GLOBAL ASYMPTOTIC STABILIZATION OF THE PROTOTYPICAL AEROELASTIC WING SECTION VIA TP MODEL TRANSFORMATION

Péter Baranyi, Péter Korondi, and Hideki Hashimoto

ABSTRACT

A comprehensive analysis of aeroelastic systems has shown that these systems exhibit a broad class of pathological response regimes when certain types of non-linearities are included. In this paper, we propose a design method of a state-dependent non-linear controller for aeroelastic systems that includes polynomial structural non-linearities. The proposed method is based on recent numerical techniques such as the Tensor Product (TP) model transformation and the Linear Matrix Inequality (LMI) control design methods within the Parallel Distributed Compensation (PDC) frameworks. In order to link the TP model transformation and the LMI's in the proposed design method, we extend the TP model transformation with a further transformation. As an example, a controller is derived that ensures the global asymptotic stability of the prototypical aeroelastic wing section via one control surface, in contrast with previous approaches which have achieved local stability or applied additional control actuator on the purpose of achieving global stability. Numerical simulations are used to provide empirical validation of the control results. The effectiveness of the controller design is compared with a former approach.

KeyWords: Aeroelasticity, tensor product model transformation, linear matrix inequality, parallel distributed compensation.

I. INTRODUCTION

In the past few years various studies of aeroelastic systems have emerged. [1] 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 [2,3], by Price et al. in [4] and [5], by Lee et al. in [6], and a complete study of a class of non-linearities is in [7], [5]. O’Neil et al. [8] examined the continuous structural non-linearity of aeroelastic systems. These papers conclude that an aeroelastic system may exhibit a variety of control phenomena such as limit cycle oscillation, flutter and even chaotic vibrations.

Control strategies have also been derived for aeroelastic systems. [9] shows that controllers, capable of stabilizing structural non-linearity over flow regimes, can be derived via classical multivariable control methods. However, while several authors have investigated the effectiveness of linear control strategies for aeroelastic systems, experimental evidence has shown that linear control methods may not be reliable when non-linear effects predominate. For example in the case of large amplitude limit cycle oscilla-
tion behavior the linear control methodologies [9] do not stabilize aeroelastic systems consistently. [10] and [9] proposed non-linear feedback control methodologies for a class of non-linear structural effects of the prototypical aeroelastic wing section [8]. Papers [10,1] develop a controller, capable of ensuring local asymptotic stability of the prototypical aeroelastic wing section, via partial feedback linearization. It has been shown that by applying two control surfaces global stabilization can be achieved. For instance, adaptive feedback linearization [11] and the global feedback linearization technique were introduced for two control actuators in the work of [1].

The primary goal of this paper is to develop a non-linear state dependent controller design method which yields a controller capable of globally and asymptotically stabilizing a given prototypical aerolelastic wing section via one control surface. Our aeroelastic model incorporates the essential and well-characterized structural non-linearities that yield limit cycle oscillation at low speeds. The controller design is based on the Tensor Product (TP) model transformation introduced in [12,37] and the Liner Matrix Inequality (LMI) control design techniques within the Parallel Distributed Compensation (PDC) frameworks [13]. In order to have an immediate link between the TP model transformation and LMI’s, we extend the TP model transformation with the NO (Normality) transformation, introduced in [14,38], which yields the tight convex hull of the elements of the TP form. Without this extension the LMI’s are not feasible in the case of the present aeroelastic system. The control results will be compared with the most recently developed partial feedback linearization technique that also utilizes one control surface [1]. Former control solutions, applying one control surfaces, do not ensure global asymptotic stability. Those control solutions which guarantee global stability apply two control surfaces. The paper is organized as: Section II defines the notation to be utilized in the paper. Section III details the analytic dynamic model of the prototypical aeroelastic wing section. Section IV and V are the main contribution of this paper. Section IV presents the control design method. Section V focuses attention on how to execute the control design method on the global stabilization problem of the prototypical aeroelastic wing section, and shows the control results by numerical simulations. Comparison to an other control solution is also discussed in Section V. Section VI concludes the paper.

II. NOMENCLATURE

This section is devoted to introduce the notations being used in this paper.
• \( \{a, b, \ldots\} \): scalar values;
• \( \{a, b, \ldots\} \): vectors;
• \( \{A, B, \ldots\} \): matrices;
• \( \{A, B, \ldots\} \): tensors;
• \( A^\dagger \): the pseudo inverse of matrix \( A \);
• \( \mathbb{R}^{h_1 \times h_2 \times \cdots \times h_n} \): 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 \( A \) at row-column number \( i, j \) is symbolized as \( (A)_{ij} = a_{ij} \). Systematically, the \( il \)th column vector of \( A \) is denoted as \( a_{il} \), i.e. \( A = [a_1, a_2, \ldots] \).
• \( \otimes_{i,j,s,\ldots} \): are indices;
• \( \otimes_{i,j,s,\ldots} \): index upper bound: for example: \( i = 1 \ldots I, j = 1 \ldots J, n = 1 \ldots N \) or \( i_s = 1 \ldots I_s \).
• \( A_{\otimes n} \): \( n \)-mode matrix of tensor \( A \in \mathbb{R}^{h_1 \times h_2 \times \cdots \times h_n} \).
• \( A \otimes U \): \( n \)-mode matrix-tensor product;
• \( A \otimes_{\otimes} U_{\otimes} \): multiple product as \( A \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_n U_n \).

Detailed discussion of tensor notations and operations is given in [15].

III. EQUATIONS OF MOTION

In this paper, we consider the problem of flutter suppression for the prototypical aeroelastic wing section as shown in Fig. 1. The airfoil is constrained to have two degrees of freedom, the plunge \( h \) and pitch \( \alpha \). The equations of motion of the system have been derived and validated in various texts (for example, see [16], and [17]), and can be written as

\[
\begin{bmatrix}
  m & mx_a b \\
  mx_a b & I_a
\end{bmatrix}
\begin{bmatrix}
  \dot{h} \\
  \dot{\alpha}
\end{bmatrix}
+ \begin{bmatrix}
  c_b & 0 \\
  0 & c_a
\end{bmatrix}
\begin{bmatrix}
  h \\
  \alpha
\end{bmatrix}
= \begin{bmatrix}
  -L \\
  M
\end{bmatrix},
\]

(1)

where \( x_a \) is the non-dimensional distance between elastic axis and the centre of mass, \( m \) is the mass of the wing; \( I_a \) is the mass moment of inertia; \( b \) is the semi-chord of the wing, \( a \) is non-dimensional distance from the midchord to the elastic axis. Further, \( c_b \) and \( c_a \) respectively are the pitch and plunge structural damping coefficients, and \( k_h \) is the plunge structural spring constant. Several classes of non-linear stiffness contributions \( k_h(\alpha) \) have been studied in papers treating the open-loop dynamics of aeroelastic systems [2,18,19,7]. For purposes of illustration, we now introduce the use of polynomial non-linearities:

\[
k_h(\alpha) = k_{h0} + \sum_{i=1}^I k_{h i} \alpha^i.
\]

The parameters of \( k_h(\alpha) \) is obtained by curve fitting on the measured displacement-moment data for non-linear spring as [20]:

\[
k_h(\alpha) = 2.82(1 - 22.1\alpha + 1315.5\alpha^2 + 8580\alpha^3 + 17289.7\alpha^4).
\]
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 [16]:

\[
L = \rho U^2 b c_{l\alpha} \left( \alpha + \frac{h}{U} + \frac{1}{2} - a \right) b \frac{\alpha}{U} + \rho U^2 b c_{l\beta},
\]

\[
M = \rho U^2 b^2 c_{m\alpha} \left( \alpha + \frac{h}{U} + \frac{1}{2} - a \right) b \frac{\alpha}{U} + \rho U^2 b c_{m\beta}. \tag{2}
\]

The above $L$ and $M$ are accurate for the class of low velocity. Wind tunnel experiments are carried out in [9,20]. In the above equation $\rho$ is the air density, $U$ is the free stream velocity, $c_{l\alpha}$ and $c_{m\alpha}$ respectively are lift and moment coefficients per angle of attack, and $c_{l\beta}$ and $c_{m\beta}$ respectively are lift and moment coefficients per control surface deflection. $\beta$ is the control surface deflection.

The equations of motion derived above exhibit limit cycle oscillation, as well as other non-linear response regimes including chaotic response [20,18,7]. The system parameters to be used in this paper are tabulated in Table 1. These data are obtained from experimental models described in full detail in works [20,1].

With the flow velocity $U = 15(m/s)$ and the initial of $\alpha = 0.1(rad)$ and $h = 0.01(m)$, the resulting time response will achieve limit cycle oscillation, in good quality agreement with the behaviour expected in this class of systems. Note that Fig. 2 does not present the complete time history to preserve clarity. Papers [20,8] have shown the relations between limit cycle oscillation, magnitudes and initial conditions or flow velocities.

Combining equations (1) and (2) one obtains:

\[
\begin{align*}
&\begin{pmatrix}
    m & n x_b b \\
    n x_b b & I_x
\end{pmatrix} \begin{pmatrix}
    \dot{h} \\
    \dot{\alpha}
\end{pmatrix} \\
&\quad + \begin{pmatrix}
    c_h + \rho U b c_{l\alpha} & \rho U b^2 c_{l\alpha} \left( \frac{1}{2} - a \right) \\
    \rho U b^2 c_{m\alpha} & c_r - \rho U^2 b c_{m\alpha} \left( \frac{1}{2} - a \right)
\end{pmatrix} \begin{pmatrix}
    h \\
    \alpha
\end{pmatrix} \\
&\quad + \begin{pmatrix}
    k_h & \rho U^2 b c_{l\alpha} \\
    0 & -\rho U^2 b c_{m\alpha} + k_\alpha (\alpha)
\end{pmatrix} \begin{pmatrix}
    h \\
    \alpha
\end{pmatrix} = \begin{pmatrix}
    \rho b c_{l\alpha} \\
    \rho b^2 c_{m\alpha}
\end{pmatrix} U^2 \beta, \tag{3}
\end{align*}
\]

For the controller design, let us convert the above equation into state-space formulation. Let

Table 1. System Parameters

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>0.135m</td>
<td>span</td>
<td>0.6m</td>
</tr>
<tr>
<td>$k_h$</td>
<td>2844.4N/m</td>
<td>$c_r$</td>
<td>27.43Ns/m</td>
</tr>
<tr>
<td>$c_{\alpha}$</td>
<td>0.036Ns</td>
<td>$\rho$</td>
<td>1.225kg/m$^3$</td>
</tr>
<tr>
<td>$c_{\beta}$</td>
<td>6.28</td>
<td>$c_{\alpha}$</td>
<td>3.358</td>
</tr>
<tr>
<td>$c_{m\alpha}$</td>
<td>$0.5 + a c_{\alpha}$</td>
<td>$c_{m\beta}$</td>
<td>-0.635</td>
</tr>
<tr>
<td>$m$</td>
<td>12.387kg</td>
<td>$x_\alpha$</td>
<td>-0.3533 - $a$</td>
</tr>
<tr>
<td>$I_x$</td>
<td>0.065</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. Open loop response for plunge $(h)$ and pitch $(\alpha)$ motion is shown for $U = 15m/s$ and $a = -0.4$. 

Fig. 1. Aeroelastic model.
\[ x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} h \\ \alpha \\ \dot{h} \\ \alpha \end{pmatrix} \quad \text{and} \quad u(t) = \beta. \]

Then we have:

\[ \dot{x}(t) = A(p(t))x(t) + B(p(t))u(t) = S(p(t)) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}, \quad (4) \]

where

\[ A(p(t)) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_1 & -(k_2U^2 + p(x_2(t))) & -c_1(U) & -c_2(U) \\ -k_3 & -(k_4U^2 + q(x_2(t))) & -c_3(U) & -c_4(U) \end{pmatrix}, \]

\[ B(p(t)) = \begin{pmatrix} 0 \\ 0 \\ g_1 U^2 \\ g_4 U^2 \end{pmatrix}, \]

where \( p(t) \in \mathbb{R}^{N-2} \) contains values \( \alpha = x_3(t) \) and \( U \), since matrix function \( A(p(t)) \) is nonlinear in respect of values \( \alpha = x_3(t) \) and \( U \). The new variables are tabulated in Table 2. One should note that the equations of motion are also dependent upon the elastic axis location \( \alpha \).

**IV. CONTROLLER DESIGN METHOD**

The recently proposed very powerful numerical methods (and associated theory) for convex optimization involving Linear Matrix Inequalities help us with the analysis and the design issues of dynamic systems models (4) in acceptable computational time [21-31]. One family of these LMI based analysis and design methods are developed under the PDC frameworks [13,32-35], and functions with the convex state-space multiple-model form. The key idea of the control design method, to be proposed in this section, is that the TP model transformation, proposed in [12,37], can be applied (after some extension, see later in this section) to represent the given state-space model in convex state-space multiple-model form with specific characteristics, whereupon LMI’s within the PDC controller design framework can immediately be executed. As a result, this section derives viable control design method for the non-linear aeroelastic system described in the previous Section III. First of all, we define basic concepts to be utilized later on.

### 4.1 Parametrically varying state-space model

Consider parametrically varying state-space model:

\[ s(x(t)) = A(p(t))x(t) + B(p(t))u(t) \]

\[ y(t) = C(p(t))x(t) + D(p(t))u(t), \]

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

\[ S(p(t)) = \begin{pmatrix} A(p(t)) & B(p(t)) \\ C(p(t)) & D(p(t)) \end{pmatrix} \in \mathbb{R}^{(d+1) \times t} \]

is a parametrically varying object, where \( p(t) \in \Omega \) is time varying \( N \)-dimensional parameter vector, where \( \Omega = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \cdots \times [\alpha_N, \beta_N] \subset \mathbb{R}^N \) is a closed hypercube. \( p(t) \) can also include some elements of \( x(t) \). Further, for a continuous-time system \( s(x(t)) = \dot{x}(t) \) holds; and for a discrete-time system \( s(x(t)) = x(t+1) \) holds.

### 4.2 Convex state-space multiple-model

Equ. (6) can be approximated for any parameter \( p(t) \) as a convex combination of the \( R \) number of LTI (Linear Time Invariant) system matrices \( S_r, \ r = 1, \ldots, R \). Matrices \( S_r \) are also termed as vertex system matrices. Therefore, one can define basis functions \( W_r(p(t)) \in [0,1] \) such that

\[
\begin{align*}
\text{Table 2. System variables} \\
\frac{d}{dt} = m(t_a - mx_a b^2) \\
k_1 = \frac{L_a}{d} \\
k_3 = \frac{L_a b^2}{d} \\
p(\alpha) = \frac{L_a b}{d} k_3(\alpha) \\
c_1(U) = \frac{L_a b c_1 + m x_a b^3 c_1}{d} \\
c_3(U) = \frac{-m x_a b c_3 - m x_a b^3 c_3 - m b^3 c_3}{d} \\
g_3 = \frac{1}{d} (-L_a b c_3 - m x_a b^3 c_3) \\
k_2 = \frac{L_a b c_2 + m x_a b^3 c_2}{d} \\
k_4 = \frac{-m x_a b c_4 - m b^3 c_4}{d} \\
q(\alpha) = \frac{1}{d} k_4(\alpha) \\
c_1(U) = \frac{L_a b c_1 (1+\omega) - m x_a b c_1 (1+\omega)}{d} \\
c_3(U) = \frac{-m x_a b c_3 (1+\omega) - m b^3 c_3 (1+\omega)}{d} \\
g_4 = \frac{1}{d} (m x_a b^3 c_3 + m b^3 c_3)
\end{align*}
\]
matrix $S(p(t))$ belongs to the convex hull of $S$, as $S(p(t)) = \{S_i \mid i \in [1, N]\}$, where $w(p(t))$ defines the basis of the convex combination. The TP model transformation utilizes univariate basis functions. The tensor product form of the convex multiple-model with univariate basis functions is:

$$
\begin{bmatrix}
    x(t) \\
    y(t)
\end{bmatrix} \approx \left\{ \sum_{i=1}^{l_1} \sum_{j=1}^{l_2} \cdots \sum_{k=1}^{l_N} w_{i_1, j_1, \ldots, k_N}(p_r(t)) S_{i_1, j_1, \ldots, k_N} \right\} x(t) + u(t).
$$

(7)

The convexity of the multiple-model is ensured by the conditions:

$$
\forall n, i, p_r(t): w_{n,i}(p_r(t)) \in [0,1]; \quad \forall n, p_r(t): \sum_{i=1}^{l_n} w_n(i)(p_r(t)) = 1.
$$

(8)

(7) is also termed TP model in [12]. Function $w_{n,i}(p_r(t))$ is the $j$-th univariate basis function defined on the $n$-th dimension of $\Omega$, and $p_r(t)$ is the $n$-th element of vector $p(t)$. $l_n$ $(n = 1, \ldots, N)$ is the number of univariate basis functions used in the $n$-th dimension of the parameter vector. The multiple index $(i_1, i_2, \ldots, i_N)$ refers to the LTI system corresponding to the $i$-th basis function in the $n$-th dimension. Hence, the number of vertex systems $S_{i_1, i_2, \ldots, i_N}$ is obviously $R = \prod_{n=1}^{N} l_n$.

One can rewrite (7) in the concise TP form as:

$$
\begin{bmatrix}
    x(t) \\
    y(t)
\end{bmatrix} \approx \left\{ \sum_{i=1}^{l_N} S_{i\otimes n}(p_r(t)) \right\} x(t) + u(t),
$$

that is $S(p(t)) \approx \sum_{i=1}^{l_N} S_{i\otimes n}(p_r(t))$. (9)

Here, $\varepsilon$ represents the approximation error between the given parameter-varying system matrix and the TP model (it can be the norm $L_2$ of the error for all $p(t)$), row vector $w_n(i) \in \mathbb{R}^{l_n}$, the convex multiple-model form is constructed from the linear constant system matrices $S_{i_1, i_2, \ldots, i_N} \in \mathbb{R}^{O \times I}$. The first $N$ dimensions of $S$ are assigned to the dimensions of $\Omega$.

$S(p(t))$ can be exactly decomposed into multiple-model form ($\varepsilon = 0$ in (9)) in many cases. However, one should face that exact finite element multiple-model (the number of LTI systems in the model is finite) representation does not exist in general ($\varepsilon > 0$ in (9)), see [36]. In this case the task is to achieve an acceptable $\varepsilon$ via increasing the number of LTI systems in the multiple-model. This, however, soon leads to the approximation trade-off. The dynamic model in the example of this paper can exactly be represented by finite element convex multiple-model form, see later in Section V. The reason of this fact is that the model described in Section III has TP structure. If possible, one may profit this fact via analytical derivations as well.

4.3 TP model transformation

4.3.1 The goal of the TP model transformation

The goal of the TP model transformation is to transform a given state-space model (5) into convex multiple-model form, namely to (9) with conditions (8). Let the synopsis of the TP model transformation be:

$$
(w_{n=1, \ldots, N}(p_r(t)), S) \approx TP_{\text{transf}}(S(p(t)), \Omega),
$$

(10)

where $S(p(t)) \in \mathbb{R}^{O \times I}$ is from (6), and $\Omega \subset \mathbb{R}^N$. It is assumed that $S(p(t))$ can be determined for all $p(t) \in \Omega$, where $S(p(t))$ can be given either by explicit analytic forms, neural network or other soft computing identification techniques. In the example of this paper the $S(p(t))$ is calculated by analytic forms defined in (4). The resulting vectors $w_n(p_r(t)) \in \mathbb{R}^{l_n}$ and tensor $S$ in the left side of (10) are defined at (9), $w_{n=1, \ldots, N}(p_r(t))$ contain the basis functions. The basis functions satisfy (8). The TP model transformation finds the tight convex hull of the LTI systems. In other aspects, the TP model transformation searches a solution for the basis functions where:

$$
\forall n, i : \exists p_{n,i} : w_{n,i}(p_{n,i}) \approx 1, \delta \to 0,
$$

(11)

namely the value of each basis functions reaches 1 or get close to 1. This simply means, in the case of (8), that $\forall i_1, i_2, \ldots, i_N : \exists p_{i_1, i_2, \ldots, i_N}$, where the LTI systems of the convex multiple-model equal to the given system over the operation point $p_{i_1, i_2, \ldots, i_N}$, where $S_{i_1, i_2, \ldots, i_N} = S(p_{i_1, i_2, \ldots, i_N})$ or they are close to the real system as possible $S_{i_1, i_2, \ldots, i_N} \approx S(p_{i_1, i_2, \ldots, i_N})$. Namely,

minimize $\|S_{i_1, i_2, \ldots, i_N} - S(p_{i_1, i_2, \ldots, i_N})\|_{L_2}$

subject to the constrain on the number of basis functions (note that if the number of basis increases to infinity one can assign each resulting LTI system to an operation point).

4.3.2 Modifications of the TP model transformation presented in [12, 37]

We propose a TP transformation method here that is an extended version of the TP model transformation proposed in [12]. We specialize the TP model transformation to yield convex multiple-model form with special characteristics whereupon LMI based techniques can immediately be executed. We added two crucial steps to the TP model transformation. First, we extend the TP model transformation with NO transformation developed in [14,38]. The NO transformation determines the basis function subject to (11). Namely, it finds the tight convex hull of the LTI vertex systems. NO transformation has crucial role, for instance, the control design of the present example in Section V fails.
if NO transformation is not applied. Furthermore, we find that when the proposed control design method leads to a solution the NO transformation is necessary in most cases. Without NO transformation the feasibility test of the LMI in the second step of the control design method, discussed later at equation (13), fails in most cases. We observed this fact via a number of examples, and our future work is to investigate and give a mathematical explanation for this fact. The second additional step helps us with generating continuous basis functions. The introduction of the TP model transformation method in [14,37] shows how to generate piece-wise linear basis functions, which implies that the resulting convex multiple-model is exact in the sense that increasing the number of pieces to infinity the exact basis is obtained. As a matter of fact infinite element cannot be computed in a numerical technique. The present improvement shows that how to determine the exact values of the basis functions over any points of the transformation space.

4.3.3 The algorithm of the TP model transformation

The algorithm has three main parts. The first one deals with the discretisation of the given model over a hyper rectangular grid. The second part extracts the LTI vertex systems of the multiple model. If the given model has finite element TP structure then the second step finds it. The third step reconstructs the basis functions from the extracted LTI systems.

**Method 1. TP model transformation**

Step 1: Define the transformation space: $p(t) \in \Omega : [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_N, b_N]$.

Step 2: Define a hyper rectangular grid by equidistantly located grid-lines: $g_{n,m} = a_n + \frac{b_n - a_n}{M_n} (m_n - 1)$, $m_n = 1 \ldots M_n$.

Step 3: Sample the given function $S(p(t))$ over the grid-points:

$$S_{n_1, n_2, \ldots, n_N} = S(p_{n_1, n_2, \ldots, n_N})$$

where $p_{n_1, n_2, \ldots, n_N} = (g_{1,n_1}, g_{2,n_2}, \ldots, g_{N,n_N})$ and superscript “$s$” means “sampled”.

Step 4: Store the sampled matrices $S_{n_1, n_2, \ldots, n_N}$ into the tensor $S' \in \mathbb{R}^{M_1 \times M_2 \times \ldots \times M_N \times O \times 1}$ then execute HOSVD (Higher Order Singular Value Decomposition) extended with NN (Non-Negativity), SN (Sum-Normalization) and NO (Normality) transformation, on the first $N$ dimension of tensor $S'$. 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 [15]. The SN and NN transformations are presented with the TP model transformation method in [14,37]. The NO transformation is introduced in [14,38]. The NO transformation has various implementations and its description is lengthy, therefore the authors refer to a detailed description [14] and [38] in this matter.

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

The result of this step is:

$$S' \approx S' \circlearrowleft_{\varepsilon} U_{\varepsilon}$$

where the size of $S$ is $(I_1 \times I_2 \times \ldots \times I_N \times O \times 1)$, where $\forall n : I_n \leq M_n$. If only zero singular values are discarded in HOSVD then $\varepsilon = 0$. If non-zero singular values are discarded too then $\varepsilon > 0$. An upper error bound of $\varepsilon$ is given in the next Step 5. Matrix $U_{\varepsilon} \in \mathbb{R}^{M_1 \times I_1}$ is resulted by the extended HOSVD. It is transformed by NN, SN and NO transformations from the singular matrices resulted by the HOSVD.

Step 5: Determine the basis from matrices $U_{\varepsilon}$. Each column $u_{n_1, n_2, \ldots, n_N}$ of matrix $U_{\varepsilon} \in \mathbb{R}^{M_1 \times I_1}$ determines one basis function $w_{n_1, n_2, \ldots, n_N}(p_d(t))$ of variable $p_d(t)$. The values $u_{n_1, n_2, \ldots, n_N}$ of one column $u_{n_1, n_2, \ldots, n_N}$ define the values of the basis function $w_{n_1, n_2, \ldots, n_N}(p_d(t))$ over the location $g_{n_1, n_2, \ldots, n_N}$ of the grid-lines:

$$w_{n_1, n_2, \ldots, n_N}(g_{n_1, n_2, \ldots, n_N}) = u_{n_1, n_2, \ldots, n_N}$$

(1) Piece-wise linear basis: In order to get the basis functions from the discrete values of the basis, let the above values $u_{n_1, n_2, \ldots, n_N}$ (in one column of $U_{\varepsilon}$) be connected by straight lines. This leads to a piece-wise linear approximation of $S(p(t))$. The advantage of the piece-wise basis is that it requires low computational cost. The less distance we have between the grid-lines $g_{n_1, n_2, \ldots, n_N}$ (namely the larger $M_n$ we define in the Step 2) the better approximations of the basis functions are obtained. Modern computer capacity can handle sampling grid dense enough for keeping the error, caused by the piece-wise approximation, dispensable in numerical sense.

(2) Continuous basis: All points of the basis can be determined by the help of the given $S(p(t))$.

Determination of the basis functions $w_a(p_d(t))$: Let $p_d(t)$ be fixed to the location of the grid-lines, for instance:

$$p_{k}(t) = g_{k, 1} \quad k = 1, ..., N, \quad k \neq n$$

Then for $p_{n}(t)$:

$$w_a(p_{n}(t)) = S(p(t)) \left( S' \circlearrowleft_{\varepsilon} U_{\varepsilon} \right)_{n, 1}$$

where superscript “$\circlearrowleft_{\varepsilon}$” denotes pseudo inverse and $u_{n, 1}$ is the first row vector of $U_{\varepsilon}$.

From the above steps we can derive an error bound for the difference between the resulting convex multiple-model form and the original model over the sampling grid defined.
in Step 2 and sampled in Step 3:

\[ S(p_{m_1, m_2, \ldots, m_N}) \approx S \bigotimes_{n=1}^{N} w_n(g_{n, m_n}), \]  

(12)

where \( p_{m_1, m_2, \ldots, m_N} = (g_{1, m_1} g_{2, m_2} \ldots g_{N, m_N}) \), and

\[ \varepsilon = \max_{p_{m_1, m_2, \ldots, m_N}} \left( \| S(p_{m_1, m_2, \ldots, m_N}) \bigotimes_{n=1}^{N} w_n(g_{n, m_n}) \|_{L_2} \right) \]

\( \varepsilon \) caused by the discarded singular values \( \sigma_i \) during the execution of the extended HOSVD:

\[ \varepsilon \leq \sum_{i} \sigma_i. \]

For further details and proofs, see [14,39]. One may ask what is the error in between the grid points? Practically, we can avoid this question by assuming that we apply a dense sampling grid, and finally we numerically check the equivalency of the original model and the resulting convex multiple-model over a huge number of points. Theoretically, we know that if the original model has finite element tensor structure then the TP model transformation finds it. As a matter of fact, the maximum number of LTI vertex systems resulted by the TP model transformation is bounded by the number of sampling grid lines. Therefore, in extreme cases, one should increase the number of grid lines to achieve exact transformation. The exact transformation means that the original model has finite element tensor product structure that is extracted by the transformation. In this case the original model and the resulting multiple model is equivalent. Since the TP model transformation is a numerical method the exactness is understood in numerical sense here. This means that the only error we have between the original and the resulting model is caused by the numerical computation (it is about \( 10^{-10} \)). If the original model does not have finite element TP structure then we cannot estimate the bound of the error in between the grid points during performing the TP model transformation. But we can decrease the error via increasing the density of the sampling points, and finally check the resulting error numerically.

It is worth noticing here again for practical cases that, if we apply a dense sampling grid, and we find zero singular values in all dimensions (the number of nonzero singular values is not increasing with the increasing number of grid lines) then we can guess that the original model has finite element TP structure and we have found the exact convex multiple-model (\( \varepsilon = 0 \) in (9)) and we can check the accuracy of the resulting model numerically. This is the case, for instance, in the example of Section V. For further details in this matter the authors refer to the examples of [14,12,37].

### 4.4 LMI based controller design under PDC framework

The PDC design techniques determine one feedback to each resulting LTI vertex model of the resulting convex multiple-model. Namely, feedback gains in tensor \( K \) is derived from tensor \( S \). For brevity, let us use the following symbolic notation:

\[ K \leftarrow PDC(S, \text{stability_theorem}). \]  

(13)

Then the control value is defined by the basis function system applied in the resulting multiple-model (9) as:

\[ u(t) = - \left( \bigotimes_{n=1}^{N} w_n(p_n(t)) \right) x(t), \]  

(14)

which ensures the system performance defined by the selected “stability_theorem” in (13). “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 [13].

For the sake of simplicity, we focuses attention on global asymptotic stability only in the control design of the aeroelastic wing section of Section V, and, therefore, we apply one of the most basic LMI theorems capable of ensuring global asymptotic stability. In order to complete the paper, we recall briefly this basic LMI theorem here. In the Step 1 of the following Method 2 we introduce the link between the result of the TP model transformation and the typical form of LMI theorems developed under the PDC design frameworks.

**Method 2. (Global and asymptotic stabilization of continuous convex multiple-model)** Assume a given state-space model in multiple-model form (9) with conditions (8).

**Step 1:** Link between the convex multiple-model and the LMI formulations

In order to have a direct link to the typical form of LMI theorems, let the following indexing be defined:

\[ \mathbf{S}_r = \begin{pmatrix} A_r & B_r \\ C_r & D_r \end{pmatrix} = S_{i_1, i_2, \ldots, i_k}, \]

where \( r = \text{ordering} \ (i_1, i_2, \ldots, i_k) \ (r = 1 \ldots R = \prod I_k) \). The function “ordering” results in the linear index equivalent of an \( N \) dimensional array’s index \( i_1, i_2, \ldots, i_k \), when the size of the array is \( I_1 \times I_2 \times \cdots \times I_k \). Let the basis functions be defined according to the sequence of \( r \):

\[ w_r(p(t)) = \prod_{n} w_{n,i_n}(p_n(t)). \]

**Step 2:** LMI theorem for global asymptotic stability

The present LMI theorem can be derived from the
Lyapunov stability theorems for global and asymptotic stability as shown in [13], and is written as:

Find \( X > 0 \) and \( M_i \), satisfying equ.

\[
-XA^T - A^TX + M_i^TB^T + B^TM_i > 0
\]

for all \( r \) and

\[
-XA^T - A^TX - XA^T - A^TX + M_i^TB^T + B^TM_i + M_i^TB_i^T + B_i^TM_i > 0
\]

for \( r < s \leq R \), except the pairs \((r, s)\) such that \( w_r(p(t)) = 0, \forall p(t) \) obtained from the solutions \( X \) and \( M_i \), as

\[
K_i = M_iX^{-1} \quad \text{and} \quad P = X^{-1}.
\]

Then, by the help of \( r = ordering(i_1, i_2, ..., i_3) \) one can define feedbacks \( K_{i_0, i_0, ..., i_0} \) from \( K_i \) obtained in (17) and store into tensor \( K \) of (14).

V. GLOBAL ASYMPTOTIC STABILISATION OF THE PROTOTYPICAL AEROELASTIC WING SECTION

The primary goal of this study is to develop a nonlinear state dependent controller capable of globally and asymptotically stabilizing a given prototypical aeroelastic wing section via one control surface in contrast with previous approaches which have achieved local stability or applied additional control actuator on the purpose of achieving global stability. The control results will be compared with the previously developed partial feedback linearization technique [1,10] that also utilizes one control surface.

To demonstrate the performance of the controlled system, numerical experiments are presented in this section. In order to be comparable to other published results, for instance to [1], the numerical examples are performed with \( a = -0.4 \) and with free stream velocity \( U = 15 \text{m/s} \) and \( U = 20 \text{m/s} \), a velocity which exceeds the linear flutter velocity \( U = 15.5 \text{m/s} \), and for initials \( h = 0.01 \text{(m)} \) and \( \alpha = 0.1 \text{(rad)} \).

5.1 Controller design

This subsection is intended to perform the proposed controller design method, discussed in the previous Section IV, to the present aeroelastic system defined in (4). First of all, let us define the transformation space \( \Omega \). We are interested in the interval \( U \in [14,25] \text{ (m/s)} \) and we presume that the interval \( a \in [-0.1,0.1] \text{ (rad)} \) is sufficiently large enough (note that these intervals can arbitrarily be set, but should not be smaller than required by vector \( p(t) \)). Therefore let: \( \Omega : [14, 25] \times [-0.1,0.1] \) in the present example. Executing the TP model transformation, namely Method 1 (with a 300 x 300 grid), yields that the dynamic model (4) can be represented exactly in convex-multiple model form (9) over a 3 times 2 basis function system. This means that the model in (4) can be described exactly by the convex combination of 3 x 2 = 6 linear vertex systems. When we numerically check the equivalency of the aeroelastic model (4) to the resulting convex multiple-model (9) over a huge number of points, we find that the difference is under 10^{-12} that is zero in numerical sense. The resulting basis functions are depicted in Fig. 3.

One should observe in Fig. 3 that the TP model transformation finds the tight convex hull of the LTI systems, meaning that all basis functions reach value one over at least one value of the \( \Omega \), except one basis function of \( U \). The LTI systems assigned to this basis function is close to (but not identical to) the real system, over any \( p(t) \).

Note that, we may try to derive the basis functions analytically from (3). The basis functions of \( a \) can be extracted from \( k_a(\alpha) \). Finding the basis functions of \( U \), however, seems to be rather complicated. The analytic derivation becomes really hard if we require to find the tight convex hull of the resulting LTI systems (without ensuring the tight convex hull the LMI of Method 2 is not feasible, hence, the controller is not derivable in the present case).

Executing the TP transformation takes a few seconds on a regular pentium computer.

Having the above resulting convex multiple-model form we can execute Method 2. The numerical solution of Method 2 (for instance in MATLAB LMI toolbox) yields that the LMI of Method 2 with the LTI systems, resulted by Method 1, is feasible:

\[
X = 10^3 \begin{bmatrix}
0.0155 & 0.0016 & -0.0281 & -0.0426 \\
0.0016 & 0.0102 & 0.0231 & -0.0842 \\
-0.0281 & 0.0231 & 4.4282 & -0.2671 \\
-0.0426 & -0.0842 & -0.2671 & 4.4726
\end{bmatrix}
\]

Equ. (17) yields 6 LTI feedback gains \( k_i \) as:

\[
\begin{align*}
\begin{bmatrix} k_{i_1} \end{bmatrix} &= \begin{bmatrix} -6.0603 & 9.1440 & -0.0552 & -0.5682 \end{bmatrix} \\
\begin{bmatrix} k_{i_2} \end{bmatrix} &= \begin{bmatrix} -5.6193 & -9.3196 & -0.1219 & -0.4941 \end{bmatrix} \\
\begin{bmatrix} k_{i_3} \end{bmatrix} &= \begin{bmatrix} -2.1869 & 4.3180 & -0.0947 & -0.1497 \end{bmatrix} \\
\begin{bmatrix} k_{i_4} \end{bmatrix} &= \begin{bmatrix} -2.4581 & -7.5076 & -0.1493 & -0.2736 \end{bmatrix} \\
\begin{bmatrix} k_{i_5} \end{bmatrix} &= \begin{bmatrix} -6.4198 & -4.6124 & -0.0343 & -1.2409 \end{bmatrix} \\
\begin{bmatrix} k_{i_6} \end{bmatrix} &= \begin{bmatrix} -6.4198 & -4.6124 & -0.0343 & -1.2409 \end{bmatrix}
\end{align*}
\]
Then the control value is computed as (8):

\[ u(t) = -\sum_{i=1}^{3} \sum_{j=1}^{2} w_{ij}(U)w_{2j}(\alpha) k_{i,j} x(t). \]

At this point we achieved the main goal of the present control design. We designed a controller that ensures global asymptotic stability for the present aeroleastic system. We should emphasize here again that the model is globally stabilized in practical sense. The model is accurate only for low speeds. Therefore the global stability has practical significance only in the parameter space \( \Omega: [14,25] \times [-0.1, 0.1] \). We should also make clear here that we restricted the transformation for this practically interesting space \( \Omega \), where the model is validated. This means that the achieved global stability is understood only in this bounded space. The TP model transformation is a numerical transformation, it can, hence, be executed over arbitrarily extended, but bounded space. The resulted global stability is obviously understood only in this extended space. Therefore, one may conclude that this is a semi-global stability.

5.2 Comparison to a former solution

We compare the control results with the previously developed partial feedback linearization technique [1,10] that also utilizes one control surface.

5.2.1 Main differences

The primary conclusion of the comparison is that:

- the proposed method leads to the solution of the global (semi-global) asymptotic stability of the prototypical aerelastic wing section. The solution applies only one control surface unlike previous works;
- further, the proposed method opens a gate to a large variety of LMI theorems, which helps us with designing controllers capable of satisfying further control specifications beyond global stability;
- once we have the program (for instance in MATLAB) of the proposed control design method the design of the controller takes a few minutes, unlike analytic works.

5.2.2 Numerical simulations

Figures 4 and 5 respectively show the control results by the derived controller for \( U = 15 \text{m/s} \) and \( U = 20 \text{m/s} \), and for initials \( h = 0.01(\text{m}) \) and \( \alpha = 0.1(\text{rad}) \). The system is stabilized in 5 sec. For comparison, Figs. 6 and 7 present the time response of the controller, developed via exact feedback linearization, for the same case, see [1]. Comparing the simulated results one can observe that the controller developed in this paper is faster, and, except the first moment, its control value is significantly smaller than the control values on Figs. 6 and 7. One can also observe that the values of \( \alpha \) converge smoothly to zero on Fig. 4 and 5 unlike to the values of \( \alpha \) on Figs. 6 and 7.

Remark 1. Note that again, for the sake of simplicity, we did not set 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 given by selecting proper LMI design theorems [13].

VI. CONCLUSION

In this paper we have proposed a numerical control design method, which is based on the TP model transformation and LMI’s within the PDC control design frameworks, to design non-linear controllers for the prototype aerelastic wing section that includes structural non-linearity. We extended the TP model transformation with two steps, which are necessary in the control design method, and proposed the immediate link between the resulting convex multiple-model form and LMI’s of the PDC framework. Without any control effort, or with linear controllers, the aerelastic system reveals various kinds of non-linear phenomenon including limit cycle oscillation as noted in various text. The derived controller globally and
Fig. 4. Time response of derived controller for $U = 15\text{ m/s}$ and $a = -0.4$.

Fig. 5. Time response of derived controller for $U = 20\text{ m/s}$ and $a = -0.4$.

Fig. 6. Time response of exact feedback linearization method for $U = 15\text{ m/s}$ and $a = -0.4$.

Fig. 7. Time response of exact feedback linearization method for $U = 20\text{ m/s}$ and $a = -0.4$. 
asymptotically stabilises the system and utilizes one control surface. A further advantage is that the proposed method is based on numerical steps. The controller can thus be determined automatically and without analytic derivations. The effectiveness of the controller has been compared with terms in the proposed design method. As a further development of this work the authors plan to design controllers for advantageous control performance via applying different LMI theorems.

REFERENCES


**Péter Baranyi** was born in Hungary in 1970. He received degrees M.Sc. in Electrical Engineering (1994), M.Sc. in Education of Engineering Sciences (1995) and Ph.D. (1999) from the Technical University of Budapest. He received the Youth Prize of the Hungarian Academy of Sciences (2000), the International Dennis Gabor Award (2000), the Japanese STA Fellowship Award (2000), and the Outstanding Young Technological Innovator of the Year 2002 II. prize in Hungary. He also received the first prize of Young Scientist Day 2003 held by SAMSUNG. He is the vice president of the Hungarian Society of IFSA (2003-). He was a founder member of the Integrated Intelligent Systems Japanese-Hungarian Laboratory. He had research positions at Chinese University of Hong Kong (1996 and 1998); The University of New South Wales (1997), Australia; CNRS LAAS Institute in Toulouse (1996), France, and Gifu Research Institute (2000-2001), Japan; The University of Hull (2002-2003), U.K.; etc.

**Péter Korondi** received Dipl. Eng. and Ph.D. degrees in electrical engineering from the Technical University of Budapest in 1984 and 1995, respectively. His research interests include tele-manipulation and motion control. Since 1986, he has been at the Technical University of Budapest (the name of University was changed to Budapest University of Technology and Economics in 2000). He teaches Motion Control, Robot control, Power Electronics. He worked for 2 years in the laboratory of Professor Harashima and Professor Hashimoto at the Institute of Industrial Science at the University of Tokyo from April of 1993 to April of 1995. His cooperation did not end upon his return to Hungary. He is in daily contact with his Japanese colleagues through the internet. He spends a month in Tokyo each year to continue their joint research. As a result of this cooperation, the Intelligent Integrated System Japanese-Hungarian Joint Laboratory was founded in 2001. He is a founding member of the International PEMC Council as well, which became a chapter of the European Power Electronic Association.
Hideki Hashimoto receive the B.Sc., M.Sc. and Dr.Eng. degrees in Electrical Engineering from the University of Tokyo, Tokyo, Japan, in 1981, 1984 and 1987, respectively. He is currently an Associate Professor with the Institute of Industrial Science, University of Tokyo. From 1989 to 1990, he was a Visiting Scientist in the Laboratory for Informational and Decision Systems and the Laboratory for Electromagnetics and Electronics Systems at the Massachusetts Institute of Technology Cambridge, MA, U.S.A. His research interests are in the areas of robotics, control, intelligent transportation, systems, and human-machine interface. He is an active member of the IEEE Industrial Electronics, IEEE Robotics and Automation Societies, and has served on the Administrative Committees of both. He is also a member of the Japanese Society of Instrument and Control Engineering (SICE), Institute of Electrical Engineers of Japan (IEEJ) and Robotics Society of Japan (RSJ).