Soft Computing Using Neural Estimation with LMI-Based Model Transformation for OMR-Based Control of the Buck Converter

Anas N. Al-Rabadi and Othman M.K. Alsmadi

Abstract - **This paper introduces a new method of intelligent control to control the Buck converter using newly developed small signal model of the pulse width modulation (PWM) switch. The new method uses recurrent supervised neural network to estimate certain parameters of the transformed system matrix [$\tilde{A}$]. Then, a numerical algorithm used in robust control called linear matrix inequality (LMI) optimization technique is used to determine the permutation matrix [P] so that a complete system transformation {[$\tilde{B}$], [$\tilde{C}$], [$\tilde{E}$]} is possible. The transformed model is then reduced using the method of singular perturbation, and state feedback control is applied to enhance system performance. The experimental simulation results show that the new control methodology simplifies the model in the Buck converter and thus uses a simpler controller that produces the desired system response for performance enhancement.**

Index Terms - **Buck Converter, Linear Matrix Inequality (LMI), Neural Network (NN), Order Model Reduction (OMR), State Feedback Control, Supervised Learning.**

1. INTRODUCTION

In recent years, small-signal modeling of dynamic behaviors of the open loop dc-to-dc power converters has received notable amount of attention, due to the fact that these models are the basis to extract accurate transfer functions [1,9] which are essential in the feedback control design. They are used to design reliable high performance regulators, by enclosing the open loop dc-to-dc power converters in a feedback loop, to keep the performance of the system as close as possible to the desired operating conditions. The purpose of this feedback loop is to counteract the outside disturbances in the: source voltages, duty ratio (the output pulses of the pulse width modulator (PWM)), and the load current, in order to regulate the output voltage [10,24].

These power converters operate in the Continuous Conduction Mode (CCM) or in the Discontinuous Conduction Mode (DCM) [1,9]. The CCM mode is desirable, as the output ripple of the dc-to-dc power converter is very small compared to the dc steady state output. A linearized small-signal model is constructed to examine the dynamic behaviors of the converter, due to the fact that disturbances are of small signal variations.

Through this model, the necessary open-loop transfer functions can be determined and plotted using Bode plots [9]. This is needed in order to use compensation to the pulse width modulation (PWM) power converters, to meet the desired nominal operating conditions, through the application of various control methods. These control methods can incorporate the approaches of: frequency analysis in the classical control theory, time analysis in the modern control theory, both frequency analysis and time analysis domains in the post modern (digital and robust) control theory, and the soft computing (fuzzy logic + neural networks + genetic algorithms) in the intelligent control theory [6,9,10,15,24]. These control methods can be applied to the models of power converters that usually work with only one specific control scheme, which is pulse width modulation (PWM) through either duty-ratio control, or current programming control [9]. In this paper, the duty-ratio control is used, in which the switch ON-time is controlled externally by comparing a sawtooth ramp with the controller voltage [1,9].

Various modeling approaches of the PWM power converters already exist. These approaches can be separated into three main categories. The first modeling category aims towards modeling the whole PWM converters. Examples for this category are: the volt-second and current-second (charge) balance approach, and the state-space averaging approach [9]. These approaches suffer from inaccurate results in the high frequency range. The second modeling category aims more specifically towards modeling what is called the converter-cell, that includes modeling the basic cell of the PWM converter, and ignoring the input (the dc voltage source) and the output (the RC filter) parts in the model (the cell includes only the PWM switch with the inductors and the capacitors associated with it). An example for this category is the averaged modeling approach [1,9]. This approach also suffers from inaccurate results in the high frequency range. The third modeling category aims more specifically to model the PWM switch, by itself, in the PWM power converters.

The previously mentioned modeling approaches utilize in general four techniques. The first technique is the sampled-data representation technique. The second

Manuscript received May 8, 2009.

A. N. Al-Rabadi is an Associate Professor with the Computer Engineering Department, The University of Jordan, Amman, Jordan (Phone: (+962) 79 644-5364; URL: http://web.pdx.edu/~psu21829/). Please send all E-mails to the corresponding author Dr. Al-Rabadi at the following E-mail address (alrabadi@yahoo.com).

O. M.K. Alsmadi is an Assistant Professor with the Electrical Engineering Department, The University of Jordan, Amman, Jordan (E-mail: othman_mk@yahoo.com).

technique is the averaged technique. The third technique is the exact small-signal analysis technique [1,9], and the fourth technique combines the averaged technique and the sampled-data technique. The averaged technique represents the easiest and the most widely used technique. It can be used to determine various impedances and transfer functions of the converter systems. The basic characteristics of this technique are: (1) It uses the averaging technique of voltages and currents and (2) It gives accurate low frequency results, but inaccurate high frequency results.

Averaged models can be produced for the nonlinear switch in the converter circuits, which is called the PWM switch, as well for the converter systems as a whole. This switch is usually a single pole double throw (SPDT) switch. It is this switch which is responsible for switching the converter from one configuration to another during each switching period. These models derived for the PWM switch are usually easier than the derivation of converter models. Yet, it has the limitation of the fact that not all the converter topologies have the same PWM switch arrangement [1].

The exact small-signal technique [1,9] is very accurate to a wide range of frequencies. This technique can be applied to any converter systems that are: periodic, time-varying, and piecewise linear. The trade off for the high accuracy occurs in the complexity of the matrix manipulations and the time consumed to produce the exact results. Yet, it has a great advantage of being automated through the use of computer aided design (CAD) software packages.

The sampled data technique is based on the generation of a difference equation that describes the propagation of a point on a converter waveform from one cycle to another. It is usually used to derive accurate response for the PWM current mode control. Yet, the price is paid again through the limitation of the upper frequency range, to be limited to half the switching frequency. The fourth modeling technique combines the averaged technique and the sampled-data technique, in an effort to gain the main benefits of each technique. However, this technique, while improved, is also inaccurate [9].

From above, it can be seen that there is a need to develop a model applicable to various regulating schemes, including the most used scheme: the PWM duty ratio and current mode control scheme. Therefore, a small-signal modeling approach which is applicable to any power converter system represented as a two-port network has been introduced [1]. This was done through the modeling of the nonlinear part in the power converter system, which is the PWM switch.

In system modeling, sometimes it is required to identify some of the system parameters. This objective maybe achieved by the use of artificial neural networks (ANN), which are considered as the new generation of information processing networks. Artificial neural systems maybe defined as physical cellular systems which have the capability of acquiring, storing, and utilizing experiential knowledge [15,29]. In fact, an ANN is a mathimatical or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases, an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase [29]. The basic processing elements of neural networks are called neurons. They perform summing operations and nonlinear function computations. Neurons are usually organized in layers and forward connections. Computations are performed in a parallel fashion at all nodes and 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. In more practical terms, neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. A neural network is an interconnected group of nodes, akin to the vast network of neurons in the human brain [5,17,28,29].

When dealing with system modeling and control analysis, some equations and inequalities require optimized solutions. A numerical algorithm, used in robust control called linear matrix inequality (LMI) serves as a source of application problems in convex optimization [6]. LMI optimization technique started by the Lyapunov theory 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 $A^T P + PA < 0$ [6]. The requirement $P > 0$, $A^T P + PA < 0$ is what is known as Lyapunov inequality on [**P**] which is a special case of an LMI. By picking any $Q = Q^T > 0$ and then solving the linear equation $A^T P + PA = -Q$ for the matrix [**P**], it is guaranteed to be positive-definite if the given system is stable. This LMI was used for stability purposes. Next significant work was done by applying Lyapunov method to some specific practical problems in control engineering. The LMIs that arise in system and control theory can be formulated as convex optimization problems that are amenable to computer solution and then solved using algorithms such as the ellipsoid algorithm. A Lyapunov function is formulated as a convex optimization problem, and then an algorithm guaranteed to solve the optimization problem is applied. In recent works, interior-point methods that apply directly to convex problems involving LMIs were developed [6].

In practical control problems, the first step is to obtain a mathematical model in order to examine the behavior of the system for the purpose of designing a proper controller [10]. Sometimes, this mathematical description involves a certain small parameter (perturbation). Neglecting this small parameter results in simplifying the order of the designed controller by

reducing the order model of the system [2,21,22]. A reduced order model 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 simplification and reduction of system modeling leads to controller cost minimization [2,7,21,22]. An example is the ICs, where increasing package density forces developers to include side effects. Knowing that these devices are often modeled by very large RLC circuits, this would be too demanding computationally due to the detailed modeling of the original system [4]. In control system, due to the fact that feedback controllers do not usually consider all the dynamics of the system, model reduction is a very important issue [26]. Model reduction leads to reducing the order of the controller, which is directly proportional to the cost [11,12].

One of the methods for order model reduction is known as singular perturbation [20,27]. Using the quadratic stabilization framework, the work in [13] presented the H_2 guaranteed cost control problem for singularly perturbed norm-bounded uncertain systems. The problem was solved by solving two Riccati equations associated with the slow and fast subsystems based on properly selected weighting matrices. The investigation of the estimator error for the slow states of singularly perturbed systems has been performed in [19]. In [21], the singular perturbation method was employed to capture the multimodel nature of interconnected systems with slow and fast dynamics. Systems strongly coupled through their slow parts and weakly coupled through their fast parts were considered.

Figure 1 illustrates the layout of the Buck-based converter control methodology used in this paper. Layer 1 (the base) is the Buck converter model using the new small signal modeling approach. Layer 2 is the neural network estimation of the transformed system matrix [$\tilde{\mathbf{A}}$]. Layer 3 is the LMI technique used in determining the permutation matrix required for system transformation {[$\tilde{\mathbf{B}}$], [$\tilde{\mathbf{C}}$], [$\tilde{\mathbf{E}}$]}. Layer 4 is the system transformation. Layer 5 presents the order model reduction. Finally, layer 6 presents the state feedback control.

State Feedback Control
Order Model Reduction
System Transformation: {[$\tilde{\mathbf{B}}$], [$\tilde{\mathbf{C}}$], [$\tilde{\mathbf{E}}$]}
LMI-Based Permutation Matrix: [P]
Neural-based State Transformation: [$\tilde{\mathbf{A}}$]
Buck Converter: {[A], [B], [C], [E]} New Small Signal Model

Figure 1. Buck-based converter hierarchical control methodology used in this paper.

Section 2 presents background on the Buck converter, recurrent supervised neural network, linear matrix inequality, model transformation, and order model reduction. Section 3 presents a detailed illustration of the recurrent neural network estimation with the LMI optimization techniques for order model reduction of the Buck converter. An implementation of the neural network estimation with the LMI optimization to the Buck converter order model reduction is presented in Section 4. Section 5 presents the application of state feedback controller on the reduced order model of the Buck converter. Conclusions and future work are presented in Section 6.

2. BACKGROUND

This section presents important background on Buck converter, supervised neural network, LMI, and order model reduction that will be used in Sections 3, 4 and 5.

2.1 Switching Mode Power Supply (SMPS): The Application of the Averaged Modeling Approach and the New Small Signal Model on the PWM Converters

There are many averaged modeling techniques used to model the PWM converters. These techniques include: volt-second and current-second balance approach, and the state-space averaging approach [1,9]. These techniques are used to model the converter systems as a whole, as well as to model the pulse width modulation (PWM) switch by itself. Yet, these techniques are valid for the low frequency range, and they give inaccurate results for the dynamic behaviors of the power converters in the high frequency ranges [9]. Another modeling approach that focuses on modeling the converter-cell, instead of the converter as a whole, is used to get averaged models for the PWM converters. This approach is also useful for the low frequency ranges, but not useful for the high frequency ranges. One major advantage of these techniques is the fact that they are easy to implement, and the results obtained are not in complicated forms.

2.1.1 The Averaged Modeling Approach and its Application on the Buck Converter

The averaged modeling approach aims to produce an averaged model for a specific cell of the PWM converters. This cell is shown in Figure 2, where this basic cell is used to explore the dc behaviors, and the ac small-signal dynamic behaviors of the PWM Buck converter.

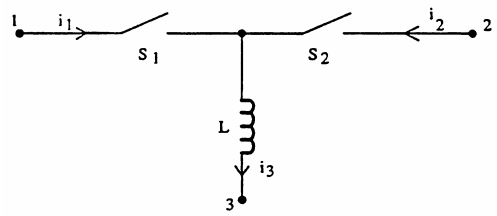

Figure 2. Basic PWM converter-cell.

It is shown that a dc and ac small-signal averaged model of the converter-cell which is shown in Figure 2, can be produced as shown in Figure 3, where D is the dc value of the duty ratio, $\hat{d}$ is the small-signal perturbation of the duty ratio, and V_{32} is the dc voltage between terminals 3 and 2.

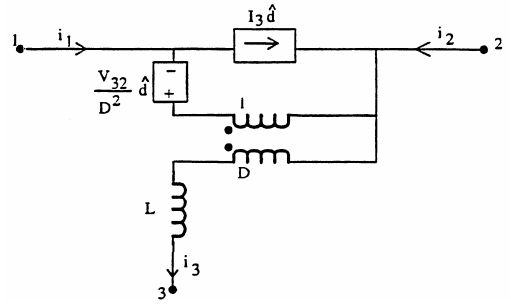

Figure 3. DC and AC small-signal model of the converter-cell shown in Figure 2.

The dc and ac small-signal averaged model, shown in Figure 3, will be used to derive the average control-to-output, input-to-output, input impedance, and the control-to-input current transfer functions for the Buck converter. The averaged model will be used to derive the input-to-output, control-to-output, input impedance, and control-to-input current average transfer functions for the PWM Buck converter. Figure 4 shows a typical Buck converter.

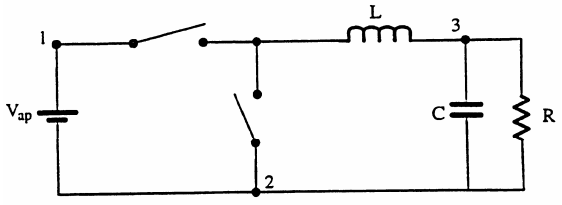

Figure 4. Typical Buck converter.

Assuming a small-signal perturbation, $\hat{v}_g$, in the dc voltage source, V_g, and that $|V_g| >> |\hat{v}_g|$. After the implementation of the averaged model that was shown previously, we get the following small-signal model for the PWM Buck converter as shown in Figure 5.

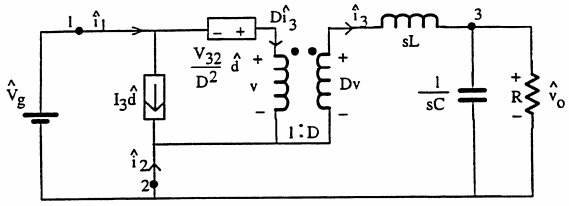

Figure 5. AC small-signal model of the Buck converter operating in the continuous conduction mode (CCM).

Nulling the input $\hat{v}_g$, we get the following control-to-output transfer function [1]:

$$\frac{\hat{v}_o}{\hat{d}} = V_g \frac{1}{1+(L/R)s+LCs^2} \tag{1}$$

Nulling the input $\hat{d}$, we get the following input-to-output transfer function [1]:

$$\frac{\hat{v}_o}{\hat{v}_g} = D\frac{1}{1+(L/R)s+LCs^2} \tag{2}$$

Other transfer functions of interest for the Buck converter are the input impedance and the control-to-input current transfer functions.

To get the input impedance ($\hat{v}_g / \hat{i}$), we null the input $\hat{d}$, so we get the following equation [1]:

$$\frac{\hat{v}_g}{\hat{i}_1} = \frac{R}{D^2}\frac{1+(L/R)s+LCs^2}{1+RCs} \tag{3}$$

To get the control-to-input current transfer function ($\hat{i}_1 / \hat{d}$), we null the input $\hat{v}_g$, so we get the following transfer function [1]:

$$\frac{\hat{i}_1}{\hat{d}} = \frac{V_g D+RI3}{R}\frac{1+(\frac{Vg\,DRC+LI3}{Vg\,D+RI3})s+(\frac{LI3RC}{Vg\,D+RI3})s^2}{1+(L/R)s+LCs^2} \tag{4}$$

2.1.2 New Method for Obtaining an Exact Model of the PWM Switch Operating in the Duty Ratio Programming Mode

In this subsection, a new approach is developed to formulate a new model for the PWM nonlinear switch [1]. The Buck converter will be used now as the basic model to extract the two-port network parameters. The main reason the Buck is used over the other PWM converters, is the fact that the Buck converter is a second order system, with a simple structure. This will be reflected upon the simplicity of the results that will be obtained.

Since the ripple voltage is comparatively much smaller than the dc voltage across the output capacitor (as the Buck converter is operating in the continuous conduction mode), the capacitor will be replaced with a constant dc voltage source $V_{c'p}$. This is illustrated in Figure 6.

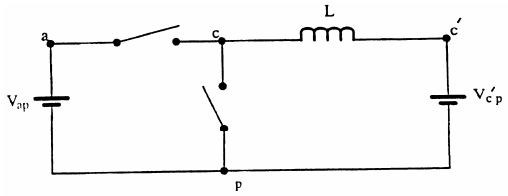

Figure 6. Alternative Buck configuration.

From the Buck converter shown in Figure 6, the two-port augmented equations can be written as follows:

$$\hat{i}_a = y_i \hat{v}_{ap} + y_{io}\hat{v}_{c'p} + i_{id}\hat{d} \tag{5}$$

$$\hat{v}_{c'p} = z_o \hat{i}_c + g_{vf}\hat{v}_{ap} + v_{od}\hat{d} \tag{6}$$

A circuit model for the two-port augmented equations, which are represented by Equations (5) and (6), can be constructed as shown in Figure 7.

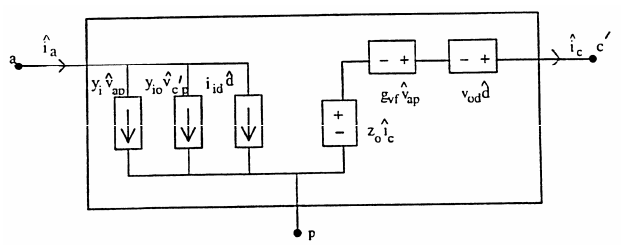

Figure 7. Circuit model for Equations (5) and (6).

The aim is to develop a new model for the PWM switch, which is the nonlinear part of the PWM converter. This model can be constructed directly by replacing the values of the parameters $\{y_i, y_{io}, i_{id}, z_o, g_{vf}, v_{od}\}$ in their simplest form [1] in Equations (5) and (6). Thus, the mathematical model will be [1]:

$$\begin{aligned}\hat{i}_a &= y_i\hat{v}_{ap} + y_{io}\hat{v}_{c'p} + i_{id}\hat{d} \\ &= \frac{\beta_1\beta_2(1+j\omega DTs)+1-j\omega DTs-\beta_1-\beta_2}{Ts\omega^2L(1-\beta_1\beta_2)}\hat{v}_{ap} \\ &\quad + \frac{jD}{\omega L}\hat{v}_{c'p} + I_x - \frac{j\beta_2(1-\beta_1)V_{ap}}{\omega L(1-\beta_1\beta_2)}\hat{d}\end{aligned} \tag{7}$$

$$\begin{aligned}\hat{v}_{c'p} &= z_o\hat{i}_c + g_{vf}\hat{v}_{ap} + v_{od}\hat{d} \\ &= -j\omega L\hat{i}_c + D\hat{v}_{ap} + V_{ap}\hat{d}\end{aligned} \tag{8}$$

By noting that the parameter z_o represents an inductor, we can "pull" out the z_o parameter outside the circuit model, which is equivalent to the mathematical model represented by Equations (7) and (8), as the z_o parameter is merely an inductor impedance, which is then multiplied by the path current $\hat{i}_c$, to form a voltage source ($z_o\hat{i}_c$) in series with the voltage sources ($g_{vf}\hat{v}_{ap}$) and ($v_{od}\hat{d}$). The result of this process is shown in Figure 8.

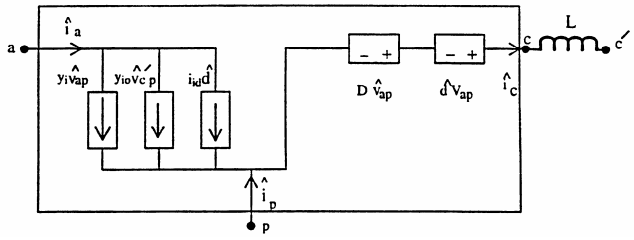

Figure 8. New circuit model.

From Figure 8, we can recognize that the circuit model between the terminals {a, p, c} is merely the switch between these terminals in the original Buck converter circuit. So, the equivalent switch model in terms of the perturbations $\{\hat{v}_{ap}, \hat{v}_{c'p}, \hat{d}\}$ is as shown in Figure 9.

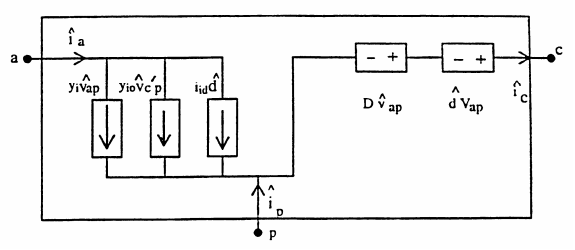

Figure 9. Circuit model for the PWM switch.

Now we need to put the switch model in terms of the perturbations $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$ instead of the perturbations $\{\hat{v}_{ap}, \hat{v}_{c'p}, \hat{d}\}$. To do so, we note from Figure 8 that:

$$\hat{v}_{c'p} = \hat{v}_{cp} - j\omega L\hat{i}_c \tag{9}$$

From Figure 8, we note that the common node (c') corresponds to the node (c') in the Buck converter in Figure 6. Multiplying both sides of Equation (9) by the parameter y_{io}, we get:

$$y_{io}\hat{v}_{c'p} = y_{io}\hat{v}_{cp} + D\hat{i}_c \tag{10}$$

So, we replace the term $\{y_{io}\hat{v}_{cp} + D\hat{i}_c\}$ instead of the term $\{y_{io}\hat{v}_{c'p}\}$ in the previously derived switch model shown in Figure 9, in order to make the new model contain the perturbations $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$ instead of the perturbations $\{\hat{v}_{ap}, \hat{v}_{c'p}, \hat{d}\}$. The new switch model will be as shown in Figure 10.

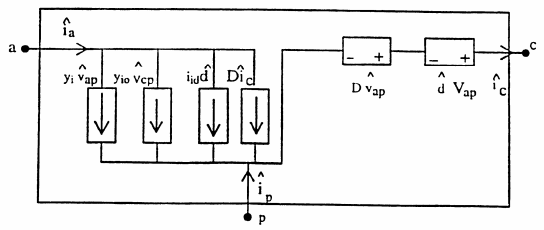

Figure 10. Alternative circuit model for the PWM switch.

In order to reduce the number of the dependent current sources that appear in the new switch model, which are four dependent current sources, we will try to reduce the number of terms in the previous mathematical switch model. We have:

$$\hat{v}_{cp} = V_{ap}\hat{d} + D\hat{v}_{ap} \tag{11}$$

and as $\hat{v}_{c'p} = z_o\hat{i}_c + g_{vf}\hat{v}_{ap} + v_{od}\hat{d}$, we obtain:

$$\therefore \hat{v}_{c'p} = V_{ap}\hat{d} + D\hat{v}_{ap} - j\omega L\hat{i}_c \quad (12)$$

To develop the first reduced mathematical switch model, we see that as: $\hat{i}_a = y_i\hat{v}_{ap} + y_{io}\hat{v}_{c'p} + i_{id}\hat{d}$. Substituting Equation (12) in Equation (5), and after the collection of the similar terms, we get the following reduced-form equation:

$$\hat{i}_a = (y_i + y_{io}D)\hat{v}_{ap} + (y_{io}V_{ap} + i_{id})\hat{d} - jy_{io}\omega L\hat{i}_c \quad (13)$$

Substituting the values of $\{y_i, y_{io}, i_{id}\}$ in Equation (13), we get the following equation [1]:

$$\hat{i}_a = \left(\frac{e^{-j\omega Ts}(1+j\omega DTs)+1-j\omega DTs-e^{-j\omega DTs}-e^{-j\omega D'Ts}}{Ts\omega^2 L(1-e^{-j\omega Ts})} + \frac{jD^2}{\omega L}\right)\hat{v}_{ap} + \left(I_x + \frac{jDVa\,p}{\omega L} - \frac{je^{-j\omega D'Ts}(1-e^{-j\omega DTs})Va\,p}{\omega L(1-e^{-j\omega Ts})}\right)\hat{d} + D\hat{i}_C \quad (14)$$

The other equation of the model is Equation (11). So, Equations (11) and (14) represent the final reduced mathematical model of the PWM switch, replacing the model represented in Figure 9. The final equivalent circuit model of the switch mathematical model (represented by the Equations (11) and (14)) is as shown in Figure 11 [1], where:

$$h_1 = \frac{e^{-j\omega Ts}(1+j\omega DTs)+1-j\omega DTs-e^{-j\omega DTs}-e^{-j\omega D'Ts}}{Ts\omega^2 L(1-e^{-j\omega Ts})} + \frac{jD^2}{\omega L} \quad (15)$$

$$h_2 = Ix + \frac{jDVa\,p}{\omega L} - \frac{je^{-j\omega D'Ts}(1-e^{-j\omega DTs})Va\,p}{\omega L(1-e^{-j\omega Ts})} \quad (16)$$

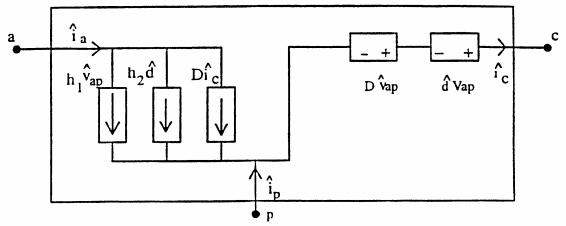

Figure 11. The new small-signal model of the PWM switch.

The new switch model in Figure 11 is expected to be an exact small-signal model, since the mathematical equations upon which the whole derivation process was built, are exact. Also, we note that two of the dependent current sources are frequency dependent, which is uncommon for current or voltage dependent sources.

2.1.3 Examining the New Small-Signal: The Implementation of the New Small-Signal Model of the PWM Switch on the Buck Converter

In this subsection, the new small-signal model of the PWM switch, that was developed in the previous subsection, will be examined on the PWM Buck converter. The control-to-output, input-to-output, input impedance, and control-to-input current transfer functions will be derived for the Buck converter, using the new small-signal model of the PWM switch. These transfer functions will be compared to the corresponding transfer functions, that were developed in previous subsections for the averaged modeling approach and the exact transfer functions for the Buck converter.

Applying the PWM switch model, that was developed previously on the Buck converter, we get the equivalent circuit model as shown in Figure 12.

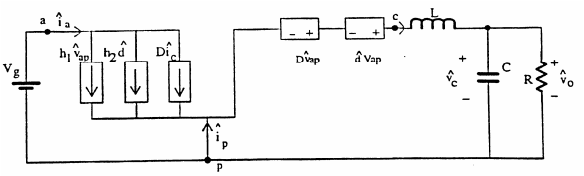

Figure 12. Equivalent circuit model of the PWM Buck converter, obtained through the application of the new small-signal model of the PWM switch.

Assuming that the input dc voltage source, V_g, has small-signal perturbation, $\hat{v}_g$, and that: $|V_g| >> |\hat{v}_g|$. To determine the system quadruple {[**A**], [**B**], [**C**], [**E**]}, for the Buck model shown in Figure 12, for the inputs $\hat{d}$ and $\hat{v}_g$, we null the dc voltage source, V_g. Then, the following equations can be developed for the Buck model shown in Figure 12, where $\hat{i}_l$ is the inductor current, $\hat{i}_c$ is the capacitor current, and $\hat{i}_o$ is the output current that flows in the output resistor. Hence, we have:

$$\hat{v}_o = \hat{v}_c,\ \hat{i}_l = \hat{i}_c + \hat{i}_o = C\dot{\hat{v}}_c + \frac{1}{R}\hat{v}_o \quad (17)$$

$$\therefore \dot{\hat{v}}_c = \frac{1}{C}\hat{i}_l - \frac{1}{RC}\hat{v}_c \quad (18)$$

$$\hat{v}_{ap} = \hat{v}_g \quad (19)$$

$$-D\hat{v}_g - \hat{d}V_g + L\dot{\hat{i}}_l + \hat{v}_c = 0$$

$$\therefore \dot{\hat{i}}_l = \frac{D}{L}\hat{v}_g + \frac{V_g}{L}\hat{d} - \frac{1}{L}\hat{v}_c \quad (20)$$

The output equation is $y = \hat{v}_o = \hat{v}_c$. For the converter states, $\boldsymbol{x}$, and the inputs, $\boldsymbol{u}$, where $\boldsymbol{x} = \begin{bmatrix}\hat{i}_l \\ \hat{v}_o\end{bmatrix}$, $\boldsymbol{u} = \begin{bmatrix}\hat{v}_g \\ \hat{d}\end{bmatrix}$, the system quadruple {[**A**], [**B**], [**C**], [**E**]} will be [1]:

$$A = \begin{bmatrix}0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC}\end{bmatrix},\ B = \begin{bmatrix}\mathrm{D/L} & \mathrm{Vg/L} \\ 0 & 0\end{bmatrix},$$

$C = [0 \quad 1]$, $E = [0 \quad 0]$.

To find the control-to-output transfer function, we null the input $\hat{v}_g$. The new system quadruple {[**A**], [**B**], [**C**], [**E**]}, will be [1]:

$$A = \begin{bmatrix} 0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC} \end{bmatrix} \tag{21}$$

$$B = \begin{bmatrix} \mathrm{V_g}/\mathrm{L} \\ 0 \end{bmatrix} \tag{22}$$

$$C = [0 \quad 1] \tag{23}$$

$$E = [0] \tag{24}$$

To find the control-to-output transfer function from the system quadruple represented by Equations (21) through (24), we apply the Laplace transformation to both sides of the state and the output equations represented by system state space equations: $\dot{x}(t) = Ax(t) + Bu(t)$ and $y(t) = Cx(t) + Eu(t)$. After re-arranging the result terms, we get the following general input-to-output transfer function:

$$\frac{y}{u} = C(s\mathbf{I} - A)^{-1}B + E \tag{25}$$

By applying Equations (21) through (24) in Equation (25), for the circuit values of: V_g = 15 V, R = 18.6 Ω , D = 0.4, f_s = 40.3 kHz, D' = 0.6, L = 58 μH, C = 5.5 μF and to investigate the accuracy of the new PWM switch model, we compare the control-to-output frequency response plots of the PWM Buck, obtained through the application of the new PWM switch small-signal model, with the exact control-to-output frequency response, the averaged control-to-output frequency response, and both the exact and the averaged control-to-output frequency responses, as shown in Figure 13.

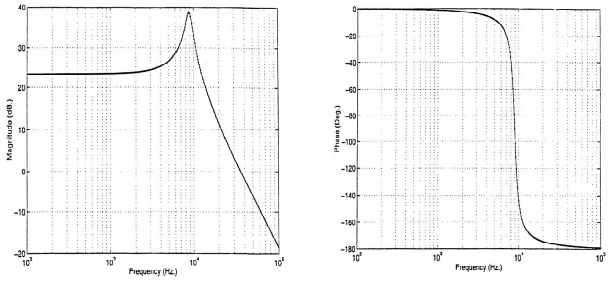

Figure 13. The control-to-output frequency response of the PWM Buck converter, operating in CCM; exact (solid line); averaged (dotted line); and the new model (dashed line).

To get the input-to-output transfer function, we null the input $\hat{d}$. The system quadruple {[**A**], [**B**], [**C**], [**E**]} will be [1]:

$$A = \begin{bmatrix} 0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC} \end{bmatrix} \tag{26}$$

$$B = \begin{bmatrix} \mathrm{D}/\mathrm{L} \\ 0 \end{bmatrix} \tag{27}$$

$$C = [0 \quad 1] \tag{28}$$

$$E = [0] \tag{29}$$

By applying Equations (26) through (29) in Equation (25), for the circuit values of: V_g = 15 V, R = 18.6 Ω , D = 0.4, f_s = 40.3 kHz, D' = 0.6, L = 58 μH, C = 5.5 μF and to investigate the accuracy of the new PWM switch model, we compare the input-to-output frequency response plots of the PWM Buck, obtained through the application of the new PWM switch small-signal model, with the exact input-to-output frequency response, the averaged input-to-output frequency response, and both the exact and the averaged input-to-output frequency responses, as shown in Figure 14.

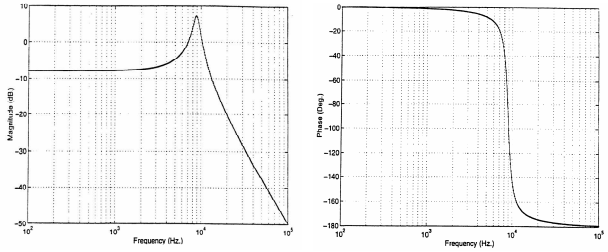

Figure 14. The input-to-output frequency response of the PWM Buck converter, operating in CCM; exact (solid line); averaged (dotted line); and the new model (dashed line).

From the previous frequency response plots for both the control-to-output and the input-to-output transfer functions of the PWM Buck converter, operating in the CCM, we see that an excellent match occurs between the exact and the new model results, as well as between the averaged and the new model results. These results indicate, for the time being, that the new small-signal model of the PWM switch is, in fact, an accurate model [1]. Yet, the effect of the new source coefficients h_1 and h_2 that exist in the new model of the PWM switch, does not appear in the case of the control-to-output and input-to-output transfer functions. So, we need the input impedance and the control-to-input current transfer functions to see the effect of the new source coefficients h_1 and h_2, respectively.

By referring to Figure 12, and considering the input current $\hat{i}_a$ to be the output, we get the following output equation: $y = \hat{i}_a$. For the converter states, $\boldsymbol{x}$, and the inputs, $\boldsymbol{u}$, where $\boldsymbol{x} = \begin{bmatrix} \hat{i}_l \\ \hat{v}_o \end{bmatrix}$, $\boldsymbol{u} = \begin{bmatrix} \hat{v}_g \\ \hat{d} \end{bmatrix}$, the system quadruple {[**A**], [**B**], [**C**], [**E**]}, will be [1]:

$$A = \begin{bmatrix} 0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC} \end{bmatrix},\ B = \begin{bmatrix} \mathrm{D/L} & \mathrm{V_g/L} \\ 0 & 0 \end{bmatrix},$$
$$C = \begin{bmatrix} \mathrm{D} & 0 \end{bmatrix},\ E = \begin{bmatrix} \mathrm{h_1} & \mathrm{h_2} \end{bmatrix}.$$

To find the control-to-input current transfer function, we null the input $\hat{v}_g$. The new system quadruple {[**A**], [**B**], [**C**], [**E**]}, will be [1]:

$$A = \begin{bmatrix} 0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC} \end{bmatrix} \quad (30)$$

$$B = \begin{bmatrix} \mathrm{V_g/L} \\ 0 \end{bmatrix} \quad (31)$$

$$C = \begin{bmatrix} \mathrm{D} & 0 \end{bmatrix} \quad (32)$$

$$E = \begin{bmatrix} \mathrm{h_2} \end{bmatrix} \quad (33)$$

To find the control-to-input current transfer function from the system quadruple represented by Equations (30) through (33), we use the general input-to-output transfer function represented by Equation (25).

By applying Equations (30) through (33) in Equation (25), for the circuit values of: V_g = 15 V, R = 18.6 Ω , D = 0.4, f_s = 40.3 kHz, D' = 0.6, L = 58 μH, C = 5.5 μF and to investigate the accuracy of the new PWM switch model, we compare the control-to-input current frequency response plots of the PWM Buck, obtained through the application of the new PWM switch small-signal model, with the exact control-to-input current frequency response, the averaged control-to-input current frequency response, and both the exact and the averaged control-to-input current frequency responses, as shown in Figure 15.

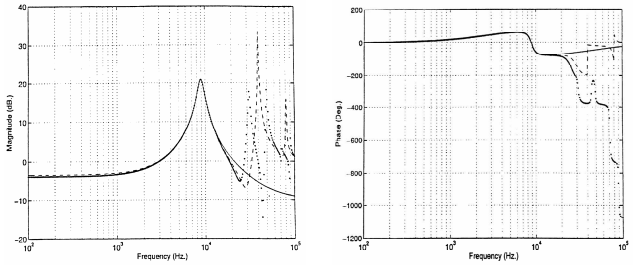

Figure 15. The control-to-input current frequency response of the PWM Buck converter, operating in CCM; exact (solid line); averaged (dotted line); and the new model (dashed line).

To get the input impedance transfer function, we null the input $\hat{d}$. The system quadruple {[**A**], [**B**], [**C**], [**E**]} will be [1]:

$$A = \begin{bmatrix} 0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC} \end{bmatrix} \quad (34)$$

$$B = \begin{bmatrix} \mathrm{D/L} \\ 0 \end{bmatrix} \quad (35)$$

$$C = \begin{bmatrix} \mathrm{D} & 0 \end{bmatrix} \quad (36)$$

$$E = \begin{bmatrix} \mathrm{h_1} \end{bmatrix} \quad (37)$$

By applying Equations (34) through (37) in Equation (25), for the circuit values of: V_g = 15 V, R = 18.6 Ω , D = 0.4, f_s = 40.3 kHz, D' = 0.6, L = 58 μH, C = 5.5 μF and to investigate the accuracy of the new PWM switch model, we compare the input impedance frequency response plots of the PWM Buck, obtained through the application of the new PWM switch small-signal model, with the exact input impedance frequency response, the averaged input impedance frequency response, and both the exact and the averaged input impedance frequency responses, as shown in Figure 16.

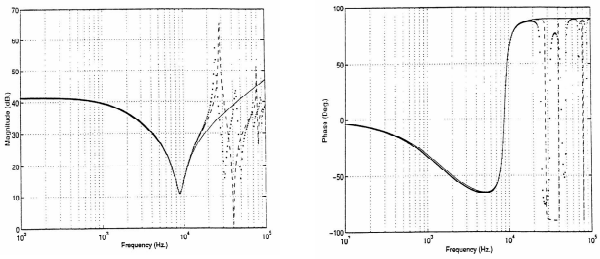

Figure 16. The input impedance frequency response of the PWM Buck converter, operating in CCM; exact (solid line); averaged (dotted line); and the new model (dashed line).

From the previous frequency response plots of the transfer functions of the PWM Buck converter, operating in the CCM, we see that a good match occurs between the exact and the new model results, as well as between the averaged and the new model results, for the frequency range up to half the switching frequency [1], although a mismatch occurs between the exact and the new model results, as well as between the averaged and the new model results, for the frequency range higher than half of the switching frequency [1]. Yet, in overall performance evaluation, the new small signal model behaves in a much accurate response than the older averaged modeling approach.

2.2 Recurrent Supervised Neural Network

An artificial neural network is is an emulation of biological neural system. The basic model of the neuron is founded upon the functionality of a biological neuron, where it is the basic signaling unit of the nervous system. The process of a neuron may be mathematically modeled as shown in Figure 17 [15].

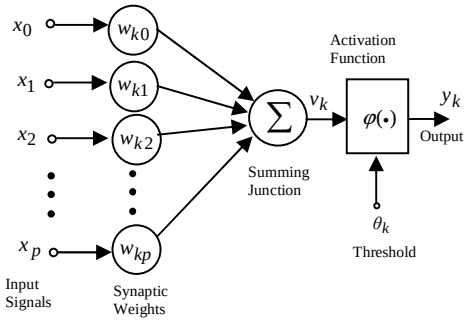

Figure 17. Mathematical model of an artificial neuron.

As seen in Figure 17, the internal activity of the neuron can be shown to be:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j \qquad (38)$$

In supervised learning, it is assumed that at each instant of time when the input is applied, the desired response of the system is available. The difference between the actual and the desired response represents an error measure and is used to correct the network parameters externally. Since the adjustable weights are initially assumed, the error measure may be used to adapt the network's weight matrix [**W**]. A set of input and output patterns called a training set is required for this learning mode. The training algorithm estimates directions of the negative error gradient and reduces the error accordingly [29].

For artificial neural network, there are several learning rules used to train the neural network. For example, in Perceptron learning rule, the learning signal is the difference between the desired and the actual neuron's response (supervised learning). Another learning rule is the Widrow-Hoff learning rule which minimizes the squared error between the desired output and the neuron's activation value. Backpropagation is also one of the important learning algorithms in neural networks [29].

The supervised recurrent neural network used for the estimation in this paper is based on an approximation of the method of steepest descent [15,28,29]. The network tries to match the output of certain neurons to the desired values of the system output at specific instant of time [28]. Consider a network consisting of a total of *N* neurons with *M* external input connections, as shown in Figure 18 for a 2nd order system with two neurons and one external input.

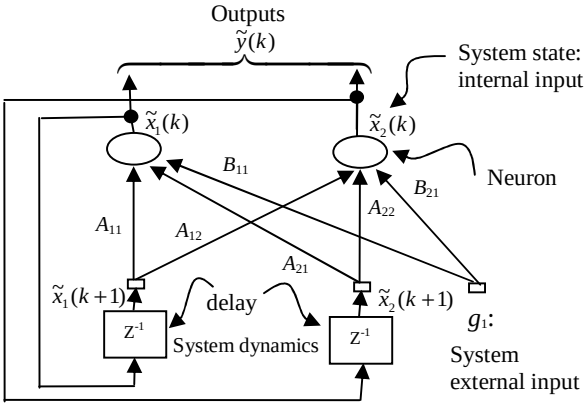

Figure 18. A second order recurrent neural network architecture, where the estimated matrices are given by: $\tilde{A}_d = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $\tilde{B}_d = \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}$ and that $W = \left[[\tilde{\mathbf{A}}_\mathbf{d}] \;\; [\tilde{\mathbf{B}}_\mathbf{d}] \right]$.

The variable $\mathbf{g}(k)$ denotes the (M x 1) external input vector applied to the network at discrete time k. 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 ß 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:

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

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

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

where $\Lambda \cup$ ß is the union of sets Λ and ß. At the next time step (k + 1), the output of the neuron j is computed by passing $v_j(k)$ through the nonlinearity $\varphi(.)$ obtaining:

$$y_j(k+1) = \varphi(v_j(k))$$

The derivation of the recurrent algorithm can be started by using $d_j(k)$ to denote the desired (target) 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:

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

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

$$E_{\text{total}} = \sum_k E(k) \text{ , where } E(k) = \frac{1}{2} \sum_{j \in \varsigma} e_j^2(k)$$

To accomplish this objective, the method of steepest descent which requires knowledge of the gradient matrix is used:

$$\nabla_{\mathbf{W}} E_{\text{total}} = \frac{\partial E_{\text{total}}}{\partial \mathbf{W}} = \sum_k \frac{\partial E(k)}{\partial \mathbf{W}} = \sum_k \nabla_{\mathbf{W}} E(k)$$

where $\nabla_{\mathbf{W}} E(k)$ is the gradient of $E(k)$ with respect to the weight matrix [**W**]. In order to train the recurrent network

in real time, the instantaneous estimate of the gradient is used $\left(\nabla_{\mathbf{W}} E(k)\right)$. For the case of a particular weight $w_{m\ell}(k)$, the incremental change $\Delta w_{m\ell}(k)$ made at time k is defined as:

$$\Delta w_{m\ell}(k) = -\eta \frac{\partial E(k)}{\partial w_{m\ell}(k)}$$

where η is the learning-rate parameter. Hence:

$$\frac{\partial E(k)}{\partial w_{m\ell}(k)} = \sum_{j\in\varsigma} e_j(k) \frac{\partial e_j(k)}{\partial w_{m\ell}(k)} = -\sum_{j\in\varsigma} e_j(k) \frac{\partial y_i(k)}{\partial w_{m\ell}(k)}$$

To determine the partial derivative $\partial y_j(k)/\partial w_{m\ell}(k)$, the network dynamics are derived. The derivation is obtained by using the chain rule which provides the following equation:

$$\frac{\partial y_j(k+1)}{\partial w_{m\ell}(k)} = \frac{\partial y_j(k+1)}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial w_{m\ell}(k)} = \dot{\varphi}(v_j(k)) \frac{\partial v_j(k)}{\partial w_{m\ell}(k)} \quad ,$$

where $\dot{\varphi}(v_j(k)) = \dfrac{\partial \varphi(v_j(k))}{\partial v_j(k)}$.

Differentiating the net internal activity of neuron j with respect to $w_{m\ell}(k)$ yields:

$$\frac{\partial v_j(k)}{\partial w_{m\ell}(k)} = \sum_{i\in\Lambda\cup\beta} \frac{\partial (w_{ji}(k) u_i(k))}{\partial w_{m\ell}(k)} = \sum_{i\in\Lambda\cup\beta} \left[w_{ji}(k) \frac{\partial u_i(k)}{\partial w_{m\ell}(k)} + \frac{\partial w_{ji}(k)}{\partial w_{m\ell}(k)} u_i(k) \right]$$

where $\left(\partial w_{ji}(k)/\partial w_{m\ell}(k)\right)$ equals "1" only when $j = m$ and $i = \ell$; otherwise, it is "0". Thus:

$$\frac{\partial v_j(k)}{\partial w_{m\ell}(k)} = \sum_{i\in\Lambda\cup\beta} w_{ji}(k) \frac{\partial u_i(k)}{\partial w_{m\ell}(k)} + \delta_{mj} u_\ell(k)$$

where δ_{mj} is a Kronecker delta equal to "1" when $j = m$ and "0" otherwise, and:

$$\frac{\partial u_i(k)}{\partial w_{m\ell}(k)} = \begin{cases} 0, & \text{if } i \in \Lambda \\ \dfrac{\partial y_i(k)}{\partial w_{m\ell}(k)}, & \text{if } i \in \beta \end{cases}$$

Having those equations provides that:

$$\frac{\partial y_j(k+1)}{\partial w_{m\ell}(k)} = \dot{\varphi}(v_j(k)) \left[\sum_{i\in\beta} w_{ji}(k) \frac{\partial y_i(k)}{\partial w_{m\ell}(k)} + \delta_{m\ell} u_\ell(k) \right]$$

The initial state of the network at time $k = 0$ is assumed to be zero:

$$\frac{\partial y_i(0)}{\partial w_{m\ell}(0)} = 0 \text{, for } \{j \in \text{ß}, m \in \text{ß}, \ell \in \Lambda \cup \beta\}.$$

The dynamical system is described by the following triply indexed set of variables ($\pi_{m\ell}^{j}$):

$$\pi_{m\ell}^{j}(k) = \frac{\partial y_j(k)}{\partial w_{m\ell}(k)}$$

For every time step k and all appropriate j, m and ℓ, system dynamics are controlled by:

$$\pi_{m\ell}^{j}(k+1) = \dot{\varphi}(v_j(k)) \left[\sum_{i\in\beta} w_{ji}(k) \pi_{m\ell}^{i}(k) + \delta_{mj} u_\ell(k) \right],$$

with $\pi_{m\ell}^{j}(0) = 0$.

The values of $\pi_{m\ell}^{j}(k)$ and the error signal $e_j(k)$ are used to compute the corresponding weight changes:

$$\Delta w_{m\ell}(k) = \eta \sum_{j\in\varsigma} e_j(k) \pi_{m\ell}^{j}(k) \tag{39}$$

Using the weight changes, the updated weight $w_{m\ell}(k+1)$ is calculated as follows:

$$w_{m\ell}(k+1) = w_{m\ell}(k) + \Delta w_{m\ell}(k) \tag{40}$$

Repeating this computation procedure provides the minimization of the cost function and the objective is achieved.

Example 1. The recurrent neural network was implemented to estimate the states and [$\mathbf{A_d}$] and [$\mathbf{B_d}$] matrices of the propulsion part of an advanced airplane 9th order model [2]. Only the 6th, 7th, 8th, and 9th states of the complete model were considered in this example. The [$\mathbf{A_d}$] and [$\mathbf{B_d}$] matrices were given by:

$$[\mathbf{A_d}] = \begin{bmatrix} -4.191 & 6.022 & -343.4 & 11.60 \\ 0.4263 & -5.707 & 27.16 & 10.40 \\ 0.2295 & 0.1155 & -90.24 & 0.8476 \\ 0.03740 & -0.1036 & -7.954 & -1.068 \end{bmatrix}$$

$$[\mathbf{B_d}] = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -43.02 & -25.83 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

This part is a fourth-order subsystem with four states to be estimated. The system was simulated for a specific command input and its corresponding output data were recorded. The generated pair of input-output data was used by the recurrent neural network to estimate the states and the [$\mathbf{A_d}$] and [$\mathbf{B_d}$] matrices of the propulsion airplane model to provide the estimated matrices within the trained weight matrix $W = [[\tilde{\mathbf{A}}_\mathbf{d}] \;\; [\tilde{\mathbf{B}}_\mathbf{d}]]$. Results of testing the estimated propulsion model showed very close state responses to the true system states. This is shown in Figure 19.

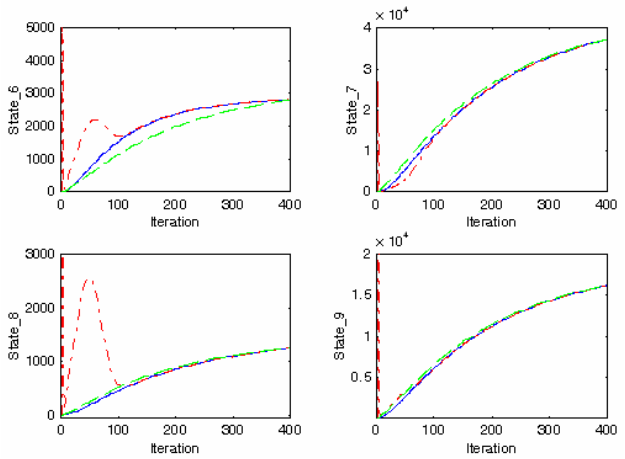

Figure 19. System state response of the original, trained, and estimated system: thick blue line: true response, thin red line: response while training, light green line: estimated response.

With the many advantages that the neural network has, it is used for parameter estimation in model transformation for the purpose of order model reduction as will be shown in the following section.

2.3 Linear Matrix Inequality (LMI) and Model Transformation

In this section, the detailed illustration of system transformation using LMI optimization will be presented. Consider the system:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{41}$$
$$y(t) = Cx(t) + Eu(t) \tag{42}$$

The state space system representation of Equations (41) and (42) may be described by the block diagram shown in Figure 20.

In order to determine the transformed [**A**] matrix, which is [$\tilde{\mathbf{A}}$], the discrete zero input response is obtained. This is achieved by providing the system with some initial state values and setting the system input to zero ($u(k) = 0$). Hence, the discrete system of Equations (41) and (42), with the initial condition $x(0) = x_0$, becomes:

$$x(k+1) = A_d x(k) \tag{43}$$
$$y(k) = x(k) \tag{44}$$

We need $x(k)$ as a neural network target to train the network to obtain the needed parameters in [$\tilde{\mathbf{A}}_\mathbf{d}$] such that the system output will be the same for [$\mathbf{A_d}$] and [$\tilde{\mathbf{A}}_\mathbf{d}$]. Hence, simulating this system provides the state response corresponding to their initial values with only the [$\mathbf{A_d}$] matrix is being used.

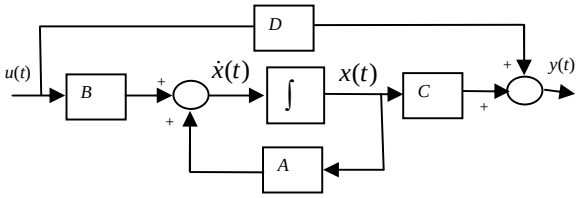

Figure 20. State space block diagram.

Once the input-output data is obtained, transforming the [$\mathbf{A_d}$] matrix is done using the NN training, as explained in Section 3. The estimated transformed [$\mathbf{A_d}$] matrix is then converted back into the continuous form which yields:

$$\tilde{A} = \begin{bmatrix} A_r & A_c \\ 0 & A_o \end{bmatrix} \tag{45}$$

Having the [**A**] and [$\tilde{\mathbf{A}}$] matrices, the permutation [**P**] matrix is determined using the LMI optimization technique, as will be illustrated in later sections. The complete system transformation can be achieved as follows: assuming that $\tilde{x} = P^{-1}x$, the system of Equations (41) and (42) can be re-written as:

$$P\dot{\tilde{x}}(t) = AP\tilde{x}(t) + Bu(t),$$
$$\tilde{y}(t) = CP\tilde{x}(t) + Eu(t), \text{ where: } (\tilde{y}(t) = y(t)).$$

Pre-multiplying the first equation above by [$\mathbf{P^{-1}}$], we obtain:

$$P^{-1}P\dot{\tilde{x}}(t) = P^{-1}AP\tilde{x}(t) + P^{-1}Bu(t),$$
$$\tilde{y}(t) = CP\tilde{x}(t) + Eu(t)$$

which yields the following transformed model:

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t) \tag{46}$$
$$\tilde{y}(t) = \tilde{C}\tilde{x}(t) + \tilde{E}u(t) \tag{47}$$

where the transformed system matrices are given by:

$$\tilde{A} = P^{-1}AP \tag{48}$$
$$\tilde{B} = P^{-1}B \tag{49}$$
$$\tilde{C} = CP \tag{50}$$
$$\tilde{E} = E \tag{51}$$

Transforming the system matrix [**A**] into the form shown in Equation (45) can be achieved based on the following definition [18].

Definition: Matrix $A \in M_n$ is called reducible if either:
(a) $n = 1$ and $A = 0$; or
(b) $n \geq 2$, there is a permutation matrix $P \in M_n$, and there is some integer r with $1 \leq r \leq n-1$ such that:

$$P^{-1}AP = \begin{bmatrix} X & Y \\ \mathbf{0} & Z \end{bmatrix} \tag{52}$$

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. However, the permutation matrix structure narrows the applicability of this method to a very limited category of applications. Some form of a similarity transformation maybe used to correct this problem; $f : R^{n\times n} \to R^{n\times n}$, where f is a linear operator defined by $f(A) = P^{-1}AP$ [18]. Hence, based on the [**A**] and [$\tilde{\mathbf{A}}$], Linear Matrix Inequalities (LMI) are used to obtain the transformation matrix [**P**]. The optimization problem is casted as follows:

$$\min_{P} \left\| P - P_o \right\| \quad Subject\ to \quad \left\| P^{-1}AP - \tilde{A} \right\| < \varepsilon \tag{53}$$

which maybe written in an LMI equivalent form as:

$$\begin{aligned} \min_{S}\ trace(S)\ \ Subject\ to & \begin{bmatrix} S & P - P_0 \\ (P - P_0)^T & I \end{bmatrix} > 0 \\ & \begin{bmatrix} \varepsilon_1^2 I & P^{-1}AP - \tilde{A} \\ (P^{-1}AP - \tilde{A})^T & I \end{bmatrix} > 0 \end{aligned} \tag{54}$$

where *S* is a symmetric slack matrix [6].

2.4 Order Model Reduction

Linear time-invariant models of many physical systems have fast and slow dynamics, which may be referred to as singularly perturbed systems [19]. Neglecting the fast dynamics of a singularly perturbed system provides a reduced slow model. This gives the advantage of designing simpler lower-dimensionality reduced order controllers based on the reduced model information.

To show the formulation of a reduced order system model, consider the singularly perturbed system [2]:

$$\dot{x}(t) = A_{11}x(t) + A_{12}\xi(t) + B_1u(t), \quad x(0) = x_0 \tag{55}$$

$$\varepsilon\dot{\xi}(t) = A_{21}x(t) + A_{22}\xi(t) + B_2u(t), \quad \xi(0) = \xi_0 \tag{56}$$

$$y(t) = C_1x(t) + C_2\xi(t) \tag{57}$$

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, { [$\mathbf{A_{ii}}$], [$\mathbf{B_i}$], [$\mathbf{C_i}$]} are constant matrices of appropriate dimensions with $i \in \{1,2\}$, and ε is a small positive constant. The singularly perturbed system in Equations (55)-(57) is simplified by setting $\varepsilon = 0$ [3,14]. In doing so, we are neglecting the fast dynamics of the system and assuming that the state variables ξ have reached the quasi-steady state. Hence, setting $\varepsilon = 0$ in Equation (56), with the assumption that [$\mathbf{A_{22}}$] is nonsingular, produces:

$$\xi(t) = -A_{22}^{-1}A_{21}x_r(t) - A_{22}^{-1}B_1u(t) \tag{58}$$

where the index *r* denotes remained or reduced model. Substituting Equations (58) in Equations (55)-(57) yields the reduced order model:

$$\dot{x}_r(t) = A_rx_r(t) + B_ru(t) \tag{59}$$

$$y(t) = C_rx_r(t) + E_ru(t) \tag{60}$$

where:

$$A_r = A_{11} - A_{12}A_{22}^{-1}A_{21} \tag{61}$$

$$B_r = B_1 - A_{12}A_{22}^{-1}B_2 \tag{62}$$

$$C_r = C_1 - C_2A_{22}^{-1}A_{21} \tag{63}$$

$$E_r = -C_2A_{22}^{-1}B_2 \tag{64}$$

Example 2. Consider the following first order system:

$$\dot{x}(t) = \begin{bmatrix} -30 & 15 \\ 8 & -40 \end{bmatrix}x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix}u(t), \quad y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix}x(t)$$

Since the system is a 2nd order, there are two eigenvalues which are {-22.9584, -47.0416}. As seen from the eigenvalues, since there are two distinct categories (fast and slow) with big difference between them, the singular perturbation reduction may be applied. The reduced 1st order model is obtained as:

$$\dot{x}_r(t) = -27x_r(t) + 1.375u(t), \quad y_r(t) = 1.2x_r(t) + 0.025u(t)$$

System output response plots of simulating this reduced order model along with simulating the original system model to a step input are shown in Figure 21. As seen in the figure, the singular perturbation has provided an acceptable response compared with the original response.

Example 3. Consider the following 3rd order system:

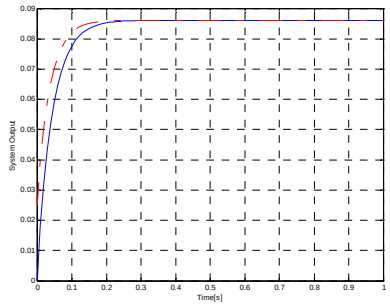

Figure 21. Output step response of the original and reduced order models (___ original model, -.-.-. reduced model).

$$\dot{x}(t)=\begin{bmatrix}-20 & 5 & -18\\ 8 & -30 & 4\\ 2 & 5 & -40\end{bmatrix}x(t)+\begin{bmatrix}1\\ 1\\ 0.5\end{bmatrix}u(t),$$
$$y(t)=\begin{bmatrix}1 & 0.2 & 1\end{bmatrix}x(t)$$

Since the system is a 3rd order, there are three eigenvalues which are {-25.2822, -22, -42.717}. Using the singular perturbation technique, the system model is reduced to the following 2nd order model:

$$\dot{x}_r(t)=\begin{bmatrix}-20.9 & 2.75\\ 8.2 & -29.5\end{bmatrix}x_r(t)+\begin{bmatrix}0.775\\ 1.05\end{bmatrix}u(t)$$
$$y_r(t)=\begin{bmatrix}1.05 & 0.325\end{bmatrix}x_r(t)\quad+\begin{bmatrix}0.0125\end{bmatrix}u(t)$$

System output response plots of the original system model and the reduced model, for a step input, are shown in Figure 22. It is seen that the reduced order model is performing very well as compared with the original system response.

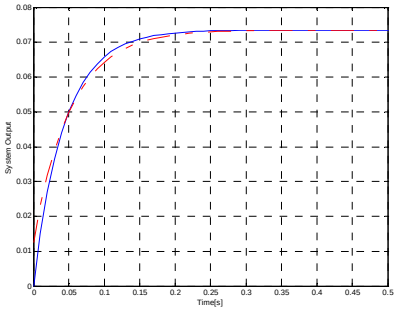

Figure 22. Output step response of the original and reduced order models (___ original model, -.-.-. reduced model).

Example 4. Consider the 5th order RLC filter shown in Figure 23 [16]. It is well known that the capacitor and the inductor are dynamical passive elements, which means that they have the ability to store energy. The dynamical equations may be derived using the Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL) [16]. It is well known that the current for the capacitor is proportional to the change of its voltage, that is:

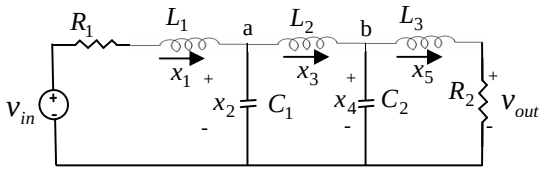

Figure 23. A 5th order RLC-based network (circuit).

$$i_{c_i}(t)=C_i\frac{dv_{c_i}(t)}{dt}$$

and that the voltage across the inductor is proportional to the change of its current, that is:

$$v_{L_i}(t)=L_i\frac{di_{L_i}(t)}{dt}$$

In order to obtain a state space model for the above system, let the dynamics of the system be designated as system states (x_i). This means that there will be a 5th order system since there are five dynamical elements in the system. The model can be obtained by assigning the following states: x_1: Current of the inductor L_1, x_2: Voltage of the capacitor C_1, x_3: Current of the inductor L_2, x_4: Voltage of the capacitor C_2, and x_5: Current of the inductor L_3.

Applying KCL at nodes (a) and (b) and KVL for the three loops starting from left to right of Figure 23 yields the following differential equations:

$$x_1(t)=C_1\frac{dx_2(t)}{dt}+x_3(t),\quad x_3(t)=C_2\frac{dx_4(t)}{dt}+x_5(t),$$
$$-v_{in}(t)+R_1x_1(t)-L_1\frac{dx_1(t)}{dt}+x_2(t)=0,$$
$$-x_2(t)+L_2\frac{dx_3(t)}{dt}+x_4(t)=0,$$
$$-x_4(t)+L_3\frac{dx_5(t)}{dt}+R_2x_5(t)=0$$

Letting $\frac{dx_i}{dt}\equiv\dot{x}_i$ and re-arranging the above equations, the following may be obtained:

$$\dot{x}_1(t)=\frac{-R_1}{L_1}x_1(t)-\frac{1}{L_1}x_2(t)+\frac{1}{L_1}v_{in}(t),$$
$$\dot{x}_2(t)=\frac{1}{C_1}x_1(t)-\frac{1}{C_1}x_3(t)$$
$$\dot{x}_3(t)=\frac{1}{L_2}x_2(t)+\frac{1}{L_2}x_4(t),$$
$$\dot{x}_4(t)=\frac{1}{C_2}x_3(t)-\frac{1}{C_2}x_5(t),$$
$$\dot{x}_5(t)=\frac{1}{L_3}x_4(t)-\frac{R_2}{L_3}x_5(t)$$

Hence, the system may be given by:

$$\dot{x}(t) = Ax(t) + Bu(t)\,,\;\; y(t) = Cx(t) + Eu(t)$$

where:

$$A = \begin{bmatrix} -\frac{R_1}{L_1} & -\frac{1}{L_1} & 0 & 0 & 0 \\ \frac{1}{C_1} & 0 & -\frac{1}{C_1} & 0 & 0 \\ 0 & \frac{1}{L_2} & 0 & -\frac{1}{L_2} & 0 \\ 0 & 0 & \frac{1}{C_2} & 0 & -\frac{1}{C_2} \\ 0 & 0 & 0 & \frac{1}{L_3} & -\frac{R_2}{L_3} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$C = [0 \;\; 0 \;\; 0 \;\; 0 \;\; R_2]\,,\; E = [0]\,.$$

Given the following values: $C_1 = C_2 = 2.57$ nF, $L_1 = L_3 = 9.82\,\mu$H , $L_2 = 31.8\,\mu$H , $R_1 = R_2 = 100\,\Omega$, the corresponding 5th order model is obtained. The eigenvalues of the system are found to be: $10^6 \times (-1.9445 \pm j5.9863, -6.2839, -5.0865 \pm j3.7029)$. Performing model reduction, the system is reduced from its 5th order to a 4th order by taking the first four rows of [**A**] as the first category represented by Equation (55) and taking the fifth row of [**A**] as the second category represented by Equation (56). Simulations of both, the original and the reduced models, are shown in Figure 24.

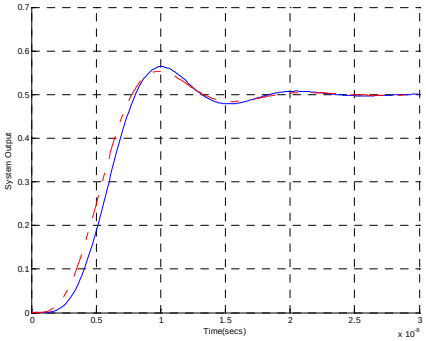

Figure 24. System output step response of the original and reduced order models (___ original model, -.-.-reduced model).

As seen in Figure 24, the reduced order model (using the singular perturbation method) has provided an acceptable response compared with the original system response.

3. Neural Network Estimation With LMI Optimization For The Buck Model Reduction

In this work, it is our objective to search for a similarity transformation that can be used to decouple a pre-selected eigenvalue set from the system matrix [**A**]. To achieve this objective, training the neural network to estimate the transformed discrete system matrix [$\tilde{\mathbf{A}}_\mathbf{d}$] is performed [15]. For the system of Equations (55)-(57), the discrete model of the Buck converter is obtained as:

$$x(k+1) = A_d x(k) + B_d u(k) \quad (65)$$

$$y(k) = C_d x(k) + E_d u(k) \quad (66)$$

The estimated discrete model of Equations (65)-(66) can be written in a detailed form (as shown in Figure 18) as:

$$\begin{bmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} u(k) \quad (67)$$

$$\tilde{y}(k) = \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} \quad (68)$$

where k is the time index. The detailed matrix elements of Equation (67)-(68) are shown in Figure 18 in the previous section.

The recurrent neural network presented in Section 2.2 can be summarized by defining Λ as the set of indices i for which $g_i(k)$ is an external input, which in the Buck converter system is one external input and by defining $ß$ as the set of indices i for which $y_i(k)$ is an internal input or a neuron output, which in the Buck converter system is two internal inputs (two system states). Also, by defining $u_i(k)$ as the combination of the internal and external inputs for which $i \in ß \cup \Lambda$. Using this setting, training the network depends on the internal activity of each neuron which is given by:

$$v_j(k) = \sum_{i \in \Lambda \cup \beta} w_{ji}(k) u_i(k) \quad (69)$$

where w_{ji} is the weight representing an element in the system matrix or input matrix for $j \in ß$ and $i \in ß \cup \Lambda$ such that $W = \left[[\tilde{\mathbf{A}}_\mathbf{d}] \;\; [\tilde{\mathbf{B}}_\mathbf{d}] \right]$. At the next time step (k +1), the output (internal input) of the neuron j is computed by passing the activity through the nonlinearity $\varphi(.)$ as follows:

$$x_j(k+1) = \varphi(v_j(k)) \quad (70)$$

With these equations, based on an approximation of the method of steepest descent, the network estimates the system matrix [$\mathbf{A_d}$] as illustrated in Equation (43) for zero input response [15]. That is, an error can be obtained by matching a true state output with a neuron output as follows:

$$e_j(k) = x_j(k) - \tilde{x}_j(k)$$

Now, the objective is to minimize the cost function given by:

$$E_{\text{total}} = \sum_{k} E(k), \text{ where } \quad E(k) = \tfrac{1}{2}\sum_{j\in\varsigma} e_j^2(k)$$

where ς denotes the set of indices j for the output of the neuron structure. This cost function is minimized by estimating the instantaneous gradient of $E(k)$ with respect to the weight matrix [**W**] and then updating [**W**] in the negative direction of this gradient [15,28]. In steps, this may be proceeded as follows:

- Initialize the weights, [**W**], by a set of uniformly distributed random numbers. Starting at the instant k = 0, use Equations (69) and (70) to compute the output values of the N neurons (where $N = ß$).

- For every time step k and all $j \in ß$, $m \in ß$, and $\ell \in ß \cup \Lambda$, compute the dynamics of the system which are governed by the triply indexed set of variables:

$$\pi_{m\ell}^{j}(k+1) = \dot{\varphi}(v_j(k))\left[\sum_{i\in ß} w_{ji}(k)\pi_{m\ell}^{i}(k) + \delta_{mj}u_\ell(k)\right]$$

with initial conditions $\pi_{m\ell}^{j}(0) = 0$ and δ_{mj} is given by $\left(\partial w_{ji}(k)/\partial w_{m\ell}(k)\right)$, which 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: $\dot{\varphi}(v_j(k)) = y_j(k+1)[1 - y_j(k+1)]$.

- Compute the weight changes corresponding to the error signal and system dynamics:

$$\Delta w_{m\ell}(k) = \eta \sum_{j\in\varsigma} e_j(k)\pi_{m\ell}^{j}(k) \tag{71}$$

- Update the weights in accordance with:

$$w_{m\ell}(k+1) = w_{m\ell}(k) + \Delta w_{m\ell}(k) \tag{72}$$

- Repeat the computation until the desired estimation is achieved.

As illustrated in Equations (43) and (44), for the purpose of estimating only the transformed system matrix [$\tilde{\mathbf{A}}$], the training is based on the zero input response. Once the training is complete, the obtained weight matrix [**W**] is the discrete estimated transformed system matrix. Transforming the estimated system back to the continuous form yields the desired continuous transformed system matrix [$\tilde{\mathbf{A}}$]. Using the LMI optimization technique illustrated in Section 2.3, the permutation matrix [**P**] is determined. Hence, a complete system transformation, as shown in Equations (46) and (47), is achieved.

To perform order model reduction, the system in Equations (46) and (47) are written as:

$$\begin{bmatrix}\dot{\tilde{x}}_r(t)\\ \dot{\tilde{x}}_o(t)\end{bmatrix} = \begin{bmatrix}A_r & A_c\\ 0 & A_o\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ \tilde{x}_o(t)\end{bmatrix} + \begin{bmatrix}B_r\\ B_o\end{bmatrix}u(t) \tag{73}$$

$$\begin{bmatrix}\tilde{y}_r(t)\\ \tilde{y}_o(t)\end{bmatrix} = \begin{bmatrix}C_r & C_o\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ \tilde{x}_o(t)\end{bmatrix} + \begin{bmatrix}E_r\\ E_o\end{bmatrix}u(t) \tag{74}$$

The following system transformation enables us to decouple the original system into retained (r) and omitted (o) eigenvalues based on the subsystems as shown in Figure 25. The retained eigenvalues are the dominant eigenvalues that produce the slow dynamics and the omitted eigenvalues are the non-dominant eigenvalues that produce the fast dynamics.

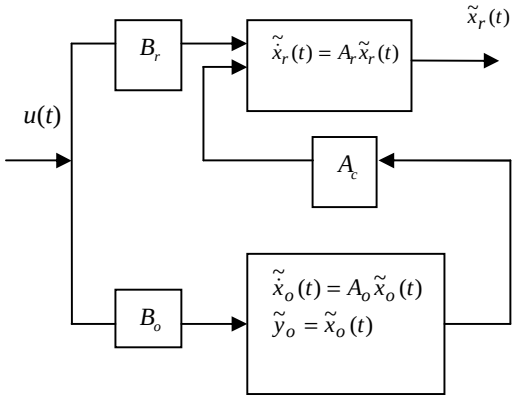

Figure 25. System decoupling process of the fast and slow eigenvalues.

Equation (73) maybe written as:

$$\dot{\tilde{x}}_r(t) = A_r\tilde{x}_r(t) + A_c\tilde{x}_o(t) + B_r u(t)$$
$$\dot{\tilde{x}}_o(t) = A_o\tilde{x}_o(t) + B_o u(t)$$

The coupling term $A_c\tilde{x}_o(t)$ maybe compensated for by solving for $\tilde{x}_o(t)$ in the second equation above by setting $\dot{\tilde{x}}_o(t)$ to zero using the singular perturbation method (by setting $\varepsilon = 0$). Doing so, the following is obtained:

$$\tilde{x}_o(t) = -A_o^{-1}B_o u(t) \tag{75}$$

Using $\tilde{x}_o(t)$, we get the reduced order model given by:

$$\dot{\tilde{x}}_r(t) = A_r\tilde{x}_r(t) + [-A_cA_o^{-1}B_o + B_r]u(t) \tag{76}$$
$$y(t) = C_r\tilde{x}_r(t) + [-C_oA_o^{-1}B_o + E]u(t) \tag{77}$$

Hence, the overall reduced order model maybe represented by:

$$\dot{\tilde{x}}_r(t) = A_{or}\tilde{x}_r(t) + B_{or}u(t) \quad (78)$$

$$y(t) = C_{or}\tilde{x}_r(t) + E_{or}u(t) \quad (79)$$

where the detail of the {[$\mathbf{A_{or}}$], [$\mathbf{B_{or}}$], [$\mathbf{C_{or}}$], [$\mathbf{E_{or}}$]} overall reduced matrices are shown in Equations (76) and (77).

4. Model Reduction Of The Buck Converter Using Neural Estimation And LMI Optimization

Investigating the proposed method of system modeling for the Buck converter using neural network with LMI and order model reduction, the input-to-output and control-to-output systems were tested on a PC platform with hardware specifications of Intel Pentium 4 CPU 2.40 GHz and 504 MB of RAM, and software specifications of MS Windows XP 2002 OS and Matlab 6.5 simulator.

4.1 Input-to-Output System

The state space model of the input-to-output system is given by the following system matrices:

$$A = \begin{bmatrix} 0 & -1/\mathrm{L} \\ 1/\mathrm{C} & -1/\mathrm{RC} \end{bmatrix} \quad (80)$$

$$B = \begin{bmatrix} \mathrm{D}/\mathrm{L} \\ 0 \end{bmatrix} \quad (81)$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (82)$$

$$E = \begin{bmatrix} 0 \end{bmatrix} \quad (83)$$

Knowing that this is a second order system, its eigenvalues should not be complex in order to perform order model reduction. As seen from the system matrix [**A**], the eigenvalues are mainly depending on the capacitor and inductor values. Therefore, different values of the capacitor and inductor were considered as it will be shown in the following examples.

As a first example, given that D = 0.4, R = 18.6 Ω, L = 5.8 H, C = 0.55 mF, the eigenvalues were found to be -3.3196 and -94.4321. Having the eigenvalue -94.4321 being much larger than the -3.3196, which shows two categories of eigenvalues, order model reduction may be performed. Thus, the system was discretized using sampling rate $\mathrm{T_s}$ = 0.0005 second. and simulated for a zero input, $\dot{x}(t) = Ax(t)$. Hence, based on the obtained simulated output data and using neural network to estimate the subsystem matrix [$\mathbf{A_c}$] of Equation (45), the following transformed system matrix [$\tilde{\mathbf{A}}$] was obtained:

$$\tilde{A} = \begin{bmatrix} A_r & A_c \\ 0 & A_o \end{bmatrix}$$

where [$\mathbf{A_r}$] is set to provide the dominant eigenvalues (slow dynamics) and [$\mathbf{A_o}$] is set to provide the non-dominant eigenvalues (fast dynamics) of the original system. It is important to notice that the system considered is a second order system. Thus, when training the system, the second state $\tilde{x}_o(t)$ of the transformed model in Equation (73) is unchanged due to the restriction of [0 A_o] seen in [$\tilde{\mathbf{A}}$]. This may lead to an undesired starting of the system response, but fast system overall convergence (as will be seen and explained in the third example). Using [$\tilde{\mathbf{A}}$] along with [**A**], the LMI is then implemented in order to obtain {[$\tilde{\mathbf{B}}$], [$\tilde{\mathbf{C}}$], [$\tilde{\mathbf{E}}$]}, which makes a complete model transformation. Finally, using the singular perturbation technique for order model reduction, the reduced order model is obtained. The input-to-output system responses to a step input of the original full order and reduced order models are seen in Figure 26.

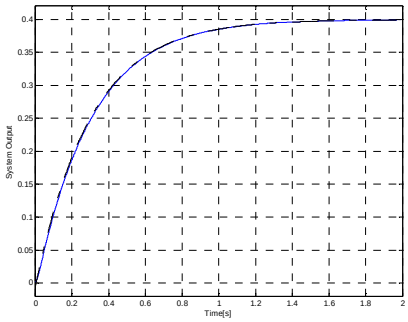

Figure 26. Input-to-output system step responses: full order system model (solid blue line), transformed reduced order model (dashed black line).

A second example is considering the values: D = 0.4, R = 18.6 Ω, L = 580 mH, C = 55 μF. The eigenvalues were found to be -33.1963 and -944.3208. Using the same procedure that was performed previously, the simulation of the full and reduced order models for a step input has generated the responses shown in Figure 27. In addition to that, the transformed reduced order model response was compared with the non-transformed reduced order model. It is seen in Figure 27 that the response of the transformed reduced order model is more accurate than the response of the non-transformed reduced order model.

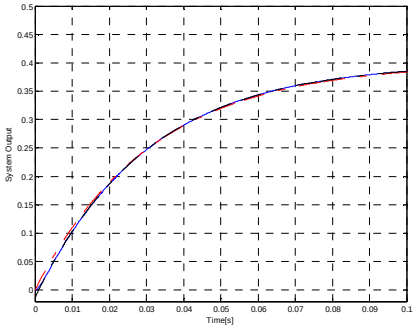

Figure 27. Input-to-output system step responses: full order system model (solid blue line), transformed reduced order model (dashed black line), and non-transformed reduced order model (dashed red line).

In a third example, different values of the capacitor, inductor, and resistor are selected such that the values of the system eigenvalues are not very high (not very fast dynamics), but still one is larger than the other. Hence, the following values were considered: D = 0.4, R = 10.1 Ω, L = 3.305 H, C = 5.5 mF. The eigenvalues were found to be -3.901 and -14.099. For a step input, simulating the original and transformed reduced order models along with the non-transformed reduced order model produced the results shown in Figure 28.

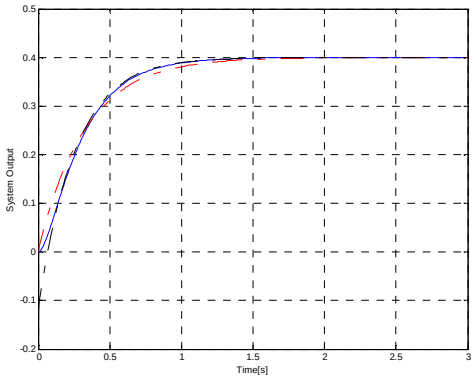

Figure 28. Input-to-output system step responses: full order system model (solid blue line), transformed reduced order model (dashed black line), and non-transformed reduced order model (dashed red line).

As seen in Figure 28, the transformed reduced order model response is starting a little off from the original system response, however, it has a faster convergence than the non-transformed reduced order model response. The cause of this is due to the construction of the output matrix C = [0 1]; given that:

$$\begin{bmatrix}\dot{\tilde{x}}_r(t)\\ \dot{\tilde{x}}_o(t)\end{bmatrix}=\begin{bmatrix}A_r & A_c\\ 0 & A_o\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ \tilde{x}_o(t)\end{bmatrix}+\begin{bmatrix}B_r\\ B_o\end{bmatrix}u(t)$$

$$\tilde{y}(t)=\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ \tilde{x}_o(t)\end{bmatrix}$$

where [$\mathbf{A_r}$] is preset to the dominant eigenvalues (slow dynamics), [$\mathbf{A_o}$] is preset to the non-dominant eigenvalues (fast dynamics), and [$\mathbf{A_c}$] is the neural-estimated sub-matrix. It is seen that $\tilde{y}(t)=\tilde{x}_o(t)$, where $\tilde{x}_o(t)$ is the solution of $\dot{\tilde{x}}_o(t)=A_o\tilde{x}_o(t)+B_o u(t)$ after setting it in the form: $\varepsilon\dot{\tilde{x}}_o(t)=\varepsilon A_o\tilde{x}_o(t)+\varepsilon B_o u(t)$ and letting $\varepsilon\dot{\tilde{x}}_o(t)=0$ using the singular perturbation method. The sub-matrix [$\mathbf{A_o}$] is set to have the fast dynamics (represented by the -14.099 eigenvalue) regardless of the system response and is independent of the neural network training. Hence, having $\tilde{y}(t)$ depending only on $\tilde{x}_o(t)$, which was independent of the training (i.e., independent of [$\mathbf{A_c}$]), makes the transformed system response less accurate. However, if the output were to depend on $\tilde{x}_r(t)$, which had the neural network training (i.e., depends on [$\mathbf{A_c}$]), such that C = [1 0], then the response would look as seen in Figure 29, which is much better and more accurate than the non-transformed reduced order model response.

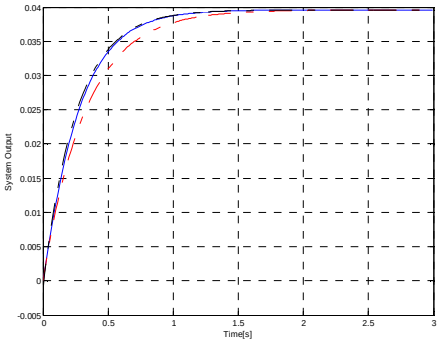

Figure 29. Input-to-output system step responses: full order system model (solid blue line), transformed reduced order model (dashed black line), and non-transformed reduced order model (dashed red line).

A fourth example that will produce complex eigenvalues and thus can not be reduced using the previously utilized singular perturbation method is as follows: if the original system element values are set to D = 0.4, R = 18.6 Ω, L = 58 μH, C = 5.5 μF, then the eigenvalues are found to be complex and given by: $(-0.4888\pm j5.5776)\times10^4$. Simulating this system to a step input produces the response shown in Figure 30. Here, since the system is a 2nd order and has complex eigenvalues, order model reduction can not be performed using the previously used singular perturbation technique.

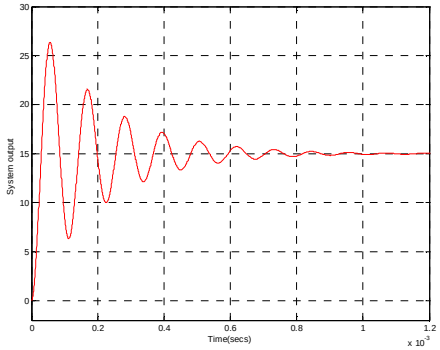

Figure 30. 2nd order (full order) original system step response.

4.2 Control-to-Output System

The state space model of the control-to-output system is given by the system matrices:

$$A=\begin{bmatrix}0 & -1/\mathrm{L}\\ 1/\mathrm{C} & -1/\mathrm{RC}\end{bmatrix} \tag{84}$$

$$B=\begin{bmatrix}\mathrm{V_g}/\mathrm{L}\\ 0\end{bmatrix} \tag{85}$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (86)$$

$$E = \begin{bmatrix} 0 \end{bmatrix} \quad (87)$$

As seen in the Equations (84)-(87), the system matrix [**A**] of the control-to-output state space model is the same as the input-to-output system. The only difference is that the input matrix [**B**] has changed to depend on the element V_g instead of D. Hence, the eigenvalues will be the same and the response will be of the same type as the input-to-output system. For example, considering the elements of the system model given by: V_g = 15 V, R = 18.6 Ω, L = 58 mH, C = 5.5 μF, the system output step response is shown in Figure 31. As previously seen, the transformed reduced model response has a faster convergence than the response of the reduced model without system transformation.

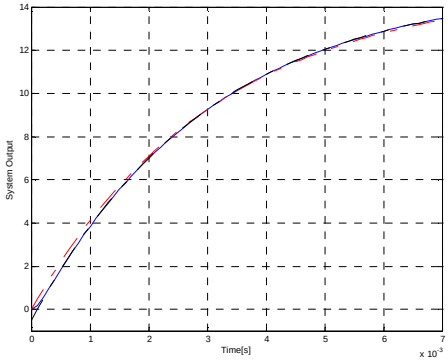

Figure 31. Control-to-output system step responses: full order system model (solid blue line), transformed reduced order model (dashed black line), and non-transformed reduced order model (dashed red line).

5. The Application Of State Feedback Controller On The Reduced Order Model Of The Buck Converter

We can apply many control techniques such as H_∞ control, robust control, stochastic control, intelligent control, etc. on the reduced order model to meet given specifications. Yet, in this paper, since the Buck system is a 2nd order system reduced to a 1st order, we will investigate system stability and enhancing performance by considering the s-domain pole replacement.

For the reduced order model in the system of Equations (78) and (79), a state feedback controller can be designed. For example, assuming that a controller is needed to provide the system with faster dynamical response, this can be achieved by replacing the system eigenvalues with new faster eigenvalues. Hence, let the control input be given by:

$$u(t) = -K\tilde{x}_r(t) + r(t) \quad (88)$$

where K is to be designed based on the desired system eigenvalues. State feedback control for the transformed reduced order model is illustrated in Figure 32.

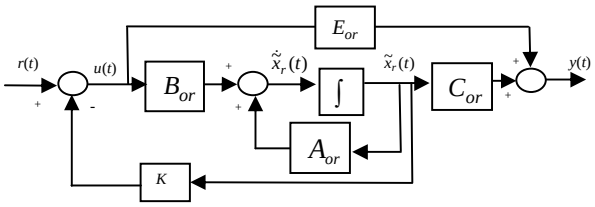

Figure 32. Block diagram of a state feedback control with {[$\mathbf{A_{or}}$], [$\mathbf{B_{or}}$], [$\mathbf{C_{or}}$], [$\mathbf{E_{or}}$]} overall reduced order system matrices.

Replacing the control input $u(t)$ in Equations (78) and (79) by the above new control input in Equation (88) yields the following reduced system equations:

$$\dot{\tilde{x}}_r(t) = A_{or}\tilde{x}_r(t) + B_{or}[-K\tilde{x}_r(t) + r(t)] \quad (89)$$

$$y(t) = C_{or}\tilde{x}_r(t) + E_{or}[-K\tilde{x}_r(t) + r(t)] \quad (90)$$

which can be re-written as:

$$\dot{\tilde{x}}_r(t) = A_{or}\tilde{x}_r(t) - B_{or}K\tilde{x}_r(t) + B_{or}r(t)$$

$$\rightarrow \dot{\tilde{x}}_r(t) = [A_{or} - B_{or}K]\tilde{x}_r(t) + B_{or}r(t)$$

$$y(t) = C_{or}\tilde{x}_r(t) - E_{or}K\tilde{x}_r(t) + E_{or}r(t)$$

$$\rightarrow \quad y(t) = [C_{or} - E_{or}K]\tilde{x}_r(t) + E_{or}r(t)$$

The overall closed-loop system model may then be written as:

$$\dot{\tilde{x}}(t) = A_{cl}\tilde{x}_r(t) + B_{cl}r(t) \quad (91)$$

$$y(t) = C_{cl}\tilde{x}_r(t) + E_{cl}r(t) \quad (92)$$

such that the closed loop system matrix [$\mathbf{A_{cl}}$] will provide the new desired system eigenvalues.

Example 5. Consider the input-to-output system presented in Section 4, for the case where the eigenvalues were -3.901 and -14.099. Using the new transformation-based reduction technique, one obtains a reduced order model given by:

$$\dot{\tilde{x}}_r(t) = [-3.901]\tilde{x}_r(t) + [-5.8051]u(t)$$

$$y_r(t) = [-0.3503]\tilde{x}_r(t) + [-0.1212]u(t)$$

with the eigenvalue of -3.901. Now, suppose that a new eigenvalue λ = -9 that will produce faster system dynamics is desired for this reduced order model. This objective can be achieved by first setting the desired characteristic equation as follows:

$$\lambda + 9 = 0$$

To determine the feedback control gain K, the characteristic equation of the closed-loop system is

needed. This can be achieved using Equations (89) through (91) which yields:

$$(\lambda \mathbf{I} - A_{cl}) = 0 \qquad \rightarrow \qquad \lambda \mathbf{I} - [A_{or} - B_{or} K] = 0$$

Knowing that A_{or} = -3.901 and B_{or} = -5.805, the closed-loop characteristic equation can be compared with the desired characteristic equation. Doing so, the feedback gain K is found to be -0.8784. Hence, the closed-loop system now has the eigenvalue of -9. As stated previously, the objective of replacing eigenvalues is to enhance system performance. Simulating the reduced order model with the new eigenvalue for the same original system input (the step input) has generated the response shown in Figure 33.

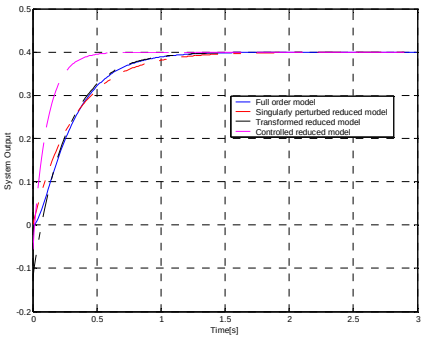

Figure 33. Enhanced system step responses based on pole placement; full order system model (solid blue line), transformed reduced order model (dashed black line), non-transformed reduced order model (dashed red line), and the controlled transformed reduced order (dashed pink line).

As it is seen in the figure, the new normalized system response is faster (enhanced) than the system response obtained without pole placement. That is, the settling time in the reduced controlled system response is about 0.4 seconds while in the uncontrolled system response is about 1.3 seconds. This shows that even simple state feedback control using the transformation-based reduced order system model can achieve the equivalent system performance enhancement that may be obtained using more complex and expensive control on the original full order system.

6. Conclusions And Future Work

A new method of intelligent control for the Buck converter using a newly developed small signal modeling of the pulse width modulation (PWM) switching is introduced in this paper. In order to achieve an intelligent control, the second order Buck system was simplified by reducing it to a first order system. This reduction was achieved by the implementation of a recurrent supervised neural network to estimate certain elements [$\mathbf{A_c}$] of the transformed system matrix [$\tilde{\mathbf{A}}$], while the other elements [$\mathbf{A_r}$] and [$\mathbf{A_o}$] are set based on the system eigenvalues such that [$\mathbf{A_r}$] contains the dominant eigenvalues (slow dynamics) and [$\mathbf{A_o}$] contains the non-dominant eigenvalues (fast dynamics). To obtain the transformed matrix [$\tilde{\mathbf{A}}$], the zero input response was used in order to obtain output data related to the state dynamics, based only on the system matrix [**A**]. After the transformed system matrix was obtained, the robust control algorithm of linear matrix inequality (LMI) optimization technique was used to determine the permutation matrix [**P**], which is required to complete system transformation matrices {[$\tilde{\mathbf{B}}$], [$\tilde{\mathbf{C}}$], [$\tilde{\mathbf{E}}$]}. The reduction process was then performed using the singular perturbation method, which operates on neglecting the faster-dynamics eigenvalues and leaving the dominant slow-dynamics eigenvalues to control the system. Simple state feedback control using pole placement was then applied on the reduced Buck model to obtain the desired Buck system response.

It is also shown in this paper that the eigenvalues of the resulting transformed reduced model are a subset of the original non-transformed full-order system, and this is important since the eigenvalues in the non-transformed reduced order model will be different from the eigenvalues of the original full-order system.

Future work will investigate the implementation of the introduced control methodology upon other converter systems such as the boost converter. Future work will also investigate the application of the introduced hierarchical control methodology used in this paper for the complex-quadruple transfer functions of control-to-input current transfer function and input impedance transfer function for the new small signal model of the Buck converter.

References

[1] A. N. Al-Rabadi, *An Approach to Exact Modeling of the PWM Switch*, M.Sc. Thesis, Electrical and Computer Engineering Department, Portland State University, 1998.

[2] O. M.K. Alsmadi, *Design of an Intelligent Neuro-Estimator for Complex Dynamic Systems*, M.Sc. Thesis, Electrical and Computer Engineering Department, Tennessee State University, 1995.

[3] P. Avitabile, J. C. O'Callahan, and J. Milani, "Comparison of System Characteristics using Various Model Reduction Techniques," 7th *International Model Analysis Conference*, Las Vegas, Nevada, February 1989.

[4] P. Benner, "Model Reduction at ICIAM'07," *SIAM News*, Volume 40, Number 8, October 2007.

[5] A. Bilbao-Guillerna, M. De La Sen, S. Alonso-Quesada, and A. Ibeas, "Artificial Intelligence Tools for Discrete Multiestimation Adaptive Control Scheme with Model Reduction Issues," *Proceedings of the International Association of Science and Technology*, Artificial Intelligence and Application, Innsbruck, Austria, 2004.

[6] S. Boyd, L. El-Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Society for Industrial and Applied Mathematics, 1994.

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

[8] J. H. Chow and Peter V. Kokotovic, "A Decomposition of

Near-Optimal Regulators for Systems with Slow and Fast Modes," *IEEE Transactions on Automatic Control*, Vol. AC-21, pp. 701-705, October 1976.
[9] R. W. Erickson, *Fundamentals of Power Electronics*, Chapman and Hall, 1997.
[10] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd Edition, Addison-Wesley, 1994.
[11] K. Gallivan, A. Vandendorpe, and P. Van Dooren, "Model Reduction of MIMO System via Tangential Interpolation," *SIAM Journal of Matrix Analysis and Applications*, Vol. 26, No. 2, pp. 328-349, 2004.
[12] K. Gallivan, A. Vandendorpe, and P. Van Dooren, "Sylvester Equation and Projection-Based Model Reduction," *Journal of Computational and Applied Mathematics*, 162, pp. 213-229, 2004.
[13] G. Garsia, J. Dfouz, and J. Benussou, "H_2 Guaranteed Cost Control for Singularly Perturbed Uncertain Systems," *IEEE Transactions on Automatic Control*, Vol. 43, pp. 1323-1329, September 1998.
[14] R. J. Guyan, "Reduction of Stiffness and Mass Matrices," *AIAA Journal*, Vol. 6, No. 7, pp. 1313-1319, 1968.
[15] S. Haykin, *Neural Networks*: *a Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.
[16] W. H. Hayt, J. E. Kemmerly, and S. M. Durbin, *Engineering Circuit Analysis*, McGraw Hill, 2007.
[17] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, pp. 504-507, 2006.
[18] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge, 1985.
[19] S. H. Javid, "Observing the Slow States of a Singularly Perturbed Systems," *IEEE Transactions on Automatic Control*, Vol. Ac-25, pp. 277-280, April 1980.
[20] H. K. Khalil, "Output Feedback Control of Linear Two Time Scale Systems," *IEEE Transactions on Automatic Control*, AC-32, pp. 784-792, 1987.
[21] H. K. Khalil and P. V. Kokotovic, "Control Strategies for Decision Makers Using Different Models of the Same System," *IEEE Transactions on Automatic Control*, Vol. AC-23, pp. 289-297, April 1978.
[22] P. Kokotovic, R. O'Malley, and P. Sannuti, "Singular Perturbation and Order Reduction in Control Theory – An Overview," *Automatica*, 12(2), pp. 123-132, 1976.
[23] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics, 2000.
[24] K. Ogata, *Discrete-Time Control Systems*, 2nd Edition, Prentice Hall, 1995.
[25] R. Skelton, M. Oliveira, and J. Han, *Systems Modeling and Model Reduction*, invited chapter of the Handbook of Smart Systems and Materials, Institute of Physics (IOP), 2004.
[26] M. Steinbuch, "Model Reduction for Linear Systems," 1st *International MACSI-net workshop on model reduction, VVV Eindhoven*, the Netherland, October 2001.
[27] A. N. Tikhonov, "On the Dependence of the Solution of Differential Equation on a Small Parameter," *Mat Sbornik* (*Moscow*), pp. 193-204, 1948.
[28] R. J. Williams and Zipser, "A Learning Algorithm for Continually Running Full Recurrent Neural Networks," *Neural Computation*, 1(2), pp. 270-280, 1989.
[29] J. M. Zurada, *Artificial Neural Systems*, West Publishing Company, New York, 1992.