Nonlinear Control Via TP Model Transformation: The TORA System Example

Gerhard Hancke[1], Ágnes Szeghegyi[2]

[1]University of Pretoria, Pretoria 0002 South Africa, ghancke@eng.up.ac.za
[2]Budapest Polytechnic 1034 Budapest, Doberdo út 6.

***Abstract* — This paper presents a case study of the TP (Tensor Product) model transformation in the control of a nonlinear benchmark problem. We design a nonlinear controller to the TORA system via TP model transformation and LMI (Linear Matrix Inequality) based controller design technique. The main contribution of the paper is to show that both numerical methods the TP model transformation and the LMI can readily be executed by computer independently on the given problem and without analytical derivations, which, hence, leads to a fast way of controller designs for a class of engineering control problems. Numerical simulation is used in the paper to provided empirical validation of the control results.**

1 Introduction

Recently a control design method was proposed by *Baranyi* 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, introduced by *Baranyi* [2], is executed. In the second step LMI's are solved under the PDC (Parallel Distributed Compensation) framework, that also includes the feasibility solution of LMI's. The method of the second step is still in active research [4] (book [4] refers to a great number of related papers). 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 3.2) whereupon design techniques of the PDC framework, can immediately be executed. The second step results in a controller according to various control specifications.

It is worth noticing here that both steps are executed numerically by computers. This implies two advantages such as: a) the controller can be derived automatically, without analytic derivations; b) 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, or algorithms based on *Rudas* type generalized operators [5, 6].

The main goal of this paper is to study, via the control design of the TORA system example, how to execute the above control design method. This control problem of TORA has a great comparative literature related to different control theories. The overview of this literature is behind the scope of this paper, but we refer the reader to [7, 8, 9, 4].

A detailed preliminary study of this work investigates the use of the TP model transformation in the control design of the inverted pendulum [10].

This paper is organized as: Section 2 introduces the notation being used in this paper. Section 3 briefly summarizes some preliminaries and discusses the concept of the control design method. Section 4 presents a detailed control design example. Section 5 concludes the paper.

2 Nomenclature

This section is devoted to introduce the notations being used in this paper: $\{a,b,\ldots\}$: scalar values. $\{\mathbf{a},\mathbf{b},\ldots\}$: vectors. $\{\mathbf{A},\mathbf{B},\ldots\}$: matrices. $\{\mathcal{A},\mathcal{B},\ldots\}$: 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. Subscript defines 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. $\mathbf{A}=\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots\end{bmatrix}$. $\diamond_{i,j,n},\ldots$: indices. $\diamond_{I,J,N},\ldots$: index upper bounds: for example: $i=1..I$, $j=1..J$, $n=1..N$ or $i_n=1..I_n$. $\mathbf{A}_{(n)}$: n-mode matrix of tensor $\mathcal{A}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. $\mathcal{A}\times_n\mathbf{U}$: n-mode matrix-tensor product. $\mathcal{A}\otimes_n\mathbf{U}_n$: multiple product as $\mathcal{A}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3..\times_N\mathbf{U}_N$. Detailed discussion of tensor notations and operations is given in [11].

3 Preliminaries

This section is intended to define the basic concepts being used in this paper.

3.1 Parametrically varying state-space model

Consider parametrically varying state-space model:

$$s\mathbf{x}(t)=\mathbf{A}(\mathbf{p}(t))\mathbf{x}(t)+\mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \tag{1}$$

$$\mathbf{y}(t)=\mathbf{C}(\mathbf{p}(t))\mathbf{x}(t)+\mathbf{D}(\mathbf{p}(t))\mathbf{u}(t),$$

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

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

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

3.2 Convex state-space TP model

Equ. (2) can be approximated for any parameter $\mathbf{p}(t)$ as a convex combination of the R number of LTI (Linear Time Invariant) system matrices $\mathbf{S}_r$, $r = 1..R$. Matrices $\mathbf{S}_r$ are also termed as vertex system matrices. Therefore, one can define basis functions $w_r(\mathbf{p}(t)) \in [0,1]$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ belongs to the convex hull of $\mathbf{S}_r$ as $\mathbf{S}(\mathbf{p}(t)) = co\{\mathbf{S}_1, \mathbf{S}_2, .., \mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}$, where $\mathbf{w}(\mathbf{p}(t))$ defines the basis of the convex combination. The control design method, to be applied in this paper, applies univariate basis functions. Thus, the explicit form of the convex combination in terms of tensor product becomes:

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

(3) is termed as TP model in [1, 2]. Function $w_{n,j}(p_n(t))$ is the j-th univariate basis function defined on the n-th dimension of Ω, and $p_n(t)$ is the n-th element of vector $\mathbf{p}(t)$. I_n (n=1,...,N) is the number of univariate basis functions used in the n-th dimension of the parameter vector. The multiple index $(i_1, i_2, ..., i_N)$ refers to the LTI system corresponding to the i_n-th basis function in the n-th dimension. Hence, the number of vertex systems $\mathbf{S}_{i_1,i_2,..,i_N}$ is obviously $R = \prod_n I_n$.

Remark 1 *Equation (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:*

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

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

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

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

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 $S \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times O \times I}$ is constructed from the linear constant system matrices $\mathbf{S}_{i_1,i_2,...,i_N} \in \mathbb{R}^{O \times I}$. The first N dimensions of S are assigned to the dimensions of Ω. The convexity of the TP form is ensured by the conditions:

$$\forall n, i, p_n(t) : w_{n,i}(p_n(t)) \in [0,1]; \quad \forall n, p_n(t) : \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \tag{5}$$

$\mathbf{S}(\mathbf{p}(t))$ can be exactly decomposed into TP model form ($\delta = 0$ in equations (3) and (4)) in many cases. However, one should face that exact finite element TP model (the number of LTI systems in the TP model is finite) representation does not exist in general ($\delta > 0$ in equations (3) and (4)), see [12]. In this case the task is to achieve an acceptable δ via increasing the number of LTI systems of the TP model. This, however, soon leads to the approximation trade-off. The dynamic model in the example of this paper can exactly be represented by TP model form.

3.3 TP model transformation [1, 2, 3]

The goal of the TP model transformation is to transform a given state-space model (1) into convex TP form, namely to (4) with conditions (5).

Let the synopsis of the TP model transformation be:

$$(\mathbf{w}_{n=1..N}(p_n(t)), \mathcal{S}) = TP\_transf(\mathbf{S}(\mathbf{p}(t)), \Omega), \tag{6}$$

where $\mathbf{S}(\mathbf{p}(t)) \in \mathbb{R}^{O \times I}$ is from (2), and $\Omega \subset \mathbb{R}^N$ denotes the bounded domain which the transformation is performed over. It is assumed that $\mathbf{S}(\mathbf{p}(t))$ can be determined for all $\mathbf{p}(t) \in \Omega$ eider by explicit analytic forms, neural network or other soft computing identification techniques. Vectors $\mathbf{w}_n(p_n(t)) \in \mathbb{R}^{I_n}$ and tensor $\mathcal{S}$ of (6) are defined at (4).

3.4 LMI based controller design under PDC framework

The PDC design techniques determine one feedback to each vertex model:

$$\mathcal{K} = PDC(\mathcal{S}, stability\_theorem), \tag{7}$$

then define the control value by the same basis function system as:

$$\mathbf{u}(t) = -\left(\mathcal{K} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t)) \right) \mathbf{x}(t), \tag{8}$$

which ensures the system performance defined by the selected "stability_theorem" in (7). "stability_theorem" is a symbolic parameter. It specifies the stability criteria and the desired control performance expressed in terms of LMI's. 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 example, discussed in Section 4 of this paper, applies one of the most basic LMI theorems to achieve a global asymptotic stability for a given dynamic system. In order to complete the paper let us recall briefly this theorem here:

Method 1 *(*Global and asymptotic stabilization of continuous TP model) *Assume a given state-space model in TP form (4) with conditions (5). In order to have a direct link to the typical form of LMI theorems, let the following indexing be defined:*

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

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

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

Then the controller design can be derived from the Lyapunov stability theorems for global and asymptotic stability as shown in [4], and is done as:

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

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} + \mathbf{M}_r^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_r > 0 \tag{9}$$

for all r and

$$\begin{gathered} -\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} - \mathbf{X}\mathbf{A}_s^T - \mathbf{A}_s\mathbf{X} + \\ +\mathbf{M}_s^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_s + \mathbf{M}_r^T\mathbf{B}_s^T + \mathbf{B}_s\mathbf{M}_r \geq 0. \end{gathered} \tag{10}$$

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 (9) and (10) 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. Numerically, this problem can be solved very efficiently by means of the most powerful tools available in the mathematical programming literature e.g. MATLAB-LMI *toolbox [13]. The feedback gains can be obtained form the solutions $\mathbf{X}$ and $\mathbf{M}_r$ as*

$$\mathbf{K}_r = \mathbf{M}_r\mathbf{X}^{-1} \quad \text{and} \quad \mathbf{P} = \mathbf{X}^{-1}. \tag{11}$$

Then, by the help of $r = ordering(i_1, i_2, .., i_N)$ one can define feedbacks $\mathbf{K}_{i_1,i_2,..,i_N}$ from $\mathbf{K}_r$ obtained in (11) and store into tensor $\mathcal{K}$ of (8).

4 Example: Control of the TORA system

Consider the system shown in Figure 1. which represents a translational oscillator with an eccentric rotational proof mass actuator (TORA) [7, 8, 9, 4]. The nonlinear coupling between the rotational motion of the actuator and the translational motion of the oscillator provides the mechanism for control. Let $x_1(t)$ and $x_2(t)$ denote the translational position and velocity of the cart with $\dot{x_1}(t) = x_2(t)$. Let $x_3(t) = \theta(t)$ and $\dot{x_3}(t) = x_4(t)$ denote the angular position and velocity of the rotational proof mass. Then the system dynamics can be described by the equation:

$$\dot{\mathbf{x}}(t) = f(x_3(t), x_4(t))\mathbf{x}(t) + g(x_3(t))u(t),$$

where u is the torque applied to the eccentric mass, and

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

and let $\varepsilon = 0.05$. The linearization around the equilibrium point has a pair of nonzero imaginary eigenvalues and two zero eigenvalues. Hence the system at the origin is an example of a critical nonlinear system. The goal of the control is to asymptotically stabilize the system. In the present example we do not go further than stabilization and apply the simple Method 2 for controller design. Note that further control specification can be guarantied by other LMI's.

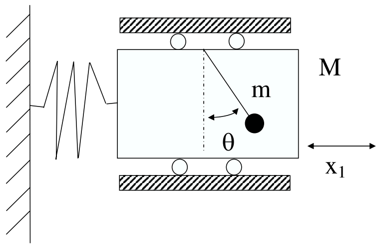

Figure 1: TORA

4.1 Controller design

Observe that the nonlinearity is caused by $x_3(t)$ and $x_4(t)$. For the TP model transformation we define the transformation space as $\mathbf{\Omega} = [-a,a] \times [a,a]$ ($x_3(t) \in [-a,a]$ and $x_4(t) \in [-a,a]$), where $a = 55/180\pi$ (note that these intervals can be arbitrarily defined). Let the density of the sampling grid be 100×100. The sampling results in $\mathbf{A}^s_{i,j}$ and $\mathbf{B}^s_{i,j}$, where $i, j = 1..100$. Then we construct matrix $\mathbf{S}^s_{i,j} = \begin{pmatrix} \mathbf{A}^s_{i,j} & \mathbf{B}^s_{i,j} \end{pmatrix}$ and tensor $\mathcal{S}^s \in \mathbb{R}^{100\times100\times2\times3}$ from $\mathbf{S}^s_{i,j}$. If we execute HOSVD on the first two dimensions of $\mathcal{S}^s$ then we find that the rank of $\mathcal{S}^s$ on the first two dimensions are 4 and 3 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, 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:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{3} \sum_{j=1}^{2} w_{1,i}(x_3(t)) w_{2,j}(x_4(t)) \left(\mathbf{A}_{i,j}\mathbf{x}(t) + \mathbf{B}_{i,j}u(t) \right). \tag{12}$$

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

Having the above resulting vertex models one can easily execute Method 2.

4.2 Control results

Figure 3 shows the state values $x_1(t)$ and $x_3(t)$ for the initial conditions $x_1(0) = 0.1m$, $x_3(0) = -20/180\pi$, when the control is switched off. Figure 4 shows the

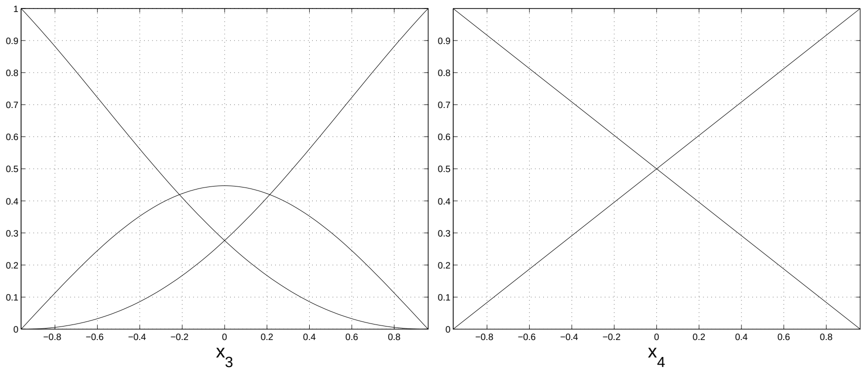

Figure 2: Basis functions on dimension $x_3(t)$ and $x_4(t)$

control result when the control is switched on at $t = 0$s. The figures show how the controller stabilize the system. Note that we applied one of the simplest LMI technique which does not consider any control specification (speed of the controller, other optimization) except global asymptotic stability.

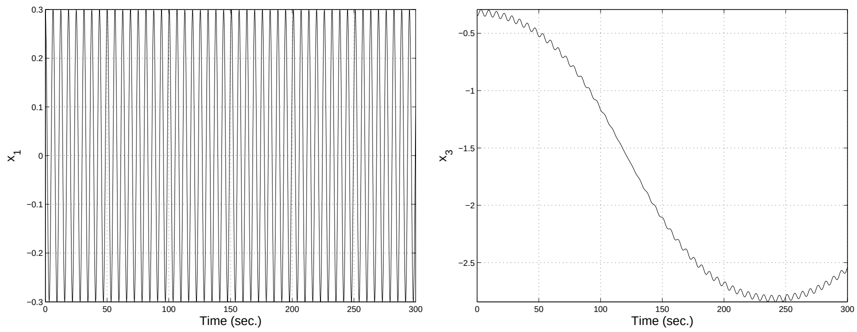

Figure 3: State values $x_1(t)$ and $x_3(t)$ without control (t=1..300s))

5 Conclusion

This paper shows that once we have a computer program, for instance in MATLAB, of the TP model transformation and the LMI solver (MATLAB LMI control toolbox [13]) then the control design method, studied in this paper, can easily and automatically be executed. The same algorithm was executed in the control design of other problems, see [3, 10]. This paper applied a simple LMI theorem in the controller design. As a matter of fact, we can select other LMIs that are capable of considering further control specifications behind global asymptotic stability. As

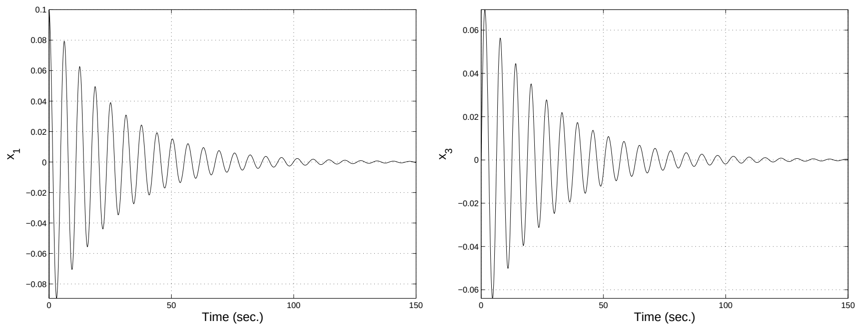

Figure 4: Controlled state values $x_1(t)$ and $x_3(t)$ (t=1..170s)

future work we apply the TP model transformation based controller design to a gas turbine exhibiting various nonlinear control phenomena.

References

[1] P. Baranyi. TP model transformation as a way to LMI based controller design. *IEEE Transaction on Industrial Electronics*, 2004. (in press).

[2] 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.

[3] P. Baranyi and R. J. Patton. A numerical control design method for prototypical aeroelastic wing section with structural non-linearity. *European Control Conference 2003*. (proc. on CD).

[4] K. Tanaka and H. O. Wang. *Fuzzy Control Systems Design and Analysis - A Linear Matrix Inequality Approach*. Hohn Wiley and Sons, Inc., 2001.

[5] I. J. Rudas and O. Kaynak. Entropy-based operations on fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 6, 1998.

[6] I. J. Rudas and O. Kaynak. Minimum and maximum fuzziness generalized operators. *Fuzzy Sets and Systems*, 98, 1998.

[7] R. T. Bupp, D. S. Bernstein, and V. T. Coppola. A benchmark problem for nonlinear control design: Problem statement, experiment testbed and passive nonlinear compensation. *American Control Conference, Seattle*, pages 4363–4367, 1995.

[8] Session on. Benchmark problem for nonlinear control design. *American Control Conference, Seattle*, pages 4337–4367, 1995.

[9] M. Jankovic, D. Fontaine, and P. Kokotovic. Tora example: Cascade and passivity control design. *American Control Conference, Seattle*, pages 4337–4351, 1995.

[10] G. Hancke and Á. Szeghegyi. Application study of the TP model transfromation in the control of an inverted pendulum. *International Conference on Computational Cybernetics, Siófok, Hungary, August 29-31*, pages 29–31, 2003.

[11] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multi linear singular value decomposition. SIAM *Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[12] D. Tikk, P. Baranyi, and R.J.Patton. Polytopic and TS models are nowere dense in the approximation model space. *IEEE Int. Conf. System Man and Cybernetics (SMC'02)*, 2002. Proc. on CD.

[13] P. Gahinet, A. Nemirovski, A.J.Laub, and M.Chilali. *LMI Control Toolbox*. The MathWorks, Inc., 1995.