TP MODEL TRANSFORMATION IN NONLINEAR CONTROL DESIGN

Péter Baranyi, Pál Michelberger
Computer and Automation Research Institute of the Hungarian Academy of Sciences
Budapest, Kende u. 13-17, Hungary, H-1111

Keywords: *Nonlinear control design, TP model transformation*

Abstract

This paper proposes a numerical controller design methodology applicable to given explicit models which eider can represent a physical model or are just the outcome of black-box identification (e.g. neural net model). The design method is capable of considering various advantageous control specifications.

The paper presents an example with numerical simulations to provide empirical validation of the proposed control design methodology. In this example a controller is derived for the prototypical aeroelastic wing section.

1 Introduction

This paper proposes a control design methodology that starts with a parameter-varying (where the parameters may also include the elements of the sate vector) state-space model, which is assumed to be linear in the inputs, and is given by either explicit analytic forms or soft-computing techniques, and then transforms the given model by TP (Tensor Product) model transformation [1, 2] into convex state-space TP model, then uses Parallel Distributed Compensation (PDC) framework [8] to reformulate the resulting TP model into Linear Matrix Inequalities (LMI) selected according to desired control specifications. The controller is resulted by solving the LMIs. All these steps can readily be executed numerically by computers in acceptable time [3].

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 [4]. Many control problems and design specifications have LMI formulations [5, 6] that comes from the fact that LMI formulations have the ability to readily combine various design constrains or objectives in a numerical tractable manner. This is especially true for Lyapunov-based analysis and design. A great list of control problems which can be solved via LMIs is addressed in [3] and [7]. There are papers [5] who claim that once a control problem is formulated in terms of LMIs then the problem is solved; and point on the fact that the LMI based controller design leads to solution even in case when analytic solution does not exist (multiple Riccati equations can not be solved analytically in general). Further developments of LMIs for the above design problems are in an area of active research. Commercialized softwares for LMI based designs are available for engineering practices [3].

The TP model describes dynamic models by the parameter varying convex combination of LTI (Linear Time Invariant) systems. By the help of the PDC framework the convex TP model can readily be reformulated in terms of LMIs according to various control design specifications [8].

The main difficulties of having the TP model of a system is that the identification techniques do not result in TP model form in most cases. Therefore, it is desired to find a transformation technique capable of transforming identified models into TP model form. There are analytic transformation techniques which can be applied to a class of analytically given models, for instance see chapter 2 of [8]. Analytic techniques, however, need problem dependent human intuition, and cannot easily be solved in many cases.

The aim of this paper is to introduce a numerical TP model transformation method that can be executed with the LMIs by computers automatically even in case when the given model is not known analytically, but is resulted by a black-box identification. If we achieve this aim we are then capable of identifying and updating models and determining and updating controllers automatically for a class of control problems in real time operations.

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. $\mathbf{A}=\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots\end{bmatrix}$. $\diamond_{i,j,n},\ldots$: are indices. $\diamond_{I,J,N},\ldots$: index upper bound: 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 [9].

3 Preliminaries

This section is intended to define some basic concepts.

3.1 Parameter-varying state-space model

Consider parameter-varying state-space model:

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

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

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

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

is a parameter-varying object, where $\mathbf{p}(t)\in\Omega$ is time varying $N-$dimensional parameter vector, where $\Omega=[a_1,b_1]\times[a_2,b_2]\times..\times[a_N,b_N]\subset\mathbb{R}^N$ is a closed hypercube. $\mathbf{p}(t)$ can also include elements of $\mathbf{x}(t)$. The control design method presented in this paper is restricted to cases when vector $\mathbf{p}(t)$ does not contain elements of $\mathbf{u}(t)$. 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

Equ. (2) can be approximated for any parameter $\mathbf{p}(t)$ as a convex combination of the R LTI 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 $w_r(\mathbf{p}(t))\in[0,1]\subset\mathbb{R}$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ belongs to the convex hull of $\mathbf{S}_r$ as $\mathbf{S}(\mathbf{p}(t))=co\{\mathbf{S}_1,\mathbf{S}_2,..,\mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}$, where vector $\mathbf{w}(\mathbf{p}(t))$ contains the basis functions $w_r(\mathbf{p}(t))$ of the convex combination. The control design methodology, to be introduced in this paper, applies univariate basis functions. Thus, the explicit form of the convex combination in terms of tensor product becomes:

$$\begin{pmatrix}s\mathbf{x}(t)\\ \mathbf{y}(t)\end{pmatrix}\approx \qquad (3)$$

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

(3) is termed as TP model in this paper. Function $w_{n,j}(p_n(t))\in[0,1]$ 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 $\mathbf{p}(t)$. 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 LTI vertex systems $\mathbf{S}_{i_1,i_2,..,i_N}$ is obviously $R = \prod_n I_n$. One can rewrite (3) in the concise TP form as:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \mathcal{S} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \qquad (4)$$

that is $\quad \mathbf{S}(\mathbf{p}(t)) \approx_{\varepsilon} \mathcal{S} \otimes_{n=1}^{N} \mathbf{w}_n(p_n(t)).$

Here, ε represents the approximation error, 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 LTI vertex 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 convex combination of the LTI vertex systems is ensured by the conditions:

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

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

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

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

Remark 1 $\mathbf{S}(\mathbf{p}(t))$ *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 [10]. In this case $\varepsilon \mapsto 0$, when the number of LTI systems involved in the TP model goes to ∞. This fact leads to complexity trade-off in case of numerical computation.*

3.3 LMI based controller design under PDC framework

The PDC design techniques determine one LTI feedback gain to each LTI vertex systems included in the 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. Having the $\mathcal{K}$ the control value is determined by the help of the same basis functions as used in (4):

$$\mathbf{u}(t) = -\left(\mathcal{K} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t)) \right) \mathbf{x}(t). \qquad (7)$$

Observe that equation (7) restricts the design to parameter-varying models where vector $\mathbf{p}(t)$ does not contains any elements of $\mathbf{u}(t)$. 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 example, discussed in Section 5 of this paper, applies one of the most basic LMI theorems to achieve 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 convex TP model (4))

Assume a given state-space model in TP form (4) with conditions (5) and (6). In order to have a direct link between the TP model (4) and the typical form of LMI theorems under the PDC frameworks, let the following indexing be defined:

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

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 [7, 8]:

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

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} + \mathbf{M}_r^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_r > 0 \quad (8)$$

for all r and

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} - \mathbf{X}\mathbf{A}_s^T - \mathbf{A}_s\mathbf{X} + \quad (9)$$

$$+\mathbf{M}_s^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_s + \mathbf{M}_r^T\mathbf{B}_s^T + \mathbf{B}_s\mathbf{M}_r \geq 0.$$

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 (8) and (9) 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 [3]. The feedback gains can be obtained form the solutions $\mathbf{X}$ and $\mathbf{M}_r$ as

$$\mathbf{K}_r = \mathbf{M}_r\mathbf{X}^{-1}. \quad (10)$$

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 (10) and store into tensor $\mathcal{K}$ of (7).

4 TP model transformation for PDC design frameworks

4.1 TP model transformation

Detailed description of the TP model transformation is given in recent papers [1, 2, 11]. The goal of the TP model transformation is to transform a given state-space model (1) into convex TP model (4) and solve the complexity trade-off if necessary, see Remark 1. It is important to emphasize here that, in order to start with the TP model transformation, one needs an explicit model, such that the system matrix $\mathbf{S}(\mathbf{p}(t))$ can be sampled for all possible values of the parameters $\mathbf{p}(t) \in \Omega$. Whether $\mathbf{S}(\mathbf{p}(t))$ is a physical model, or are just the outcome of black-box identification is irrelevant.

The TP model transformation is a numerical method and has three key steps. The first step is the discreatisation 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 discretisation 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 defines the continuous basis functions to the LTI vertex systems.

Method 2 ***TP model transformation***

Step 1) Discretisation

a) Define the transformation space Ω as: $\mathbf{p}(t) \in \Omega : [a_1, b_1] \times [a_2, b_2] \times .. \times [a_N, b_N]$.

b) Define a hyper rectangular grid by equidistantly located grid-lines: $g_{n,m_n} = a_n + \frac{b_n - a_n}{M_n - 1}(m_n - 1)$, $m_n = 1..M_n$. The numbers of the grid lines in the dimensions are M_n.

c) Sample the given function $\mathbf{S}(\mathbf{p}(t))$ over the grid-points:

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

where $\mathbf{p}_{m_1,m_2,..,m_N} = \begin{pmatrix} g_{1,m_1} & g_{2,m_2} & .. & g_{N,m_N} \end{pmatrix}$. Superscript "s" means "sampled".

d) Store the sampled matrices $\mathbf{S}^s_{m_1,m_2,..,m_N}$ into the tensor $\mathcal{S}^s \in \mathbb{R}^{M_1 \times M_2 \times .. \times M_N \times O \times I}$.

Step 2) Extracting the LTI vertex systems

This step uses Higher Order Singular Value Decomposition (HOSVD), extended with transformations NN, SN and NO. The studies of HOSVD can be found in a large varieties of publications. This paper uses the concept and tensor notation of HOSVD as discussed in [9]. The HOSVD extended with SN, NN and NO transformations are introduced in [12] and [13].

This step executes this extended HOSVD on the first N dimensions of tensor $\mathcal{S}^s$. During performing the extended 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:

$$\gamma = \left(\|\mathcal{S}^s - \mathcal{S} \underset{n}{\otimes} \mathbf{U}_n\|_{L_2} \right)^2 \leq \sum_k \sigma_k^2. \qquad (11)$$

The resulting tensor $\mathcal{S}$, with the size of $(I_1 \times I_2 \times .. \times I_N \times O \times I)$, 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 (5) and (6). The transformation NO ensures that the resulting LTI systems form the tight convex hull of the sampled systems.

The software implementation of the extended HOSVD is rather simple, for instance, in MATLAB programming.

Step 3) Constructing continuous basis functions

a) One can determine the discretised points of the basis easily from matrices $\mathbf{U}_n$. The i_n-th column vector $\mathbf{u}_{n,i_n=1..I_n}$ of matrix $\mathbf{U}_n \in \mathbb{R}^{M_n \times I_n}$ determines one discretised 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 $p_n(t) = g_{n,m_n}$:

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

b) 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 $\mathbf{w}_d(p_d(t))$, let $p_k(t)$ be fixed to the grid-lines as:

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

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 $\mathbf{p} = \begin{pmatrix} g_{1,1} & g_{2,1} & \dots & p_d & \dots & g_{N,1} \end{pmatrix}$, 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.

Property 1 ***Exact transformation:*** The TP model transformation is capable of finding the finite TP model form if it exists (for instance see the example in Section 5.).

Property 2 ***non-exact transformation and complexity trade-off:*** If finite TP model form of the original model does not exist, then the TP model transformation results in a TP model that is an approximation of the given model. The direct sampling and finding a tensor product based approximation by the help of identification techniques may lead to large-sized problems which are not possible (or difficult) to handle with standard solvers or use the result in real time applications. Therefore, the proposed TP model transformation involves complexity trade-off that leads to a tractable LMI problems with well-preserved dynamical behavior of the controlled system. The approximation accuracy can be improved by discarding as small number of nonzero singular values as possible. This, however leads to a complexity problem soon. The error bound in (11) helps us with the complexity trade-off.

Whatever we estimate for the error bound during the complexity trade-off, the final error of the resulting TP model can always be simply checked numerically. We can select a huge number of points in Ω and evaluate the error between the original model and the resulting TP model over the selected set. Then one can define wether the model is acceptable or the increase of the number of resulting LTI systems is necessary.

4.2 Summary: control design based on the TP model transformation

The previous subsection shows how we can transform a given model into TP model form. The TP model transformation starts with $\mathbf{S}(\mathbf{p}(t))$ and Ω and results in $\mathbf{w}_{n=1..N}(p_n(t))$ and $\mathcal{S}$. $\mathbf{S}(\mathbf{p}(t)) \in$

$\mathbb{R}^{O\times I}$ is from (2), and $\Omega \subset \mathbb{R}^N$ denotes the bounded domain which the transformation is performed over. Vectors $\mathbf{w}_n(p_n(t)) \in \mathbb{R}^{I_n}$ and tensor $\mathcal{S}$ are defined at (4). If we find that the resulting TP model is not exact then we can check the accuracy of the resulting TP model numerically as discussed at Property 2. If we accept the TP model then we can apply LMI based control design under PDC framework as discussed in subsection 3.3. As a result we have the LTI feedback systems in $\mathcal{K}$ for (7). The control value is computed by (7). The basis functions in $\mathbf{w}_n(\mathbf{p}_n(t))$ are computed by the 3th step of the TP model transformation for the actual $\mathbf{p}(t)$.

5 Validation of the TP model transformation in the control design of the prototypical aeroelastic wing section

The example treats the question of state variable feedback control of prototypical aeroelastic wing section. This type of model has been traditionally used for the theoretical as well as experimental analysis of two- dimensional aeroelastic behavior. The model investigated in the example describes the nonlinear plunge and pitch motion of a wing, and incorporates essential and well-characterized structural non-linearities that yields limit cycle oscillation at low speeds. This section derives a controller capable of globally and *asymptotically stabilizing* the aeroelastic wing section via *single control surface*. This case study has been detailed in recent papers, for instance in [11]. In these papers the control results are compared with former control solutions as well.

5.1 Background

In the past few years various studies of aeroelastic systems have emerged. [14] presents a detailed background and refers to a number of papers dealing with the modelling and control of aeroelastic systems. The following provides a brief summary of this background.

Regarding the properties of aeroelastic systems one can find the study of free-play non-linearity by Tang and Dowell in [15, 16], by Price et al. in [17] and [18], by Lee et al. in [19], and a complete study of a class of non-linearities is in [20], [18]. O'Neil et al. [21] examined the continuous structural non-linearity of aeroelastic systems. These papers conclude that an aerolesatic system may exhibit a variety of control phenomena such as *limit cycle oscillation, flutter* and even *chaotic vibrations.*

Control strategies have been derived for prototypical aeroelastic wing section, for instance, in papers [22, 23, 14]. It has been shown that by applying an additional control surface global stabilization can be achieved. For instance, adaptive feedback linearization [24] and the global feedback linearization technique were introduced with two control actuators in the work of [14].

5.2 Equations of motion

In this paper, we consider the problem of flutter suppression for the prototypical aeroelastic wing section as shown in Figure 1. The aerofoil is constrained to have two degrees of freedom, the plunge h and pitch α. The equations of motion of the system have been derived in many references (for example, see [25], and [26]), and can be written as

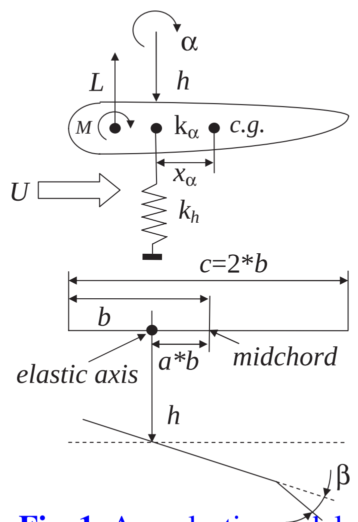

Fig. 1 Aeroelastic model

$$\begin{pmatrix} m & mx_\alpha b \\ mx_\alpha b & I_alpha \end{pmatrix} \begin{pmatrix} \ddot{h} \\ \ddot{\alpha} \end{pmatrix} + \begin{pmatrix} c_h & 0 \\ 0 & c_\alpha \end{pmatrix} \begin{pmatrix} \dot{h} \\ \dot{\alpha} \end{pmatrix} + \tag{12}$$

$$+\begin{pmatrix} k_h & 0 \\ 0 & k_\alpha(\alpha) \end{pmatrix}\begin{pmatrix} h \\ \alpha \end{pmatrix} = \begin{pmatrix} -L \\ M \end{pmatrix},$$

where

$$L = \rho U^2 b c_{l_\alpha}\left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a\right) b \frac{\dot{\alpha}}{U}\right) + \qquad (13)$$

$$+\rho U^2 b c_{l_\beta}\beta,$$

$$M = \rho U^2 b^2 c_{m_\alpha}\left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a\right) b \frac{\dot{\alpha}}{U}\right) +$$

$$+\rho U^2 b c_{m_\beta}\beta,$$

and where x_α is the non-dimensional distance between elastic axis and the centre of mass; m is the mass of the wing; I_α is the mass moment of inertia; b is semi-chord of the wing, and c_α and c_h respectively are the pitch and plunge structural damping coefficients, and k_h is the plunge structural spring constant. Traditionally, there have been many ways to represent the aerodynamic force L and moment M, including steady, quasi-steady, unsteady and non-linear aerodynamic models. In this paper we assume the quasi-steady aerodynamic force and moment, see work [25]. It is assumed that L and M are accurate for the class of low velocities concerned. Wind tunnel experiments are carried out in [27]. In the above equation ρ is the air density, U is the free stream velocity, c_{l_α} and c_{m_α} respectively, are lift and moment coefficients per angle of attack, and c_{l_β} and c_{m_β}, respectively are lift and moment coefficients per control surface deflection, and a is non-dimensional distance from the mid-chord to the elastic axis. β is the control surface deflection.

Several classes of non-linear stiffness contributions $k_\alpha(\alpha)$ have been studied in papers treating the open-loop dynamics of aeroelastic systems [15, 20, 28, 29]. For the purpose of illustration, we now introduce the use of polynomial non-linearities. The non-linear stiffness term $k_\alpha(\alpha)$ is obtained by curve-fitting the measured displacement-moment data for non-linear spring as [30]:

$$k_\alpha(\alpha) = 2.82(1 - 22.1\alpha + 1315.5\alpha^2 +$$

$$+8580\alpha^3 + 17289.7\alpha^4).$$

The equations of motion derived above exhibit limit cycle oscillation, as well as other non-linear response regimes including chaotic response [20, 28, 30]. The system parameters to be used in this paper are given in [1] and are obtained from experimental models described in full detail in works [14, 30].

With the flow velocity $u = 15(m/s)$ and the initial conditions of $\alpha = 0.1(rad)$ and $h = 0.01(m)$, the resulting time response of the nonlinear system exhibits limit cycle oscillation, in good qualitative agreement with the behaviour expected in this class of systems. Papers [21, 30] have shown the relations between limit cycle oscillation, magnitudes and initial conditions or flow velocities.

Let the equations (12) and (13) be combined and reformulated into state-space model form: $\mathbf{x}(t) = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}^T = \begin{pmatrix} h & \alpha & \dot{h} & \dot{\alpha} \end{pmatrix}^T$ and $\mathbf{u}(t) = \beta$.

Then we have:

$$\dot{\mathbf{x}}(t) = \qquad (14)$$

$$= \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) = \mathbf{S}(\mathbf{p}(t))\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix},$$

where

$$\mathbf{A}(\mathbf{p}(t)) =$$

$$= \begin{pmatrix} x_3 \\ x_4 \\ -k_1x_1 - (k_2U^2 + p(x_2))x_2 - c_1x_3 - c_2x_4 \\ -k_3x_1 - (k_4U^2 + q(x_2))x_2 - c_3x_3 - c_4x_4 \end{pmatrix}$$

$$\mathbf{B}(\mathbf{p}(t)) = \begin{pmatrix} 0 & 0 & g_3U^2 & g_4U^2 \end{pmatrix}^T,$$

where $\mathbf{p}(t) \in \mathbb{R}^{N=2}$ contains values x_2 and U. The new variables are tabulated in Table 1. One should note that the equations of motion are also dependent upon the elastic axis location a.

5.3 Controller design by the proposed methodology

5.3.1 TP model transformation

The values of the parameters in (13) are given in [14]. First of all, according to Method 2, let

Table 1 System variables

$d = m(I_\alpha - m x_\alpha^2 b^2)$
$k_1 = \frac{I_\alpha k_h}{d}$
$k_2 = \frac{I_\alpha \rho b c_{l_\alpha} + m x_\alpha b^3 \rho c_{m_\alpha}}{d}$
$k_3 = \frac{-m x_\alpha b k_h}{d}$
$k_4 = \frac{-m x_\alpha b^2 \rho c_{l_\alpha} - m \rho b^2 c_{m_\alpha}}{d}$
$p(\alpha) = \frac{-m x_\alpha b}{d} k_\alpha(\alpha)$
$q(\alpha) = \frac{m}{d} k_\alpha(\alpha)$
$c_1(U) = \frac{I_\alpha (c_h + \rho U b c_{l_\alpha}) + m x_\alpha \rho U^3 c_{m_\alpha}}{d}$
$c_2(U) = \frac{I_\alpha \rho U b^2 c_{l_\alpha}(\frac{1}{2} - a) - m x_\alpha b c_\alpha + m x_\alpha \rho U b^4 c_{m_\alpha}(\frac{1}{2} - a)}{d}$
$c_3(U) = \frac{-m x_\alpha b c_h - m x_\alpha \rho U b^2 c_{l_\alpha} - m \rho U b^2 c_{m_\alpha}}{d}$
$c_4(U) = \frac{m c_\alpha - m x_\alpha \rho U b^3 c_{l_\alpha}(\frac{1}{2} - a) - m \rho U b^3 c_{m_\alpha}(\frac{1}{2} - a)}{d}$
$g_3 = \frac{1}{d}(-I_\alpha \rho b c_{l_\beta} - m x_\alpha b^3 \rho c_{m_\beta})$
$g_4 = \frac{1}{d}(m x_\alpha b^2 \rho c_{l_\beta} + m \rho b^2 c_{m_\beta})$

us define the transformation space Ω. We are interested in the interval $U \in [14, 25](m/s)$ and we presume that the interval $\alpha \in [-0.1, 0.1](rad)$ is sufficiently large enough. Therefore, let: $\Omega : [14, 25] \times [-0.1, 0.1]$ in the present example (note that these intervals can arbitrarily be defined). Let the grid density be defined as $M_1 \times M_2$, $M_1 = 300$ and $M_2 = 300$. The resulting basis functions $w_{1,i}(U)$ and $w_{2,j}(\alpha)$ are shown on Figure 2.

In conclusion, the analytic dynamic model (14) can be described exactly in finite convex TP form of 6 vertex LTI models. Note that, one may try to derive the basis functions analytically from (14). The basis functions of α can be extracted from $k_\alpha(\alpha)$. Finding the basis functions of U, however, seems to be rather complicated. In spite of this, the computation of the TP model transformation takes a few seconds.

5.3.2 LMI based controller design under PDC frameworks

Let the above obtained LTI vertex systems be substituted into Method 1. The LMI solvers shows that equ. (8) and (9) are feasible in the present case. Equ. (10) yields 6 linear feedback systems $\mathbf{k}_{i,j}$. Then the control value is computed

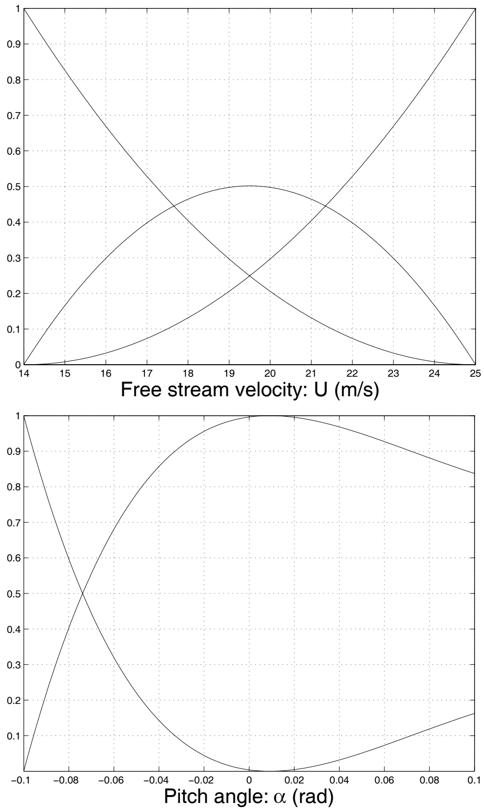

Fig. 2 Basis functions on the dimensions U and α.

by (7):

$$u(t) = -\left(\sum_{i=1}^{3}\sum_{j=1}^{2} w_{1,i}(U) w_{2,j}(\alpha) \mathbf{k}_{i,j}\right) \mathbf{x}(t).$$

5.4 Control results

To demonstrate the performance of the controlled system, numerical experiments are presented in this subsection. In order to be comparable to other published results, for instance to [14], the numerical examples are performed with $a = -0.4$ and with free stream velocity $U = 15m/s$ and for initials $h = 0.01m$ and $\alpha = 0.1rad$. Figure 3 shows the time response of the controlled system. The system is stabilized asymptotically. Note that we did not define any specific control performances, except global asymptotic stability, in the design process. We applied one of the basic LMI theorems. If the design requirements extend beyond stability, various performance specifications can be readily ensured by selecting proper

LMI design theorems.

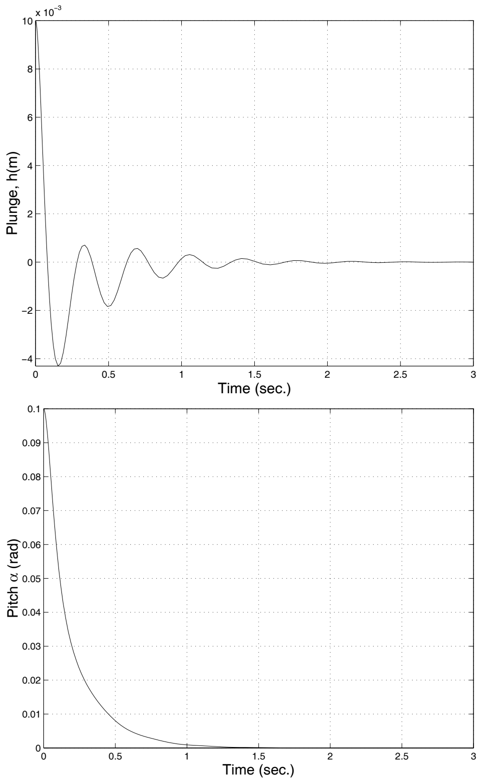

Fig. 3 Time response of derived controller for $U = 15m/s$ and $a = -0.4$.

6 Conclusion

This paper proposed a numerical TP model transformation capable of transforming a given state-space model to the TP model form whereupon the PDC controller design frameworks can be immediately executed. The TP model transformation is executable on explicit models, such that the behaviour of the system can be sampled for all possible values of the parameters. Whether these models represent a physical model, or are just the outcome of black-box identification (e.g. neural net model) is irrelevant. The TP model transformation can hence be viewed as a uniform gateway between various identification techniques and LMI based analysis and design.

References

[1] 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.

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

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

[4] Y. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programing: Theory and Applications*. SIAM Books, Philadelphia, 1994.

[5] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. Linear matrix inequalities in system and control theory. *Philadelphia PA:SIAM, ISBN 0-89871-334-X*, 1994.

[6] L. E. Ghaoui and S. I. Niculescu. *Advances in Linear Matrix Inequality Methods in Control*. SIAM Books, Philadelphia, l.e.ghaoui and s.i.niculescu, eds.: advances in design and control edition, 2000.

[7] C. W. Scherer and S. Weiland. *Linear Matrix Iequalities in Control*. DISC course lecture notes. DOWNLOAD: http://www.cs.ele.tue.nl/SWeiland/lmid.pdf, 2000.

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

[9] 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.

[10] 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.

[11] P. Baranyi, P. Korondi, R.J. Patton, and H. Hashimoto. Global asymptotic stabilisation of the prototypical aeroelastic wing section via TP model transfromation. *Asian Journal of Control*, (to be printed in Vol. 7, No. 2, 2004).

[12] Y. Yam, P. Baranyi, and C. T. Yang. Reduction of fuzzy rule base via singular value decom-

position. *IEEE Transaction on Fuzzy Systems*, 7(2):120–132, 1999.

[13] 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.

[14] J. Ko, A. J. Kurdila, and T. W. Strganac. Nonlinear control of a prototypical wing section with torsional nonlinearity. *Journal of Guidance, Control, and Dynamics*, 20(6):1181–1189, November–December 1997.

[15] D. M. Tang and E. H. Dowell. Flutter and stall response of a helicopter blade with structural nonlinearity. *Journal of Aircraft*, 29:953–960, 1992.

[16] D. M. Tang and E. H. Dowell. Comparison of theory and experiment for nonlinear flutter and stall response of a helicopter blade. *Journal of Sound and Vibration*, 165(2):251–276, 1993.

[17] S. J. Price, H. Alighanbari, and B. H. K. Lee. Postinstability behavior of a two dimensional airfoil with a structural nonlinearity of aircraft. *Journal of Aircraft*, 31(6):1395–1401, 1994.

[18] S. J. Price, H. Alighanbari, and B. H. K. Lee. The aeroeloasric response of a two dimensional airfoil with bilinear and cubic structural nonlinearities. *Proc. of the 35th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 94-1646*, pages 1771–1780, 1994.

[19] B. H. K. Lee and P. LeBlanc. Flutter analysis of two dimensional airfoil with cubic nonlinear restoring force. *National Aeronautical Istbalishment, Aeronautical Note-36, National Research Council, Ottava, Canada*, (25438), 1986.

[20] L. C. Zhao and Z. C. Yang. Chaotic motions of an airfoil with nonlinear stiffness in incompressible flow. *Journal of Sound and Vibration*, 138(2):245–254, 1990.

[21] T. O'Neil, H. C. Gilliat, and T. W. Strganac. Investigatiosn of aeroelsatic response for a system with continuous structural nonlinearities. *Proc. of the 37th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 96-1390*, 1996.

[22] J. Ko, A. J. Kridula, and T. W. Strganac. Nonlinear control theory for a class of structural nonlinearities in a prototypical wing section. *Proc. of the 35th AIAA Aerospace Scinece Meeting and Exhibit, AIAA paper 97-0580*, 1997.

[23] J. Ko, A. J. Kurdila, and T. W. Strganac. Nonlinear dynamics and control for a structurally nonlinear aeroelastic system. *Proc. of the 38th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 97-1024*, 1997.

[24] J. Ko, A. J. Kurdila, and T. W. Strganac. Adaptive feedback linearization for the control of a typical wing section with structural nonlinearity. *Journal of Guidance, Control, and Dynamics*, 21(5):718–725, September–October 1998.

[25] Y. C. Fung. *An Introduction to the Theory of Aeroelasticity*. John Wiley and Sons, New York, 1955.

[26] E. H. Dowell (Editor), H. C. Jr. Curtiss, R. H. Scanlan, and F. Sisto. *A Modern Course in Aeroelasticity*. Stifthoff and Noordhoff, Alpen aan den Rijn, The Netherlands, 1978.

[27] J. J. Block and T. W. Strganac. Applied active control for nonlinear aeroelastic structure. *Journal of Guidance, Control, and Dynamics*, 21(6):838–845, November–December 1998.

[28] E. H. Dowell. Nonlinear aeroelasticity. *Proc. of the 31th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 97-1024*, pages 1497–1509, 1990.

[29] Z. C. Yang and L. C. Zhao. Analysis of limit cycle flutter of an airfoil in incompressible flow. *Journal of Sound and Vibration*, 123(1):1–13, 1988.

[30] T. O'Neil and T. W. Strganac. An experimental investigation of nonlinear aeroelastic respons. *AIAA paper 95-1404, proc. 36th AIAA / ASME / ASCE/ AHS / ASC Structures, Structural Dynamics, and Materials Conference, New Orleans, Lousiana*, pages 2043–2051, 1995.