PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 49, NO. 1–2, PP. 109–122 (2005)

REFERENCE SIGNAL TRACKING CONTROL OF THE TORA SYSTEM: A CASE STUDY OF TP MODEL TRANSFORMATION BASED CONTROL

Zoltán PETRES and Péter BARANYI

Department of Telecommunication and Telematics,
Budapest University of Technology,
H–1117 Budapest, Magyar Tudósok körútja 2.

Received: April 15, 2005

Abstract

The paper presents a case study of the TP (Tensor Product) model transformation in the control of a non-linear benchmark problem. We design a non-linear controller of translational oscillation with an eccentric rotational proof mass actuator (TORA) system via TP model transformation and LMI (Linear Matrix Inequality) based controller design technique that is also capable of the reference signal tracking control. The main contribution of the paper is to show that both numerical methods the TP model transformation and the LMI can readily be executed computer independently on the given problem and without analytical derivations, that, hence, lead to a fast way of controller designs for a class of engineering control problems. Numerical simulation is used in the paper to provide empirical validation of the control results.

Keywords: non-linear control design, tensor product model, linear matrix inequalities, parallel distributed compensation, TORA system.

1. Introduction

Recently a control design method was proposed for the stabilization of parameter varying non-linear state-space models [1, 2, 3]. This method is based on two numerical steps. In the first step the TP model transformation [2] is executed, while in the second step LMIs are solved under the PDC (Parallel Distributed Compensation) framework, that also includes the feasible solution of LMIs'. The book [4] refers to a great number of related papers dealing with PDC design framework. The first step is capable of transforming a given state-space model into a tensor product form (which is identical with a class of the Takagi–Sugeno inference operator based fuzzy model, see in Section 4) whereupon design techniques of the PDC framework can immediately be executed. The second step results in a controller according to various different control specifications.

It is worth noticing here that both steps are executed numerically by computers. This implies two advantages such as:

1. the controller can be derived automatically, without analytic derivations;
2. the identified model which the control design method starts with can be defined either by analytical equations or by other soft-computing techniques,

for instance by neural networks, fuzzy logic systems, or algorithms based on *Rudas*-type generalized operators [5, 6].

The main goal of this paper is to study, via the control of the TORA system example, how to execute the TP model transformation based control design method and to show its performance in case of reference signal tracking control. This control problem has a great comparative literature related to different control theories, and also a special issue of the *International Journal of Robust and Non-Linear Control* was devoted to describe the control problem and to present several control design methods including optimal control theory, Lyapunov backstepping, passivity theory, fuzzy logic, computing with words, etc. The overview of this literature is behyond the scope of this paper, but we refer the reader to [7, 8, 9, 4, 10]

The rest of the paper is organized as follows: Section 2 introduces the notation being used in this paper. Section 3 briefly summarizes some preliminaries and defines the convex state-space TP model. Section 4 presents the TP model transformation and Section 5 describes the LMI based controller design. Section 6 illustrates the case study of this paper, the TORA system and the introduced controller design theory is applied. Finally Section 7 concludes the paper.

2. Nomenclature

This section is devoted to introduce the notations being used in this paper.

- $\{a, b, \dots\}$ = scalar values
- $\{\mathbf{a}, \mathbf{b}, \dots\}$ = vectors
- $\{\mathbf{A}, \mathbf{B}, \dots\}$ = matrices
- $\{\mathcal{A}, \mathcal{B}, \dots\}$ = tensors
- $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ = vector space of real valued $(I_1 \times I_2 \times \cdots \times I_N)$-tensors
- $\diamond_{i,j,n}, \dots$ = indices, they define lower order: for example, an element of matrix $\mathbf{A}$ at row-column number i, j is symbolized as $(\mathbf{A})_{i,j} = a_{i,j}$. Systematically, the ith column vector of $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e. $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots\end{bmatrix}$
- $\diamond_{I,J,N}, \dots$ = index upper bound: for example: $i = 1 \dots I$, $j = 1 \dots J$, $n = 1 \dots N$ or $i_n = 1 \dots I_n$
- $\mathbf{A}_{(n)}$ = n-mode matrix of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$
- $\text{rank}_n(\mathcal{A})$ = n-mode rank of tensor $\mathcal{A}$
- $\mathcal{A} \times_n \mathbf{U}$ = n-mode matrix-tensor product
- $\mathcal{A} \otimes_n \mathbf{U}_n$ = multiple product as $\mathcal{A} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_N \mathbf{U}_N$
- $\mathbf{A}^+$ = the pseudo inverse of matrix $\mathbf{A}$

Detailed discussion of tensor notations and operations is given in [11].

3. Basic Concepts of Tensor Product Model Transformation

3.1. Parameter-varying State-space Model

Consider parameter-varying state-space model:

$$\begin{aligned} s\mathbf{x}(t) &= \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t), \end{aligned} \quad (1)$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state vector $\mathbf{x}(t)$. The system matrix

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{pmatrix} \in \mathbb{R}^{O \times I} \quad (2)$$

is a parameter-varying object, where $\mathbf{p}(t) \in \Omega$ is time varying N-dimensional parameter vector, where $\Omega = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_N, b_N] \subset \mathbb{R}^N$ is a closed hypercube. $\mathbf{p}(t)$ can also include some elements of $\mathbf{x}(t)$. Further, for a continuous-time system $s\mathbf{x}(t) = \dot{\mathbf{x}}(t)$; and for a discrete-time system $s\mathbf{x}(k) = \mathbf{x}(k+1)$ holds.

3.2. Convex State-space TP Model

Eq. (2) can be approximated for any parameter $\mathbf{p}(t)$ as a convex combination of the R linear time-invariant (LTI) system matrices $\mathbf{S}_r$, $r = 1 \dots R$. Matrices $\mathbf{S}_r$ are also termed as vertex system matrices. Therefore, one can define basis functions $w_r(\mathbf{p}(t)) \in [0, 1] \subset \mathbb{R}$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ belongs to the convex hull of $\mathbf{S}_r$ as $\mathbf{S}(\mathbf{p}(t)) = \text{co}\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}$, where vector $\mathbf{w}(\mathbf{p}(t))$ contains the basis functions $w_r(\mathbf{p}(t))$ of the convex combination. This kind of approximation is termed as, for instance, basis function based approximation, B-spline approximation, or tensor product approximation, see Chapter 3.2 of [12] and [13], and one can find the above model as *polytopic* model in control theories. The control design methodology, to be applied in this paper, uses univariate basis functions. Thus, the explicit form of the convex combination in terms of tensor product becomes:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n(t)) \mathbf{S}_{i_1,i_2,\dots,i_N} \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}. \quad (3)$$

The *Eq.* (3) is termed as TP model in this paper. Function $w_{n,j}(p_n(t)) \in [0, 1]$ is the jth univariate basis function defined on the nth dimension of Ω, and $p_n(t)$ is the nth element of vector $\mathbf{p}(t)$. The I_n $(n = 1, \dots, N)$ is the number of univariate basis functions used in the nth dimension of the parameter vector $\mathbf{p}(t)$. The multiple index $(i_1, i_2, \dots, i_N)$ refers to the LTI system corresponding to the i_nth basis function in the nth dimension. Hence, the number of LTI vertex systems $\mathbf{S}_{i_1,i_2,\dots,i_N}$ is obviously $R = \prod_n I_n$.

Remark 1 *Eq. (3) is also known as the explicit inference form of the Takagi–Sugeno inference operator based fuzzy model (TS fuzzy model for brevity). For instance, (3) is defined by fuzzy rules:*

$$\textbf{IF} \quad w_{1,i_1}(p_1(t)) \quad \textbf{AND} \quad w_{2,i_2}(p_2(t)) \quad \dots \quad w_{N,i_N}(p_N(t)) \quad \textbf{THEN} \quad \mathbf{S}_{i_1,i_2,\dots,i_N},$$

where functions $w_{n,i_n}(p_n(t))$ represent the antecedent fuzzy sets and $\mathbf{S}_{i_1,i_2,\dots,i_N}$ represents the consequent systems.

One can rewrite (3) in the concise TP form as:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \underset{\delta}{\approx} \left(\mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)) \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{4}$$

that is

$$\mathbf{S}(\mathbf{p}(t)) \underset{\varepsilon}{\approx} \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)).$$

Here, row vector $\mathbf{w}_n(p_n) \in \mathbb{R}^{I_n}$ contains the basis functions $w_{n,i_n}(p_n)$, the $N+2$-dimensional coefficient tensor $S \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N \times O \times I}$ is constructed from the LTI vertex system matrices $\mathbf{S}_{i_1,i_2,\dots,i_N} \in \mathbb{R}^{O \times I}$. The first N dimensions of S are assigned to the dimensions of Ω. The convex combination of the LTI vertex systems is ensured by the conditions:

Definition 1 *The TP model (4) is convex if:*

$$\forall n, i, p_n(t) : w_{n,i}(p_n(t)) \in [0, 1]; \tag{5}$$

$$\forall n, p_n(t) : \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \tag{6}$$

This simply means that $\mathbf{S}(\mathbf{p}(t))$ is within the convex hull of LTI vertex systems $\mathbf{S}_{i_1,i_2,\dots,i_N}$ for any $\mathbf{p}(t) \in \Omega$.

Remark 2 $\mathbf{S}(\mathbf{p}(t))$ *has finite TP model representation in many cases ($\varepsilon = 0$ in (4)). However, one should face that exact finite element TP model representation does not exist in general ($\varepsilon > 0$ in (4)), see [14]. In this case $\varepsilon \to 0$, when the number of LTI systems involved in the TP model goes to ∞.*

We define here a further characteristic of the convex TP model.

Definition 2 *The LTI vertex systems form a tight convex hull if their corresponding basis functions have the following feature:*

$$\forall n, i_n; \max_{p_n(t)}(w_{n,i_n}(p_n(t))) \underset{\delta_{n,i_n}}{\approx} 1, \tag{7}$$

where $\forall\delta_{n,i_n}$ *is as small as possible. For instance, the basis functions are determined subject to*

$$\text{minimize}(\|\delta\|_{L_2}),$$

where vector δ *consists of all* δ_{n,i_n}.

4. Tensor Product Model Transformation

The goal of the TP model transformation is to transform a given state-space model (1) into convex TP model, in which the LTI systems form a tight convex hull. Namely, the TP model transformation results in (4) with conditions (5) and (6), and searches the LTI systems as points of a tight convex hull of $\mathbf{S}(\mathbf{p}(t))$, see (7).

The TP model transformation is a numerical method and has three key steps. The first step is the discretization of the given $\mathbf{S}(\mathbf{p}(t))$ via the sampling of $\mathbf{S}(\mathbf{p}(t))$ over a huge number of points $\mathbf{p} \in \Omega$. The sampling points are defined by a dense hyper rectangular grid. In order to loose minimal information during the discretization we apply as dense grid as possible. The second step extracts the LTI vertex systems from the sampled systems. This step is specialized to find the minimal number of LTI vertex systems as the vertex points of the tight convex hull of the sampled systems. The third step constructs the TP model based on the LTI vertex systems obtained in the second step. It defines the continuous basis functions to the LTI vertex systems.

Method 1 (TP Model Transformation)
Step 1) Discretization

- *Define the transformation space* Ω *as:* $\mathbf{p}(t) \in \Omega : [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_N, b_N]$.
- *Define a hyper rectangular grid by equidistantly located grid-lines:* $g_{n,m_n} = a_n + \frac{b_n - a_n}{M_n - 1}(m_n - 1)$, $m_n = 1 \ldots M_n$. *The numbers of the grid lines in the dimensions are* M_n.
- *Sample the given function* $\mathbf{S}(\mathbf{p}(t))$ *over the grid-points:*

$$\mathbf{S}^s_{m_1,m_2,\ldots,m_N} = \mathbf{S}(\mathbf{p}_{m_1,m_2,\ldots,m_N}) \in \mathbb{R}^{O\times I},$$

  where $\mathbf{p}_{m_1,m_2,\ldots,m_N} = \begin{pmatrix} g_{1,m_1} & \ldots & g_{N,m_N}\end{pmatrix}$. *Superscript 's' means 'sampled'.*
- *Store the sampled matrices* $\mathbf{S}^s_{m_1,m_2,\ldots,m_N}$ *into the tensor* $\mathcal{S}^s \in \mathbb{R}^{M_1\times M_2\times\cdots\times M_N\times O\times I}$

Step 2) Extracting the LTI vertex systems
This step uses Higher-Order Singular Value Decomposition (HOSVD), and transformations Non-negativeness (NN), Sum Normalization (SN) and Normalization (NO). The studies of HOSVD can be found in a large variety of publications. This

paper uses the concept and tensor notation of HOSVD as discussed in [11]. The SN, NN and NO transformations are introduced in [15] and [16].

This step executes HOSVD, extended with NN, SN and NO transformations, on the first N dimensions of tensor $\mathcal{S}^s$. During performing the HOSVD we discard all zero or small singular values σ_k and their corresponding singular vectors in all dimensions. As a result we have

$$\mathcal{S}^s \underset{\gamma}{\approx} \mathcal{S} \underset{n}{\otimes} \mathbf{U}_n,$$

where the error γ is bounded as:

$$\gamma = \left(\left\| \mathcal{S}^s - \mathcal{S} \underset{n}{\otimes} \mathbf{U}_n \right\|_{L_2} \right)^2 \leq \sum_k \sigma_k^2. \tag{8}$$

The resulting tensor $\mathcal{S}$, with the size of $(I_1 \times I_2 \times \cdots \times I_N \times O \times I)$, where $\forall n : I_n \leq M_n$, contains the LTI vertex systems, and is immediately substitutable into (4). The NN and SN transformations guarantee that the resulting LTI vertex systems form a convex hull of the sampled systems in $\mathcal{S}^s$. When the transformation NO is executed the resulting LTI systems form the tight convex hull of the sampled systems.

The software implementations of HOSVD, NN, SN and NO are rather simple, for instance, in MATLAB.

Step 3) Constructing continuous basis system

- *One can determine the discretized points of the basis easily from matrices $\mathbf{U}_n$. The i_nth column vector $\mathbf{u}_{n,i_n=1\ldots I_n}$ of matrix $\mathbf{U}_n \in \mathbb{R}^{M_n \times I_n}$ determines one discretized basis function $w_{n,i_n}(p_n(t))$ of variable $p_n(t)$. The values u_{n,m_n,i_n} of column i_n define the values of the basis function $w_{n,i_n}(p_n(t))$ over the grid-lines $p_n(t) = g_{n,m_n}$:*

$$w_{n,i_n}(g_{n,m_n}) = u_{n,m_n,i_n}.$$

- *The basis functions can be determined over any points by the help of the given $\mathbf{S}(\mathbf{p}(t))$. In order to determine the basis functions in vector $\mathbf{w}_d(p_d)$, let p_k be fixed to the grid-lines as:*

$$p_k = g_{k,1} \quad k = 1\ldots N, \quad k \neq d.$$

Then for p_d:

$$\mathbf{w}_d(p_d) = (\mathbf{S}(\mathbf{p}))_{(3)} \left(\left(\mathcal{S} \underset{k}{\otimes} \mathbf{u}_{k,1} \right)_{(n)} \right)^+,$$

where vector $\mathbf{p}$ consists of elements p_k and p_d as $\mathbf{p} = g_{1,1} \ldots p_d \ldots g_{N,1}$, and superscript '$+$' denotes pseudo inverse and $\mathbf{u}_{k,1}$ is the first row vector of $\mathbf{U}_k$. The third-mode matrix $(\mathbf{S}(\mathbf{p}))_{(3)}$ of matrix $\mathbf{S}(\mathbf{p})$ is understood such that *matrix* $\mathbf{S}(\mathbf{p})$ *is considered as a three-dimensional tensor, where the length of the third dimension is one. This practically means that the matrix* $\mathbf{S}(\mathbf{p})$ *is stored into one row vector by placing the rows of* $\mathbf{S}(\mathbf{p})$ *next to each other, respectively.*

5. Determination of Controllers via PDC Design Framework

This section briefly introduces the main concept of the LMI design and calls LMI design theorems involving different control purposes. These LMI design theorems will be applied in the second part of this section to the TORA system.

As a result of the dramatic and continuing growth in computer power, and the advent of very powerful algorithms (and associated theory) for *convex optimization*, we can now solve very rapidly many *convex optimization* problems involving LMIs [17]. Many control problems and design specifications have LMI formulations [18, 19] what comes from the fact that LMI formulations have the ability to readily combine various design constraints or objectives in a numerical tractable manner. This is especially true for Lyapunov-based analysis and design.

As an alternative way of LMI based control design the PDC framework was introduced by TANAKA and WANG [4]. The PDC design framework determines one LTI feedback gain to each LTI vertex systems of a given convex TP model. The framework starts with the LTI vertex systems $\mathcal{S}$, and results in the vertex LTI gains $\mathcal{K}$ of the controller. The $\mathcal{K}$ is computed by the LMI based stability theorems. After having the $\mathcal{K}$, the control value $\mathbf{u}(t)$ is determined by the help of the same basis functions as used in (4):

$$\mathbf{u}(t) = -\left(\mathcal{K} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t))\right)\mathbf{x}(t). \tag{9}$$

The LMI theorems, to be solved under the PDC framework, are selected according to the stability criteria and the desired control performance. For instance, the speed of response, constraints on the state vector or on the control value can be considered via properly selected LMI based stability theorems. The present control design applies different LMI theorems to achieve global asymptotic stability and to enforce constraint on the control value.

In order to complete the paper let us recall briefly those LMI theorems, which will be applied in this paper. The derivations and the proofs of these theorems are fully detailed in [4].

Before dealing with the LMI theorems, we introduce a simple indexing technique in order to have direct link between the TP model form and the typical form of LMI formulations:

Method 2 (Index Transformation) *Let be*

$$\mathbf{S}_r = \begin{pmatrix} \mathbf{A}_r & \mathbf{B}_r \\ \mathbf{C}_r & \mathbf{D}_r \end{pmatrix} = \mathbf{S}_{i_1,i_2,\ldots,i_N},$$

where $r = \text{ordering}(i_1, i_2, \ldots, i_N)$ *(*$r = 1 \ldots R = \prod_n I_n$*). The function "ordering" results in the linear index equivalent of an* N*-dimensional array's index* $i_1, i_2, \ldots, i_N$*, when the size of the array is* $I_1 \times I_2 \times \cdots \times I_N$*. Let the basis functions be defined according to the sequence of* r*:*

$$w_r(\mathbf{p}(t)) = \prod_n w_{n,i_n}(p_n(t)).$$

First we call one of the simplest LMI design theorems. The controller design can be derived from the Lyapunov stability theorems for global and asymptotic stability as shown in [20, 4]:

Theorem 1 *(Global and asymptotic stabilization of the convex TP model (4))Assume a given state-space model in TP form (4) with conditions (5) and (6).*

Find $\mathbf{X} > 0$ *and* $\mathbf{M}_r$ *satisfying eq.*

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} + \mathbf{M}_r^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_r > 0 \tag{10}$$

for all r *and*

$$\begin{gathered} -\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} - \mathbf{X}\mathbf{A}_s^T - \mathbf{A}_s\mathbf{X} + \\ +\mathbf{M}_s^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_s + \mathbf{M}_r^T\mathbf{B}_s^T + \mathbf{B}_s\mathbf{M}_r \geq 0. \end{gathered} \tag{11}$$

for $r < s \leq R$*, except the pairs* (r, s) *such that* $w_r(\mathbf{p}(t))w_s(\mathbf{p}(t)) = 0, \forall \mathbf{p}(t)$*.*

Since the above conditions (10) and (11) are LMI's with respect to variables $\mathbf{X}$ and $\mathbf{M}_r$, we can find a positive definite matrix $\mathbf{X}$ and matrix $\mathbf{M}_r$ or determine that no such matrices exist. This is a convex feasibility problem. This numerical problem can be solved very efficiently by means of the most powerful tools available in the mathematical programming literature e.g. MATLAB LMI Control Toolbox [21]. The feedback gains can be obtained form the solutions $\mathbf{X}$ and $\mathbf{M}_r$ as

$$\mathbf{K}_r = \mathbf{M}_r\mathbf{X}^{-1} \tag{12}$$

Then, by the help of $r = \text{ordering}(i_1, i_2, \ldots, i_N)$ in Method 2 one can define feedbacks $\mathbf{K}_{i_1,i_2,\ldots,i_N}$ from $\mathbf{K}_r$ obtained in (12) and store into tensor $\mathcal{K}$ of (9).

In order to set constraints on the control value we add the following LMIs to (10) and (11):

Theorem 2 *Constraint on the control value Assume that* $\|\mathbf{x}(0)\| \leq \phi$*, where* $\mathbf{x}(0)$ *is unknown, but the upper bound* ϕ *is known. The constraint* $\|\mathbf{u}(t)\| \leq \mu$ *is enforced at all times* $t > 0$ *if the LMIs*

$$\begin{aligned} \phi^2\mathbf{I} &\leq \mathbf{X} \\ \begin{pmatrix} \mathbf{X} & \mathbf{M}_i^T \\ \mathbf{M}_i & \mu^2\mathbf{I} \end{pmatrix} &\geq 0 \end{aligned}$$

hold. We obtain the feedback gains as above (12) by solving all the LMIs.

6. TORA System

The Translational Oscillations with a Rotational Actuator (TORA) system[1] was developed as a simplified model of a dual-spin spacecraft [13]. Later, Bernstein and his colleagues at the University of Michigan Ann Arbor, turned it into a benchmark problem for non-linear control [7, 22, 23].

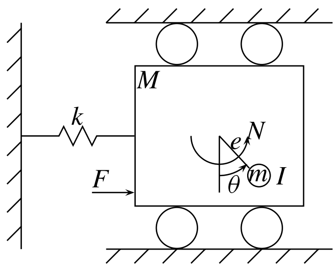

Fig. 1. TORA system

The system shown in *Fig. 1* represents a translational oscillator with an eccentric rotational proof-mass actuator. The oscillator consists of a cart of mass M connected to a fixed wall by a linear spring of stiffness k. The cart is constrained to have one-dimensional travel. The proof-mass actuator attached to the cart has mass m and moment of inertia I about its center of mass, which is located at distance e from the point about which the proof mass rotates. The motion occurs in a horizontal plane, so that no gravitational forces need to be considered. In *Fig. 1*, N denotes the control torque applied to the proof mass, and F is the disturbance force on the cart.

Let q and $\dot{q}$ denote the translational position and velocity of the cart, and let θ and $\dot{\theta}$ denote the angular position and velocity of the rotational proof mass, where $\theta = 0\,\text{deg}$ is perpendicular to the motion of the cart, and $\theta = 90\,\text{deg}$ is aligned with the positive q direction. The equations of motion are given by

$$\begin{aligned}(M+m)\ddot{q} + kq &= -me(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + F \\ (I + me^2)\ddot{\theta} &= -me\ddot{q}\cos\theta + N\end{aligned}$$

With the normalization

$$\xi \triangleq \sqrt{\tfrac{M+m}{I+me^2}}\,q, \qquad \tau \triangleq \sqrt{\tfrac{k}{M+m}}\,t,$$
$$u \triangleq \tfrac{M+m}{k(I+me^2)}N, \quad w \triangleq \tfrac{1}{k}\sqrt{\tfrac{M+m}{I+me^2}}\,F,$$

the equation of motion become

$$\begin{aligned}\ddot{\xi} + \xi &= \varepsilon\left(\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta\right) + w \\ \ddot{\theta} &= -\varepsilon\ddot{\xi}\cos\theta + u\end{aligned}$$

[1] Also referred to as the rotational/translational proof-mass actuator (RTAC) system.

where ξ is the normalized cart position, and w and u represent the dimensionless disturbance and control torque, respectively. In the normalized equations, the symbol $(\cdot)$ represents differentiation with respect to the normalized time τ. The coupling between the translational and rotational motions is represented by the parameter ε which is defined by

$$\varepsilon \triangleq \frac{me}{\sqrt{(I + me^2)(M + m)}}$$

Letting $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}^T = \begin{pmatrix} \xi & \dot{\xi} & \theta & \dot{\theta} \end{pmatrix}^T$, the dimensionless equations of motion in first-order form are given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u + \mathbf{d}(\mathbf{x})w, \tag{13}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-1}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon \cos x_3}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{-\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix},$$

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix}, \qquad \mathbf{d}(\mathbf{x}) = \begin{pmatrix} 0 \\ \frac{1}{1-\varepsilon^2\cos^2 x_3} \\ 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix}.$$

Note that u, the control input, is the normalized torque N and w, the disturbance, is the normalized force F. In the followings consider the case of no disturbance. The parameters of the simulated system are given in *Table 1*.

6.1. Determination of the Convex State-space TP Model Form of the TORA System

Observe that the non-linearity is caused by $x_3(t)$ and $x_4(t)$. For the TP model transformation we define the transformation space as $\Omega = [-a, a] \times [-a, a]$ ($x_3(t) \in$

Table 1. Parameters of the TORA system

Description	Parameter	Value	Units
Cart mass	M	1.3608	kg
Arm mass	m	0.096	kg
Arm eccentricity	e	0.0592	m
Arm inertia	I	0.0002175	kg m^2
Spring stiffness	k	186.3	N/m
Coupling parameter	ε	0.200	—

$[-a, a]$ and $x_4(t) \in [-a, a]$), where $a = \frac{45}{180}\pi$ rad (note that these intervals can be arbitrarily defined). Let the density of the sampling grid be 101×101. The sampling results in $\mathbf{A}^s_{i,j}$ and $\mathbf{B}^s_{i,j}$, where $i, j = 1 \ldots 101$. Then we construct the matrix $\mathbf{S}^s_{i,j} = \begin{pmatrix} \mathbf{A}^s_{i,j} & \mathbf{B}^s_{i,j} \end{pmatrix}$, and after that the tensor $\mathcal{S}^s \in \mathbb{R}^{101\times 101\times 4\times 4}$ from $\mathbf{S}^s_{i,j}$. If we execute HOSVD on the first two dimensions of $\mathcal{S}^s$, we find that the rank of $\mathcal{S}^s$ on the first two dimensions are 4 and 2 respectively. This means that the TORA system can be exactly given as convex combination of $4\times 2 = 8$ linear vertex model. In the present case the fourth singular value of the first dimension is very small comparing to the other three ($\sigma_1 = 202.3062$, $\sigma_2 = 1.4580$, $\sigma_3 = 0.6665$, $\sigma_4 = 0.0018$), therefore we discard it. Consequently, we reduce the rank of the first dimension to three, which causes a dispensable error. In conclusion, the TP model transformation describes TORA system as:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{3}\sum_{j=1}^{2} w_{1,i}(x_3(t))w_{2,j}(x_4(t))\left(\mathbf{A}_{i,j}\mathbf{x}(t) + \mathbf{B}_{i,j}u(t)\right). \tag{14}$$

The basis functions $w_{1,i}(x_3(t))$ and $w_{2,j}(x_4(t))$ are depicted in *Fig. 2*.

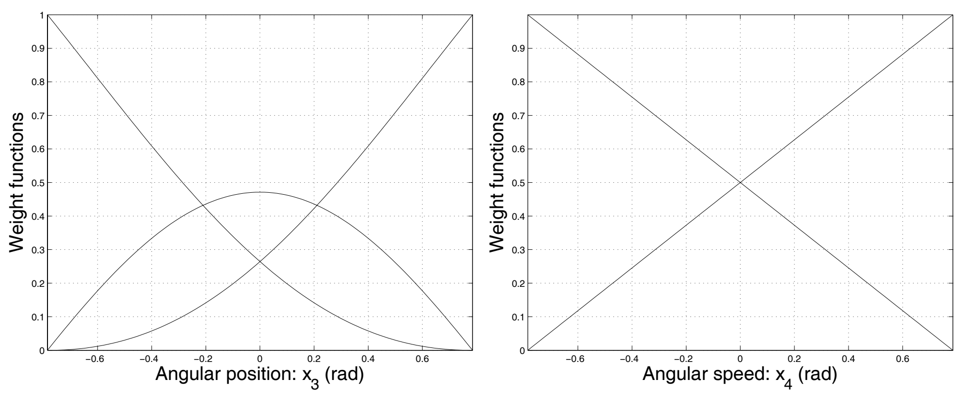

Fig. 2. Basis functions on dimensions $x_3(t)$ and $x_4(t)$

6.2. Evaluation of the Derived Controllers

To demonstrate the performance of the controlled system numerical experiments are presented in this section. The control values are computed by (9) as

$$u(t) = -\left(\sum_{i=1}^{3}\sum_{j=1}^{2} w_{1,i}(x_3(t))w_{2,j}(x_4(t))\mathbf{K}_{i,j}\right)\mathbf{x}(t)$$

in all cases of the simulations. Vectors $\mathbf{K}_{i,j}$ are resulted by LMIs discussed above.

6.2.1. *Controller 1: Global and Asymptotic Stabilization of the TORA System*

Let the resulting LTI vertex systems be substituted into the LMIs of the Theorem 1. The LMI solver shows that *Eq.* (10) and (11) are feasible in the present case. *Eq.* (12) yields 6 LTI feedback gains $\mathbf{K}_{i,j}$.

In order to show the performance of the controller we generated a sinusoidal reference signal $f(t)$ with the following parameters: amplitude $\frac{15}{180}\pi$ rad and frequency $0.01\,\frac{\text{rad}}{\text{sec}}$. Thus, the input x_3 of the controller became $x_3(t) - f(t)$. The response of the reference signal tracking control is shown in *Fig. 3*. It shows the state values $x_1(t)$, $x_3(t)$ (solid line) and $f(t)$ (dashed line), and the control value $u(t)$ for the initial conditions $x_1(0) = 0.1$ m, $x_3(0) = \frac{20}{180}\pi$ rad.

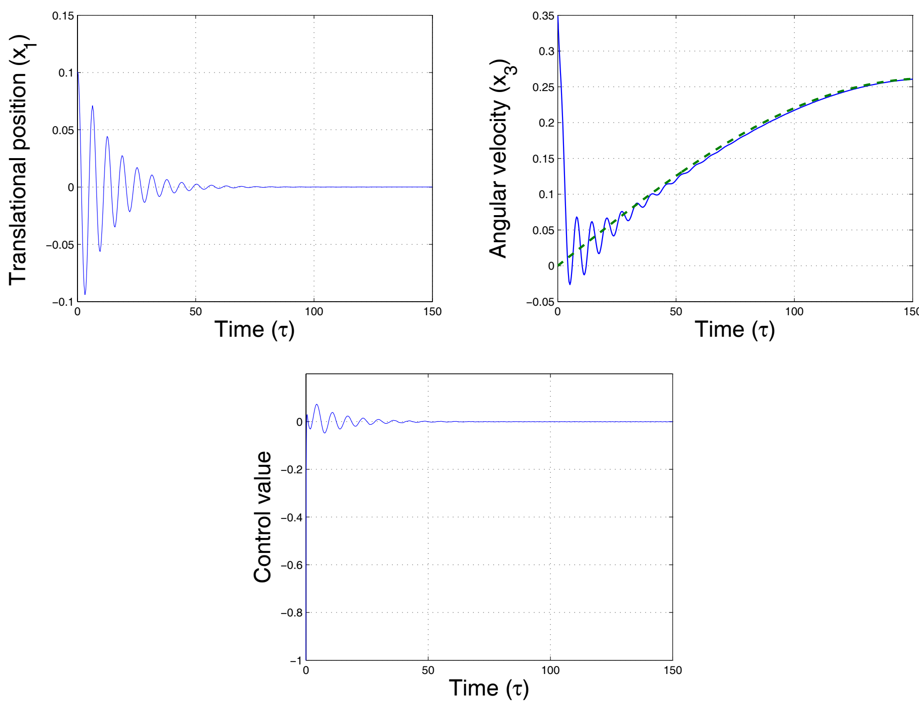

Fig. 3. Controller 1: Global and asymptotic stabilization of the TORA system

6.2.2. *Controller 2: Constraint on the Control Value*

In order to be capable of bounding the control values we apply Theorem 2. In the case of Controller 2 we define the minimal control value whereas the LMIs are still feasible. The response of the resulting controller is presented in *Fig. 4*. The control value in the second case ($\max(\|u\|) = 0.1972$) is significantly smaller than in the first case ($\max(\|u\|) = 1.0593$) while only a slight difference can be seen on the simulation results.

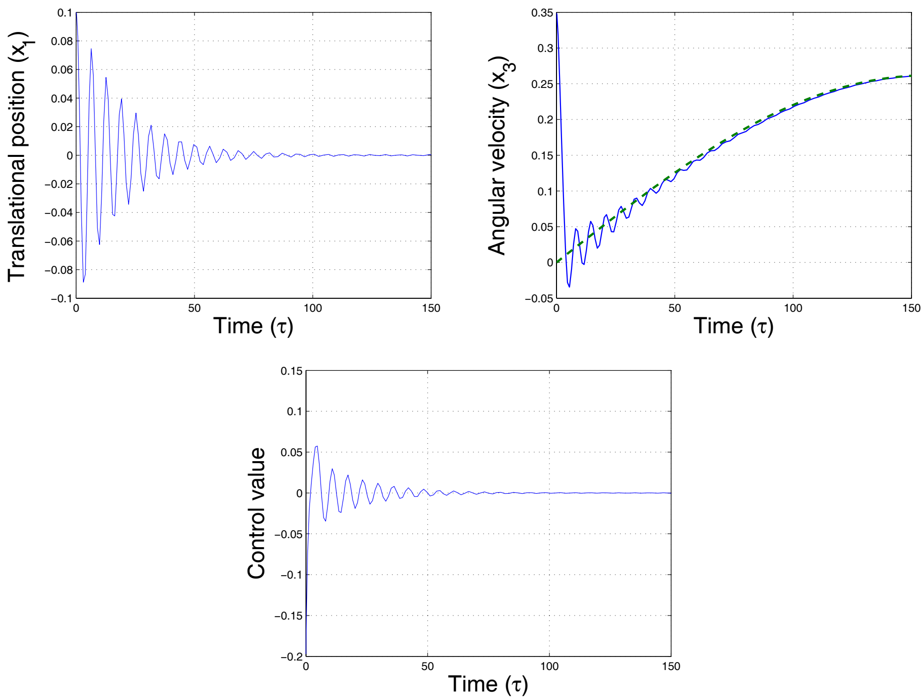

Fig. 4. Controller 2: Constraint on the control value

7. Conclusion

This paper shows that once we have a computer programme, for instance in MATLAB, of the TP model transformation and an LMI solver (MATLAB LMI Control Toolbox [21]), the control design method, studied in this paper, can easily and automatically be executed. This paper shows an example when we want to achieve more than the global and asymptotic stability but also we want to define some constraint on the control value. The derived controllers' performance is shown in a reference signal tracking case. This paper applieds rather simple LMI theorems in the controller design, but by applying more advanced theorems other control specifications can be taken into consideration during the controller design.

References

[1] BARANYI, P., TP Model Transformation as a Way to LMI Based Controller Design. *IEEE Transaction on Industrial Electronics*, **51**, (2, April 2004) pp. 387–400.

[2] BARANYI, P. – TIKK,D. – YAM, Y. – PATTON, J., From Differential Equations to PDC Controller Design via Numerical Transformation. *Computers in Industry, Elsevier Science*, **51** 2003, pp. 281–297.

[3] BARANYI, P.– KORONDI, P.– PATTON,R. J. – HASHIMOTO, H., Global Asymptotic Stabilization of the Prototypical Aeroelastic Wing Section via TP Model Transformation. *Asian Journal of Control*, **7**(2), 2004. (to be printed).

[4] TANAKA, K. – WANG, H. O., *Fuzzy Control Systems Design and Analysis — A Linear Matrix Inequality Approach*. John Wiley and Sons, Inc., 2001.

[5] RUDAS, I. J. – KAYNAK, O., Entropy-based Operations on Fuzzy Sets, *IEEE Transactions on Fuzzy Systems*, **6**, 1998, pp. 33–40.

[6] RUDAS, I. J. – KAYNAK, O., Minimum and Maximum Fuzziness Generalized Operators. *Fuzzy Sets and Systems*, **98** 1998, pp. 83–94,

[7] BERNSTEIN, D. S., Special issue: A Non-linear Benchmark Problem, *International Journal of Robust and Non-Linear Control*, **8** 1998.

[8] JANKOVIC, M. – FONTAINE, D. – KOKOTOVIC, P. V., TORA example: Cascade and Passivity Control Design, *IEEE Transaction on Control System Technology*, **4**, May 1996, pp. 292–297.

[9] ALLEYNE, A.. Physical Insights on Passivity-based TORA Control Designs, *IEEE Transaction on Control System Technology*, **6**, May 1998, pp. 436–439.

[10] MARGALIOT, M. – LANGHOLZ, G., Fuzzy Control of a Benchmark Problem: A Computing with Words Approach, *IEEE Transaction on Fuzzy Systems*, **12** (2 April 2004), pp. 230–235.

[11] LATHAUWER, L. D. – MOOR, B. D. – VANDEWALLE, J., A Multi Linear Singular Value Decomposition, *SIAM Journal on Matrix Analysis and Applications*, **21** (4), 2000, pp. 1253–1278.

[12] RAMSAY, J. O. – SILVERMAN, B. W., *Functional Data Analysis*. Springer Series in Statistics. Springer-Verlag, 1997.

[13] FARIN, G., *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press Professional, San Diego, USA, 1990.

[14] TIKK, D.– BARANYI, P. – PATTON, R. J., Polytopic and TS Models are Nowhere Dense in the Approximation Model Space, *IEEE International Conference on System, Man, and Cybernetics (SMC'02)*, 2002. Proc. on CD.

[15] YAM, Y. – BARANYI, P. – YANG, C. T., Reduction of Fuzzy Rule Base via Singular Value Decomposition, *IEEE Trans. Fuzzy Systems*, **7** (2) 1999, pp. 120–132.

[16] YAM, Y. – YANG, C. T. – BARANYI, P., *Singular Value Based Fuzzy Reduction with Relaxed Normalization Condition*, **28** of *Studies in Fuzziness and Soft Computing, Interpretability Issues in Fuzzy Modelling*, pp. 325–354. Springer-Verlag, 2003.

[17] NESTEROV, Y. – NEMIROVSKY, A., *Interior-point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM Books, Philadelphia, 1994.

[18] BOYD, S. – EL GHAOUI, L. – FERON, E. – BALAKRISHNAN, V. Linear Matrix Inequalities in System and Control Theory, *Philadelphia PA:SIAM, ISBN 0-89871-334-X*, 1994.

[19] GHAOUI L. E. – NICULESCU, S. I. Advances in Linear Matrix Inequality Methods in Control, In L. E. Ghaoui and S. I. Niculescu, editors, *Advances in Design and Control*. SIAM Books, Philadelphia, 2000.

[20] SCHERER C. W. – WEILAND, S., *Linear Matrix Inequalities in Control*, DISC course lecture notes, 2000. http://www.cs.ele.tue.nl/sweiland/lmi.htm.

[21] GAHINET, P. – NEMIROVSKI, A. – LAUB, A. J. – CHILALI, N., *LMI Control Toolbox*. The MathWorks, Inc., 1995.

[22] BUPP, R. T. – BERNSTEIN, D. S. – COPPOLA, V. T., A Benchmark Problem for Non-linear Control Design, *International Journal of Robust and Non-Linear Control*, **8**, 1998, pp. 307–310.

[23] BUPP, R. T.– BERNSTEIN, D. S.–COPPOLA, V. T., Experimental Implementation of Integrator Backstepping and Passive Non-linear Controllers on the RTAC Testbed. *International Journal of Robust and Non-Linear Control*, **8**, 1998, pp. 435–457.