TP Tool - a MATLAB Toolbox for TP Model Transformation

Szabolcs Nagy, Zoltán Petres, Péter Baranyi

Systems and Control Laboratory
Computar and Automation Research Institute
Hungarian Academy of Sciences
Kende u. 13-17, H-1111 Budapest, Hungary
E-mail: nszabolcs@gmail.com, {petres,baranyi}@sztaki.hu

Abstract: The Tensor Product model transformation is a numerical method that is capable of uniformly transforming LPV (linear parameter-varying) dynamic models into polytopic forms, both in a theoretical and algorithmic context. Using the TP model transformation, different optimization and convexity constraints can be considered, and transformations can be executed without any analytical interactions, within a reasonable amount of time (irrespective of whether the model is given in the form of analytical equations resulting from physical considerations, as an outcome of soft computing based identification techniques such as neural networks or fuzzy logic based methods, or as a result of a black-box identification). Thus, the transformation replaces the usual analytical and oftentimes complex conversions with a numerically tractable and straightforward series of operations.

The TP model transformation generates two kinds of polytopic models. Firstly, it numerically reconstructs the HOSVD (Higher Order Singular Value) based canonical form of LPV models. This is a new and unique polytopic representation. This form extracts the unique structure and various important properties of a given LPV model in the same sense as the HOSVD does for matrices and tensors. Secondly, the TP model transformation generates various convex polytopic forms, upon which LMI (Linear Matrix Inequality) based multi-objective control design techniques can immediately be executed in order to satisfy the given control performance requirements.

TP tool is a MATLAB Toolbox that implements the Tensor Product Model Transformation based Control Design framework. It is available at http://tptool.sztaki.hu.

Keywords: LPV, LMI, polytopic, HOSVD, TP, MATLAB

1 Introduction

1.1 Motivation and Background

The development of the Higher Order Singular Value Decomposition (HOSVD) based canonical form of Linear Parameter Varying (LPV) dynamic models and the TP model transformation was motivated by the significant paradigm changes in control theory, mathematics, system modeling and identification theories, appearing almost simultaneously in the last few decades.

1.1.1 Multi-Objective Non-Linear Control Theory

The Linear Parameter Varying (LPV) representations and Linear Matrix Inequality (LMI) based analysis and system control design are in the focus of modern control theories. LPV systems appear in the form of Linear Time Invariant (LTI) state-space representations where the elements of the S(ρ) system matrices can depend on an unknown but at any time instant measurable vector parameter ρ. This parameter can be a function of time or state variables. The parameters may represent constant but unknown uncertainties or external time signals. These properties show relations to the theory of uncertain systems with parametric uncertainties and to the theory of LPV systems, too. The application of LPV system representations appeared in relation to aerospace control and it represents a systematic approach to gain scheduling control for nonlinear systems (Shamma and Athans, 1991). Passivity and H∞ theory have been extended to design robust controllers for LPV systems, see eg. Lim and How (2002), Becker and Packard (1994). Moreover, the study of LPV systems provides additional insights into some longstanding and sophisticated problems in robust adaptive control (see Athans et al, 2005), switching control systems (see Hespanha et al, 2003) and in intelligent control (see Feng and Ma, 2001, Ravindranathan and Leitch, 1999). The appearance of Lyapunov-based stability criteria in the form of LMIs made a significant improvement. From this point on, stability questions were formulated in a new representation, and the feasibility of Lyapunov-based criteria was reinterpreted as a convex optimization problem, as well as extended to an extensive model class. The pioneers Gahinet, Bokor, Balas, Chilai, Boyd, and Apkarian were responsible for establishing this new concept. Soon, it was also proved that this new representation could be used for the formulation of different control performances beyond the stability issues together with the optimization problem. Ever since, the number of papers about LMIs are increasing drastically in various topics such as optimal LQ control, robust H∞ control/H∞ synthesis, μ-analysis, quadratic stability, Lyapunov-based stability, multi-model and multi-objective state-feedback control of parameter-dependent systems, control of stochastic systems. Boyd's paper states that it is true of a wide class of control problems that if the problem is formulated in the form of LMIs, then the problem is practically solved. In parallel, efficient numerical mathematical methods and algorithms were developed for solving convex optimization problems—thus LMIs (Nesterov and Nemirovski). As a result, with the usage of numerical methods of convex optimization we consider a large set of problems that require the resolution of a large number of convex algebraic Ricatti-equations solved today, in spite of the fact that the result of the obtained solution is not a closed (in its classical sense) analytical equation.

1.1.2 System Modeling and Identification Theory

In the last decade, various new representations of dynamic models have emerged in systems theory. The origins of this paradigm shift can be linked with the famous speech given by Hilbert in Paris, in 1900. Hilbert listed 23 conjectures, hypotheses, concerning unsolved problems which he believed would prove to be the biggest challenge in the 20th Century. According to Hilbert's 13th conjecture, there exist continuous multi-variable functions which cannot be decomposed as the finite superposition of continuous functions of a smaller number of variables. In 1957, Arnold disproved this hypothesis. Moreover, in the same year, Kolmogorov formulated a general representation theorem, along with a constructive proof, which allows for a decomposition into one-dimensional functions (see also Sprecher and Lorentz). This proof justified the existence of 'universal approximators'. Based on these results, starting from the 1980s, it has been proved that universal approximators exist within the categories defined by approximation tools such as biologically inspired neural networks and genetic algorithms, as well as fuzzy logic. As a result, these approximators have appeared in the identification models of systems theory, and turned out to be effective tools even for systems that can hardly be described in an analytical way.

As a result of the above, we have various effective identification techniques today. However, the identified models obtained as a result of these alternative techniques are described in forms that are quite far form the models given by analytical closed formulas derived via physical considerations engendered by the system under scrutiny.

1.1.3 Tensor Algebra

During the last 150 years several mathematicians (Beltrami, Jordan, Sylvester, Schmidt and Weyl, to name a few) were responsible for establishing the foundations of the Singular Value Decomposition (SVD) and for developing its theory, which is one of the most fruitful developments in linear algebra. A very recent result is the Higher-Order generation of the SVD (HOSVD) to tensors (Lathauwer, 2000, SIAM journal). The Workshop on Tensor Decompositions and Applications held in Luminy, Marseille, France, 2005 was the first event where the key topic was HOSVD. Its very unique power comes from the fact that it can decompose a given N-dimensional tensor into a full orthonormal system in a special ordering of higher order singular values, expressing the rank properties of the tensor in the order of L2-norm. In effect, the HOSVD is capable of extracting the very clear and unique structure underlying the given tensor.

1.1.4 TP Model Transformation

We can conclude that on the one hand we have powerful optimization and control design techniques for polytopic and affine forms of LPV models, and on the other hand we have a large variety of identification techniques. However, we can hardly link these two aspects because of the lack of a uniform representation. Therefore, there is a great demand for automatic and uniform ways to convert various alternative representations into a unique polytopic or affine form.

The TP model transformation was proposed as a possible solution. It is capable of transforming given LPV models into proper polytopic forms, upon which LMI-based design techniques are immediately executable. The result of the TP model transformation is a TP model that belongs to the class of polytopic models, where the parameter-dependent weighting of the vertex systems are one-dimensional functions of the elements of the parameter vector. This form offers a relatively simple way to describe various convex hull generations in terms of matrix operations.

2 TP Tool – MATLAB Toolbox for TP Model Transformation

TP Tool is a MATLAB Toolbox that implements the Tensor Product Model Transformation based Control Design framework. In the following sections we provide a brief introduction to its usage, for more details we refer to the homepage of the toolbox [1].

2.1 HOSVD-based Canonical Form of an LPV Model

This simple example shows how to obtain the higher order singular value decomposition (HOSVD) based canonical form of a linear parameter-varying (LPV) model. (This example is also included in the downloadable source files of the toolbox)

Consider the following LPV model:

$$\dot{\mathbf{x}}(t) = \mathbf{S}(\mathbf{x}(t))\mathbf{x}(t)\,, \tag{1}$$

where

$$\mathbf{S}(\mathbf{x}(t)) = \begin{pmatrix} 2\cos(4x_1(t)) & x_2(t) \\ 3 & x_2^3(t)\sin(x_1(t)) \end{pmatrix}, \tag{2}$$

Our goal is to determine the HOSVD based canonical form (in the closed hypercube of the parameter space) by the TP model transformation:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{I}\sum_{j=1}^{J} w_{1,i}(x_1(t))w_{2,j}(x_2(t))\mathbf{S}_{i,j} = \left(S \otimes_{n=1}^{2} \mathbf{w}_n^T(x_n(t))\right)\mathbf{x}(t), \tag{3}$$

where the weighting functions and the LTI systems define the invariant characteristics of the given LPV model, see the definition of the HOSVD based canonical form.

2.1.1 Implementation in MATLAB with the TP Toolbox

First we define the parameter domain and grid size (I, J).

```
% intervals
Omega = [-0.5 0.5; -0.5 0.5];
% grid
M = [20, 20];
```

so both x_1 and x_2 are in the [-0.5, 0.5] interval and we will sample these intervals over a 20x20 grid.

Now we have to provide the LPV model of our system as a cell array of functions. In this case we can use MATLAB's anonymous function handles to describe the system easily. (Note that these functions can be the result of complex algorithms or calculated from measurement data.)

```
LPV = {...
      @(p)2*cos(4*p), @(p)p;
      @()3,           @(p)p(2)^3*sin(p(1));
};
```

We use dep 3 dimensional matrix to have to tell the toolbox what are the function arguments for each element of the LPV matrix.

```
% LPV{i,j} will be called with x(dep(i,j)==1) argumentum
dep = zeros([size(LPV) 2]);
dep(1,1,:) = [1 0];
dep(1,2,:) = [0 1];
dep(2,2,:) = [1 1];
```

We can simply execute the TP model transformation to get the LTI systems (S) and the discretized weighting functions (U). (the last parameter is the weighting function type, 'canonic' means canonical form here)

```
[S, U] = tptrans(LPV, dep, M, Omega, 'canonic');
```

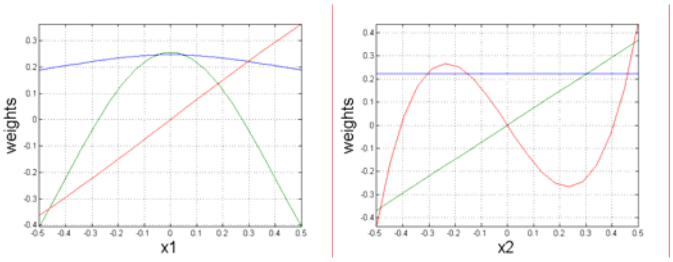
We can numerically check (over thousands of random points) the difference between the original model and the numerically reconstructed HOSVD based canonical form.

```
[maxerr, meanerr] = tperror(2000, S, U, LPV, dep, M, Omega);
disp('max error:'); disp(maxerr(2));
disp('mean error:'); disp(meanerr(2));
```

Conclusion: the maximum error and the root mean square error over 2000 random points is in the range 10e-15 which is caused by the numerical computation. Thus we can conclude that the two models are equivalent (numerically) in the present example.

Let us see the univariate othonormed functions for each parameter

```
plotw(U, M, Omega, {'x1', 'weights'; 'x2', 'weights'});
```



2.2 Convex TP Model Form

Let us consider the same system as in the previous example. Now our goal is to generate convex tp model form. (For instance for LMI based analysis and design.)

Since the TP model form is not unique, thus the toolbox contains various methods to adjust the shape of the resulting convex weight functions (we refer to papers listed under 'Different types of convex hulls' in the publication list).

The only difference in the code from the previous example is the wtype parameter in the tptrans function call. The possible values of the wtype parameter and its meaning can be found in the documentation also see the publications about the different types of convex hulls.

2.2.1 Implementation in MATLAB with the TP Toolbox

Specifying the model

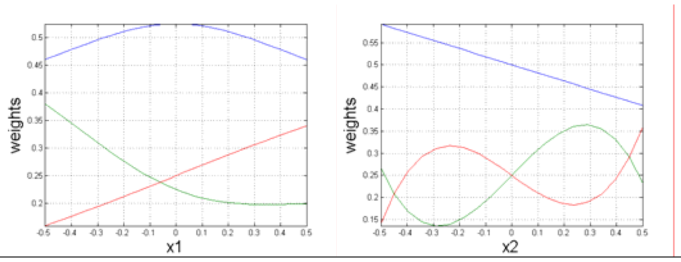
```
% intervals
Omega = [-0.5 0.5; -0.5 0.5];
% grid
M = [20, 20];

% LPV model and parameter dependencies
LPV = {...
       @(p)2*cos(4*p), @(p)p;
       @()3,           @(p)p(2)^3*sin(p(1));
};
dep = zeros([size(LPV) 2]);
dep(1,1,:) = [1 0];
dep(1,2,:) = [0 1];
dep(2,2,:) = [1 1];
```

TP model transformation with different weight function types.

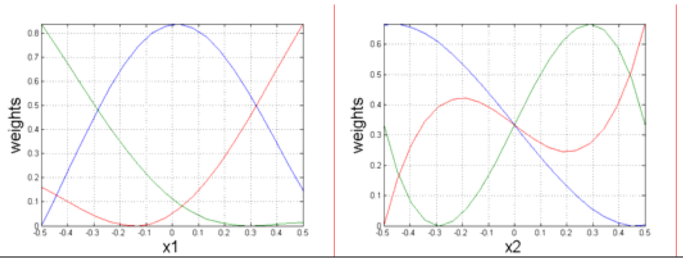
```
% wtype can be 'canonic', 'snnn', 'irno', 'cno'
[S, U] = tptrans(LPV, dep, M, Omega, wtype);

% numerical error check
[maxerr, meanerr] = tperror(2000, S, U, LPV, dep, M, Omega);
disp('max error:'); disp(maxerr(2));
disp('mean error:'); disp(meanerr(2));

% plotting the weighting functions
plotw(U, M, Omega, {'x1', 'weights'; 'x2', 'weights'});
```

2.2.2 Different Type of Weighting Functions

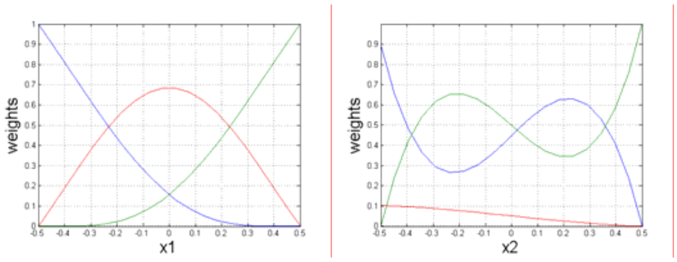
```
wtype = 'snnn';
```



```
wtype = 'irno';
```



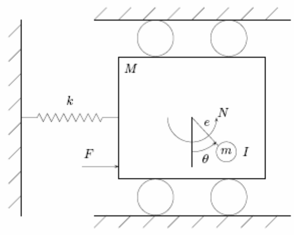
```
wtype = 'cno';
```



2.3 Stabilization of a TORA System (a Fourth Order Benchmark Problem)

For detailed derivation of this example with controller and observer design see [1], [6], [17].

Consider the control of a Translational Oscillations with an Eccentric Rotational Proof Mass Actuator (TORA) system, which was originally studied as a simplified model of a dual-spin spacecraft with mass imbalance to investigate the resonance capture phenomenon.



TORA system

The system model can be written in the following form:

$$\dot{\mathbf{x}}(t) = f(x_3(t), x_4(t))\mathbf{x}(t) + g(x_3(t))u(t)\,, \tag{4}$$

$$f(\mathbf{x}(t)) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \dfrac{-1}{1-\varepsilon^2\cos^2(x_3(t))} & 0 & 0 & \dfrac{\varepsilon x_4(t)\sin(x_3(t))}{1-\varepsilon^2\cos^2(x_3(t))} \\ 0 & 0 & 0 & 1 \\ \dfrac{\varepsilon\cos(x_3(t))}{1-\varepsilon^2\cos^2(x_3(t))} & 0 & 0 & \dfrac{-\varepsilon x_4(t)\sin(x_3(t))}{1-\varepsilon^2\cos^2(x_3(t))} \end{pmatrix}, \tag{5}$$

$$g(\mathbf{x}(t)) = \begin{pmatrix} 0 \\ \dfrac{-\varepsilon\cos(x_3(t))}{1-\varepsilon^2\cos^2(x_3(t))} \\ 0 \\ \dfrac{1}{1-\varepsilon^2\cos^2(x_3(t))} \end{pmatrix}, \tag{6}$$

We use TP model transformation to yield convex TP model (CNO type) representation of the model, then we apply LMI design on the resulting TP model to derive a controller.

2.3.1 Controller Design

Model specification:

```
% Model constants
epsilon = 0.2;

% LPV model
LPV = {...
    @()0     @()1     @()0     @()0     @()0;
    @(x)-1/(1-epsilon^2*cos(x)^2)  @()0     @()0
      @(x)(epsilon*x(2)*sin(x(1)))/(1-epsilon^2*cos(x(1))^2)
      @(x)(-epsilon*cos(x))/(1-epsilon^2*cos(x)^2);
    @()0     @()0     @()0     @()1     @()0;
      @(x)(epsilon*cos(x))/(1-epsilon^2*cos(x)^2) @()0 @()0
      @(x)(-x(2)*epsilon^2*cos(x(1))*sin(x(1)))/(1-
epsilon^2*cos(x(1))^2)
      @(x)1/(1-epsilon^2*cos(x)^2);
    @()0     @()0     @()1     @()0     @()0;
    @()0     @()0     @()0     @()1     @()0;
};

% parameter dependencies
dep = zeros([size(LPV) 2]);
dep(2,1,:) = [1 0];
dep(2,4,:) = [1 1];
dep(2,5,:) = [1 0];
dep(4,1,:) = [1 0];
dep(4,4,:) = [1 1];
dep(4,5,:) = [1 0];
```

TP model transformation:

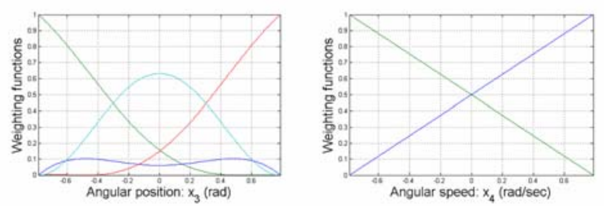
```
filename='tora_data';
A_size = 4;

% intervals
Omega = [-45/180*pi 45/180*pi; -45/180*pi 45/180*pi];
% grid size
Mgrid = [137, 137];
% weight function type: 'cno', 'snnn', 'canonic'
wtype = {'cno' 'cno'};

% TP model transformation
reply = input('TP model transformation? Y/N [Y]: ', 's');
if isempty(reply) || lower(reply)=='y'
       disp('Step 1: Sampling the LPV model');
       tic
       [S, U] = tptrans(LPV, dep, Mgrid, Omega, wtype, 1);
       toc
       disp('L_2 norm and mean square error');
       [maxerr, meanerr] = tperror(100, S, U, LPV, dep, Mgrid,
Omega);
       disp(maxerr);
       disp(meanerr);

       save(filename, 'S', 'U', 'Omega', 'Mgrid');
else
       load(filename, 'S', 'U');
```

```
end

% Plotting
reply = input('Draw weight functions? Y/N [Y]: ', 's');
if isempty(reply) || lower(reply)=='y'
    labels{1,1}='Angular position: x_3 (rad)';
    labels{1,2}='Weighting functions';
    labels{2,1}='Angular speed: x_4 (rad/sec)';
    labels{2,2}='Weighting functions';
      plotw(U, Mgrid, Omega, labels);
end

reply = input('Print TP model? Y/N [Y]: ', 's');
if isempty(reply) || lower(reply)=='y'
    printtp(S);
end
```

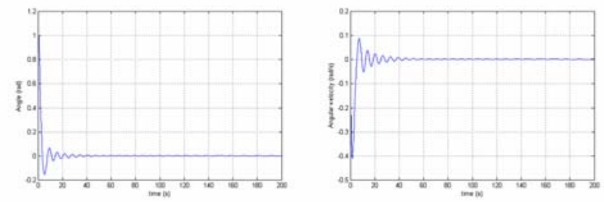
Controller design (constrains (input and state) and asymptotic stabilization, for details about including further constrains and decay rate control design in this example we refer to publications):

```
reply = input('PDC controller design? Y/N [Y]: ', 's');
if isempty(reply) || lower(reply)=='y'
    mu = 17450;
    phi = 1.1;

    K = lmi_asymptotic_const(S, A_size, mu, phi);
    save(filename, '-APPEND', 'K');
end

reply = input('Print TP controller? Y/N [Y]: ', 's');
if isempty(reply) || lower(reply)=='y'
    printtp(K);
end
```

Resulting CNO type weighting functions of the convex TP model: (For further different types of weighting functions of this example we refer to publications.)



Simulation results:



Further Information

For the latest version of the toolbox and for further information see the homepage of the toolbox [1].

Acknowledgement

This research was supported by the Hungarian National Found OTKA No. T 049838 and T 042896. Dr. Petres was supported by Zoltán Magyart Postdoctroal Scholarship, Dr. Baranyi was supported by János Bolyai Postdoctoral Scholarship.

References

TP Toolbox

[1] Sz. Nagy, Z. Petres, P Baranyi. TP Toolbox homepage url: http://tptool.sztaki.hu

Background of the TP Model Transformation

[2] P. Baranyi: "TP Model Transformation as a Way to LMI-based Controller Design", IEEE Transaction on Industrial Electronics, Vol. 51, No. 2, April, 2004, pp. 387-400 (ISSN 0278-0046)

[3] P. Baranyi, D. Tikk, Y. Yam, R. J. Patton: "From Differential Equations to PDC Controller Design via Numerical Transformation", Computers in Industry, Elsevier Science, Vol. 51, 2003, pp. 281-297

[4] L. Szeidl, P. Baranyi, Z. Petres, P. Várlaki: „Numerical Reconstruction of the HOSVD-based Canonical Form of Polytopic Dynamic Models", in proc. of 3rd International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII 2007), Agadir, Morocco, March 28-30, 2007, pp. 111-116 (1-4244-1158-0/07)

[5] J. Bokor, P. Baranyi, P. Michelberger, P. Várlaki: „TP Model Transformation in Non-Linear System Control", 3rd IEEE International Conference on Computational Cybernetics (ICCC 2005), Mauritius, April 13-16, 2005, (Plenary Lecture), pp. 111-119, ISBN 9637154 37 X

[6] Polytopic Decomposition of Linear Parameter-Varying Models by Tensor-Product Model Transformation. Zoltán Petres. PhD thesis, *Doctorate School of Computer Science and Information Technology, Budapest University of Technology and Economics*, November 2006 (url: http://alpha.tmit.bme.hu/~petres/phd.pdf)

Different Types of Convex Hulls

[7] P. Baranyi, Z. Petres, P. L. Várkonyi, P. Korondi, Y. Yam: „Determination of Different Polytopic Models of the Prototypical Aeroelastic Wing Section by TP Model Transformation", Journal of Advanced Computational Intelligence, Vol. 10, No. 4, 2006, pp. 486-493 (ISSN 1343-0130)

[8] P. Baranyi: „Output Feedback Control of 2-D Aeroelastic System", Journal of Guidance, Control, and Dynamics, Vol. 29, No. 3, May-June 2006, pp. 762-767 (ISSN 0731-5090)

[9] Z. Petres, P. Baranyi, P. Gáspár, P. Várlaki, P. Michelberger: „Different Polytopic Decomposition of the Model of Heavy Vehicles by TP Model Transformation" in proc. of 11th International Conference on Intelligent Engineering Systems (INES 2007), Budapest, Hungary, June 29-July 1, 2007, pp. 265-270 (ISBN 1-4244-1148-3)

[10] Sz. Nagy, Z. Petres, P. Baranyi, L. Szeidl, P. Michelberger: „Politopic Decomposition of the Linear Parameter-varying Model of the Parallel-Type Double inverted Pendulum" in proc. of 11th International Conference on Intelligent Engineering Systems (INES 2007), Budapest, Hungary, June 29-July 1, 2007, pp. 271-275 (ISBN 1-4244-1148-3)

[11] Z. Petres, P. L. Várkonyi, P. Baranyi, P Korondi: „Different Affine Decomposition of the TORA System by TP model transformation", IEEE 9th International Conference on Intelligent Engineering Systems 2005, (INES 2005), Athens, Greece, September 16-19, 2005, ISBN: 0-7803-9474-7

[12] P. Baranyi, P. L. Várkonyi, P. Korondi, Y. Yam: „Different Affine Decomposition of the Model of the Prototypical Aeroelastic Wing Section by TP Model Transformation", IEEE 9th International Conference on Intelligent Engineering Systems 2005, (INES 2005), Athens, Greece, September 16-19, 2005, ISBN: 0-7803-9474-7

Complexity-relaxed Implementation

[13] P. Baranyi, Z. Petres, P. Korondi, Y. Yam, H. Hashimoto: „Complexity Relaxation of the Tensor Product Model Transformation for Higher Dimensional Problems", Asian Journal of Control, Vol. 9, No. 2, June 2007 (in press)

[14] Sz. Nagy, Z. Petres, P. Baranyi, H. Hashimoto: „Computational-relaxed TP Model Transformation by Restriction of the Computation to Subspaces of the Dynamic Model", in proc. of 3rd International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII 2007), Agadir, Morocco, March 28-30, 2007, pp. 99-104 (1-4244-1158-0/07)

[15] Z. Petres, P. Baranyi, H. Hashimoto: „Decrease of the Computational Load of TP Model Transformation" Proc. of 3rd IEEE International Conference on Mechatronics (ICM 2006), Budapest, Hungary, July 3-5, 2006, pp. 655-659

Control Design Examples

[16] Z. Petres, P. Baranyi, H. Hashimoto: „Trajectory Tracking by TP Model Transformation: Case Study of a Benchmark Problem", IEEE Transaction in Industrial Electronics, Vol. 54, No. 3, June 2007, pp. 1654-1663

[17] P. Baranyi, Z. Petres, P. Várlaki, P. Michelberger: "Observer and Control Law Design to the TORA System via TPDC Framework", WSEAS Transactions on Systems, Issue 1, Volume 5, January 2006, pp. 156-163 (ISSN 1109-2777)

[18] P. Baranyi: „Tensor-Product Model-based Control of Two-Dimensional Aeroelastic System", Journal of Guidance, Control, and Dynamics, Vol. 29, No. 2, March-April, 2006, pp. 391-400

[19] P. Baranyi: „Output Feedback Control of 2-D Aeroelastic System", Journal of Guidance, Control, and Dynamics, Vol. 29, No. 3, May-June 2006, pp. 762-767 (ISSN 0731-5090)