2004.

[85] G. Hancke and A. Szeghegyi, "Application study of the tp model transformation in the control of an inverted pendulum," in *Proceedings of the in International Conference on Computational Cybernetics (ICCC)*, 2003.

[86] S. Ileš, F. Kolonić, and J. Matuško, "Linear matrix inequalities based H∞ control of gantry crane using tensor product transformation," in *Proceedings of the 18th International Conference on Process Control*, pp. 92–99, Tatranská Lomnica, Slovakia, 2011.

[87] A. Rövid, L. Szeidl, and P. Várlaki, "On tensor-product model based representation of neural networks," in *Proceedings of the 15th International Conference on Intelligent Engineering Systems, INES 2011*, pp. 69–72, Slovakia, June 2011.

[88] 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 Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 911–922, 2009.

[89] K. Tanaka, H. Ohtake, T. Seo, M. Tanaka, and H. O. Wang, "Polynomial fuzzy observer designs: A sum-of-squares approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 5, pp. 1330–1342, 2012.