A new algorithm for RNO-INO type Tensor Product model representation

Péter Várkonyi, Domonkos Tikk, Péter Korondi and Péter Baranyi
Budapest University of Technology and Economy;
Computer and Automation Research Institute of the Hungarian Academy of Sciences,
E-mail: vpeter@mit.bme.hu, baranyi@sztaki.hu.

***Abstract* — A recently developed technique of modelling parameter varying dynamical systems is the Tensor Product model representation, where a system is decomposed to convex combination of several parameter-invariant models. Effectiveness and tractability of the representation depends strongly on the type of the applied convex combination. This paper presents an algorithm, which generates a special type of convex combination, the RNO-INO representation. An example is also introduced.**

I. INTRODUCTION

The Tensor Product (TP) model form is a representation of parameter-dependent dynamical systems, which has been recently used in model control [3, 4]. At the TP model representation, a parameter-dependent model is replaced by a parameter-dependent convex combination of several parameter-invariant models. The combination is defined by weight functions of each parameter of the model. Many different TP representations of a specific model can be constructed, and the feasibility and behavior of the representation depends heavily on the type of the time-invariant models and the weight functions. No general algorithm has been developed to find a proper representation of a given dynamical model. Instead, many types of representations have been proposed by different authors (eg. [6, 7]) and it is usually decided empirically, which kind of representation to be used in a given case.

The type of a TP representation is primarily characterized by properties of the weight functions. One of these is the 'RNO-INO', proposed in a slightly different form by [8]. They also presented an algorithm to construct this type of representation. Our paper presents a different, *improved algorithm* for producing TP model representations with RNO-INO type weight-functions. The new algorithm provides a more compact model representation than [8] (some details about the difference are highlighted in Section VI.).

The more or less heuristic conditions such as the 'RNO-INO' are only useful if they provide advantageous representations of some models of practical interest. The description of a specific example, namely the observer design of a prototypical aeroelastic wing section can be found in the same proceedings [5]).

In Section II. some definitions are listed. In part III., the matrix decomposition problem associated with the RNO-INO type TP model representation is shown. Section IV. presents the algorithm for the matrix decomposition problem in a special case and based on this, the general case is discussed in Section V.. Finally, Section VI. is devoted to applications of the algorithm.

II. DEFINITIONS

Sum Normalization (SN) A matrix is called SN type if the sum of the elements in each row is 1. Note that erasing the last column of an SN type matrix results no loss of information. Such a truncation of an arbitrary SN type matrix $\mathbf{X}$ is denoted by $\overline{\mathbf{X}}$.

Non negativeness (NN) A matrix is called NN type if all its elements are non-negative.

Normalization (NO) A matrix is called NO type if it is SN and NN type and the maximum of each column is 1.

Relaxed normalization (RNO) A matrix is called RNO type if it is SN and NN type and the maximum of each column is equal.Note that these maxima are between 0 and 1.

Inverse normalization (INO) A matrix is called INO type if it is SN and NN type, and the minimum of each column is 0.

III. RNO-INO TYPE TP REPRESENTATION

The basic elements of TP model transformation are not introduced here, but one can find a brief summary of this framework in the same book [5]). We only use the fact that constructing a TP representation corresponds to the following simple matrix decomposition problems associated to each parameter of the model:

$$\mathbf{S} = \mathbf{U} \cdot \mathbf{Z} \tag{1}$$

$\mathbf{S}$ is a given $n \times r$ 'system matrix' of the parameter varying system, usually with $n >> r$. The $n \times r$ matrix $\mathbf{U}$ and the $r \times r$ size $\mathbf{Z}$ have to be constructed. The former one corresponds to the weight functions and the later one is a 'system matrix' of the parameter-invariant models. The matrix $\mathbf{U}$ is always SN, which corresponds to the fact that the sum of the weight functions is 1, and it is also NN, since $\mathbf{U}$ defines a *convex combination* of the parameter invariant models. Further properties of $\mathbf{U}$ characterize special types of TP representations. In our case, an RNO-INO representation corresponds to *an RNO and INO type matrix* $\mathbf{U}$.

The rest of our paper deals with the following problem: given is an initial TP representation of a specific model, ie. a decomposition of the form (1), where $\mathbf{U}$ is SN and NN, but not RNO-INO type, we construct another decomposition $\mathbf{S} = \mathbf{U}' \cdot \mathbf{Z}'$, where $\mathbf{U}'$ is SN, NN, RNO and INO. As a further simplification we only deal with the following decomposition of $\mathbf{U}$:

$$\mathbf{U} = \mathbf{U}' \cdot \Theta, \tag{2}$$

where the sizes of $\mathbf{U}'$ and Θ are $n\times r$ and $r\times r$, respectively. Equation (2), substituted in (1) provides the final, RNO-INO type decomposition (with $\mathbf{Z}' = \Theta\cdot\mathbf{Z}$).

One can easily verify that the SN property of Θ in eq. (2) follows from the SN property of $\mathbf{U}'$ and $\mathbf{U}$. What is more, (2) is equivalent of the following truncated form (see the meaning of 'overlines' at the definition of SN type matrices):

$$\overline{\mathbf{U}} = \mathbf{U}'\cdot\overline{\Theta}. \tag{3}$$

We will refer to a geometrical interpretation of the above equation, which has been proposed by [7]: hence $\mathbf{U}'$ is SN and NN, the row vectors of $\overline{\mathbf{U}}$ are generated by eq. (3) as convex combinations of the row vectors of $\overline{\Theta}$, ie. the simplex determined by the latter vectors bounds the points associated to the rows of $\overline{\mathbf{U}}$. Instead of exact algebraic proofs, we will refer to this geometrical interpretation several times. Notice that an NO type matrix $\mathbf{U}'$ would geometrically mean that $\overline{\Theta}$ is the *exact convex hull* of $\overline{\mathbf{U}}$, which shows that the NO condition usually can not be satisfied. The milder INO property means that $\overline{\Theta}$ bounds $\overline{\mathbf{U}}$ so, that each face of the simplex contains at least one of the row vectors of $\overline{\mathbf{U}}$. The meaning of the RNO condition is less illustrative.

We continue the paper by a further simplification of the problem. In section IV. and V. an invertible, inhomogenous linear transformation T is constructed, for which $\mathrm{T}(\overline{\mathbf{U}}) = \overline{\mathbf{U}}'$ (and $\mathbf{U}'$ is SN, NN, INO and RNO). Finding T provides the solution of the original problem: let the $r\times r$ unit matrix be denoted by $\mathbf{I}_r$. It follows straightforward from the above shown geometrical interpretation that the trivial equation $\mathbf{U}' = \mathbf{U}'\mathbf{I}_r$ is equivalent of the following ones:

$$\overline{\mathbf{U}}' = \mathbf{U}'\overline{\mathbf{I}}_r \tag{4}$$
$$T^{-1}(\overline{\mathbf{U}}') = \mathbf{U}'\cdot T^{-1}(\overline{\mathbf{I}}_r) \tag{5}$$
$$\mathbf{U} = \mathbf{U}'\cdot T^{-1}(\mathbf{I}_r) \tag{6}$$

Equation (6) is the desired (2) type decomposition of $\mathbf{U}$.

The description of the proposed algorithm is split to two parts. Section IV. deals with the decomposition of $\mathbf{U}$ matrices with 2 columns; matrices, which have more than two columns are discussed in Section V..

IV. MATRICES WITH TWO COLOUMNS

If $\mathbf{U}$ is an SN type matrix with two columns, a transformation T_0, which makes the $\overline{\mathbf{U}}$ *vector* RNO and INO is the following:

$$\mathbf{q} = T_0(\mathbf{p}) : q_k = \frac{p_k - \min_i(p_i)}{\max_i(p_i) - \min_i(p_i)} \tag{7}$$

where $\mathbf{p}_k$ and $\mathbf{q}_k$ denote the k^{th} elements of the corresponding vectors. If $\overline{\mathbf{U}}' = T_0(\overline{\mathbf{U}})$, $\mathbf{U}'$ is obviously SN,NN, INO and RNO (the maxima of both columns are 1).

V. MATRICES WITH MORE THAN TWO COLUMNS

In this section, we show an invertible linear transformation, which makes $\overline{\mathbf{U}}$ INO-RNO, if $\mathbf{U}$ has more than two columns. Subsection V./A. deals with a basic step of the algorithm, Subsection V./B. contains the proof of two lemmas, and finally Subsection V./C. shows the course of the complete algorithm.

A. *Basic step of the algorithm*

Let a be a non-negative constant and consider the product T of the following four inhomogenous linear transformations:

$$\mathbf{Q} = T_A(\mathbf{P},a) : \quad Q_{i,j}(a) = P_{i,j} + \tfrac{a-1}{\rho}\left(\textstyle\sum_{k=1}^{\rho} P_{i,k} - 1\right) \tag{8}$$
$$\mathbf{Q} = T_B(\mathbf{P}) : \quad Q_{i,j} = P_{i,j} - \min_l(P_{l,j}) \tag{9}$$
$$\mathbf{Q} = T_C(\mathbf{P}) : \quad Q_{i,j} = \frac{P_{i,j}}{\max_l(P_{l,j})} \tag{10}$$
$$\mathbf{Q} = T_D(\mathbf{P}) : \quad Q_{i,j} = \frac{P_{i,j}}{\max_l(\sum_{k=1}^{\rho} P_{l,k})} \tag{11}$$

where ρ is the number of columns in the input matrix $\mathbf{P}$, and the $(i,j)^{th}$ element of any matrix $\mathbf{X}$ is denoted by $X_{i,j}$. See also a geometrical illustration of the above transformations in Figure 1.

Let $\mathbf{W}$ be an arbitrary SN type matrix with $\rho > 2$ columns. If $\overline{\mathbf{W}}' = T(\overline{\mathbf{W}})$, $\mathbf{W}'$ is obviously SN; it is NN and INO, because T_B changes the smallest element of the first $r-1$ columns to 0 and T_C and T_D preserve this property, while the smallest element of the r^{th} column is 0 because T_D and the SN property of $\mathbf{W}'$. Finally, $\mathbf{W}'$ is almost RNO: the maxima of the az $1., 2., \dots, (r-1)^{th}$ columns are equal due to T_C, but the maximum of the last one is different. The RNO condition is completely satisfied if

$$f(a) = \max_k\left(W'_{k,r}(a)\right) - \max_k\left(W'_{k,1}(a)\right) =$$
$$= 1 - \min_l\left(\sum_{k=1}^{\rho-1} W'_{l,k}(a)\right) - \max_k\left(W'_{k,1}(a)\right) = 0. \tag{12}$$

We will show later

Lemma 1:

$$\lim_{a\to\infty} f(a) = 1 - 0 - \frac{1}{\rho-1} \tag{13}$$

and also

Lemma 2: If $T_A(\overline{\mathbf{W}},0)$ is INO and RNO, then

$$f(0) = 1 - 1 - \max_m\left(W'_{m,l}(0)\right) \tag{14}$$

If the condition of *Lemma 2* holds for $\overline{\mathbf{W}}$, (12) has a positive solution a_0, because $f(a)$ is continuous and $f(0) < 0 < lim_{a\to\infty} f(a)$. This value can be found numerically and $\mathbf{W}$ can be transformed to RNO-INO form.

B. *Proof of two lemmas*

Proof of Lemma 1: If a approaches infinity, all row vectors of $T(\overline{\mathbf{W}},a)$ lie approximately on the $x_1 = x_2 = \ldots = x_\rho$ line of the ρ dimensional space (due to T_A and T_B, see Figure 2). One row will be the zero vector (because of T_B), another will be approximately $[1/\rho, 1/\rho, \dots, 1/\rho]$ (because T_D) and all the rest will be between these two vectors. The statement of *Lemma 1* follows straightforward from these facts.

Proof of Lemma 2: If $T_A(\overline{\mathbf{W}},0)$ is INO, then T_B, applied on this matrix, becomes identity and because the RNO property, the product of T_C and T_D is also identity. Thus, in this case $\overline{\mathbf{W}}'(0) = T_A(\overline{\mathbf{W}},0)$. Moreover, this matrix is SN because of $a=0$, (cf. Figure 2) ie.

$$\min_l\left(\sum_{k=1}^{\rho-1} W'(0)_{l,k}\right) = 1 \tag{15}$$

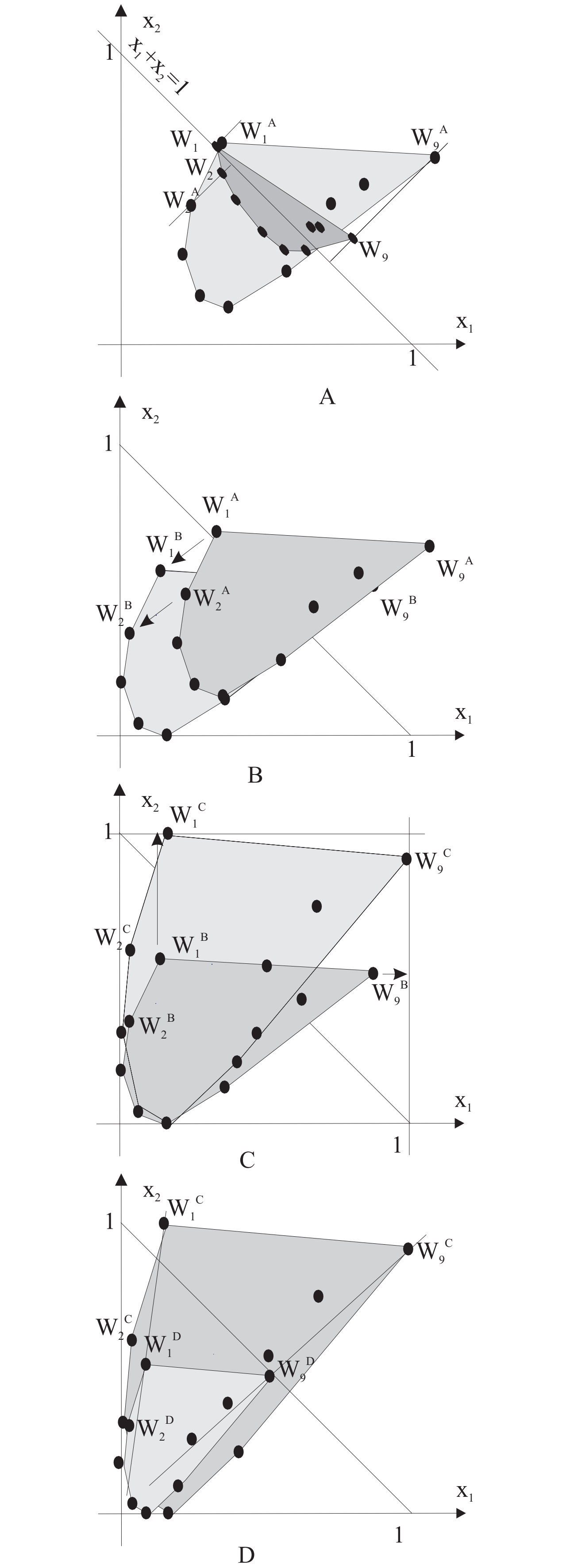

Fig. 1: A geometrical illustration of the transformation T: the four panels show the effect of the successive application of T_A, T_B, T_C and T_D on the example of a 9×2 size matrix $\overline{\mathbf{W}}$. Point W_i^X correspond to the i^{th} row vector of $\overline{\mathbf{W}}$ after transformation T_X, $X\in\{A,B,C,D\}$. The geometrical effect of the transformation T_X is: A: perpendicular stretching by a from the line d ($a=4$ in the figure), which would mean projection to d if $a=0$. B: shifting to the axis' C: perpendicular stretching from the two coordinate axis', to fill the unit square. D: enlarging from the origin to hit line $x_1+x_2=1$.

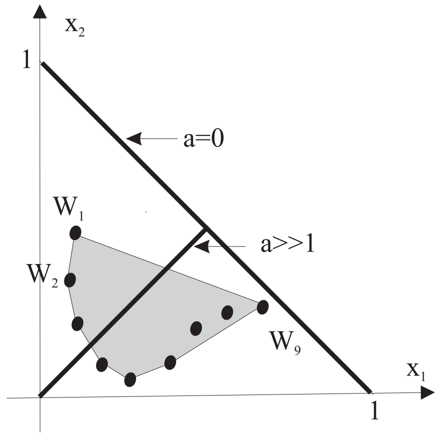

Fig. 2: A geometrical illustration of the transformations T applied on a 9×2 size matrix $\overline{\mathbf{W}}$ if $a>>1$, as well as if $a=0$ and the condition of *Lemma 2* is satisfied. In both cases, the points are transformed to a line. Notations are the same as in Figure 1.

Thus, eq. (14) holds. Q.E.D.

C. *The complete algorithm*

It was shown in Section V./A., that we can transform an arbitrary SN type matrix $\mathbf{W}$ to RNO-INO type, if $T_A(\overline{\mathbf{W}},0)$ is RNO-INO type (cf. *Lemma 2*). If the former matrix has ρ columns, notice that the latter has only $\rho-1$. This way the problem can be reduced step-by-step to the trivial case $r=2$ (see Section IV.). In this subsection, the complete algorithm is presented.

Apply the notation $\mathbf{U}^{(1,r)}=\mathbf{U}$ and create $\mathbf{U}^{(1,k-1)}$ ($k=r,r-1,\ldots,3$) from the

$$\mathbf{U}^{(1,k-1)}=T_A(\overline{\mathbf{U}}^{1,(k)},0) \tag{16}$$

recursion. The matrix $\mathbf{U}^{(1,2)}$ is SN type and it is of size $n\times 2$. It can be transformed to RNO-INO type by the transformation T_0, as shown in Section IV.. Let us apply the notation $T_0(\mathbf{U}^{(1,2)})=\mathbf{U}^{(3,2)}$. After this,repeat the following steps $r-2$ times with $k=3,4,\ldots r$ respectively:

- 1: Let $\mathbf{U}^{(2,k)}$ be created from $\mathbf{U}^{(1,k)}$ as

$$\overline{\mathbf{U}}^{(2,k)}=T^*(\overline{\mathbf{U}}^{(1,k)})=\mathbf{U}^{(3,k-1)}-\mathbf{U}^{(1,k-1)}+\overline{\mathbf{U}}^{(1,k)} \tag{17}$$

- 2: Notice that the T_A and T^* transformations are commutative, which is easy to see from their geometrical meaning: e.g. if $a=0$, T_A projects the row vectors of the input matrix orthogonally to the subspace $\sum x_i=1$ of the ρ dimensional space (cf. Figure 1/A), and T^* shifts the row vectors, *parallel* to this subspace. Because the commutativity of the two transformations (see also eqs. (16), (17)) ,

$$\begin{aligned} T_A(\overline{\mathbf{U}}^{(2,k)},0) &= T_A(T^*(\overline{\mathbf{U}}^{(1,k)}),0)= \\ =T^*(T_A(\overline{\mathbf{U}}^{(1,k)},0)) &= T^*(\overline{\mathbf{U}}^{(1,k-1)})=\mathbf{U}^{(3,k-1)} \end{aligned} \tag{18}$$

which is SN, RNO and INO type. Hence the condition of *Lemma 2* is satisfied, $\overline{\mathbf{U}}^{(2,k)}$ can be transformed to RNO-INO type as shown in Section V./A.. Let the image of $\overline{\mathbf{U}}^{(2,k)}$ be $\overline{\mathbf{U}}^{(3,k)}$.

Finally, the matrix $\mathbf{U}' = \mathbf{U}^{(3,r)}$ is RNO-INO type and it has been constructed from $\mathbf{U}$ by an invertible, linear, inhomogenous transformation.

VI. APPLICATION OF THE RNO-INO TYPE MODEL REPRESENTATION

As already mentioned in the introductory part, the Tensor Product model representation is a general framework of modelling parameter-dependent dynamical systems. It provides a good approximation of many models even in cases where analytical handling would be too complicated. Many applications [5] show, that special types of Tensor Product representation produce tractable results in most cases. The RNO-INO type is one of the advantageous types of representations, A former algorithm for an eq. (2) type RNO-INO decomposition has been published in [7]. Their algorithm produced an $n \times 2 \cdot r$ size matrix $\mathbf{U}'$ and a $2 \cdot r \times r$ size Θ, which means that the original system was decomposed to the convex combination of $2r$ parameter-invariant systems. The improved algorithm of this paper produces $n \times r$ size $\mathbf{U}'$ and an $r \times r$ size Θ, which corresponds to the convex combination of only r systems. Further decrease in the above matrix sizes is not possible (hence an $r-1$ dimensional non-degenerated convex hull has at least r points).

The main difficulty of creating a TP model representation is the lack of guarantee that any of the proposed types of representations is tractable at a specific model. That is the main point, while several types of representations are in use at the same time.

The proposed RNO-INO algorithm has been used at a recently analyzed example: a prototypical wing-section, which produces spontaneous limit-cycle oscillation but should be stabilized for safety reasons. Two parts of the stabilizer have been designed, a controller unit and an observer unit [1, 2, 5]. At the latter one, the RNO-INO type representation gave the only tractable solution among several others. Hence this model is the subject of another paper in the same proceedings, we do not present here any details about the model, the representation or the simulation results. At the same time, we hope that the presented algorithm will help modelling several other examples.

VII. ACKNOWLEDGEMENT

This research was supported by the Hungarian Scientific Research Fund (OTKA) Grant No. TS 49885 and F049838.

REFERENCES

[1] P. Baranyi. Output-feedback design of 2-D aeroelastic system. *Journal of Guidance, Control, and Dynamics (in Press)*.

[2] P. Baranyi. Tensor product model based control of 2-D aeroelastic system. *Journal of Guidance, Control, and Dynamics (in Press)*.

[3] P. Baranyi. TP model transformation as a way to LMI based controller design. *IEEE Transaction on Industrial Electronics*, 51(2):387–400, April 2004.

[4] P. Baranyi, D. Tikk, Y. Yam, and R. J. Patton. From differential equations to PDC controller design via numerical transformation. *Computers in Industry, Elsevier Science*, 51:281–297, 2003.

[5] P. Baranyi, P. Várkonyi, and Y. Yam. Different affine decomposition of the model of the prototypical aeroelastic wing section by TP model transformation. Part I. In *this Proceedings*.

[6] Y. Yam. *Fuzzy Identification with SVD and Subdomain Normality.* World Scientific, theory and practice of control and systems, A. Tornambé, G. Conte, A. M. Perdon (Eds.) edition, 1999.

[7] Y. Yam, P. Baranyi, and C. T. Yang. Reduction of fuzzy rule base via singular value decomposition. *IEEE Transaction on Fuzzy Systems*, 7(2):120–132, 1999.

[8] Y. Yam, C. T. Yang, and P. Baranyi. *Singular Value-Based Fuzzy Reduction with Relaxed Normalization Condition*, volume 128 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, interpretability issues in fuzzy modeling, J. Casillas, O. Cordón, F.Herrera, L.Magdalena (Eds.) edition, 2003.