MODEL-ORDER REDUCTION OF SINGULARLY PERTURBED SYSTEMS BASED ON ARTIFICIAL NEURAL ESTIMATION AND LMI-BASED TRANSFORMATION

Othman M-K. Alsmadi, Za'er S. Abo-Hammour
Department of Electrical Engineering, University of Jordan, Amman, Jordan
Department of Mechatronics Engineering, University of Jordan, Amman, Jordan

Mohammad S. Saraireh
Department of Computer Engineering, Mutah University, Karak, Jordan

Keywords: Model order reduction, System transformation, Artificial neural networks, Linear matrix inequality (LMI), Singular perturbation.

Abstract: A new method for model order reduction with eigenvalue preservation is presented in this paper. The new technique is formulated based on the system state matrix transformation which preserves the system eigenvalues and is accomplished using an artificial neural network training. A linear matrix inequality (LMI) numerical algorithm technique is used to obtain the complete system transformation. Model order reduction is then obtained utilizing the singular perturbation method. Simulation results show that the LMI-based transformed reduced model order is superior to other reduction methods.

1 INTRODUCTION

The objective of any control system is to obtain a desired response. In order to achieve this objective, a dynamical model is usually developed based on a set of differential equations (Franklin, 1994). The obtained mathematical model may have a certain parameter, called perturbation, that has a little effect on the system performance (Kokotovic, 1986) (Zhou, 2009). Neglecting this parameter results in simplifying the order of the designed controller based on reducing the system model order. A reduced model order can be obtained by neglecting the fast dynamics (i.e., non-dominant eigenvalues) of the system and focusing on the slow dynamics (i.e., dominant eigenvalues). This method is referred to as singular perturbation. Simplification and reduction of a system model leads to controller cost minimization (Garsia, 1998). An example is the ICs, where increasing package density forces developers to include side effects. Knowing that these devices are often modeled by large RLC circuits, this would be too demanding computationally and practically due to the detailed modeling of the original system (Benner, 2007). In control system, due to the fact that feedback controllers do not usually consider all the dynamics of the system, model reduction becomes a very important issue (Bui-Thanh, 2005).

For a reduced model order that will best mimic the performance of its original system, system transformation is performed. In the process of system transformation, some system parameters are required to be identified. This objective maybe achieved by the use of artificial neural networks (ANN) (Alsmadi, 2007), which are considered as the new generation of information processing networks (Hinton, 2006). Artificial neural systems maybe defined as physical cellular systems which have the capability of acquiring, storing and utilizing experiential knowledge. They can be represented as mathematical or computational models based on biological neural networks. An artificial neural network consists of an interconnected group of artificial neurons and processes information. They perform summing operations and nonlinear function computations. Neurons are usually organized in layers and forward connections where computations are performed in a parallel fashion at all nodes and

M-K. Alsmadi O., S. Abo-Hammour Z. and S. Saraireh M. (2009).
MODEL-ORDER REDUCTION OF SINGULARLY PERTURBED SYSTEMS BASED ON ARTIFICIAL NEURAL ESTIMATION AND LMI-BASED TRANSFORMATION .
In *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Intelligent Control Systems and Optimization*, pages 173-180
DOI: 10.5220/0002187801730180

connections. Each connection is expressed by a numerical value called a weight. The learning process of a neuron corresponds to a way of changing its weights. An artificial neural network can be used to model complex relationships between inputs and outputs of different systems (Haykin, 1994) (Zurada, 1992) (Williams, 1989).

In obtaining the overall transformed model, which leads to control design advantages, part of the transformation requires some optimized solution. This is accomplished using what is called the linear matrix inequality (LMI), which serves application problems, in convex optimization (Boyd, 1994). The LMI is based on the Lyapunov theory of showing that the differential equation $\dot{x}(t) = Ax(t)$ is stable if and only if there exists a positive definite matrix [**P**] such that $A^T P + PA < 0$. The requirement { $P > 0$, $A^T P + PA < 0$ } is what is known as Lyapunov inequality on [**P**]. The LMIs that arise in systems and control theory can be formulated as convex optimization problems that are amenable to computer solution and then solved using different algorithms (Boyd, 1994).

This paper is organized as follows: Section 2 presents background on model order reduction and artificial neural networks. A detailed illustration of the ANN transformed system state matrix estimation and the LMI-based complete system transformation is presented in Section 3. Section 4 presents a practical implementation of the ANN transformation training, LMI-based transformation, and singular perturbation reduction along with simulation comparative results. Conclusions are presented in Section 5.

2 PRELIMINARY

Many of linear time-invariant (LTI) systems have fast and slow dynamics, which are referred to as singularly perturbed systems (Kokotovic, Khalil, and O'Reilly, 1986). Neglecting the fast dynamics gives the advantage of designing simpler lower-dimensionality reduced order controllers. To show the formulation of a reduced model order, consider the following system:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

$$y(t) = Cx(t) + Du(t) \tag{2}$$

As a singularly perturbed system (with slow and fast dynamics), Equations (1) - (2) may be formatted as:

$$\dot{x}(t) = A_{11}x(t) + A_{12}\xi(t) + B_1 u(t), \quad x(0) = x_0 \tag{3}$$

$$\varepsilon\dot{\xi}(t) = A_{21}x(t) + A_{22}\xi(t) + B_2 u(t), \quad \xi(0) = \xi_0 \tag{4}$$

$$\mathrm{y}(t) = C_1 x(t) + C_2\xi(t) + Du(t) \tag{5}$$

where $x \in \Re^{m_1}$ and $\xi \in \Re^{m_2}$ are the slow and fast state variables respectively, $u \in \Re^{n_1}$ and $y \in \Re^{n_2}$ are the input and output vectors respectively, $\{[\mathbf{A_{ii}}], [\mathbf{B_i}], [\mathbf{C_i}], [\mathbf{D}]\}$ are constant matrices of appropriate dimensions with $i \in \{1, 2\}$, and ε is a small positive constant. The singularly perturbed system in Equations (3)-(5) is simplified by setting $\varepsilon = 0$. That is, the fast dynamics of the system are being neglected and the state variables ξ are assumed to have reached their quasi-steady state. Hence, setting $\varepsilon = 0$ in Equation (4), with the assumption that $[\mathbf{A_{22}}]$ is nonsingular, produces:

$$\xi(t) = -A_{22}^{-1}A_{21}x_r(t) - A_{22}^{-1}B_1 u(t) \tag{6}$$

where the index r denotes the remained or reduced model. Substituting Equation (6) in Equations (3)-(5) yields the following reduced model order:

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t) \tag{7}$$

$$y(t) = C_r x_r(t) + D_r u(t) \tag{8}$$

Where the new matrices: $A_r = A_{11} - A_{12}A_{22}^{-1}A_{21}$, $B_r = B_1 - A_{12}A_{22}^{-1}B_2$, $C_r = C_1 - C_2A_{22}^{-1}A_{21}$, and $D_r = D - C_2A_{22}^{-1}B_2$.

The system in Equations (1) and (2) maybe estimated by an ANN. In this paper, a recurrent neural network based on an approximation of the method of steepest descent is used for the estimation of the system state matrix. The network tries to match the output of certain neurons with the desired values of the system output at specific instant of time (Haykin, 1994) (Williams, 1989). Hence, consider the discrete system given by:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{9}$$

$$y(k) = x(k) \tag{10}$$

which, for a system with two eigenvalue categories (slow and fast), can be represented as:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} u(k) \tag{11}$$

$$y(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{12}$$

where k is the time index. Using the recurrent neural network, the system in Equations (11) and (12) for a 2nd model order can be estimated as illustrated in Figure 1.

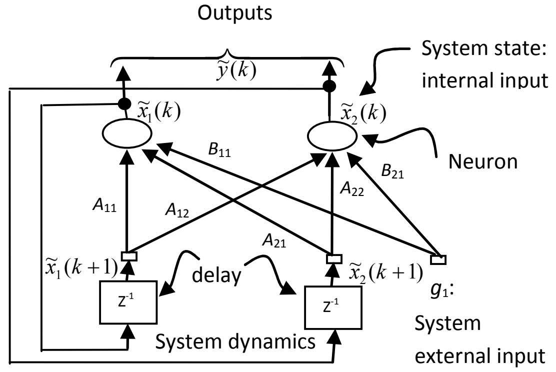

Figure 1: A second order recurrent neural network architecture.

As a general case, consider a network consisting of a total of N neurons with M external input connections, as shown in Figure 1 for a 2nd model order. Let the variable $\mathbf{g}(k)$ denotes the (M x 1) external input vector applied to the network at discrete time k and the variable $\mathbf{y}(k + 1)$ denotes the corresponding (N x 1) vector of individual neuron outputs produced one step later at time (k + 1). The input vector $\mathbf{g}(k)$ and one-step delayed output vector $\mathbf{y}(k)$ are concatenated to form the ((M + N) x 1) vector $\mathbf{u}(k)$, whose i^{th} element is denoted by $u_i(k)$. If Λ denotes the set of indices i for which $g_i(k)$ is an external input, and β denotes the set of indices i for which $u_i(k)$ is the output of a neuron (which is $y_i(k)$), the following is true:

$$u_i(k) = \begin{cases} g_i(k), & \text{if } i \in \Lambda \\ y_i(k), & \text{if } i \in \beta \end{cases} \tag{13}$$

The (N x (M + N)) recurrent weight matrix of the network is represented by the variable [$\mathbf{W}$]. The net internal activity of neuron j at time k is given by:

$$v_j(k) = \sum_{i \in \Lambda \cup \beta} w_{ji}(k) u_i(k) \tag{14}$$

At (k + 1), the output of the neuron j is computed by passing $v_j(k)$ through the nonlinearity $\varphi(.)$:

$$y_j(k+1) = \varphi(v_j(k)) \tag{15}$$

The derivation of the recurrent algorithm maybe obtained by using $d_j(k)$ to denote the desired response of neuron j at time k, and $\varsigma(k)$ to denote the set of neurons that are chosen to provide externally reachable outputs. A time-varying (N x 1) error vector $\boldsymbol{e}(k)$ is defined whose j^{th} element is given by the following relationship:

$$e_j(k) = \begin{cases} d_j(k) - y_j(k), & \text{if } j \in \varsigma(k) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The objective is to minimize the cost function E_{total} which is obtained by:

$$E_{\text{total}} = \sum_k E(k) = \sum_k [\frac{1}{2} \sum_{j \in \varsigma} e_j^2(k)] \tag{17}$$

This cost function will be minimized by estimating the instantaneous gradient, which is the error at each instant of time k with respect to the weight matrix [$\mathbf{W}$] and then updating [$\mathbf{W}$] in the negative direction of this gradient (Haykin, 1994). As a result:

$$W = \left\lfloor [\tilde{A}_d] \ [\tilde{B}_d] \right\rfloor \tag{18}$$

where $\tilde{\mathrm{A}}_{\mathrm{d}}$ and $\tilde{\mathrm{B}}_{\mathrm{d}}$ are the estimates of Equation (9).

3 SYSTEM TRANSFORMATION AND ORDER REDUCTION

In the new reduction technique, the system is transformed before the model order is reduced. System transformation is achieved by transforming the system state matrix [$\mathbf{A}$] based on the ANN estimation and then transforming the [$\mathbf{B}$], [$\mathbf{C}$], and [$\mathbf{D}$] matrices of Equations (1) and (2) using the LMI-based transformation.

3.1 ANN System State Matrix Transformation

In this paper, one objective is to search for a transformation that decouples different categories of system eigenvalues. In the transformed system presented in this paper, the dominant eigenvalue category is selected as a subset of the original system eigenvalues. This is accomplished by transforming the system state matrix [$\mathbf{A}$] in Equation (1) into [$\hat{\mathrm{A}}$] (for all real eigenvalues) as follows:

$$\hat{A} = \begin{bmatrix} \lambda_1 & a_{12} & \cdots & a_{1n} \\ 0 & \lambda_2 & \cdots & a_{2n} \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \tag{19}$$

This is an upper triangular matrix that has the original system eigenvalues preserved in the diagonal, seen as λ_i, and has the elements to be identified, seen as (a_{ij}). It is set as such for the purpose of eliminating the fast dynamics and sustaining the slow dynamics through model order reduction. In order to evaluate the (a_{ij}) elements, first, the system of Equations (1) and (2) is discretized as shown in Equations (9) and (10), second, the [A_d] in Equation (9) is transformed into [$\tilde{A}_d$] (similar to the form seen in Equation (19)), third, the recurrent neural network estimates the required elements of [$\tilde{A}_d$], fourth, (a_{ij}) are then evaluated once the continuous form is obtained from the estimated discrete system.

In this estimation, the interest is to estimate or obtain the [$\tilde{A}_d$] only without the estimation of the [$\tilde{B}_d$] matrix, where this [$\tilde{B}_d$] matrix is automatically obtained in the recurrent network as seen in Figure 1 and Equation (18). In order to achieve this objective, the zero input ($u(k) = 0$) response is obtained where the input/output data is basically generated based on the initial state conditions only. Hence, the discrete system of Equations (9) and (10), with initial state conditions $x(0) = x_0$, becomes:

$$x(k+1) = A_d x(k), \qquad x(0) = x_0 \tag{20}$$

$$y(k) = x(k) \tag{21}$$

Now based on Equations (20) and (21), where the initial states are the system input and the obtained states are the system output, a set of input/output data is obtained and the neural network estimation is applied (Haykin, 1994). In steps:

Step 1. Initialize the weights [**W**] by a set of uniformly distributed random numbers. Starting at the instant $k = 0$, use Equations (14) and (15) to compute the output values of the N neurons (where $N = \beta$).

Step 2. For every time step k and all $j \in \beta$, $m \in \beta$ and $\ell \in \beta \cup A$, compute the system dynamics which are governed by the triply indexed set of variables:

$$\pi^j_{m\ell}(k+1) = \dot{\varphi}(v_j(k))\left[\sum_{i\in\beta} w_{ji}(k)\pi^i_{m\ell}(k) + \delta_{mj}u_\ell(k)\right] \tag{22}$$

with initial conditions $\pi^j_{m\ell}(0) = 0$ and δ_{mj} given by $\left(\partial w_{ji}(k)/\partial w_{m\ell}(k)\right)$ is equal to "1" only when $j = m$ and $i = \ell$; otherwise it is "0". Notice that for the special case of a sigmoidal nonlinearity in the form of a logistic function, the derivative $\dot{\varphi}(\cdot)$ is given by $\dot{\varphi}(v_j(k)) = y_j(k+1)[1 - y_j(k+1)]$.

Step 3. Compute the weight changes corresponding to the error signal and system dynamics:

$$\Delta w_{m\ell}(k) = \eta \sum_{j\in\varsigma} e_j(k)\pi^j_{m\ell}(k) \tag{23}$$

Step 4. Update the weights in accordance with:

$$w_{m\ell}(k+1) = w_{m\ell}(k) + \Delta w_{m\ell}(k) \tag{24}$$

Step 5. Repeat the above 4 steps for final desired estimation.

Training the network as illustrated, produces the discrete transformed system state matrix [$\tilde{A}_d$]. This new discrete matrix is then converted to the continuous form to give the transformed system state matrix [$\hat{A}$] as actually seen in Equation (19).

3.2 LMI-based Complete System Transformation

The transformation in Equation (19) is motivated by the matrix reducibility concept illustrated as follows (Boyd, 1994) (Horn, 1985):

*Definition***.** A matrix $A \in M_n$ is called reducible if either:

(a) $n = 1$ and $A = 0$; or

(b) $n \ge 2$, there is a permutation matrix $P \in M_n$, and some integer r with $1 \le r \le n-1$ such that:

$$P^{-1}AP = \begin{bmatrix} X & Y \\ \mathbf{0} & Z \end{bmatrix} \tag{25}$$

where $X \in M_{r,r}$, $Z \in M_{n-r,n-r}$, $Y \in M_{r,n-r}$, and $\mathbf{0} \in M_{n-r,r}$ is a zero matrix.

The attractive features of the permutation matrix [**P**] such as being orthogonal and invertible have made this transformation easy to carry out. Based on the LMI technique, the optimization problem is casted as follows:

$$\min_{P} \left\| P - P_o \right\| \text{ subject to } \left\| P^{-1}AP - \hat{A} \right\| < \varepsilon \tag{26}$$

which maybe written in an LMI equivalent form as:

$$\min_{S} trace(S) \text{ subject to}$$

$$\begin{bmatrix} S & P - P_o \\ (P - P_o)^T & I \end{bmatrix} > 0$$
$$\begin{bmatrix} \varepsilon_1^2 I & P^{-1}AP - \hat{A} \\ (P^{-1}AP - \hat{A})^T & I \end{bmatrix} > 0 \tag{27}$$

where S is a symmetric slack matrix (Boyd, 1994).

The Linear Matrix Inequalities (LMI) are applied to the [**A**] and [$\hat{\mathrm{A}}$] matrices and the transformation matrix [**P**] is then obtained, which is necessary for obtaining the complete system transformation {[$\hat{\mathbf{B}}$], [$\hat{\mathbf{C}}$], [$\hat{\mathbf{D}}$]}. Complete system transformation can be achieved as follows: assuming that $\hat{x} = P^{-1}x$, the system of Equations (1) and (2) can be re-written as:

$$P\,\dot{\hat{x}}(t) = AP\,\hat{x}(t) + Bu(t) \tag{28}$$

$$\hat{y}(t) = CP\,\hat{x}(t) + Du(t) \tag{29}$$

Pre-multiplying Equation (28) by [$\mathbf{P}^{-1}$] yields:

$$P^{-1}P\,\dot{\hat{x}}(t) = P^{-1}AP\,\hat{x}(t) + P^{-1}Bu(t)$$
$$\therefore\ \dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t) \tag{30}$$

and $\hat{y}(t) = CP\,\hat{x}(t) + Du(t)$

$$\therefore\ \hat{y}(t) = \hat{C}\hat{x}(t) + \hat{D}u(t) \tag{31}$$

where the transformed system matrices are:
$\hat{A} = P^{-1}AP$, $\hat{B} = P^{-1}B$, $\hat{C} = CP$, and $\hat{D} = D$.

3.3 Model Order Reduction

Model order reduction will now be applied to the system of Equations (30) and (31) which has the following format:

$$\begin{bmatrix} \dot{\hat{x}}_r(t) \\ \dot{\hat{x}}_o(t) \end{bmatrix} = \begin{bmatrix} A_{rn} & A_c \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \hat{x}_r(t) \\ \hat{x}_o(t) \end{bmatrix} + \begin{bmatrix} B_r \\ B_o \end{bmatrix} u(t) \tag{32}$$

$$\hat{y}(t) = \begin{bmatrix} C_{rn} & C_o \end{bmatrix} \begin{bmatrix} \hat{x}_r(t) \\ \hat{x}_o(t) \end{bmatrix} + \hat{D}u(t) \tag{33}$$

Notice that in the new formulation, the dominant eigenvalues (slow dynamics) which are presented in A_{rn} are now decoupled from the non-dominant eigenvalues (fast dynamics) which are presented in A_o. Hence, as illustrated in Equations (3) and (4) for order reduction, Equation (32) is written as:

$$\dot{\hat{x}}_r(t) = A_{rn}\hat{x}_r(t) + A_c\hat{x}_o(t) + B_r u(t) \tag{34}$$

$$\dot{\hat{x}}_o(t) = A_o\hat{x}_o(t) + B_o u(t) \tag{35}$$

By neglecting the system fast dynamics (setting $\dot{\hat{x}}_o(t) = 0$ by setting $\varepsilon = 0$)), the coupling term $A_c\hat{x}_o(t)$ is evaluated by solving for $\hat{x}_o(t)$ in Equation (35). That is, $\hat{x}_o(t) = -A_o^{-1}B_o u(t)$ and the reduced model order becomes:

$$\dot{\hat{x}}_r(t) = A_{rn}\hat{x}_r(t) + [-A_c A_o^{-1}B_o + B_r]u(t) \tag{36}$$

$$\hat{y}(t) = C_r\hat{x}_r(t) + [-C_o A_o^{-1}B_o + D]u(t) \tag{37}$$

Hence, the overall transformed reduced model order is given by:

$$\dot{\hat{x}}_r(t) = A_{or}\hat{x}_r(t) + B_{or}u(t) \tag{38}$$

$$\hat{y}(t) = C_{or}\hat{x}_r(t) + D_{or}u(t) \tag{39}$$

where the details of the {[$\mathbf{A_{or}}$], [$\mathbf{B_{or}}$], [$\mathbf{C_{or}}$], [$\mathbf{D_{or}}$]} overall reduced matrices are shown in Equations (36) and (37).

4 SIMULATIONS AND RESULTS

The proposed method of reduced order system modeling based on neural network estimation, LMI-based transformation, and model order reduction is investigated the following case studies.

Case Study. Consider the system of a high-performance tape transport shown in Figure 2 (Franklin, 1994). The system is designed with a small capstan to pull the tape past the read/write heads with the take-up reels turned by DC motors. In the static equilibrium, the tape tension equals the vacuum force $T_o = F$ and the torque from the motor equals the torque on the capstan $K_t i_o = r_1 T_o$. Please notice that all the variables are defined in (Franklin, 1994).

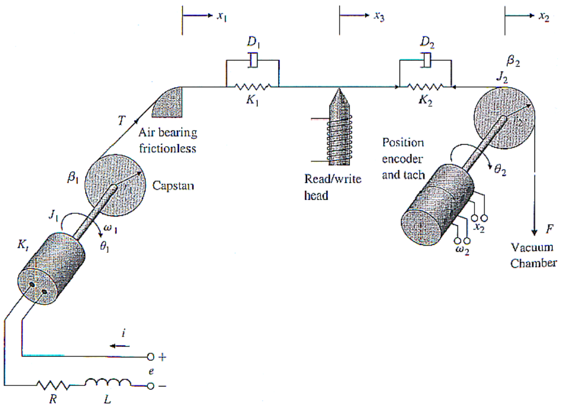

Figure 2: Tape-drive system schematic control model.

The variables are defined as deviations from the equilibrium. The system equations of motion are given as follows:

$$J_1 = \frac{d\omega_1}{dt} + \beta_1\omega_1 - r_1 T + K_t i\,,\ \dot{x}_1 = r_1\omega_1$$

$$L\frac{di}{dt} Ri + K_e\omega_1 = e\,,\ \ \dot{x}_2 = r_2\omega_2$$

$$J_2\frac{d\omega_2}{dt} + \beta_2\omega_2 + r_2 T = 0$$

$$T = K_1(x_3 - x_1) + D_1(\dot{x}_3 - \dot{x}_1)$$

$$T = K_2(x_2 - x_3) + D_2(\dot{x}_2 - \dot{x}_3)$$

$$x_1 = r_1\theta_1\,,\ \ x_2 = r_2\theta_2\,,\ \ x_3 = \frac{x_1 - x_2}{2}\,,$$

The state space model is derived from the system equations, where there are (i) one input, which is the applied voltage, (ii) three outputs, which are: (1) tape position at the head, (2) tape tension, and (3) tape position at the wheel, (iii) five states: (1) tape position at the air bearing, (2) drive wheel speed, (3) tape position at the wheel, (4) tachometer output speed, and (5) capstan motor speed. For dynamical testing of the new reduction technique validity, different cases of this practical system were investigated.

As a first example, a system with all real eigenvalues is considered:

$$\dot{x}(t) = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ -0.1 & -1.35 & 0.1 & 04.1 & 0.75 \\ 0 & 0 & 0 & 5 & 0 \\ 0.35 & 0.4 & -1.4 & -5.4 & 0 \\ 0 & -0.03 & 0 & 0 & -10 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ -0.2 & -0.2 & 0.2 & 0.2 & 0 \end{bmatrix} x(t)$$

with the eigenvalues {-9.9973, -3.9702, -1.8992, -0.677, -0.2055}. Since there are two categories of eigenvalues, slow {-1.8992, -0.6778, -0.2055} and fast {-9.9973, -3.9702}, model order reduction may be applied.

Discretizing this system with a sampling period T_s = 0.1s, simulating the discrete system for 200 input/output data points, and training it with learning rate of $\eta = 1 \times 10^{-4}$ and initial weights for $[\tilde{\mathbf{A}}_\mathbf{d}]$:

$$w = \begin{bmatrix} 0.0048 & 0.0039 & 0.0009 & 0.0089 & 0.0168 \\ 0.0072 & 0.0024 & 0.0048 & 0.0017 & 0.0040 \\ 0.0176 & 0.0176 & 0.0136 & 0.0175 & 0.0034 \\ 0.0055 & 0.0039 & 0.0078 & 0.0076 & 0.0051 \\ 0.0102 & 0.0024 & 0.0091 & 0.0049 & 0.0121 \end{bmatrix}$$

produces the transformed system matrix:

$$\hat{A} = \begin{bmatrix} -0.2051 & -0.0367 & -0.0068 & 0.0762 & 0.2074 \\ 0 & -0.6782 & 0.0513 & 0.0156 & 0.0554 \\ 0 & 0 & -1.8986 & 0.2282 & 0.0537 \\ 0 & 0 & 0 & -3.9708 & 0.0920 \\ 0 & 0 & 0 & 0 & -9.9963 \end{bmatrix}$$

with estimated eigenvalues -9.9963, -3.9708, -1.898, -0.6782, -0.2051. This was achieved by decoupling the fast eigenvalue category from the slow one, which simply was done by first placing the slow eigenvalue category in λ_i of Equation (19) and then the fast category. As observed in $\hat{A}$ above, the eigenvalues are almost identical with the original system with little difference due to discretization. Using the LMI-based system transformation, the complete transformed system is obtained. Considering the {-9.9963, -3.9708} as the fast category eigenvalue, the 3rd order reduced model is determined. Simulation results based on (i) model order reduction without system transformation, (ii) model order reduction with ANN transformation (estimation of $[\tilde{\mathbf{A}}_\mathbf{d}]$ and $[\tilde{\mathbf{B}}_\mathbf{d}]$ matrices only as presented in Equations (11) and (12)), (iii) model order reduction with LMI-based complete system transformation, and (iv) the original 5th order system are all shown in Figure 3.

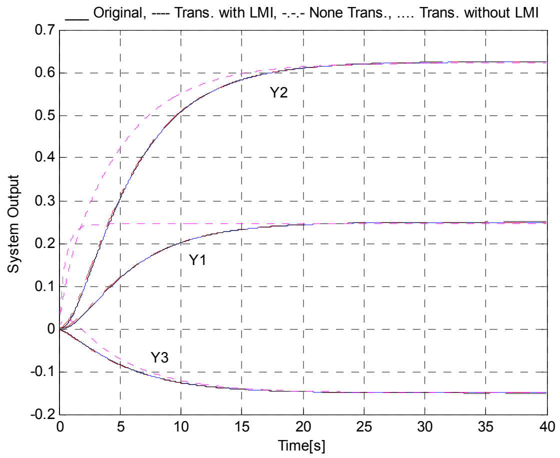

Figure 3: Reduced 3rd model orders (Pink….. transformed with ANN estimation only, Red-.-.-.- non-transformed, Black---- transformed with LMI) output responses to a step input along with the non reduced (Blue____ original) system output response. The LMI-transformed curve fits almost exactly on the original response.

For more rigorous testing of the new reduction technique, the 5th model order is reduced to a 2nd order assuming that the -1.8986 belongs to the fast eigenvalue category. Hence, the 2nd order reduced model with its eigenvalues preserved as desired is obtained:

$$\dot{\hat{x}}_r(t) = \begin{bmatrix} -0.2051 & -0.0367 \\ 0 & -0.6782 \end{bmatrix} \hat{x}_r(t) + \begin{bmatrix} -1.9672 \\ -2.1764 \end{bmatrix} u(t)$$

$$\hat{y}(t) = \begin{bmatrix} -0.0436 & 0.0451 \\ -0.1055 & 0.1029 \\ 0.0217 & -0.0140 \end{bmatrix} \hat{x}_r(t) + \begin{bmatrix} 0.0018 \\ 0.0043 \\ 0.0005 \end{bmatrix} u(t)$$

Simulating this reduced 2nd model order as performed for the 3rd model order, provided the results shown in Figure 4 where the new reduction technique results in responses are identical to the original system's.

As a second example, the system considered here consists of two complex eigenvalues and three real,

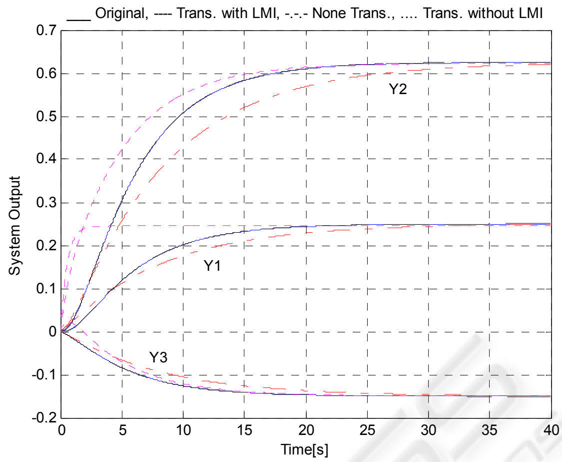

Figure 4: Plots of Pink…. 3rd order transformed with ANN estimation only and reduced 2nd model orders (Red-.-.-.- non-transformed, Black---- transformed with LMI) output responses to a step input along with the non reduced (Blue____ original) system output response. The LMI-transformed curve fits almost exactly on the original response.

where two of the real eigenvalues produce fast dynamics. The system is given by:

$$\dot{x}(t) = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ -1.1 & -1.35 & 1.1 & 3.1 & 0.75 \\ 0 & 0 & 0 & 5 & 0 \\ 1.35 & 1.4 & -2.4 & -11.4 & 0 \\ 0 & -0.03 & 0 & 0 & -10 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t),$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ -0.2 & -0.2 & 0.2 & 0.2 & 0 \end{bmatrix} x(t)$$

The five eigenvalues are {-10.5772, -9.999, -0.9814, -0.5962 ± j0.8702}. Considering the {-10.5772, -9.999} as the fast eigenvalue category, model order reduction is performed.

Discretizing the system with T_s = 0.1s, using a step input with a learning time T_l = 15s, and training the ANN for the input/output data with η = 0.001 learning rate produces the transformed system matrix:

$$\hat{A} = \begin{bmatrix} -0.5967 & 0.8701 & -1.4633 & -0.9860 & 0.0964 \\ -0.8701 & -0.5967 & 0.2276 & 0.6165 & 0.2114 \\ 0 & 0 & -0.9809 & 0.1395 & 0.4934 \\ 0 & 0 & 0 & -9.9985 & 1.0449 \\ 0 & 0 & 0 & 0 & -10.5764 \end{bmatrix}$$

As observed, all the system eigenvalues have been preserved. Based on this transformed matrix, using the LMI technique, the permutation matrix [**P**] is computed and then used for obtaining the [$\hat{\mathbf{B}}$], [$\hat{\mathbf{C}}$], and [$\hat{\mathbf{D}}$] matrices. Since there are two eigenvalues that produce fast dynamics, the following 3rd order reduced model is obtained:

$$\dot{\hat{x}}_r(t) = \begin{bmatrix} -0.5967 & 0.8701 & -1.4633 \\ -0.8701 & -0.5967 & 0.2276 \\ 0 & 0 & -0.9809 \end{bmatrix} \hat{x}_r(t) + \begin{bmatrix} 35.1670 \\ -47.3374 \\ -4.1652 \end{bmatrix} u(t)$$

$$\hat{y}(t) = \begin{bmatrix} -0.0019 & 0 & -0.0139 \\ -0.0024 & -0.0009 & -0.0088 \\ -0.0001 & 0.0004 & -0.0021 \end{bmatrix} \hat{x}_r(t) + \begin{bmatrix} -0.0025 \\ -0.0025 \\ 0.0006 \end{bmatrix} u(t)$$

The reduced model has also preserved the original system dominant eigenvalues {-0.9809, -0.5967± j0.8701}, which achieves the proposed objective. Investigating the performance of this reduced model order compared with the other reduction techniques shows again its superiority as seen in Figure 5. The LMI-based transformed responses are almost identical to the 5th order original systems'.

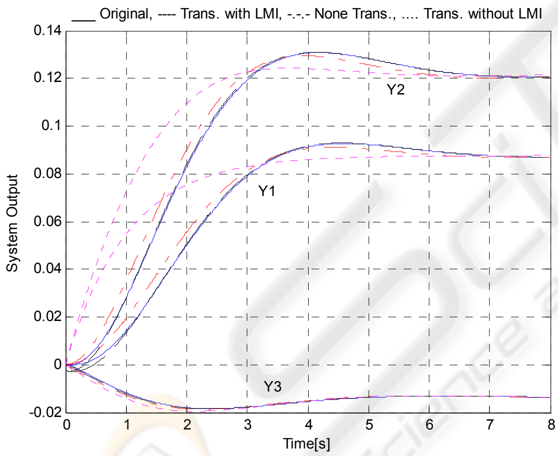

Figure 5: Reduced 3rd model orders (Pink…. transformed with ANN estimation only, Red-.-.-.- non-transformed, Black---- complete transformation with LMI) output responses to a step input along with the non reduced (Blue____ original) system output response. The LMI-transformed curve fits almost exactly on the original response.

5 CONCLUSIONS

In this paper, a new method of dynamic systems model order reduction is presented that has the following advantages. First, in the transformed model, a decoupling of the slow and fast dynamics is achieved. Second, in the reduced model order, the eigenvalues are preserved as a subset of the original system. Third, the reduced model order shows responses that are usually almost identical to the original full order system. Hence, observing the simulation results, it is clear that modeling of dynamic systems using the new LMI-based reduction technique is superior to those other reduction techniques.

REFERENCES

Alsmadi, O., Abdalla, M., 2007. Model order Reduction for Two-Time-Scale Systems Based on Neural Network Estimation. *Proceedings of the 15th Mediterranean Conference on Control & Automation*. Athens, Greece.

Benner, P., 2007. Model Reduction at ICIAM'07. *SIAM News*, V. 40, N. 8.

Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V., 1994. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics (SIAM).

Bui-Thanh, T., Willcox, K., 2005. Model Reduction for Large-Scale CFD Applications using the Balanced Proper Orthogonal Decomposition. *17th American Institute of Aeronautics and Astronautics Computational Fluid Dynamics Conference*. Toronto, Canada.

Franklin, G., Powell, J., Emami-Naeini, A., 1994. *Feedback Control of Dynamic Systems*, Addison-Wesley, 3rd edition.

Garsia, G., Dfouz, J., Benussou, J., 1998. H2 Guaranteed Cost Control for Singularly Perturbed Uncertain Systems. *IEEE Transactions on Automatic Control*, Vol. 43, pp. 1323-1329.

Haykin, S., 1994. *Neural Networks*: *a Comprehensive Foundation*, Macmillan College Publishing Company, New York.

Hinton, G., Salakhutdinov, R., 2006. Reducing the Dimensionality of Data with Neural Networks. *Science*, pp. 504-507.

Horn, R., Johnson, C., 1985. *Matrix Analysis*. Cambridge, UK.

Kokotovic, P., Khalil, H., O'Reilly, J., 1986. *Singular Perturbation Methods in Control: Analysis and Design*, Academic Press. Orlando, Florida.

Williams, R., Zipser, D., 1989. A Learning Algorithm for Continually Running Full Recurrent Neural Networks. *Neural Computation,* 1(2), pp. 270-280.

Zhou, W., Shao, S., Gao, Z., 2009. A Stability Study of the Active Disturbance Rejection Control Problem by a Singular Perturbation Approach. *Applied Mathematical Science*, Vol. 3, no. 10, pp. 491-508.

Zurada, J., 1992. *Artificial Neural Systems,* West Publishing Company, New York.