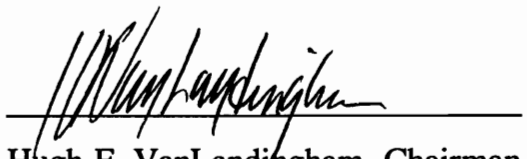
Nonlinear System Identification and Control Using a Neural Network Approach

by

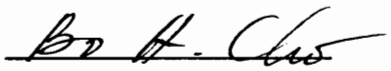
Ju-Yeop Choi

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

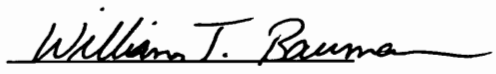
APPROVED:



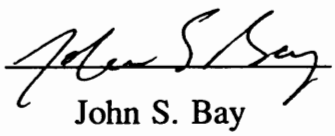
Hugh F. VanLandingham, Chairman



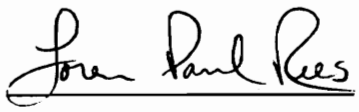
Bo H. Cho



William T. Baumann



John S. Bay



Loren P. Rees

August, 1994
Blacksburg, Virginia

Nonlinear System Identification and Control

Using a Neural Network Approach

by

Ju-Yeop Choi

Hugh F. VanLandingham, Chairman

Electrical Engineering

(ABSTRACT)

In this thesis, the plant identification, state estimation based on the identified plant and also the design of a neuro-controller using multi-layer perceptrons (MLPs) for a complex system are presented. The quasi-linear system to be controlled is both unstable and nonlinear. The complete nonlinear feedback control system is designed without *a priori* information of the plant dynamics, using only measured input/output data. The first design step is to combine a conventional method of multivariable system identification with a dynamic multi-layer perceptron (MLP) to achieve a constructive method of system identification. Based on the identified linear model of the system, states will be estimated and converted to more appropriate state for control in the second design step. The class of quasi-linear nonlinear systems is assumed to operate nominally around an equilibrium point in the neighborhood of which a linearized model exists to represent the system, although normal operation is not limited to the linear region. The results presented here provide an accurate discrete-time nonlinear model, which is used in the design of a nonlinear state estimator. The controller design is derived from a switched-linear feedback controller from the estimated states using the identified linearized model of the system around each suitable operating point, as a role model for the neuro-controller in the initial phase. Finally, using the partially

trained controller, the neuro-controller can be further trained "on-line" using a selected performance index to guide the learning. A prototype problem, an inverted pendulum system, is simulated as a physical system to be identified and to be controlled. Simulation results indicate that the present design method is very reliable comparing with other methods and hence is suitable for both identifying and controlling critical industrial processes. The prominent feature of this method is that no specific model information is initially required throughout the identification and control of the nonlinear plant. As an application of identifying an unknown plant in power electronics systems, an empirical data modeling approach which aims at generating small-signal equivalent models and also nonlinear models for a general class of converters, including resonant converters, and subsystems in a distributed power system is presented.

To my Mother in Heaven

Acknowledgements

First of all, I sincerely would like to thank my advisor, Dr. Hugh F. VanLandingham for his full support throughout this work, as well as his academic advice for the direction of this research. Furthermore, without his warm heart and endless patience, I could not come to this point.

I also would like to thank Dr. Bo H. Cho for his all kinds of tremendous efforts in helping to make this possible and Dr. William T. Baumann, Dr. John S. Bay and Dr. Loren P. Rees for their suggestions and critical review for serving as my committee members.

I have been very fortunate to have good friends and families around to share the moments of graduate study in Virginia Tech. Especially, I would like to thank Dr. Dong S. Ha for his valuable comments about the attitude of being a graduate student. Also, thanks to Phuong Huynh, Michael Tran, A. Tsoukkas and Lichun Guo for sharing their knowledge throughout research. Among so many good friends, in particular, I should thank Dr. C.J. Chung, Dr. Hyung-Gi Lee, Dr. Geun-Hee Rhim, Dr. Chang-Hyun Cho, Mr. Shi-Young Lee, Suk-Won Kim, Young-Min Kim, Kyu- Bong Doh, Jahng-Soon Park, Paul Gendron, David Kapp and Arnie Descastro for allowing me to be myself and, all others those who were willing to drop by *Macado's*, *Maxwell*'s and *Bogen's* for a couple of refreshment after midnight.

Besides the academic side, I will really miss the day of *Foxridge Tennis Court* playing tennis together with my younger brother Gene Kim and the "beautiful and also miserable" day of *Tech Golf Course* with Bong-Soo Choi. In addition, I must thank all my friends in Dallas, Kathy Lee, Jong-Seok Hong, Jeong-Seon Yoon and Ziad for their friendship and continuing to include me in the fun. I also would like to thank all my friends and supporters of E-Hwa Electrical Eng. Co. including Mr. Taek-Yong Kim, Mr. Jeong-Hee Rhim and Mr. Byeong-Geun Kim.

Finally, and most importantly, I would like to express my gratitude to my family, including aunts Ran-Soon Choi, Yun-Soon Choi those who have never hesitated in helping me to attain this little goal and also my sister, S-Mi Choi, who kept reminding me of my mother every time when I was out of track. I also thank my wife, Young-Sook Park for her help and understanding.

If it is indeed possible, I wish to give my all accomplishment to Ju-Hoon Choi, my younger brother for his role in our family during my stay in U.S.A.

Table of Contents

List of Figures

1. Introduction

1.1. Background

Real-world industrial processes have always been of considerable interest for control theorists and practioners. These processes are typically characterized with partially understood nonlinear system dynamics as well as lack of knowledge of true system parameters, noises, and uncertainties in the interactions between process and its environment. Over the last decade, theorists of classical and modern control have been promoting new and sophisticated techniques, under the heading of adaptive control to meet some of these difficult problems. However, rigorous mathematical treatment is constrained usually by a set of assumptions with various degrees of validity. To ensure applicability of certain theories, continuous verification of underlying assumptions must be enforced, which from a practical viewpoint is often impossible.

There has been ongoing research on using neural networks for identification and control of unknown nonlinear dynamic systems since the early 1960's. The idea of using neural networks for controlling physical systems has more recently seen a great deal of attention. One of the main objectives is to remove the requirements of having an exact detailed mathematical model for the system. The underlying promise is that through recursive learning, a neural network will be able to mimic the essential dynamical characteristics of the physical system. Until the advent of recent neurocomputing techniques almost all feedback control designs required a detailed model of the system to be controlled. Since real-world systems cannot be modeled in precise mathematical terms due to unmodeled dynamics and, typically,

a noisy environment, it is very difficult to determine an exact model for a complex nonlinear system. If there are significant plant dynamics that are not included in design model, then feedback control system will perform worse than expected and may be unstable around the operating point. Consequently, there is a need for a non-classical technique which has the ability to accurately model these physical processes to prevent the controller's failure due to modeling errors. The control of an unknown plant becomes even more complicated, especially when the state of the plant is not always measurable and control must be done using only input/output data. But, it has been shown that a multi-layer perceptron (MLP), one of the many forms of artificial neural networks (ANNs) is a universal function approximator, i.e. with sufficient training on appropriate input/output data, an MLP can represent arbitrarily closely any continuous vector map. Later this model is used to train a neuro-controller for nonlinear plants [1,2]. One of the challenges for future research in the field of control is to develop robust, adaptive, and fault-tolerant controllers. So far, significant progress has been made in the theory and applications of adaptive control. It has become a promising approach to achieve high performance of advanced control systems. However, current adaptive control approaches have their limitations. For one, these methods usually make use of a structured type of uncertainty in which the plant model has a known form, but with unknown parameters. Moreover, adaptive control systems designed according to existing theory could become unstable due to the excitation of the inevitable unmodeled dynamics and in the presence of unmeasurable output disturbances. Therefore, it is important to develop an approach in which the structure of the plant model could be well identified during an identification process.

Although the theory of linear system identification may now be considered to be a mature discipline, new techniques, particularly for *nonlinear* system

identification, continue to be of interest. In this research such a method is addressed in the context of using neural networks [3]. Neural networks of various types and structures (paradigms) have been found to be efficient tools for identifying nonlinear systems, e.g. through Volterra series models, group method of data handling (GMDH) models, self-organizing neural nets (SONN) models and radial basis functions [4,5,6,7]. Although there are many techniques available for the corresponding *linear* identification problem, MLPs may be regarded as a non-classical technique which can accomplish similar results using only input/output data, i.e. without prior model information. Most importantly, MLPs do not require the usual assumption of linearity. Thus, although it is true that neural networks can offer little, if any, improvement over existing methods of identification of linear systems, they do present a potential for capturing the complex nonlinearities of industrial processes of all kinds in a universal manner never before imagined [8].

Among the several researchers of control community using ANNs over the past two decades, Narendra has used dynamic ANNs as components in dynamical systems, concentrating on system identification and control of the nonlinear plants [9,10,11]. Pao introduced functional-link net which constructs a nonlinear mapping into the input layer to reduce the complexity of ANNs [12]. Psaltis et al. introduced a modified error-back propagation algorithm based on propagation of the output error through the plant which is considered an additional unmodifiable layer of the ANN, using its partial derivatives at the operating point [13]. Guez presented trainable adaptive controllers which consist of a teacher, the trainable controller and a plant. The teacher may be automated as a linear or nonlinear control law, or it may be a human expert to provide the knowledge of the system dynamics through the analysis of the controlled process [14]. Baird III et al. developed a hybrid controller which is a combination of an adaptive controller and ANNs to cope with

time-varying dynamics [15]. One of the main trends in training an ANN is to learn the system's inverse assuming that the system is invertible, and then the desired system output is achieved using the control input generated by the system's inverse. But Gu and Cui et al. pointed out that even if the system is invertible, the inverse control scheme may not be acceptable due to possible internal instability in a non-minimum phase system [16,17]. Another trend is borrowing the concept of linear optimal theory in ANNs using, not only quadratic errors, but also a more general cost function (or performance criterion) to reduce system output errors [18,19,20]. Since ANNs are used to take into account nonlinear effects of the system to the conventional linear optimal controller, ANNs broaden the range of control beyond the limited range of using linear optimal control law alone. In designing the above mentioned controllers the most important thing is to develop an efficient training algorithm. Usually, the "training" of an ANN is typically not straightforward. There is still a high degree of "art" associated with selecting and training of an ANN. However, in this work we will emphasize systematic steps used to achieve a nonlinear feedback control design with an explanation of the state estimation technique based on the identified model.

MLPs are regarded as a non-classical tool for identification and control of nonlinear systems using only input/output data; however, there are many difficult problems to overcome, such as when the nonlinear system is found to be both complex and unstable. This latter condition complicates the "training" of the MLP [21]. One approach is to stabilize the system locally. Such stabilization of a nonlinear dynamic system can be done for systems which are controllable near an equilibrium state, i.e. stabilizing the linearized model near the equilibrium point with linear feedback [22,23,24,25]. From the extension of well-known linear system theory, if all the states are available through measurements, both theory and

application indicate that "locally controllable" systems in the domain of our interest can be controlled by forcing proper inputs to the system. Unfortunately, there is no general theory regarding performance if estimated states (instead of actual states) are used in the feedback process of the nonlinear system. In cases such as all the states of the system are not accessible, but only outputs can be measured, the task of control becomes more complicated. Therefore, the observability of the nonlinear system is a critical issue. After checking the observability of the identified system near an equilibrium point, we can define a "locally observable" system in the domain of interest, whose concept is similar to that of a "locally controllable" system in the neighborhood of the equilibrium point. In many control problems, often it is inconvenient to measure every system state due to various restrictions, but it would be desirable to feedback all the states of the system if it were possible to generate the state variations in some indirect way.

In the group of researchers who are interested in constructing nonlinear observers for feedback linearizable nonlinear systems, Hunt and Verma insisted that a properly designed observer can work well together with the controller if the system is feedback linearizable [26]. But they admitted that their result is local because their domain of interest is sufficiently small. Therefore, there is little difference from the well-known linear observer of a linearized model of the nonlinear system. Dhingra et al designed an estimator by assigning "additional state variables" as outputs of the nonlinear blocks. After all, the plant is re-modeled as an "extended state space" model which permits a computationally efficient state estimator to be devised [27]. Even though the estimation procedure, which is similar to a Kalman filter algorithm, exhibits good structural robustness properties, the system nonlinearities are assumed to be known in advance. During the process of the fabrication of the controllers, Cheok and Beck generated state estimates by

a delayed-measurement observer which compares with Luenberger reduced-order observer [28]. The observer which was originally designed by Loh and his colleagues looks simpler and more efficient than that of Luenberger since it has no dynamics [29]. But it depends heavily on the quality of the measurements. Furthermore, they assumed that they already knew which state should be estimated (either discrete integration or differentiation). Therefore, there is no generality applicable to the system unless we are familiar with the internal structure of the system.

The design approach proposed here involves three main steps: The first steps are to determine an equilibrium point and identify the linearized system about this equilibrium point by a combination of the connectionist approach (with back-propagation supervised learning) and a conventional method of multivariable system identification. This classical approach can examine all the admissible structures of the system in order to obtain a (linear) model which optimally generalizes over the available input/output data around the equilibrium point. This linear model is, in turn, incorporated into the MLP model as that part of the system corresponding to the linear feedthrough terms of the MLP (using a linear output activation function). The resulting identified model is called the ***modified dynamic* MLP** model with inputs representing both the actual inputs and delayed versions of the inputs and outputs to capture both the nonlinearities and the dynamics of the system. This modified dynamic MLP will be restricted to a simple single hidden layer form so that analytical information can be readily obtained from the derived model. The reader is referred to Reference [8] for details. Since the order or the structure of higher-order neural networks can be tailored to the order or structure of the problem from the linear modeling stage, a neural network designed for a particular class of problems such as quasi-linear systems can be specialized

and very efficient in solving those problems.

To demonstrate the validity of the technique, this method is used on certain power electronics systems in order to provide discrete-time (D-T) small signal models not only for pulse-width modulation (PWM) converters, but also for resonant type converters. The resulting small-signal model describes the converter as a linear time invariant system and the knowledge of the identified linear system can be applied to switching converters. But switching regulators are inherently nonlinear, only small-signal methods are generally used in order to apply linear control theory. However, this small-signal approximation cannot represent the nonlinear characteristics of the regulator, which becomes significant for large perturbations. Due to an inadequate modeling of a switching regulator, the resulting feedback controller might fail to control the regulator beyond the assumed small-signal model boundary. Therefore, an MLP network with inputs representing both the actual inputs and delayed versions of both the inputs and outputs is needed to capture both the nonlinearities and the dynamics of the system in large-scale simulation.

The second step is to generate the estimates of the state variables of the system based on the identified linear model in order to construct an optimal linear state feedback controller from the estimated states $\hat{\mathbf{x}}$ near an equilibrium point, e.g. $\mathbf{x} = \mathbf{0}$.

After completion of identifying and controlling the linear part of the plant, the remaining nonlinear part of the system will be identified and controlled, separately. At least conceptually, and for relatively tractable systems, the remaining nonlinear part of the system can be captured with a single hidden layer having a number of processing elements (neurons) with nonlinear memoryless activation functions embedded in a linear dynamic system. In order to have a network of

minimal complexity, pruning is done by removing connections whose weights have small values, including connections to neurons in the hidden layer if their contribution is insignificant.

The final step is to design a neuro-controller using the information from the identified nonlinear parts of the system. The (constant) control gains are calculated for each operating point chosen selectively—to be used for simulation and data collection. The data is obtained using a switched-linear controller for the purpose of developing a neural net replacement. The neural net is needed for implementation and interpolation, two inherent and attractive properties of feedforward multilayer networks. Utilizing the designed switched-linear state feedback controller as an initial role model for the neuro-controller, this partially trained neuro-controller can be trained completely based on a desired performance index in the domain of interest. This final step ensures that the controller goes beyond its previous training stage and captures the necessary control actions for the nonlinear plant.

1.2. Problem Statement

A general class of discrete dynamical systems can be represented by the following state space model:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] \\ \mathbf{y}(k) &= \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \end{aligned} \tag{1}$$

where $\mathbf{u}$ and $\mathbf{y}$ are the system input and output vectors, respectively, and the state $\mathbf{x}$ is some combination of delayed inputs and outputs. The nonlinear functions $\mathbf{f}$ and

g are assumed to have continuous partial derivatives. The class of nonlinear systems for this dissertation is called ***quasi-linear systems***, meaning that around some operating point a linearized model describes the local system stability; and, further, the order of the system is assumed to be invariant over the desired (nonlinear) operating region.

Given that neither the order of the system (dimension of **x**) , nor the vector maps **f** and **g** are known, an MLP can be trained to represent the system by assuming only that the inputs and corresponding outputs of the system are measurable. Using the initial knowledge of the first MLP, a second MLP will be introduced as a feedback controller based on the state estimates $\hat{\mathbf{x}}$ instead of **x**, trained on the minimization of a selected cost functional. Fig. 1 illustrates the structure of the method. Since the ultimate control objective is to regulate the plant output to zero asymptotically, for simplicity, the reference input **r** is taken to be zero, i.e. for the present purpose of the regulation problem.

Initially, the identification is carried out using a dynamic MLP which consists of delayed inputs and outputs as additional inputs to the MLP. Dynamics are incorporated into the model using the standard assumption that the outputs of the network, which correspond to the outputs of the actual plant, are used to derive part of the network inputs, which include the system inputs as well as delayed versions of the system inputs and outputs. This structure which is called a ***modified dynamic* MLP** is presented in Fig. 2, with an arbitrary number, *h*, of delays. The extent to which the outputs are delayed depends on the order of the unknown system. Since all network inputs are connected through to the neurons in hidden layer, keeping the number of network inputs small is very important in minimizing the complexity of MLPs with desirable accuracy.

This work was motivated by the desire to extend the models of systems

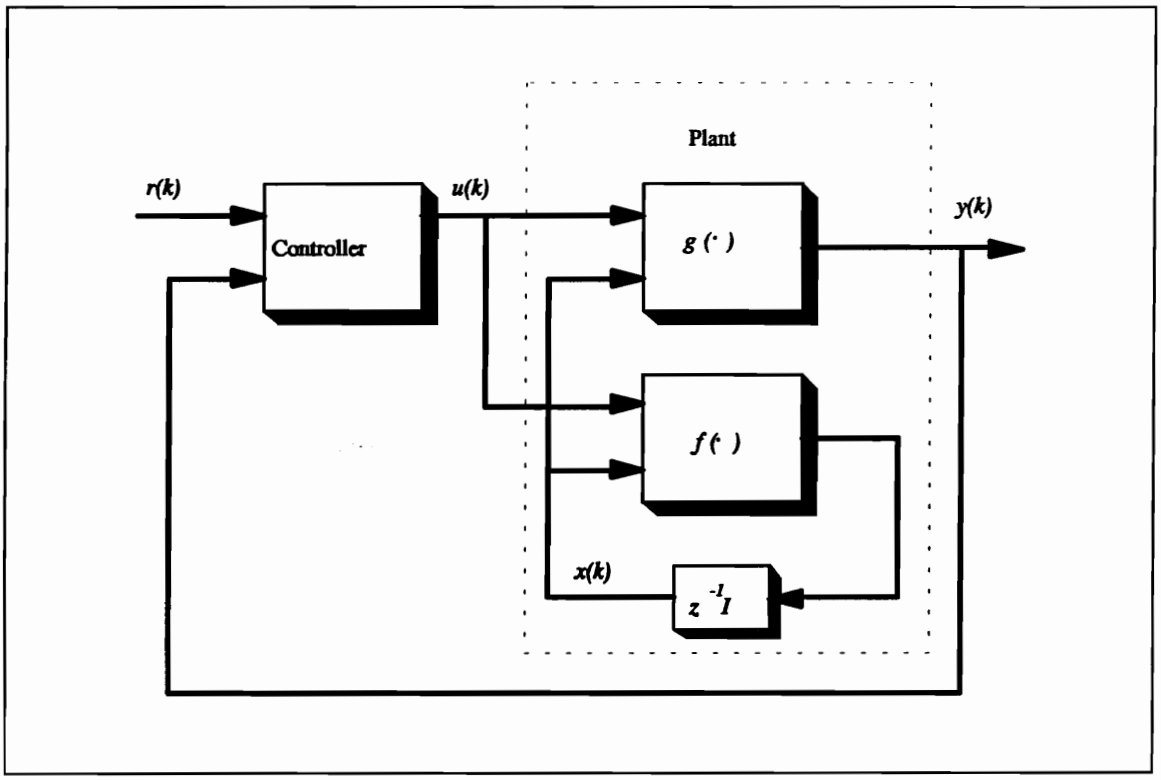


Fig. 1. Control from Identified Plant

which are linear (for small signal variations around an operating point) well into their nonlinear range. Since constructing a linear controller is based on the linearized model of the nonlinear plant around an equilibrium point, the linearization must be done to reduce the dynamics of a nonlinear system to a linear model. However, when the system goes deeper into its nonlinear range where the assumption of linearity is not valid any more, the system with linear controller may be unstable. In this research, our ultimate goal is to design a neuro-controller which can stabilize the system far beyond the linearized region.

An inverted pendulum system is selected as a prototype example of this class of nonlinear dynamic systems in Eqn. (1) for two reasons: (1) the system is

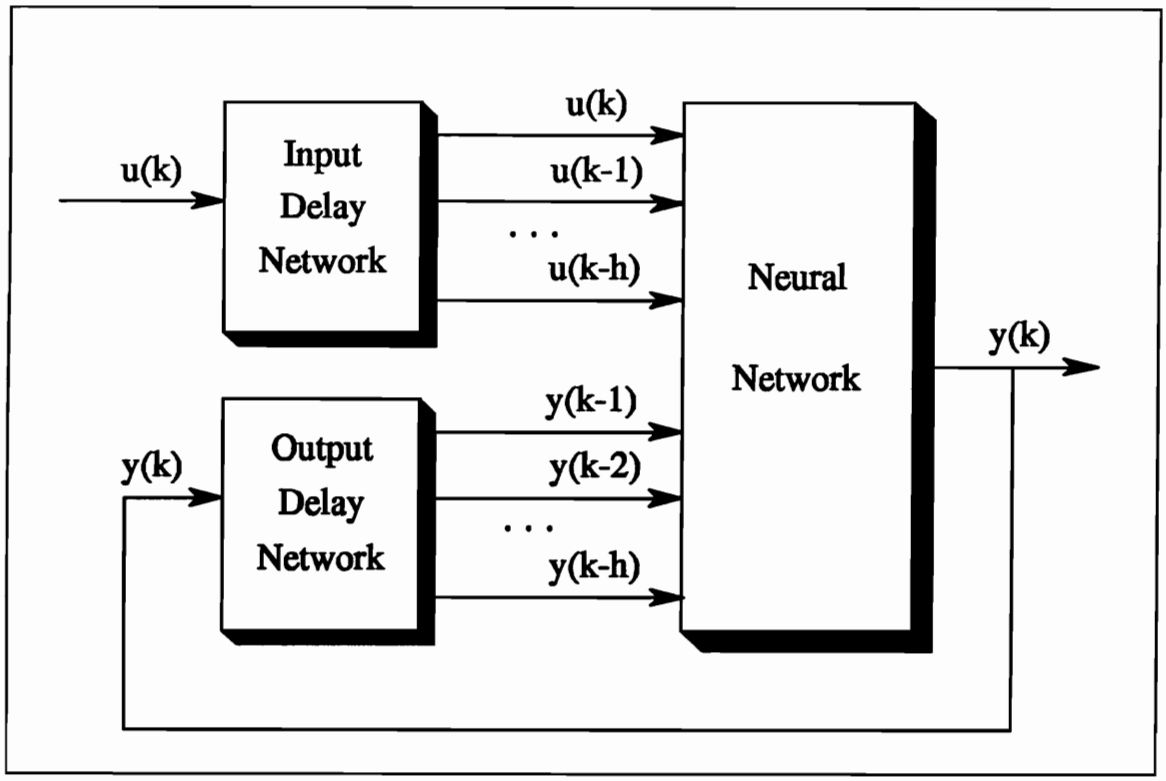


Fig. 2. Modified Dynamic Neural Network

a classical unstable system with severe nonlinear dynamics and therefore representative of this class of difficult control problems; and, (2) so many previous researchers have investigated this problem with their own assumptions that it has often been considered a "benchmark system" in the control literature [30,31,32]. The control objective is to balance the pole in the upright position while maintaining cart at the center of the tracks as quickly as possible even with a large initial deflection. Fig. 3 illustrates the cart within its position boundaries. Four state variables represent the state of the system dynamics: θ (the angular position of the pole), $\dot{\theta}$ (the angular velocity of the pole), p (the position of the cart on the track), $\dot{p}$ (the velocity of the cart on the track). The system dynamic equations are the following two second-order differential equations, derived from first principles:

$$\ddot{p} = \frac{u + \frac{ml}{2}(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta)}{M + m}$$

$$\ddot{\theta} = \frac{2g\sin\theta - 2\cos\theta \dfrac{u + \frac{ml}{2}\dot{\theta}^2 \sin\theta}{M + m}}{l(\frac{4}{3} - \frac{m}{M + m}\cos^2\theta)} \qquad (2)$$

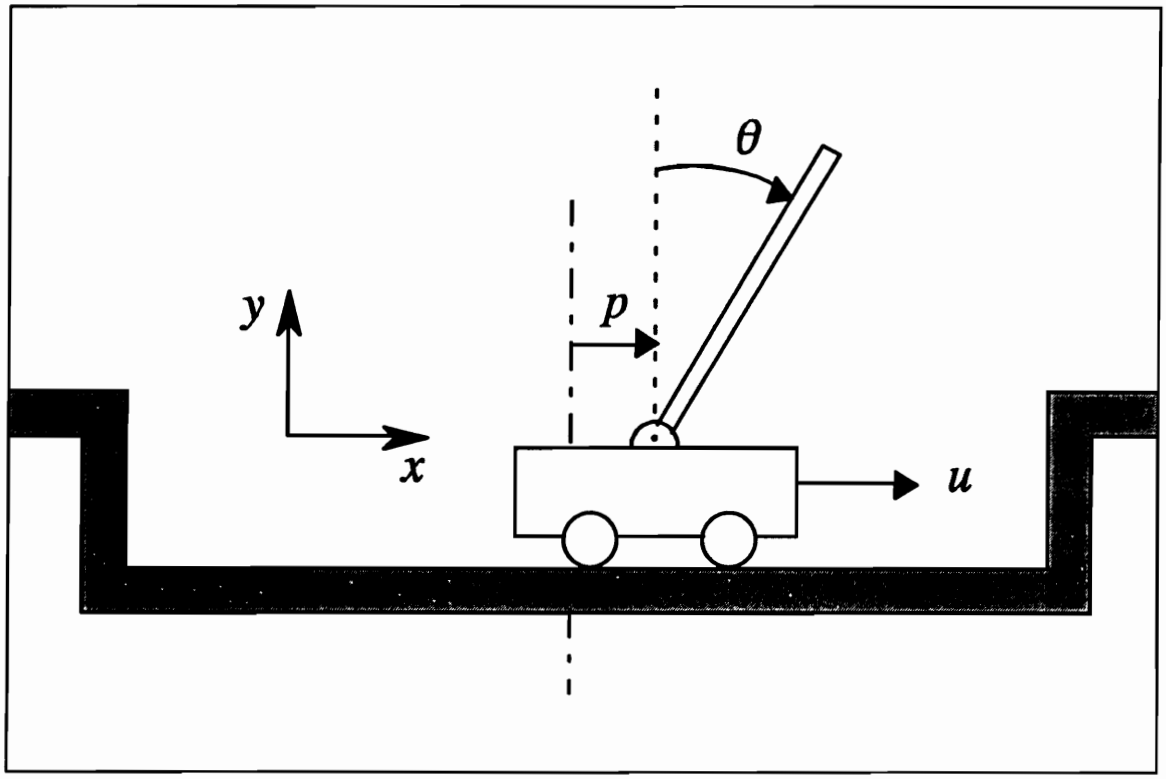


Fig. 3. Cart/Pendulum System

where the parameters M, the cart mass; m, the pole mass; l, the pole length, have the values 1, 0.1 and 1, respectively, all in SI units (g = 9.81 m/s^2).

1.3. Organization of Dissertation

The dissertation is organized in the following way. Chapter 2 begins with a section which describes pre-processing of the input/output data to prepare the data for training the MLP. More than simple scaling and shifting of the data, pre-processing seeks to determine indications of relationships between the various signals, e.g. involving a study of the statistical correlation between input and output data. Also care must be taken to obtain data that is representative of the system's entire range of response, since an MLP can only learn to model the system to the extent that the data is complete. The details of a (hybrid) linear system identification technique for the linearized model of nonlinear system around its equilibrium point to extract the linear part of the unstable system are explained in the remainder of Chapter 2. The conventional method of multivariable system identification, which utilizes all possible observable structures of the system to achieve a linearized model, is presented briefly. This method is used to optimally generalize over the available input/output data around an equilibrium point. Chapter 3 addresses the topic of modeling of a highly nonlinear power electronics system. The first section deals with small-signal modeling of power converters including an open-loop boost converter, a series-resonant converter and a forward multi-resonant converter, which is used to design feedback controllers. The following section is concerned with large-signal modeling of both open-loop boost converters and closed-loop buck converters. Due to the highly nonlinear characteristics of power converter systems, it is not easy to check transient responses of the system in various step-load (or line) changes. An MLP, which was trained without transient response data, is used to identify the transient response of the plants in the large-scale system simulation. The well-trained MLP shows the nonlinear

behavior of the system outside the linear region and also gives a close approximation of the boundaries where the system might go unstable, i.e. beyond the capability of the linear feedback controller. Identification of the inverted pendulum system is discussed in Chapter 4. The method and algorithm of generating the estimates of the states variables based on the robust hybrid linear system identification technique is explained in Section 4.1.A and 4.1.B. Identification and state estimation of the linearized inverted pendulum system is discussed in Section 4.1.C. The structure of the final MLP (based on the identified linear model of the system) completing the nonlinear part identification is given in Section 4.1.D. Among others, these details are concerned with determining the number of neurons needed to capture the nonlinearities of the system. Pruning is used to ensure a network which is large enough to generalize, yet small enough to implement easily. Section 4.2.A is concerned with designing a switched-linear controller to stabilize the system initially. Using an optimal switched-state feedback controller as an initial role model of the neuro-controller, final training of the neuro-controller according to the performance index is discussed in Section 4.2.B. Finally, a discussion of the simulation results and some directions for future work are presented in Chapter 5.

2. Robust Hybrid Identification

2.1. Introduction

Two widely differing identification techniques for MIMO systems are presented in order to provide a basis for comparison and selection [8]. The first technique is a novel identification procedure which uses input/output data to establish, not only the system order, but also the minimal number of parameters required for representation [33]. The second technique uses a connectionist approach along with back-propagation learning to establish the identified model [34]. After a development of each method, an example, which is necessarily restricted to be linear, is presented to illustrate relative numerical accuracy as well as advantages and disadvantages of each method. To identify a system, the classical approach requires sequential data of the system, and identifies the parameters as well as the initial state for different structures. Unless the state is initialized appropriately, this model does not work well for a different set of sequential data of the unstable system. In contrast, back-propagation learning does not need specific initial states of the system in training, but does need information such as the approximate order of the given system and the most appropriate structure of state space form. Hence these two methods are not completely disparate and can be combined to capture the advantages of both for a ***robust hybrid identification technique***. Particularly when the system is unstable, there is very little choice in generating input/output data except choosing different initial conditions and stopping the process before the states of the system leave the domain of interest or beyond the limit of safety. Therefore, the initial stage of data selection for

identifying the unstable system is crucial and is discussed in Section 2.2. The basic modeling structure using an MLP with an explanation of nonlinear squashing function is addressed in Section 2.3. After a review of the robust hybrid identification approaches in the Section 2.4, 2.5, and 2.6, there follows a brief summary of the identification process as applied to D-T systems in Section 2.7. This hybrid method was developed to use only input/output data to determine both the structure and the system parameters with no other system information.

2.2. Pre-processing

The initial stage of data selection for training an MLP is very important. For example, it is not only the range of data, but its distribution, that is necessary to define a good mapping. The reason for this is that the system will typically occupy a very small region, i.e. the neighborhood around either a stable or unstable (with controller) equilibrium point, for the vast majority of the response time. One solution to this problem is to excite the system with varying amounts of pseudo-random disturbance, up to the point where the system can be controlled by the controller, so that the data collected will be representative of the entire range of the system. Acquiring input/output data which show nonlinear characteristics of the system, within safe operating conditions of an industrial process, is very important in training MLPs.

Once a satisfactory statistical distribution of the data achieved, it is normally found that the ranges of the unprocessed input/output data are not appropriate for training a neural network. A common cause of problems stems from presenting data to a back-propagation algorithm as raw values, rather than in values that have been suitably scaled to the neuro-dynamic functions being used. The MLP network

commonly uses a sigmoid, or hyperbolic tangent, activation function. For present purposes the hyperbolic tangent function, which responds in a nearly linear fashion to summations between about -2 to +2, will begin to saturate when its input approaches either extreme. When the hyperbolic tangent is used as a nonlinear squashing function, it produces outputs between -1.0 and 1.0, similar to outputs between 0 to 1 for a standard sigmoid function. Therefore, the desired outputs should lie within these ranges. When very large input values are presented to a network with even very small weights, the summations can be huge in which case the squashing function such as sigmoid or hyperbolic tangent will become saturated. This saturation, in turn, causes the learning process to stall since the back-propagation algorithm uses the derivative of the hyperbolic tangent activation function (which would then be nearly zero) as a multiplier in the weight update equation. Thus, learning stops for any processing element with such a large summation signal into its activation function.

Specification of the real world data ranges and also target network ranges is needed to define linear mapping which map the minimum and maximum values of the real world ranges to those of the network range. The basic idea is to choose a range that produces summations which will not initially saturate the transfer function. In the notation below r_I, the minimum input range; R_I, the maximum input range; r_D, the minimum desired output range; R_D, the maximum desired output range; i_k, the network input corresponding to the real world value f_k; d_k, the network desired output corresponding to the real world value f_k; o_k, an actual network output; and g_k, corresponding to the real world output. Then, the corresponding mapping from the real world to the MLP network are as follows:

Input:

$$i_k = \frac{(R_I - r_I)f_k + (M_k r_I - m_k R_I)}{(M_k - m_k)} \tag{3}$$

Desired Output:

$$d_k = \frac{(R_D - r_D)f_k + (M_k r_D - m_k R_D)}{(M_k - m_k)} \tag{4}$$

After the network produces scaled output, the mapping from network output to the real world is "descaled" by:

$$g_k = \frac{(M_k - m_k)o_k + (R_D m_k - r_D M_k)}{(R_D - r_D)} \tag{5}$$

Values outside of the given ranges are mapped linearly outside the low and high values, using the same scale and offset values.

2.3. Modeling Structure

In order to control the nonlinear cart-pendulum problem of Fig. 3, we first need to identify the system dynamics accurately. In general, there are several different approaches that can be used, but for present purposes the MLP structure of Fig. 4 was used. MLPs consist of a number of *layers* of *processing elements (PEs)*. The basic PE is an idealized version of a *neuron* as found in the nervous systems of animals. The basic model was first introduced by McCulloch and Pitts [35]. Fig. 5 illustrates the basic PE (neuron model) with inputs $\{u_i\}$, $0 \le i \le n$.

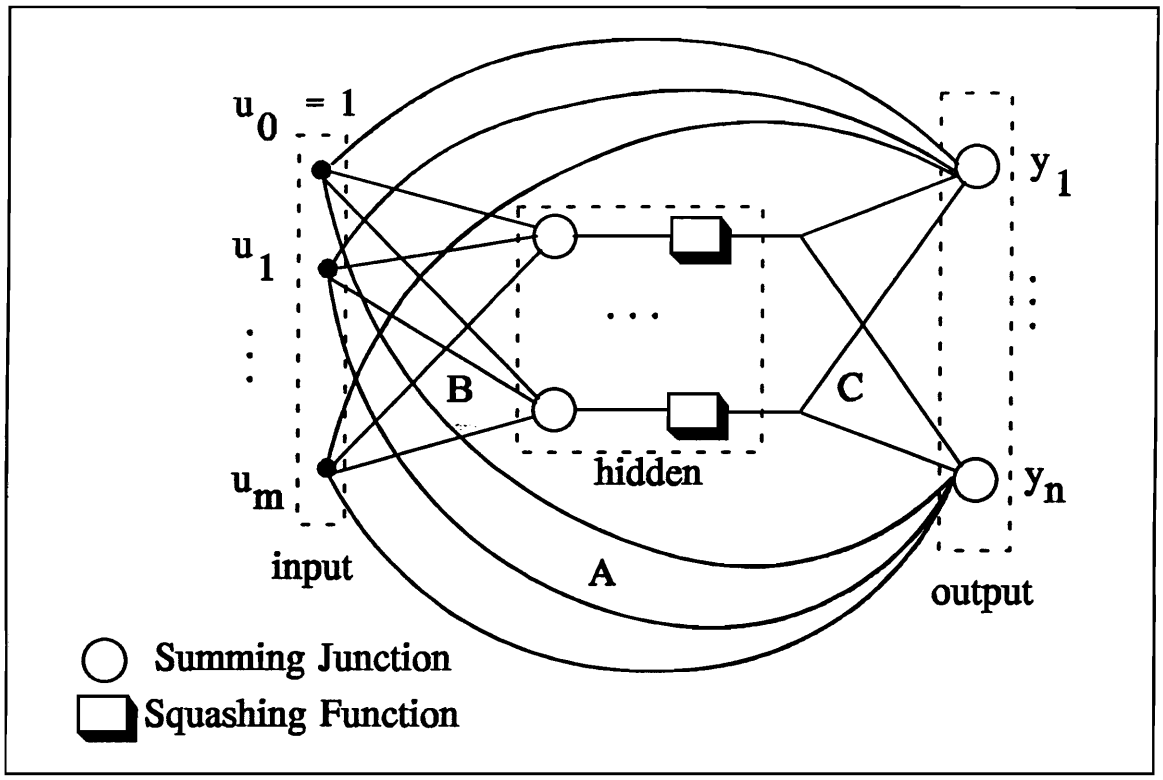


Fig. 4. MLP Structure for Identification

The weighted sum of the inputs is an intermediate variable, shown as x in Fig. 5; x is subsequently passed through a function block to obtain the PE output, $Y = f(\cdot)$. The weights are adjusted during training so that the overall effect of the network is to identify the underlying mapping that corresponds to the input/output data. The function shown in Fig. 5 is called the activation function. Its purpose is to limit the signal amplitude at the output of the processing element. The function $f(\cdot)$ can be any of several types. For present purposes we will either use a *linear* function, $f(x) = x$, or a specific nonlinear function, the *hyperbolic tangent* function,

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6}$$

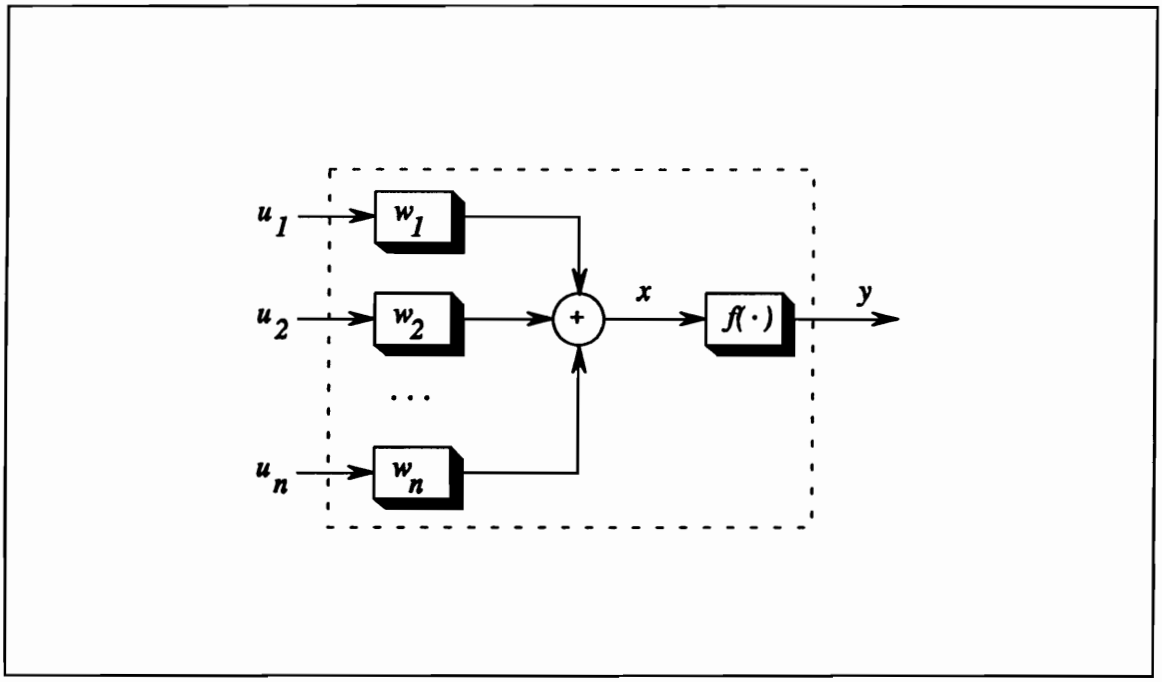


Fig. 5. The McCulloch-Pitts Neuron

The hyperbolic tangent is a type of *squashing function* since its range, (-1, 1), limits the output signal amplitude.

The network topology of a multilayer perceptron network can vary from one application to another. In a feedforward net signals are fed only in a direction from input to output, i.e. there are no feedback paths. With multilayer nets individual layers (or even selected PEs within a layer) can have different PE functions. Fig. 4 illustrates a useful topology for purposes of system identification. Without counting the input layer, since it merely acts to transfer the signals into the MLP, the network of Fig. 4 would be called a two layer topology, one *hidden* layer and one output layer with linear feedthrough weights. The circles in Fig. 4 are summing elements and the rectangular blocks indicate hyperbolic tangent functions. Thus, the hidden (center) layer contains a standard squashing function, while the

output layer has effectively a linear function. By using a linear transfer function for the output elements, the system dynamics can be matched to a convenient mathematical form of Eqn. (7). The hidden layer activation functions are hyperbolic tangents. The connection weights from inputs to outputs, inputs to hidden layer, and hidden layer to outputs are the arrays **A**, **B**, and **C**, respectively. The purpose of this particular structure is to have a convenient, yet general, mathematical form which is capable of representing the nonlinearity of the system as well as linear part [36]. Specifically, this form is given by

$$\mathbf{Y} = \mathbf{A}\,\mathbf{U} + \mathbf{C}\tanh(\mathbf{B}\,\mathbf{U}) \tag{7}$$

where **Y** is the vector of system outputs and **U**, the vector of all MLP inputs, including the delayed signals. In this manner the system model can be extracted from the trained MLP. Experimental results indicates that two hidden layer feedforward networks are more prone to fall into bad local minima, but that one and two hidden layer networks perform similarly in all other aspects [37]. Furthermore, single hidden layer networks which have the structure of Eqn. (7) can give more analytical insight of the nonlinear plant to the system designer. Therefore, there seems to be no reason to use two hidden layer networks in preference to single hidden layer nets except very sophisticated applications [38].

2.4. MIMO Structure Determination

We now consider the case where only input/output data is available, without a given system model. The process of creating a system model from the data is called system identification. A deterministic D-T system identification will be performed by calculating an observable form state space model $R_o = \{\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o,$

$\mathbf{D}_o\}$ from a set of input and corresponding output data with the restriction that the input signals are "persistently exciting," i.e. that the system is sufficiently excited to exhibit all of its modes in the corresponding output signals. This algorithm can be considered symbolically as $\{\mathbf{u}(t), \mathbf{y}(t)\} \Rightarrow \mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o, \mathbf{D}_o$, which simply applies the relationship between an input/output data stream and a corresponding state space description in a selected pseudo-observable form to obtain an identified system model. The identification process is shown in Fig. 6.

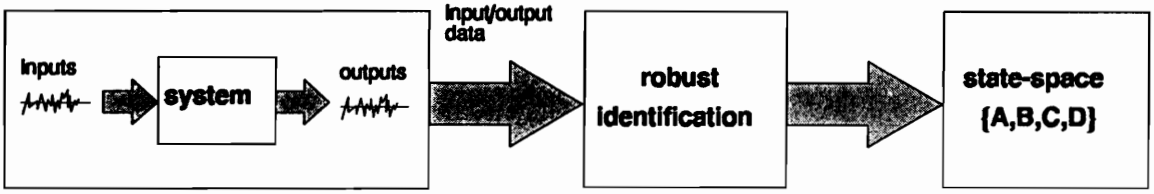


Fig. 6. Identification Process

In his often referenced paper Luenberger established nominal structures for a multi-input, multi-output (MIMO) system [39]. The method is based on the concept of either controllability or observability indices. For present purposes only the observability form will be discussed. We will assume that sampled data from the unknown C-T system has been made available. Beginning with an assumed D-T state space model,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{8}$$

where $\mathbf{x}$ is an $(n\times 1)$ vector, $\mathbf{u}$ is an $(m\times 1)$ vector, $\mathbf{y}$ is a $(p\times 1)$ vector and the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ have corresponding compatible dimensions. In particular,

suppose that the following input/output vector data pairs are given for a given system:

$$\{ \mathbf{u}(k), \mathbf{y}(k) \} \quad \text{for} \quad k = [0, N] \tag{9}$$

where N is sufficiently large for identification purposes. The input and output vectors are of dimensions $m \times 1$ and $p \times 1$, respectively, where $m, p \geq 1$. An initial step is used to re-organize a part of the data as follows, typically with $l < q < N$:

$$\mathbf{Z}_l = \begin{bmatrix} \mathbf{U}_l \\ \text{----} \\ \mathbf{Y}_l \end{bmatrix} \tag{10}$$

where

$$\mathbf{U}_l = \begin{bmatrix} \mathbf{u}(0) & \mathbf{u}(1) & \cdots & \mathbf{u}(q) \\ \mathbf{u}(1) & \mathbf{u}(2) & \cdots & \mathbf{u}(1+q) \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}(l) & \mathbf{u}(l+1) & \cdots & \mathbf{u}(l+q) \end{bmatrix} \tag{11}$$

and

$$\mathbf{Y}_l = \begin{bmatrix} \mathbf{y}(0) & \mathbf{y}(1) & \cdots & \mathbf{y}(q) \\ \mathbf{y}(1) & \mathbf{y}(2) & \cdots & \mathbf{y}(1+q) \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{y}(l) & \mathbf{y}(l+1) & \cdots & \mathbf{y}(l+q) \end{bmatrix} \tag{12}$$

An effort is made to select q large enough so that as additional rows are added, the number of columns will continue to exceed the number of rows of $\mathbf{Z}$, as will be explained in the following.

Starting with a "small" value of l, the procedure calls for the rank of $\mathbf{Z}_l$ to be checked, followed by an augmentation of $\mathbf{Z}_l$ to $\mathbf{Z}_{l+1}$, i.e. with rows [$\mathbf{u}(l+1)$ ... $\mathbf{u}(l+q+1)$] and [$\mathbf{y}(l+1)$ ... $\mathbf{y}(l+q+1)$] appended to $\mathbf{U}_l$ and $\mathbf{Y}_l$, respectively, and a subsequent check of the rank of $\mathbf{Z}_{l+1}$. More specifically, suppose that

$$\text{rank}\,\mathbf{Z}_l = r_l \quad \text{and} \quad \text{rank}\,\mathbf{Z}_{l+1} = r_{l+1} \tag{13}$$

then if

$$d_l = d_{l+1} \quad \text{where} \quad d_l = r_l - m(l+1) \tag{14}$$

the system order is $n = d_l$, and if Eqn. (14) is not satisfied, the augmentation step is repeated. In this manner, starting from the first input/output pair and sequentially augmenting additional pairs until the d_l ceases to increase, the system order is determined. Note that the effect of rank due to the input vectors is subtracted out. For a "sufficiently rich" input the rank of $\mathbf{Z}$ would continue to increase with additional augmentation. Once the system order, n, and the number of linearly independent rows of $\mathbf{Z}$ arising from the i^{th} output, n_i, have been determined, the *observability index*, n_x, defined by

$$n_x = \max_{i=[1,p]} \{ n_i \} \tag{15}$$

can also be determined. In fact, as it will be shown later, n_x is given by $n_x = l+1$, where l is the smallest integer satisfying Eqn. (14). In other words, it is equal to the number of blocks of outputs $\mathbf{y}(i)$, $i=[0, n_x-1]$, containing at least one linearly independent row, $y_j(i)$, $j=[1,p]$. The set $\{n_i\}$ is referred to as the unique set of observability indices or, as will be explained later, as a set of admissible pseudo-observability indices. A specific example will help to illustrate the procedure.

Consider a system with order $n=7$, $m=2$ inputs and $p=3$ outputs. A typical instance of the above augmentation process might result in the array $\mathbf{Z}_3$ being constructed after $\mathbf{Z}_2$ with the determination that $d_2 = d_3$ from Eqn. (14). It is found that the rank of $\mathbf{Z}_3$ is *15* which, after subtracting $m(l+1)=8$, corresponding to the number of linearly independent rows of input vectors, gives for the system order $n=7$. Two rows among the those beginning with $\mathbf{y}(1)$ and $\mathbf{y}(2)$, are found to be linearly dependent; and all $p=3$ rows beginning with $\mathbf{y}(3)$ are linearly dependent.

If we are interested in the unique set of observability indices, we should determine which two rows among those beginning with $\mathbf{y}(1)$ and $\mathbf{y}(2)$ are linearly dependent. Assuming that the particular rows beginning with $y_2(1)$ and $y_2(2)$ (output #2) are, in fact, the dependent rows, then, according to the definition of observability indices [40,41], it may be concluded that this case leads to the set of observability indices given by

$$\{ n_i \} = \{ 3, 1, 3 \}$$

It has recently been shown that the use of this unique set of observability indices does not necessarily lead to the most convenient system representation, and that the use of so called admissible sets of pseudo-observability indices offers more flexibility in choosing the appropriate model [42,43]. For these reasons in the sequel we will pursue the selection of the most convenient set of (pseudo) observability indices.

Knowing that the system order is *7* from the rank calculations and that the observability index is *3*, corresponding to the minimum number of output vectors needed to achieve that rank, there are several possible observable form structures that may be considered. These pseudo-observable indices are given by

Case	1	2	3	4	5	6
Pseudo-Observability Indices	{3,2,2}	{2,3,2}	{2,2,3}	{3,3,1}	{3,1,3}	{1,3,3}

In all six cases $d_3 = d_2 = 7$, but in the first three cases $d_1 = 6$, whereas in the last three cases $d_1 = 5$. Note that in each case the "observability indices" sum to $n=7$. Let us further assume that, in fact, $d_1 = 5$, i.e. cases 4, 5 and 6. We can use a *crate* diagram to represent each of these three cases [41].

{3,3,1}		
1	1	1
1	1	0
1	1	
0	0	

{3,1,3}		
1	1	1
1	0	1
1		1
0		0

{1,3,3}		
1	1	1
0	1	1
	1	1
	0	0

Crate diagrams are simply a graphical method of visualizing the selection of linearly independent rows from the given output data. For example, with the columns of the crate being associated with particular output strings, the center crate above indicates that the independent elements are rows beginning with $y_1(0)$, $y_2(0)$, $y_3(0)$, $y_1(1)$, $y_3(1)$, $y_1(2)$, $y_3(2)$.

From the crate diagrams several related "selection vectors" are generated:

- By omitting the first row of, say the center diagram, corresponding to the indices {3,1,3}, the vector $\mathbf{v}_i$ is created by selecting the non-blank elements row-wise:

$$\mathbf{v}_i = [\ 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0\ ]^T \tag{16}$$

- From $\mathbf{v}_i$ the binary complement is formed, and denoted as $\mathbf{v}_a$:

$$\mathbf{v}_a = [\ 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1\ ]^T \tag{17}$$

- By considering the blank elements to be zeros, $\mathbf{v}_{li}$ is formed in like manner, but with row 1 included:

$$\mathbf{v}_{li} = [\ 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0\ ]^T \tag{18}$$

- Finally, $\mathbf{v}_{ld}$ is formed by again including the first row, but now taking the blank elements of the diagram to be unit valued, and finally taking the binary complement, leading to:

$$\mathbf{v}_{ld} = [\ 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1\ ]^T \tag{19}$$

The above selector vectors are uniquely determined by the particular set of pseudo-observability indices, or equivalently, the location of the unity elements in the corresponding crate diagram. As will be shown later, these selector vectors greatly facilitate calculation of the observable forms based on the chosen set of observability indices. In particular, the "selector matrices" given below in Eqn. (20), which are derived from the associated selector vectors by a corresponding selection of columns from an appropriately dimensioned identity matrix, are used directly in obtaining the observable form. The selection of rows (or columns) of a

matrix may be accomplished by a pre- (or post-) multiplication of a corresponding "selector" matrix. Thus, for instance from Eqn. (20), since $\mathbf{S}_i$ is a (7×4) selector matrix, the product $\mathbf{S}_i^T\mathbf{M}$, where $\mathbf{M}$ is a (7×7) matrix results in the "selection" of rows 1, 3, 4 and 5 from $\mathbf{M}$ into the (4×7) product. It will be clear in a later development how useful the selector matrices of Eqn. (20) are in the formulation of the identification algorithm.

$$\mathbf{S}_i = \begin{bmatrix} 1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \end{bmatrix}^T, \quad \mathbf{S}_{li} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \end{bmatrix}^T \tag{20}$$

$$\mathbf{S}_a = \begin{bmatrix} 0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \end{bmatrix}^T, \quad \mathbf{S}_{ld} = \begin{bmatrix} 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}^T$$

2.5. MIMO System Parameterization

In this section the structural details described in the previous section will be used in the process of obtaining the system parameters. The eventual representation of the identified system is a state space observable form $R_o = \{\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o, \mathbf{D}_o\}$ where $\mathbf{C}_o$ and $\mathbf{A}_o$ have the following structure, (continuing to use the *{3,1,3}* example from Section 2.4):

$$\mathbf{A}_o = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ x & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

and (21)

$$\mathbf{C}_o = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The structure of the pair $\{\mathbf{A}_o, \mathbf{C}_o\}$ is characterized by the following points:

- $\mathbf{C}_o$ consists of the first $p=3$ rows of the $(n \times n)$ identity matrix $\mathbf{I}_n$.
  At locations specified by the unities in the selector vector $\mathbf{v}_i$, the matrix $\mathbf{A}_o$ contains the last $n-p = 4$ rows of $\mathbf{I}_n$.
- At locations specified by the $p=3$ unities in the selector vector $\mathbf{v}_a$, the matrix $\mathbf{A}_o$ contains rows of elements which are not necessarily of zero or unit value.
- The "observability matrix" $\mathbf{Q}_{oo}$ of the pair $\{\mathbf{A}_o, \mathbf{C}_o\}$, i.e.

$$\mathbf{Q}_{oo} = \left[\mathbf{C}_o^T \quad (\mathbf{C}_o\mathbf{A}_o)^T \quad \cdots \quad (\mathbf{C}_o\mathbf{A}_o^{n_x})^T \right]^T \tag{22}$$

  contains all n rows of $\mathbf{I}_n$ at locations specified by the $n=7$ unities in the selector vector $\mathbf{v}_{li}$.

- The $p=3$ rows of $\mathbf{A}_o$ containing not necessarily zero or unit elements appear in $\mathbf{Q}_{oo}$ at locations specified by the unities in the selector vector $\mathbf{v}_{ld}$.

The results of Eqn. (21) derive from the basic similarity transformation, or change of state,

$$\begin{aligned} \mathbf{A}_o &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1}, \quad & \mathbf{B}_o &= \mathbf{T}\mathbf{B} \\ \mathbf{C}_o &= \mathbf{C}\mathbf{T}^{-1}, \quad & \mathbf{D}_o &= \mathbf{D} \end{aligned} \tag{23}$$

where $R = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ is an arbitrary n^{th} order observable state space representation. In order to obtain $\mathbf{A}_o$ and $\mathbf{C}_o$ given by Eqn. (21), the transformation matrix $\mathbf{T}$ in Eqn. (23), corresponding to the observability indices *{3,1,3}*, is given by

$$\mathbf{T} = \left[\mathbf{c}_1^T \ \mathbf{c}_2^T \ \mathbf{c}_3^T \ (\mathbf{c}_1\mathbf{A})^T \ (\mathbf{c}_3\mathbf{A})^T \ (\mathbf{c}_1\mathbf{A}^2)^T \ (\mathbf{c}_3\mathbf{A}^2)^T \right]^T \tag{24}$$

It may be verified that all $n=7$ rows of $\mathbf{T}$ are located in the observability matrix $\mathbf{Q}_o$ of the pair $\{\mathbf{A}, \mathbf{C}\}$, i.e.

$$\mathbf{Q}_o = \left[\mathbf{C}^T \quad (\mathbf{C}\mathbf{A})^T \quad \cdots \quad (\mathbf{C}\mathbf{A}^{n_x})^T \right]^T$$

at locations specified by $n=7$ unities in the selector vector $\mathbf{v}_{li}$.

The difference between a POF and the corresponding Luenberger form is that Luenberger re-ordered the selected rows by the columns of the crate such as given below the similarity transformation $\mathbf{T}_L$

$$\mathbf{T}_L = \left[\mathbf{c}_1^T \ (\mathbf{c}_1\mathbf{A})^T \ (\mathbf{c}_1\mathbf{A}^2)^T \ \mathbf{c}_2^T \ \mathbf{c}_3^T \ (\mathbf{c}_3\mathbf{A})^T \ (\mathbf{c}_3\mathbf{A}^2)^T \right]^T$$

The above step is not only unnecessary, but counter productive in that the resulting structure is more complex!

The idea behind the POFs is that the selection of the n linearly independent rows of $\mathbf{Q}_o$ can be done in many ways, according to the indices $\{n_1,n_2,...,n_p\}$, representing the number of units in the p-columns of the crate. The indices must, of course, sum to n. Each possibility must be checked for "*admissibility degree*" of a full rank matrix, i.e. that the resulting n-rows are, in fact, linearly independent. The admissible POFs are then all possible structures for the MIMO system. Investigation of the various POFs for a particular system indicates that some forms are better than others in terms of the condition number of the transformation matrix **T**. A poorly conditioned transformation matrix typically results in a large range of parameter values in the POF, as well as loss of numerical accuracy in the model.

2.6. MIMO System Parameter Determination

In Section 2.4 the POF was introduced. The key is in the set of indices specified for the POF in that everything related to the system structure is determined from them. In practice it is useful to establish an algorithm which will construct the POF given a basic state space model and the information of the indices.

System identification from input/output data assumes that the input signals are "persistently exciting," i.e. that the system is sufficiently excited to exhibit all of its modes in the corresponding output signals. In addition, it is clear that only the controllable and observable part of the system can be identified from input/output data. In Section 2.5 the existence of an observable state space representation having the structure Eqn. (21) has been established. In other words,

by a similarity transformation **T**, given by Eqn. (24), any observable realization $R = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ can be transformed into the realization $R_o = \{\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o, \mathbf{D}_o\}$ as described in Eqn. (21). In this section we describe the *Identification Identity* which relates the input/output data generated from the different initial conditions to the matrices of R_o. Since $\mathbf{C}_o$ is completely specified, as is the structure of $\mathbf{A}_o$, it remains to relate the data to matrices $\mathbf{B}_o$, $\mathbf{D}_o$ and the unspecified p rows of $\mathbf{A}_o$.

To this end, consider the desired result of the identification, namely, an order-n *D-T* system with m-inputs and p-outputs:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_o\mathbf{x}(k) + \mathbf{B}_o\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_o\mathbf{x}(k) + \mathbf{D}_o\mathbf{u}(k) \end{aligned} \tag{25}$$

where $\{\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o, \mathbf{D}_o\}$ is in the observable form corresponding to a set of admissible (pseudo) observability indices $n_o = \{n_i\}$, $i=[1,p]$. From Eqn. (25) we may write

$$\begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \\ \vdots \\ \mathbf{y}(k+r) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_o \\ \mathbf{C}_o\mathbf{A}_o \\ \vdots \\ \mathbf{C}_o\mathbf{A}_o^{\,r} \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \mathbf{D}_o & 0 & \cdots & 0 & 0 \\ \mathbf{C}_o\mathbf{B}_o & \mathbf{D}_o & \cdots & 0 & 0 \\ & & \cdots & & \\ \mathbf{C}_o\mathbf{A}_o^{\,r-1}\mathbf{B}_o & \cdots & \mathbf{C}_o\mathbf{A}_o\mathbf{B}_o & \mathbf{C}_o\mathbf{B}_o & \mathbf{D}_o \end{bmatrix} \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+r) \end{bmatrix} \tag{26}$$

Now we let $r = n_x$. Clearly, Eqn. (26) holds for any $k = [0, N-r]$ and can be rewritten as

$$\mathbf{y}_k = \mathbf{Q}_\infty \mathbf{x}(k) + \mathbf{H}\mathbf{u}_k \tag{27}$$

where $\mathbf{y}_k$ and $\mathbf{u}_k$ are $(n_x+1)p$ and $(n_x+1)m$ dimensional columns containing output

and input vectors $\mathbf{y}(k+j)$ and $\mathbf{u}(k+j)$, $j = [0, n_x]$. The matrix $\mathbf{Q}_{\infty}$ is the observability matrix of the pair $\{\mathbf{A}_o, \mathbf{C}_o\}$, while $\mathbf{H}$ is the $(r+1)p \times (r+1)m$ lower block triangular matrix containing along the main diagonal the $(p \times m)$ blocks $\mathbf{D}_o$. The other nonzero blocks of $\mathbf{H}$ are the $p \times m$ dimensional *Markov parameters*:

$$\mathbf{C}_o\mathbf{A}_o^j\mathbf{B}_o \ , \quad \text{for } j = [0, n_x - 1] \tag{28}$$

Our goal is to eliminate from Eqn. (26) the $\mathbf{x}(k)$ terms, thereby obtaining the *Identification Identity*, which relates the available sampled data to the unknown elements in R_o.

Equation (26) can be considered to represent $(n_x+1)p$ scalar equations in the samples

$$y_{ij} = y_i(k+j) \tag{29}$$

i.e. the i^{th} element of the output vector $\mathbf{y}(k+j)$, $i=[1, p]$, $j=[0, n_x]$. In Section 2.5 it was shown that $\mathbf{Q}_{\infty}$ has n rows of an identity matrix and p rows that correspond to the unknown rows of $\mathbf{A}_o$. Furthermore, the location of these rows are determined by the selector vectors $\mathbf{v}_{li}$ and $\mathbf{v}_{ld}$, respectively.

Premultiplying Eqn. (27) by the selector matrices $\mathbf{S}_{li}^{\mathrm{T}}$ and $\mathbf{S}_{ld}^{\mathrm{T}}$ defined by Eqn. (20), we obtain, respectively,

$$\mathbf{y}_{1k} = \mathbf{x}(k) + \mathbf{H}_1\mathbf{u}_k \ , \quad \text{and} \quad \mathbf{y}_{2k} = \mathbf{A}_r\mathbf{x}(k) + \mathbf{H}_2\mathbf{u}_k \tag{30}$$

where

$$\mathbf{y}_{1k} = \mathbf{S}_{li}^T\mathbf{y}_k \ , \ \mathbf{y}_{2k} = \mathbf{S}_{ld}^T\mathbf{y}_k \quad \text{with} \quad \mathbf{H}_1 = \mathbf{S}_{li}^T\mathbf{H} \ , \ \mathbf{H}_2 = \mathbf{S}_{ld}^T\mathbf{H}$$

Eliminating $\mathbf{x}(k)$ from Eqn. (30),

$$\mathbf{y}_{2k} = \begin{bmatrix} (\mathbf{H}_2 - \mathbf{A}_r\mathbf{H}_1) & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{y}_{1k} \end{bmatrix} \tag{31}$$

The matrix $\mathbf{A}_r$ in Eqn. (30) and Eqn. (31) is a $(p \times n)$ matrix containing the unknown rows of $\mathbf{A}_o$, whose locations in $\mathbf{A}_o$ are specified by the selector vector $\mathbf{v}_a$. Equation (31) may be written in a more concise form given by

$$\mathbf{y}_{2k} = \begin{bmatrix} \mathbf{N}_r & \mathbf{A}_r \end{bmatrix} \mathbf{z}_k \tag{32}$$

$$\mathbf{Z}_k = \begin{bmatrix} \mathbf{U}_k \\ \mathbf{Y}_{1k} \end{bmatrix} \begin{matrix} \} \ (n_x+1)m \\ \} \ n \end{matrix} \tag{33}$$

where $\mathbf{N}_r = \mathbf{H}_2 - \mathbf{A}_r\mathbf{H}_1$ is a $p \times (n_x+1)m$ matrix and $\mathbf{z}_k$ is an h-dimensional vector of data where $h = (n_x+1)m + n$. Note that the matrix $\mathbf{U}_l$, for $l=n_x$, given in Eqn. (11) is equal to $\mathbf{U}_k$ in Eqn. (33). Also, $\mathbf{Y}_{1k}$ and $\mathbf{Y}_{2k}$ in Eqn. (33) could be obtained from $\mathbf{Y}_l$ in Eqn. (12) by premultiplying it with selector matrices $\mathbf{S}_{li}^T$ and $\mathbf{S}_{ld}^T$, respectively. In other words, $\mathbf{Y}_{1k}$ and $\mathbf{Y}_{2k}$ are obtained from $\mathbf{Y}_l$ in Eqn. (12) "according to" the selector vectors $\mathbf{v}_{li}$ and $\mathbf{v}_{ld}$, respectively. Equation (32) is referred to as the ***Identification Identity*** since it relates input/output data samples arranged into columns $\mathbf{y}_{2k}$ and $\mathbf{z}_k$ to the unknown parameters of the state space representation R_o, i.e. in the matrices $\mathbf{A}_o$, $\mathbf{B}_o$ and $\mathbf{D}_o$.

We now consider the case where only input/output data is available, without a given system model. The process of creating a system model from the data is called system identification. A deterministic D-T system identification will be performed by calculating an observable form state space model $R_o = \{\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o,$

$\mathbf{D}_o\}$ from a set of input and corresponding output data with the restriction that the input signals are "persistently exciting." The technique is based on the ***Identification Identity***, Eqn. (32). In order to determine $\mathbf{N}_r$ and $\mathbf{A}_r$, which contains the parameter information for R_o, Eqn. (32) can be solved in terms of the *pseudoinverse* of $\mathbf{z}_k$; that is,

$$[\ \mathbf{N}_r \quad \mathbf{A}_r\ ] = \mathbf{y}_{2k}\mathbf{z}_k^t(\mathbf{z}_k\mathbf{z}_k^t)^{-1} \tag{34}$$

In practice, however, nonsensical values are obtained if the determinant of $\mathbf{z}_k\mathbf{z}_k^t$ is vanishingly small. After trying all possible sets of POIs for a better conditioning of $\mathbf{z}_k$, the following back-propagation algorithm is suggested.

The back-propagation algorithm, or *generalized delta rule*, is a gradient descent procedure in *weight space*. There are three stages to developing a neural net system model. The first and most important one is determining the dynamic structure. The other two involve the interconnections that are internal to the MLP, namely, the number and type of layers and the number of elements in each layer. Fig. 4 illustrates a general form that has proven to be useful for system identification. In the *linear case*, i.e. with no nonlinear elements included in the ANN, Fig. 4 simply implements an *ARMA* model of order *h*, similar to the *LMS algorithm*, Widrow and Stearns [44], which corresponds to the linear difference equation

$$\mathbf{y}(k) = \mathbf{A}_1\,\mathbf{y}(k-1) + \cdots + \mathbf{A}_{n_i}\,\mathbf{y}(k-n_i) + \mathbf{N}_0\,\mathbf{u}(k) + \cdots + \mathbf{N}_{n_x}\,\mathbf{u}(k-n_x) \tag{35}$$

where the *observability index*, n_x, already defined by

$$n_x = \max_{i=[1,p]} \{\, n_i \,\}$$

From the above Eqn. (35) the system order n is

$$n = \sum_{i=[1,p]} \{ n_i \}$$

Since much information is available on the structure of strictly linear systems, as discussed in Sections 2.4 and 2.5, we should be careful not to over simplify the neural net approach. But in very basic terms the required order of the ANN dynamic model, as shown in Fig. 4, can be obtained by the conventional method of system identification technique instead of "training" the net for a selected value of h, which is typically fairly low initially, and subsequently incrementing h if the "training" results are not satisfactory. Thus, the ANN "learns" the matrices $\{\mathbf{A}_i\}$, $1 \le i \le n_i$, and $\{\mathbf{N}_i\}$, $0 \le i \le n_x$ in Eqn. (35), or, more generally, Eqn. (36) below. Using the z-domain description, Eqn. (35) may be rewritten as

$$(\mathbf{I} - \mathbf{A}_1 z^{-1} - \cdots - \mathbf{A}_{n_i} z^{-n_i})\, \boldsymbol{y}(z) = (\mathbf{N}_0 + \mathbf{N}_1 z^{-1} + \cdots + \mathbf{N}_{n_x} z^{-n_x})\, \boldsymbol{u}(z) \qquad (36)$$

The representation of Eqn. (36) is sometimes referred to as *Matrix Fraction Description (MFD)*. We will use this form to relate ANN identification to the conventional method.

It is relatively easy to transform an *ARMA* (or *MFD*) model to the state space realization R_o. Thus, the problem, even in the linear case, is to choose h. In a more general case the structure of the ANN would include a nonlinear layer as shown in Fig. 4. This would be done to capture any nonlinearities in the system. The present situation is a restriction to a linear model and, as such, obviates the inclusion of nonlinear elements at this initial modeling stage. One useful technique to obtain the model of a nonlinear system would be to begin with only a linear MLP and subsequently add an increasing number of nonlinear processing elements

in the process of "matching" the model to the data. There is very little difference in the difficulty of training an MLP with, or without, a nonlinear layer.

The "training" of the ANN is divided into two phases, deciding on the internal structure (number of hidden layer nodes, learning parameters, etc.), and using the input/output data to adjust the network weights. Only at this last stage is the *back-propagation algorithm* utilized. Back-propagation can be described by considering a single pair of data consisting of an *input vector*, which from Fig. 5 would be of dimension $(n_x + 1)m + n$, which must be based on the best POF for the given system, and a *desired output vector*,of dimension p. This structure is what we have called a ***modified dynamic* MLP**.

First the input is passed through the network, which was initiated with small random weights. Next the output **y** is compared to the "desired output", the second part of the data. The *error, E,* can be defined in different ways, but for simplicity we will assume that it is defined as

$$E = \|\mathbf{y}_{2k} - \mathbf{y}_d\|^2 = \sum_{i=1}^{p} |y_{2ki} - y_{di}|^2 \tag{37}$$

Then, if we assign the symbol w to be a generic weight from either the array **A**, **B**, or **C** shown in Fig. 4, each weight is incremented proportionally to its effect on reducing the error, E,

$$w_{new} = w_{old} + \Delta w \ , \quad \text{where} \quad \Delta w = -\frac{\partial E}{\partial w} \tag{38}$$

Running the back-propagation algorithm using the available data, "automatically" adapts the weights (with guidance from the designer) to the "correct" values, i.e. identifies the D-T system model.

In the next section a step-by-step procedure is suggested for extracting the required arrays.

2.7. Algorithm

After determining the system order, n, and the observability index, n_x, one must select an admissible set of (pseudo) observability indices, $n_o = \{n_i\}$; see Section 2.5. Using back-propagation learning method, the matrices $\mathbf{A}_r$ and $\mathbf{N}_r$ defined in Eqn. (32) are estimated. This process can be symbolized as

$$\mathbf{U}, \mathbf{Y}_1, \mathbf{Y}_2 \rightarrow \mathbf{A}_r, \mathbf{N}_r \tag{39}$$

where the k subscript notation has been dropped for convenience. Once $\mathbf{A}_r$ and $\mathbf{N}_r$ have been determined, R_o may be found, as formalized in the following steps:

1. Set $\mathbf{I}_n \Rightarrow$ $\begin{bmatrix} \mathbf{C}_o \\ \mathbf{I}_2 \end{bmatrix} \begin{matrix} \} p \\ \} n-p \end{matrix}$

2. Set $\mathbf{S}_{li}\, \mathbf{I}_2 + \mathbf{S}_a\, \mathbf{A}_r \Rightarrow \mathbf{A}_o$

3. Partition $\mathbf{B}_r$ into $p \times m$ blocks:

   $\mathbf{N}_r \Rightarrow [\mathbf{X}_0\ \mathbf{X}_1\ \cdots\ \mathbf{X}_{nx}]$ where each block has m-columns,

   and similarly,

   Partition $\mathbf{A}_r$ into two blocks:

   $\mathbf{A}_r \Rightarrow [\mathbf{A}_{r0}\ \mathbf{A}_{r1}]$ where the block $\mathbf{A}_{r0}$ has

dimensions $p \times p$.

4. Concatenate $\begin{bmatrix} \mathbf{X}_0 \\ \vdots \\ \mathbf{X}_{nx} \end{bmatrix}$ $\Rightarrow \mathbf{B}_c$, a $(n_x+1)p \times m$ matrix

5. Calculate the $n \times (n_x+1)m$ controllability matrix $\mathbf{Q}_c$ of the pair $\{\mathbf{A}_o, \mathbf{S}_a\}$ having n_x+1 blocks, i.e.

$$\mathbf{Q}_c = \left[\mathbf{S}_a \quad \mathbf{A}_o\mathbf{S}_a \quad \cdots \quad (\mathbf{A}_o)^{n_x}\mathbf{S}_a\right]$$

6. Set $\mathbf{Q}_c \mathbf{B}_c \Rightarrow \mathbf{B}_o$
7. Set $\mathbf{C}_o \mathbf{A}_o^{-1} \mathbf{B}_o - \mathbf{X}_o \mathbf{A}_{ro}^{-1} \Rightarrow \mathbf{D}_o$

This algorithm can be considered symbolically as $\{ \mathbf{u}(k), \mathbf{y}(k) \} \Rightarrow \mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o, \mathbf{D}_o$, which simply applies the relationship between an input/output stream and a corresponding state space description in a selected pseudo-observable form to obtain an identified system model. The steps involved in the above algorithm may all be verified by relatively straightforward, but rather tedious, matrix calculations. In the following chapter the technique for system identification applied to the power electronics systems is explained.

3. System Identification of Highly Nonlinear Power Electronics System

3.1. Introduction

To provide an efficient power conversion, a power electronics converter is composed of high speed semiconductor switches and reactive elements. The basic function of the converter, delivering energy from source to load in an efficient and regulated manner, is achieved by a switchable configuration of reactive elements and on-off action of switches. The existence of switches allows the converter to have more than one circuit structure. Depending on the state of switches, the converter may have entirely different circuit configurations.

The area of modeling and analysis of power electronics systems, owing to their inherent nonlinear nature, has been a very difficult task in view of the lack of adequate analysis tools at the disposal of the circuit designer working in the field. Due to the increased speed, accuracy and smaller size of today's high performance regulators, new and more complex converter topologies have been continuously developed. It becomes an even more challenging task to develop a generalized modeling tool to analyze and design new circuit topologies.

This chapter addresses the topic of modeling of highly nonlinear power electronics systems. As an application of identifying an unknown plant in power electronics systems, an empirical data modeling approach is presented which aims at generating small-signal linear equivalent models as well as large-signal nonlinear models for a general class of converters, which includes resonant and PWM type converters.

Small-signal modeling is used to provide the circuit designer with analytical tools which are accurate enough for some practical purpose, yet simple enough to give him or her powerful tools for design-oriented analysis. This analysis, through appropriate linear circuit models, provides the necessary insight which can lead to near optimum performance.

In contrast, large-signal modeling is used to simulate the dynamics of , say, converters under a disturbance-in-large such as step line voltage or step load current. As mentioned above, a small-signal model enables one to apply linear system control theories to design a feedback controller which provides optimum small-signal characteristics. However, the small-signal model cannot describe the behavior of the converter during a large transient.

Section 3.2 presents small-signal modeling of various type of converters employing the ***robust hybrid identification technique***. Several approaches, known as state-space averaging, discrete-time modeling, harmonic-balancing and describing function method have been employed in an attempt to generalize small-signal models for PWM type converters and also resonant converters [45,46,47,48,49,50]. However, the results often tend to be complicated and difficult to apply. The proposed modeling technique adopts a data analysis point of view; i.e. an input/output approach, treating the system of interest as a black box and identifying the unknown model based on its input/output time-domain response [51]. The approach is simple and is suitable for identification of various type of converters and subsystems. The robust hybrid identification technique of the previous chapter is used as an identification tool for small-signal modeling of subsystems in a distributed power system.

To demonstrate the validity of the technique, this method is used on power

electronics systems in order to provide discrete-time small-signal models not only for PWM type converters, but also for resonant type converters. The resulting small-signal model describes the converter as a linear time invariant system, and the knowledge of the identified linear system can be applied to the switching converters for constructing feedback controllers. In this section small-signal modeling of power converters, including an open-loop boost converter, a series-resonant converter (SRC) and a forward multi-resonant converter (FMRC) is presented. The identification results are compared with the analytical model and experimental data.

Section 3.3 presents large-signal modeling of converters, including all the nonlinearities. The small-signal approximation cannot represent the nonlinear characteristics of the system, which become significant for large perturbations. Due to the limitation of small-signal modeling of a power electronics system, the resulting feedback controller might fail to control the system beyond the assumed small-signal boundary. This work was motivated by the desire to extend the models of systems, which are linear for small signal variation around an operating point, well into their nonlinear range. Since constructing a linear feedback controller is based on the linearized model of the nonlinear system around an equilibrium point, the linearization must be done to reduce the dynamics of a nonlinear system to a linear model. However, when the system operates deeper into its nonlinear range where the assumption of linearity is no longer valid, the system with a linear controller may be unstable. In Section 3.3, the approach is to use an MLP network with inputs representing both the actual inputs and delayed versions of both the inputs and outputs to capture the nonlinearities and the dynamics of the system for an approximation of boundaries into the possible unstable region in large-scale

simulation. As examples, a boost converter and a buck converter which are stable for small signals but found to be unstable or oscillatory for large perturbations are simulated as physical systems to be identified and the comparisons of the large-signal stability analysis of the quiescent operating point between the real system and the identified one using the MLP are reported [52]. In this section, large-signal modeling of both an open-loop boost converter and a closed-loop buck converter is presented. Due to the highly nonlinear characteristics of power converter systems, an MLP is used to identify the transient response of the converters in the large-scale system simulation. A well-trained MLP shows the nonlinear behavior of the system beyond the linear region and also provides a close approximation of the boundaries where the system goes unstable with the linear feedback controller.

A prominent feature of this empirical data modeling method is that no specific model information is required throughout the identification process. The simulation results indicate that this modeling method is well suited for identifying complex nonlinear power electronics systems.

3.2. Small-Signal Modeling of Power Converters

Switching converters are inherently nonlinear oscillatory systems. A switching converter consists of linear resistors, inductors, capacitors, as well as nonlinear magnetic components and semiconductor switches. Especially, due to the severe nonlinear characteristics of magnetic components and switching devices, it is very difficult to design stable feedback controllers using exact mathematical descriptions of switching converters. Usually, switching converters have too many complex nonlinear differential equations to be solved. Therefore, it is generally not feasible to construct design guidelines to regulate a converter in a large-signal domain. Instead, small-signal models are commonly used to provide dynamic information of the switching converters for control purposes, where the converter can be linearized around a specific operating point. Since the control issues of switching converters can be treated very effectively by small-signal analysis, the resulting small-signal models are very useful to design engineers on the ground that all of the relatively simple techniques of linear system control theory can be applied easily to the small-signal model. Therefore, practicing engineers may acquire the physical insight of the given system for developing a proper feedback controller. From this small-signal model important specifications such as audio susceptibility, loop-gain and output impedance are calculated. Additionally, these specifications can be easily measured whenever the small-signal model and/or the controller based on this model needs to be verified experimentally.

For the past decades, state-space averaging is a commonly used modeling approach for small-signal modeling of switching converters. This method was originally proposed to model PWM converters. For properly designed PWM converters, the natural frequencies of each linear circuit are much lower than the

switching frequency. This provides justification of the linear ripple assumption. Under the assumption that the natural frequency of the converter power stage is well below the switching frequency, the averaging technique can provide approximate linear solutions of a nonlinear averaged state equation. Then, the small-signal model can be derived by "persistently exciting" input signals around a particular operating point. The obtained small-signal model has a continuous form. The model can predict the dynamics of PWM type converter power stages accurately up to the half of the switching frequency. The analysis of state-space averaging is simplified by using a circuit averaging technique based on three-terminal PWM switch model [53]. However, this averaging concept does not apply for resonant converters and multi-resonant converters where the energy of state variables is carried mainly by switching harmonics but not by the low frequency components as in the case of PWM type converters. For resonant converters and multi-resonant converters, the dynamics are often determined by the interaction between the switching frequency and the natural resonant frequency of the converter [54]. This interaction cannot be investigated using averaging concept because it eliminates the switching frequency information.

Another systematic modeling method to obtain small-signal models for switching converters is a discrete-time (D-T) or a sampled-data modeling approach. By solving the nonlinear state equations in the time-domain, a steady-state analysis can be done under given operating conditions. Perturbation of this nonlinear equation around a specific operating point provides the small-signal dynamics with a sample interval the same as the switching frequency.

In this section, small-signal modeling of a PWM type boost converter, a series-resonant converter and a forward multi-resonant converter are chosen to apply the robust hybrid identification technique. This empirical data modeling

approach is generally simpler and independent of type of converters. Also, since this approach is a model-free identification, internal structure need not be known as long as one can obtain the data either through a time-domain simulation or a hardware measurement. This approach is also very effective to generate a reduced order model to represent a complex subsystem in a distributed power system.

3.2.A. Open-Loop Boost Converter

As an example of the small-signal modeling of nonlinear dynamic systems under study, an open-loop boost converter is selected. Since existing state-space averaged model is quite accurate up to the half of the switching frequency, the proposed modeling approach can be compared and verified its effectiveness and accuracy. Fig. 7 illustrates a typical two-state boost converter example.

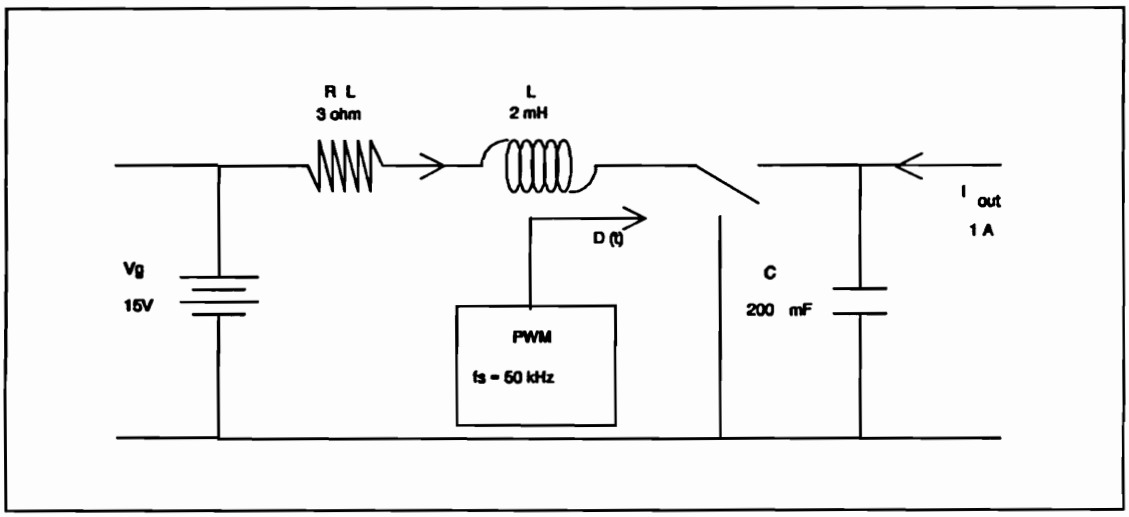


Fig. 7. Boost Converter (with PWM Control over the Switch)

Three input variables and two output variables represent the state of the system dynamics: $\hat{v}_g$ (the variation of input voltage), $\hat{i}_o$ (the variation of input current), and

$\hat{d}$ (the variation of duty cycle), $\hat{i}_l$ (the variation of output inductor current), $\hat{v}_c$ (the variation of output capacitor voltage). This converter was designed to operate at a nominal duty ratio of 0.6 with an efficiency of 70.5%. The exact discrete state-space equation including all the nonlinearities are used for the time-domain simulation.

As described in detail in Chapter 2, the modeling procedure is summarized as follows:

Step 1: A small range of elaborate input perturbations around a nominal equilibrium point is injected at the inputs of the boost converter, such as, $\hat{v}_g$, $\hat{i}_o$ and $\hat{d}$, and then the corresponding output responses are measured in physical unit. Generally, small-signal modeling of an unknown system, unlike the above boost converter, must be done using a circuit simulation tool such as SPICE or the measurement data from the hardware directly. Therefore, extracting information from data is not a straightforward task. In addition to the decisions required for model structure selection and generalization, the collected data need to be handled carefully for the robust identification process. The levels in these raw inputs and outputs should be matched in a consistent way. The mean levels must be subtracted from the input and output sequences before the estimation. The best way is to match the mean levels corresponding to a system equilibrium.

Step 2: The second step is to determine a nominal range of the system order with the restriction that the input signals are "persistently exciting." From the assumption that the order of the system is unknown, to determine the system order from raw data, a rank test is done. However, due to the nonlinearity of the system with added noise, a rank test may not be reliable.

Step 3: The third step is to construct an ARMA model with inputs representing both the present inputs and delayed versions of the inputs and outputs

to capture the dynamics of the systems. The method is determining an equilibrium point to identify a linearized system about this equilibrium point. A combination of the connectionist approach and a classical method of multivariable system identification which utilizes the possible structures of the system in order to achieve a model that optimally generalizes over the available input/output data.

Step 4: The final step, a deterministic D-T system identification, is performed by calculating an observable form state-space model $R_o = \{A, B, C, D\}$ from the identified ARMA model. The small-signal modeling process is shown in Fig. 8.

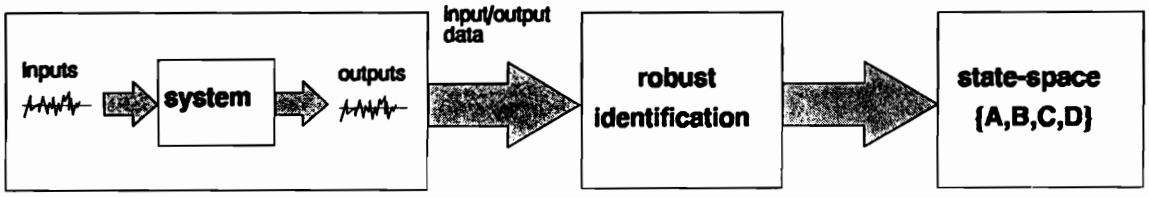


Fig. 8. Small-Signal Modeling Process

Using the proposed identification technique, a small-signal model of the boost converter is developed as the following ARMA model:

$$y(k) = \sum_{i=1}^{n} a_{k-i} y(k-i) + \sum_{i=0}^{n} b_{k-i} u(k-i) \qquad (40)$$

where $y_1 = \hat{i}_l$, $y_2 = \hat{v}_c$, $u_1 = \hat{v}_g$, $u_2 = \hat{i}_o$ and $u_3 = \hat{d}$. From Eqn. (40) it is noted that delayed inputs and outputs contribute to the "predicted" output. Since the boost converter is second order, the ARMA model of the linearized system is expected to have $u_1(k)$, $u_1(k-1)$, $u_2(k)$, $u_2(k-1)$, $u_3(k)$, $u_3(k-1)$, $y_1(k-1)$ and $y_2(k-1)$ terms. After training 8 inputs and 2 outputs with the standard back-propagation learning rule, the results are shown in Figs. 9 and 10.

In order to check the generalization of the trained MLP over the unused data in training, the responses of two output variables, $\hat{i}_l$ (the variation of output inductor current), $\hat{v}_c$ (the variation of output capacitor voltage) of the boost converter to samples of three different random input variables, $\hat{v}_g$ (the variation of input voltage), $\hat{i}_o$ (the variation of input current), and $\hat{d}$ (the variation of duty cycle) in Fig. 11 are shown in Figs. 12 and 13. For reference, the eigenvalues of the C-T equivalent model of Eqn. (40) are {-1166.6, -347.5} comparing with those of state-space averaged model of the exact system equation, {-1153.1, -346.9}. When duty cycle, $\hat{d}$, is modulated, the magnitude and the phase of the control-to-output transfer function of the identified model are compared against the state-space averaged model of the system in Figs. 14 and 15, respectively. The obtained small-signal model is accurate up to the half of the switching frequency, i.e. 25KHz (Nyquist frequency).

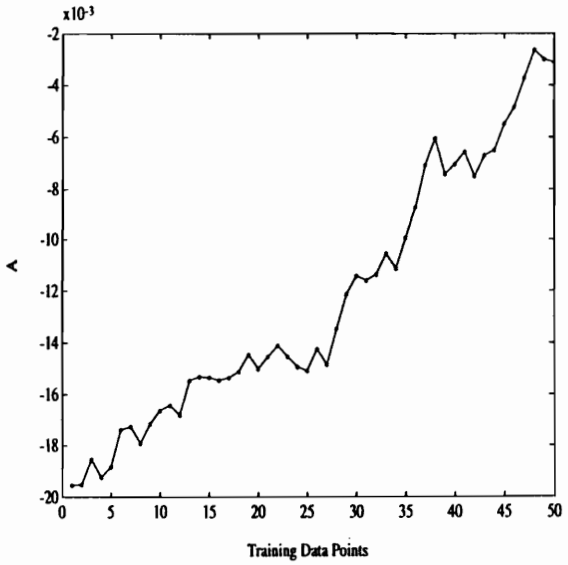


Fig. 9. Identified Training Model:
Given $\hat{i}_l$ (solid), Model $\hat{i}_l$ (*)

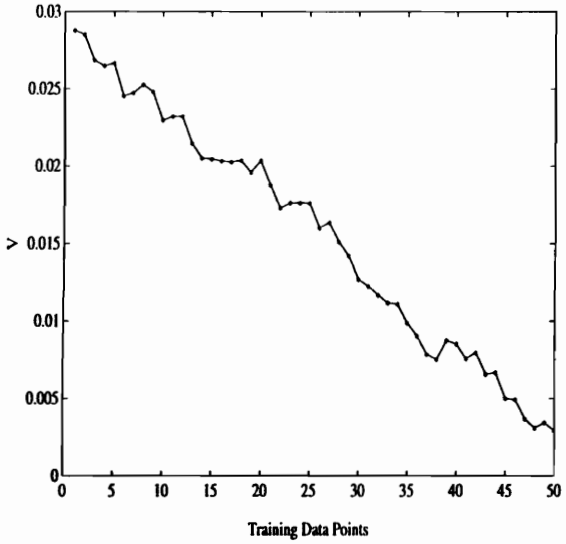


Fig. 10. Identified Training Model: Given $\hat{v}_c$(solid), Model $\hat{v}_c$ (*)

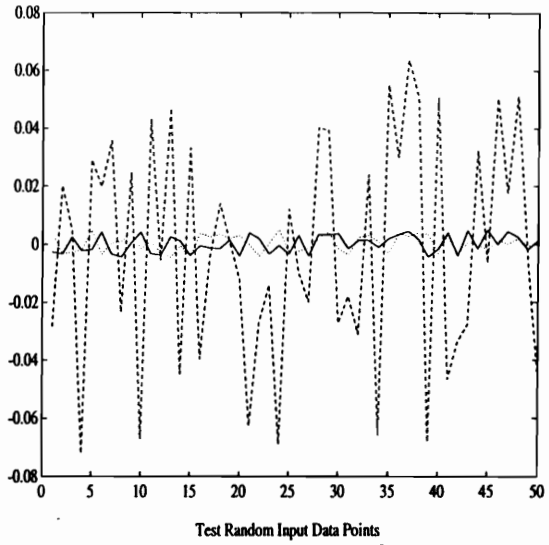


Fig. 11. Input Signals: $\hat{d}$ (solid), $\hat{v}_g$ (dashed), $\hat{i}_o$ (dotted)

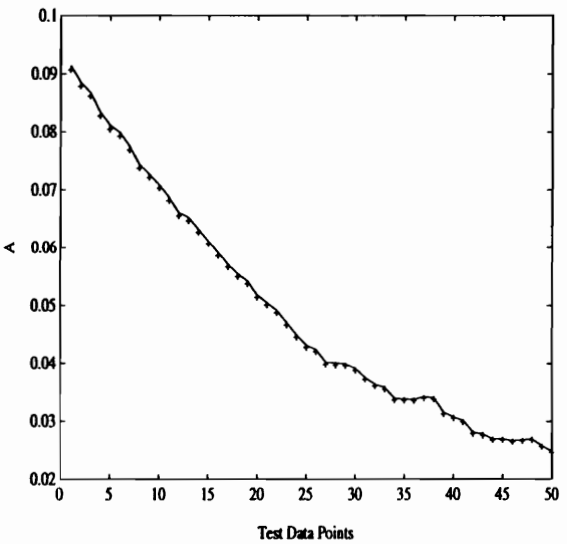


Fig. 12. Identified Test Model: Given $\hat{i}_l$ (solid), Model $\hat{i}_l$ (+)

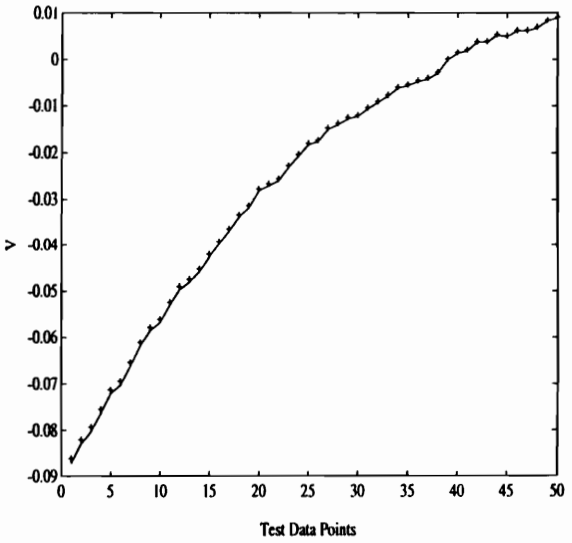


Fig. 13. Identified Test Model: Given $\hat{v}_c$ (solid), Model $\hat{v}_c$ (+)

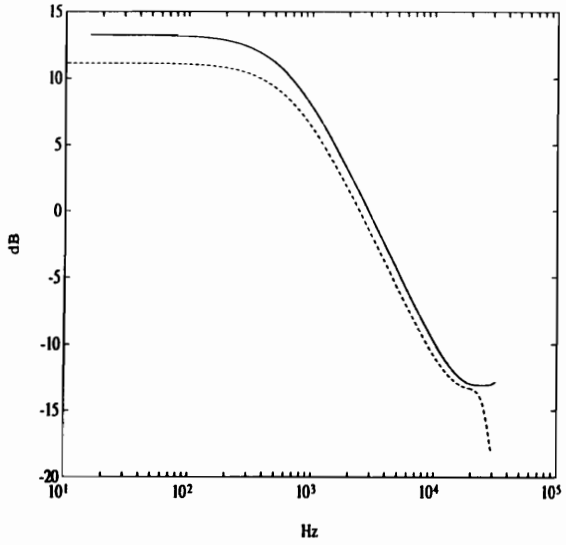


Fig. 14. Magnitude: Identified (solid), State-Space Averaging (dotted)

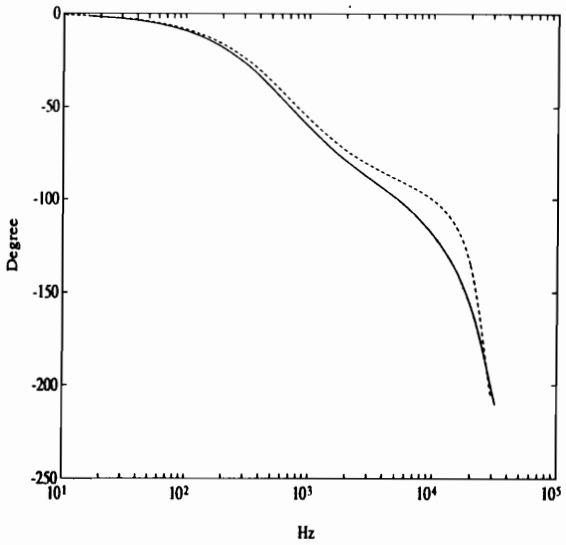


Fig. 15. Phase: Identified (solid), State-Space Averaging (dotted)

3.2.B. Series-Resonant Converter (SRC)

The small-signal modeling approach for a series-resonant converter (SRC) based on the proposed robust hybrid identification technique is discussed in this section. Among several approaches for modeling a SRC, the well-known state-space averaging technique does not show promising results in modeling for resonant converters, where the energy of the system is carried mainly by the switching frequency harmonics (not by the low frequency components as in the case of PWM converters). Since the dynamics are often determined by the interaction between the switching frequency and the natural frequency of the resonant converter, state-space averaging eliminates the useful information of this interaction between both frequencies. Therefore, the previous identification procedure was applied to the input/output data streams of nonlinear system

equations of a SRC. The identified model was compared with the analytical result to verify the correctness of the procedure. The circuit diagram of the SRC is shown in Fig. 16.

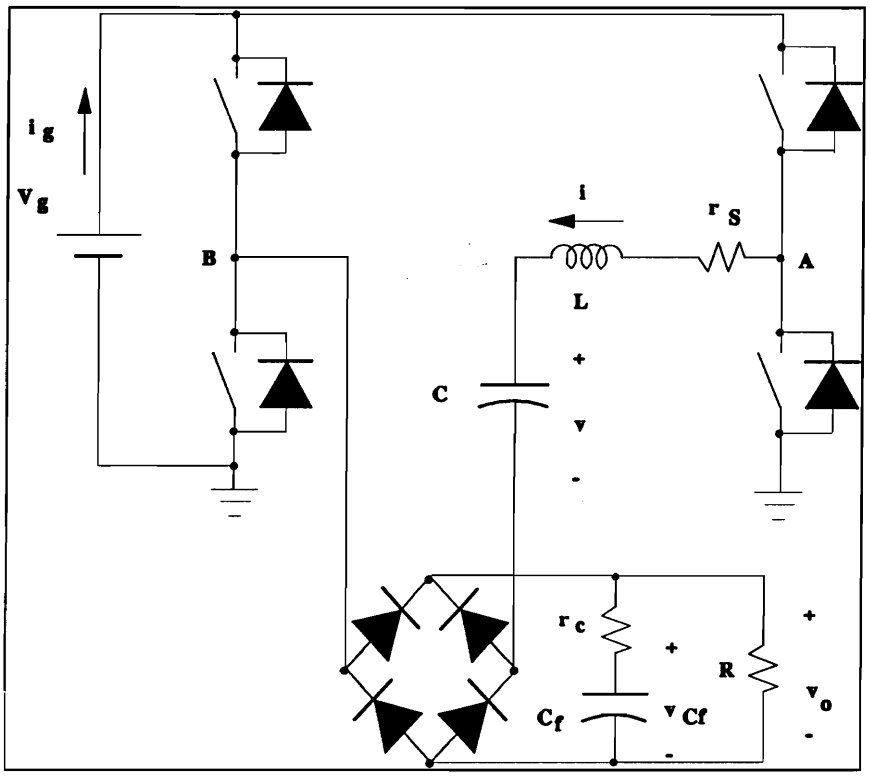


Fig. 16. Series Resonant Converter

The active switch network generates a quasi-square voltage, v_{AB}, applied to the resonant tank. By assuming the continuous mode of the inductor current (tank current, i), the SRC can be modeled as in the following nonlinear state equations [55]:

$$L\frac{di}{dt} + v + sgn(i)v_o = v_{AB}$$

$$C\frac{dv}{dt} = i \qquad (41)$$

$$C_f\, d\frac{v_o}{dt} + \frac{v_o}{R} = |\, i\,|$$

There are three input variables, $\hat{v}_g$ (the variation of input voltage), $\hat{i}_o$ (the variation of output current), and $\hat{f}$ (the variation of switching frequency) and three output variables, $\hat{i}_g$ (the averaged input current), $\hat{v}_{Cf}$ (the capacitor, C_f, output voltage) and $\hat{v}_o$ (the output voltage) of the power stage. In this configuration, the output voltage is regulated by modulating the switching frequency, $\hat{f}$. The circuit parameters and the operating point are the following:

$L = 197\ \mu H,$	$C = 51\ nF$	
$C_f = 32\ \mu F,$	$r_c = 0,$	$r_s = 0$
$F_o = 50.2\ KHz,$	$Z_o = 62.1\ \Omega$	
$V_g = 62.15\ V,$	$D = 0.95$	
$F_s/F_o = 0.85,$	$Q_s = Z_o/R = 2.5$	

The resulting small-signal model of the SRC is given below as a state-space representation form $R_o = \{A, B, C, D\}$:

$$\begin{bmatrix} \mathbf{A} \\ \text{---} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} .3367 & -.0094 & -.0011 \\ 23.5832 & .7523 & .3472 \\ 1.4497 & -.0345 & .9702 \\ \text{-----} & \text{-----} & \text{-----} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and (42)

$$\begin{bmatrix} \mathbf{B} \\ \text{---} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} .4396 & .0014 & .0110 \\ -24.2318 & .4943 & -.2752 \\ 2.6661 & .0113 & -.6898 \\ \text{-----} & \text{-----} & \text{-----} \\ .3282 & .0083 & .0074 \\ -79.4499 & .1109 & -.3381 \\ .6759 & .0108 & -.4351 \end{bmatrix}$$

Figs. 17 and 18 show the control-to-output transfer function of the SRC compared against the measured data. The numerical results are in good agreement with the measured data.

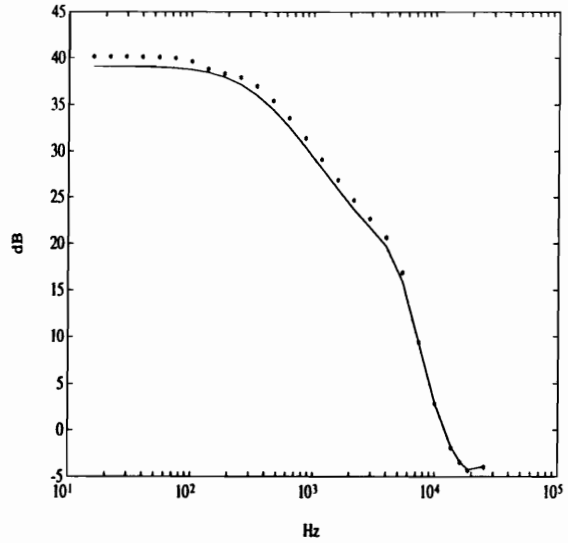


Fig. 17. Magnitude:
Identified (solid), Measured (*)

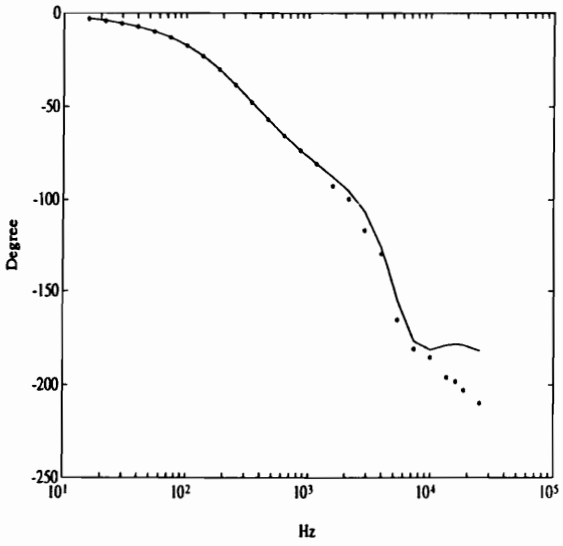


Fig. 18. Phase:
Identified (solid), Measured (*)

3.2.C. Forward Multi-Resonant Converter

In this section, a forward multi-resonant converter (FMRC) is selected to demonstrate the proposed modeling method. The FMRC has strong oscillatory nature, where switching frequency plays important roles for energy delivery and the interactions exist between the natural resonant frequencies and the switching frequency. The FMRC is especially suitable for distributed power systems, where small volume, high frequency operation and low EMI are required. Despite the advantages of this converter, it is very complex circuit because of its structure and operation. Therefore, the FMRC is a very challenging circuit for small-signal modeling because of its complexity. The circuit diagram of the FMRC is shown in Fig. 19.

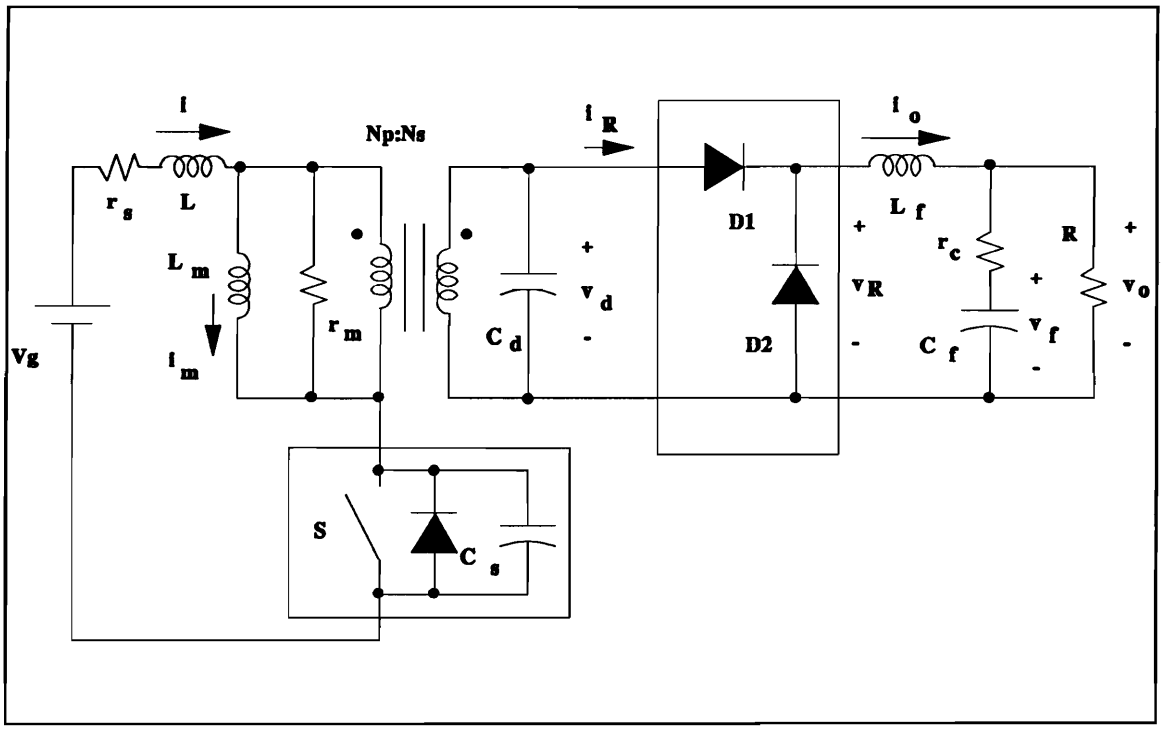


Fig. 19. Forward Multi-Resonant Converter

The whole circuit has six energy storage elements and the resonant tank formed by L, C_s and C_d is the third order. The magnetizing inductance is denoted by L_m. The FMRC is controlled by varying the turn-on time of the active switch (S) and the turn-off time is fixed. The advantage of this topology is that it absorbs the major circuit parastics such as the output capacitance of the switching devices, the junction capacitance of the rectifier diodes and the leakage inductance of the transformer. Furthermore, all of the semiconductor devices can be operated under zero voltage switching condition with proper design. Since several operating modes exist for this circuit with respect to different loads and switching frequencies and therefore, it is not feasible to obtain dc characteristics analytically. Especially, there are two resonant frequencies for the FMRC, corresponding to the turn-on period and the turn-off period, respectively. This makes the FMRC a more difficult circuit than the other resonant converters for modeling. The FMRC are built as the following parameters:

$L = 45\ \mu H,$	$C_s = 48.2\ nF,$	$L_m = 139\ \mu H$
$C_d = 661\ nF,$	$L_f = 82\ \mu H,$	$C_f = 34\ \mu F$
$n = 2,$	$r_s = 0.63\ \Omega,$	$r_c = 206\ m\Omega$
$r_m = 7.3\ K\Omega,$	$F_o = 50.2\ KHz,$	$Z_o = 62.1\ \Omega$
$F_o = 108.1\ KHz,$	$Z_o = 30.6\ \Omega$	

The nonlinear state equations of the FMRC are the following:

$$L\frac{di}{dt} + i\,r_s + v_s + \frac{v_d}{n} = V_g$$

$$C_s\frac{dv_s}{dt} = S_a\,i$$

$$L_m\frac{di_m}{dt} = \frac{v_d}{n}$$

$$C_d\frac{dv_d}{dt} = n\left(i - i_m - \frac{n\,v_d}{r_m}\right) i_R \qquad (43)$$

$$L_f\frac{di_o}{dt} = v_r - v_o$$

$$C_f\,d\frac{v_f}{dt} = i_o - \frac{v_o}{R}$$

where S_a represents the switching action of the MOSFET:

$S_a = 0$ (MOSFET or its body diode ON)

$S_a = 1$ (otherwise)

The parastics related with losses are defined by:

r_s = conduction loss of the tank

r_m = core loss of the transformer

r_c = ESR of the output capacitor

With the above large-signal nonlinear model available, time domain simulation is done with PSPICE. There are three input variables, $\hat{v}_g$, $\hat{i}_o$, d and six output variables, $\hat{i}_l$, $\hat{i}_{lm}$, $\hat{i}_{ls}$, $\hat{v}_{cs}$, $\hat{v}_{cd}$ and $\hat{v}_o$ of the power stage. Figs. 20 and 21 show the control-to-output transfer function of the small-signal model of the FMRC compared with the measured data. The identified small-signal model of the FMRC

is given below as a state-space representation form $R_o = \{A, B, C, D\}$:

$$\begin{bmatrix} \mathbf{A} \\ \text{---} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} -2.3167 & .4516 & .5953 & 293.3722 & .0367 & -.3943 \\ .6309 & .5219 & -.3427 & -412.1493 & -.0271 & .1871 \\ -.8650 & -.8106 & 1.2722 & 933.7580 & -.0466 & .3579 \\ -.0021 & .0005 & .0005 & 2.7181 & .0000 & -.0003 \\ -15.7170 & -2.9882 & -2.2787 & 1370.8563 & -.0942 & -3.1327 \\ .0167 & -.0786 & .3058 & -2.9637 & -.0043 & .9329 \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and (44)

$$\begin{bmatrix} \mathbf{B} \\ \text{---} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} -.0651 & .0258 & -9.7244 \\ .0009 & -.0058 & -1.0201 \\ .0691 & -.0026 & -1.3769 \\ -.0001 & .0000 & -.0094 \\ -.0101 & .6498 & -73.8649 \\ .0214 & .1051 & 1.0132 \\ \text{-----} & \text{-----} & \text{-----} \\ -.0973 & -.1201 & -3.4953 \\ .0488 & .0234 & 7.9658 \\ .0575 & .0083 & 4.8434 \\ -.0001 & -.0001 & -.0035 \\ -.4763 & -.5441 & 156.3498 \\ .0046 & .1186 & .2315 \end{bmatrix}$$

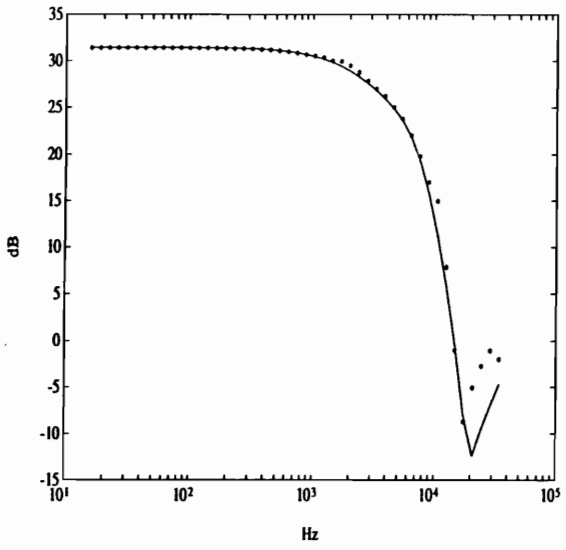


Fig. 20. Magnitude:
Identified (solid), Measured (*)

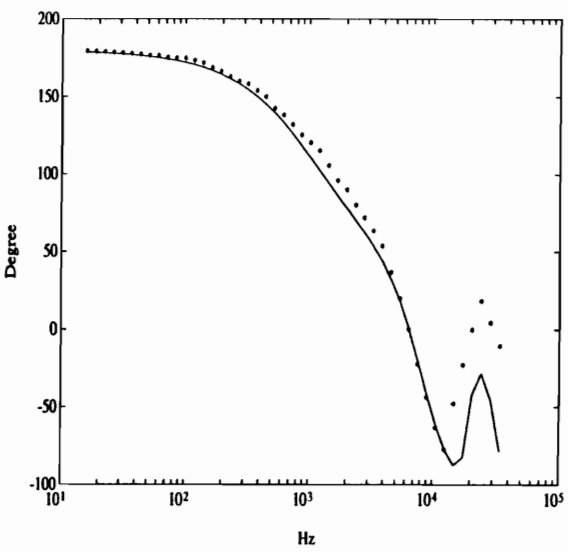


Fig. 21. Phase:
Identified (solid), Measured (*)

The identified results are close to the measured data except near the one-half of the switching frequency, which shows the difficulty of this problem.

3.3. Large-Signal Modeling of Power Converters

A method of using MLPs for modeling a complex nonlinear power electronics system is investigated in this section. In order to achieve better control of unknown plant dynamics, an accurate model of a nonlinear physical system should be developed. The importance of pre-processing at the initial modeling stage is explained, and a method of classifying the input/output data set into different categories for training-data selection to capture both the linearity and the nonlinearity of the power converters is included. As examples, a boost converter and a buck converter which are stable for small signals but found to be unstable for large perturbations are simulated as physical systems to be identified and the comparisons between real system and the identified one using the MLP are demonstrated.

The one of the most popular method of identifying a *quasi-linear* system using an MLP is to determine an equilibrium point and generate a small-signal model for the linear model of the system to extract the linear part of the system (by *robust hybrid identification technique*). Then, this linear part of the system is represented by the linear feedthrough terms of the MLP with a linear output activation function. At least conceptually, and for relatively tractable systems, the remaining nonlinear part of the system can be captured with a single hidden layer having a number (to be determined) of processing elements (neurons) with nonlinear activation functions. The required number of neurons in the hidden layer that provide satisfactory results are determined systematically. Since the order or the structure of higher-order neural networks can be tailored to the order or structure of the problem from the linear modeling stage, the MLP designed for a particular class of problems, such as an open-loop boost converter, can be used for

an entire class of problems.

3.3.A. Open-Loop Boost Converter

The second stage is to identify the nonlinearity of the system by training the network with the hidden layer in place, but with the linear feedthrough terms fixed. The McCulloch-Pitts neurons are added and the back-propagation algorithm is applied. In order to identify the system, the MLP structure consists of a standard single hidden layer feedforward net with linear feedthrough terms. By using a linear activation function for the output elements, the system dynamics can be matched to a convenient mathematical form. The hidden layer activation functions are hyperbolic tangents. The connection weights from inputs to outputs, inputs to hidden layer, and hidden layer to outputs are the arrays ***An***, ***Bn***, and ***Cn***, respectively. Specifically, this form is given by

$$\boldsymbol{Y} = \boldsymbol{An} * \boldsymbol{U} + \boldsymbol{Cn} * \mathit{TANH}(\, \boldsymbol{Bn} * \boldsymbol{U} \,) \tag{45}$$

where **Y** is the vector of system outputs and **U**, the vector of all MLP inputs, including the delayed signals.

In the previous stage, the linear feedthrough terms are determined and fixed for a 2 unit delay *ARMA* model depicting the plant's linear characteristics. In order to get the desired accuracy, 6 hidden neurons of a single hidden layer result in representing the system dynamics of the boost converter system with reasonable accuracy. After pruning 10 percent of the weights of smallest absolute magnitude, a total of 12 weights of the MLP were eliminated. Output functions from the identified function are of the form:

Fig. 22 shows the response of the final version of the MLP using frozen linear

$$y = \sum_{i=0}^{8} a_i u_i(k) + \sum_{i=1}^{6} c_i \tanh[\sum_{i=0}^{8} b_i u_i(k)] \quad (46)$$

feedthrough terms and 6 hidden neurons of a single hidden layer. To test the accuracy of the identified boost converter, the MLP open-loop step response and large-signal stability of the quiescent operating point in closed-loop were checked and compared to those of the original system. State-plane trajectories of both systems are shown in Fig. 23. Inductor current i_l is plotted vs. capacitor voltage v_c. Since the feedback controller was designed using the linear part of the boost converter, which is nearly the same as the linearized version of state-space averaging method, neglecting the nonlinearity of the system, it shows instability beyond some radius **r** from the quiescent operating point, $\hat{\mathbf{x}}_e = \mathbf{0}$. For small perturbations, inside the circle, the converter behaves as predicted by the small-signal model, and transients converge as expected. However, for large perturbations, the nonlinear terms become significant and the performance of the resulting controller outside this range is unacceptable. Figs. 24, 25 and 26 show the open loop step-response of the measured boost converter system and the identified one. Although the system was trained using a medium range of input data, the resulting output error was quite small over input perturbations (twice larger than training range), and the boundary of small-signal behavior, in which the system is stable, was predicted to be nearly the same as the original boost converter.

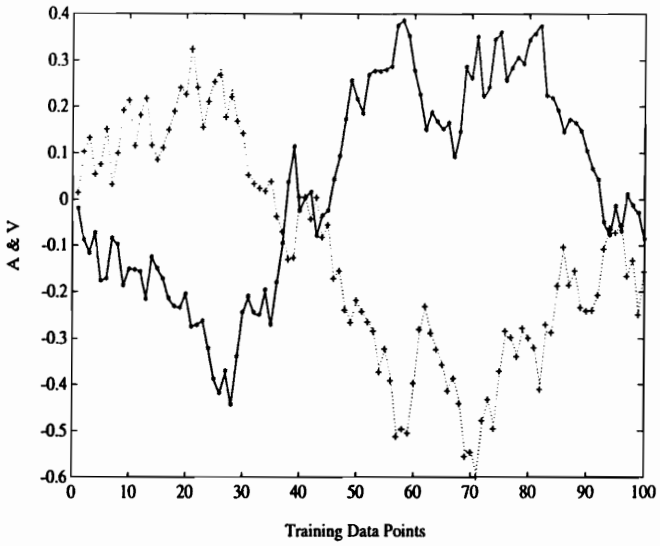


Fig. 22. Identified Model: Given i_l (*), Model i_l (solid), Given v_c (+), Model v_c (:)

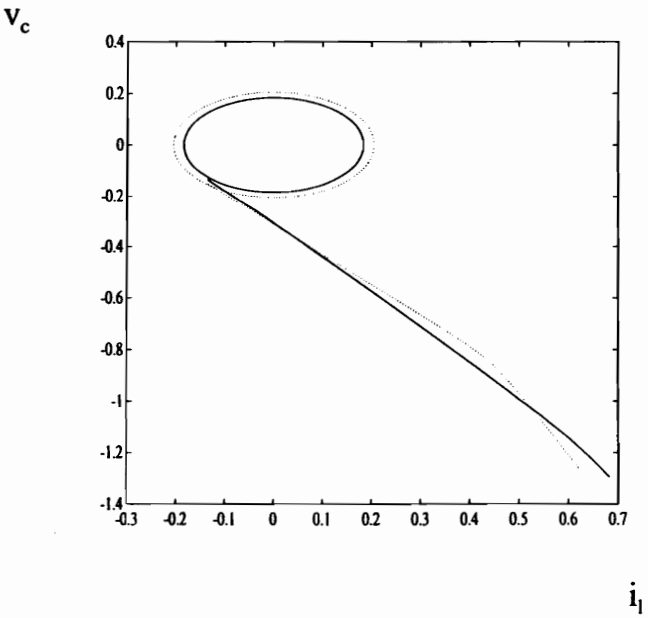


Fig. 23. State-Plane Trajectories: Given i_l, v_c (solid), model i_l, v_c (:)

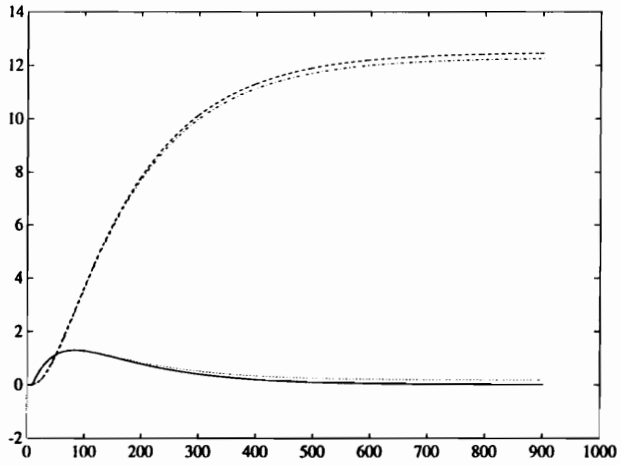


Fig. 24. Step-Response (vi=5V,io=d=0):
Given il (-), vc (--), Model il (:), vc (-.)

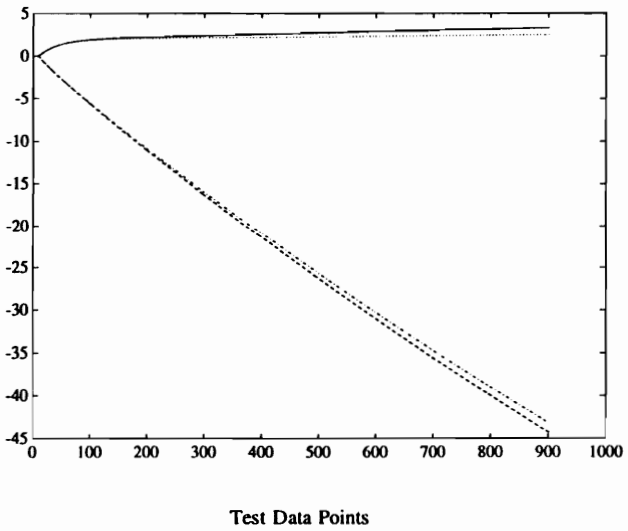


Fig. 25. Step-Response (vi=io=0,d=0.3):
Given i_l (-), v_c (--), Model i_l(:), v_c (-.)

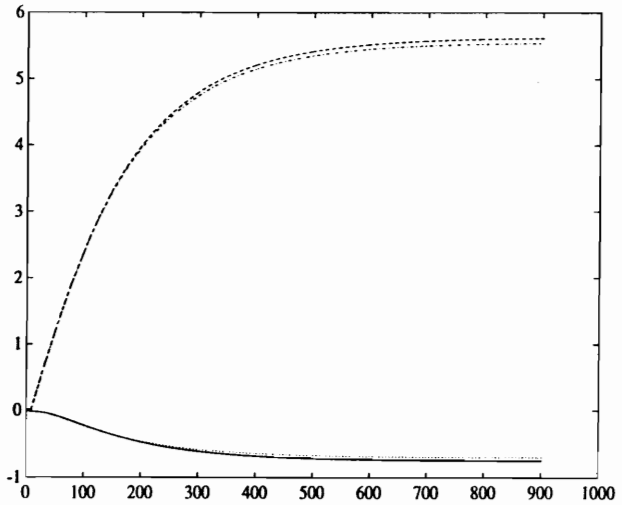


Fig. 26. Step-Response (vi=d=0,io=.3A):
Given i_l (-), v_c (--), Model i_l (:), v_c (-.)

3.3.B. Closed-Loop Buck Converter

As an final example of the modeling of highly nonlinear power electronics system, both current and voltage-loop closed buck converter is selected. Large-signal modeling along with linear model of the buck converter using an MLP will be reported and also the simulation results combined with the single stage filter of constant power load are presented. Two-input two-output closed-loop buck converter is shown in Fig.27.

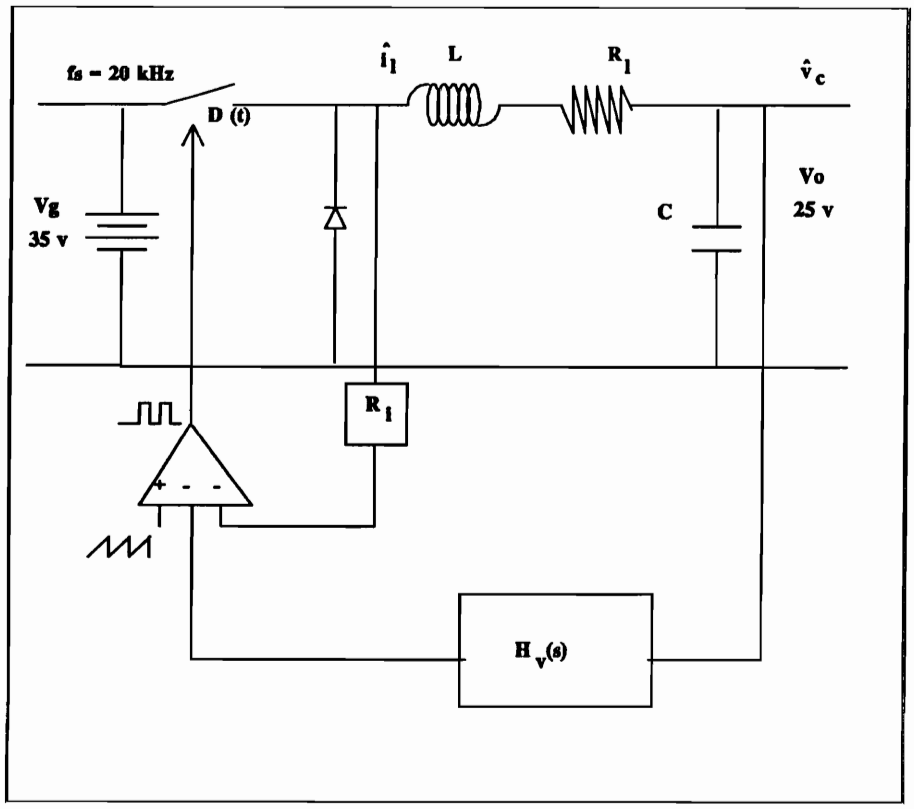


Fig. 27. Current and Voltage Loop-Closed
Buck Converter with PWM Control over the Switch

Two input variables, $\hat{v}_g$ (the variation of input voltage), $\hat{i}_o$ (the variation of output current) and two output variables, $\hat{i}_{in}$ (the variation of input current), $\hat{v}_o$ (the variation of output voltage) represent the state of the system dynamics. The current-loop gain, R_i and voltage-loop transfer function, $H_v(s)$ are given as the following:

$$R_i = 0.1$$

$$H_v(s) = \frac{8000\,(1 + \frac{s}{10000})}{(s + \frac{s}{60000})} \tag{47}$$

The circuit parameters and operating points are given by:

L = 80e-6 H,	R_l = 0.05 Ω
C = 300e-6 F,	V_g = 35 V
I_o = 1.2 A,	V_{ref} = 5 V
V_o = 25 V,	V_{ramp} = 0.4 V
C_1 = 2e-9 F,	C_2 = 10e-9 F
R_1 = 10416 Ω,	R_2 = 10000 Ω
R_s = 2604 Ω	

This system has two different modes of operation including continuous inductor current, $\hat{i}_l$, conduction mode (CCM) and discontinuous inductor current, $\hat{i}_l$, conduction mode (DCM). While operating in CCM, the system is found to be marginally stable for resistive load but oscillatory for reactive load. This converter was designed to operate at a nominal duty ratio of 0.72 in CCM. The input inductor current is sensed for current control and the output voltage is fed back for voltage control.The nonlinear state equation of the buck converter consists of both on-time model $\{A_o, B_o, C_o, D_o\}$ and off-time model $\{A_f, B_f, C_f, D_f\}$, which are given as the following equations:

$$A_o = \begin{bmatrix} -\frac{R_l}{L} & -\frac{1}{L} & 0 & 0 \\ \frac{1}{C} & 0 & 0 & 0 \\ 0 & \frac{1}{R_1 C_1} & -\frac{1}{R_2 C_1} & \frac{1}{R_2 C_1} \\ 0 & 0 & \frac{1}{R_2 C_2} & -\frac{1}{R_2 C_2} \end{bmatrix}$$

$$B_o = \begin{bmatrix} \frac{1}{L} & 0 & 0 \\ 0 & -\frac{1}{C} & 0 \\ 0 & 0 & -\frac{1}{R_1 C_1} - \frac{1}{R_s C_1} \\ 0 & 0 & 0 \end{bmatrix} \qquad (48)$$

$$C_o = [0 \;\; 0 \;\; -1 \;\; 0], \qquad D_o = [0 \;\; 0 \;\; 1]$$

$$A_f = A_o , \qquad C_f = C_o , \qquad D_f = D_o$$

$$B_f = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\frac{1}{C} & 0 \\ 0 & 0 & -\frac{1}{R_1 C_1} - \frac{1}{R_s C_1} \\ 0 & 0 & 0 \end{bmatrix} \qquad (49)$$

Since the system consists of two different modes of operation, the initial stage of data selection for training an MLP is very important. For example, it is not only the range of data, but its distribution, that is necessary to define a good mapping. The reason for this is that the system will typically occupy a very small region, e.g. the neighborhood around a stable equilibrium point, for the vast majority of the response time. One solution to this problem is to excite the system with varying amounts of pseudo-random disturbance so that the data collected will be representative of the full range of the system. Using the proposed identification technique, a linear time invariant (LTI) model of the closed-loop buck converter around the operating point (in CCM) is developed to show how the system becomes marginally stable for the amounts of input perturbations. Figs. 28 and 29 show the identification results of the system for small signals. The eigenvalues of the 4th order of the system are {-0.3099, **0.9760**, 0.5774 +/- 0.3804i}.

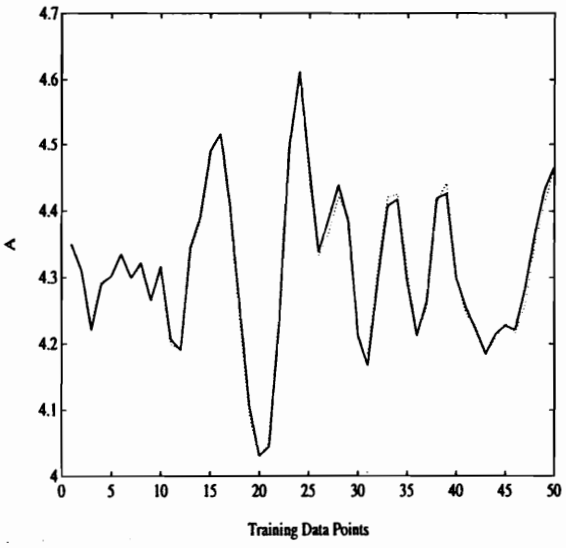


Fig. 28. Input Current (i_{in}): Identified (dotted), Measured (solid)

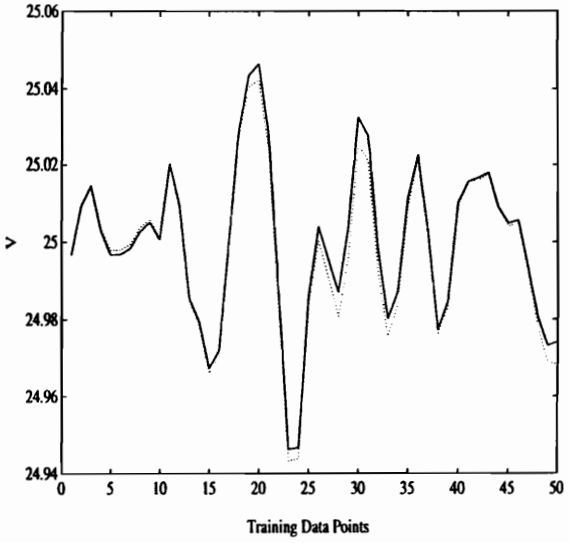


Fig. 29. Output Voltage (v_o): Identified (dotted), Measured (solid)

In comparison with the above figures, the identification results of the system for large variations of the input signals are given in Figs. 30 and 31.

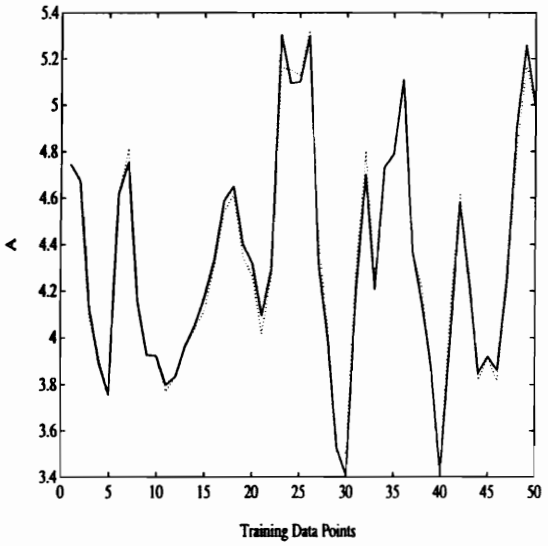


Fig. 30. Input Current (i_{in}): Identified (dotted), Measured (solid)

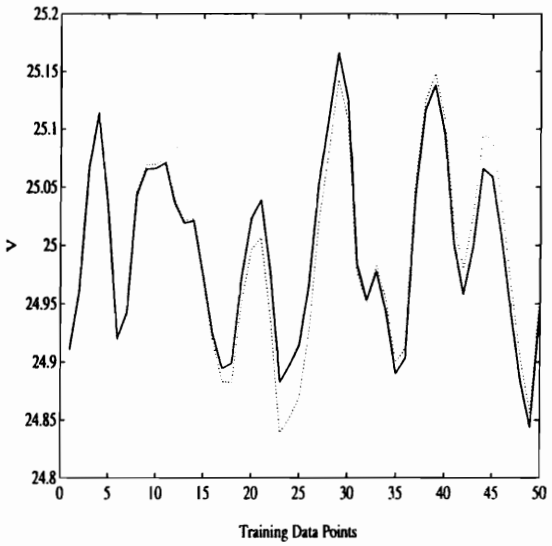


Fig. 31. Output Voltage (v_o): Identified (dotted), Measured (solid)

The resulting eigenvalues of the system are {-0.2870, **0.9987**, 0.5704 +/- 0.3735i}. Due to large input perturbations, Figs. 30 and 31 are less accurate than Figs. 28 and 29. However, the large-scale behavior of the system is discovered, which is very critical in a distributed power system and, therefore, should be known in advance. After investigating the eigenvalues of two different identified system, the key eigenvalue of both system which makes the system unstable is found to be near unity. Especially, in the presence of large amount of input perturbations, this eigenvalue is becoming close to unity, which causes the system to be oscillatory in parallel with nonlinear reactive load shown in Fig. 32.

The parameters of single stage filter with constant load are given by:

L = 3e-6 H, $R_l = 0.2\ \Omega$

C = 30e-6 F, $V_{in} = 25\ V$

C_l = 100e-6 F, $R = v_c^2/Power\ \Omega$

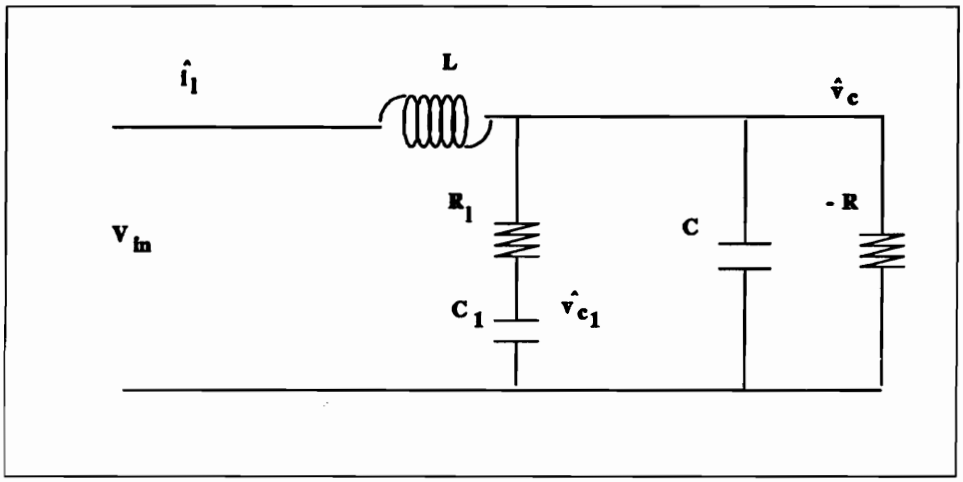


Fig. 32. Single Stage Filter with Constant Power Load

Figs. 33 and 34 show the response of the identified system using 30 hidden neurons of one hidden layer with linear feedthrough terms in step change of resistive load (R_l = 0.2 Ω) from 8A to 4A.

Since R_l becomes smaller, the converter begins to operate in parallel with resistive load toward reactive load and, eventually, starts to oscillate at the point of R_l, 0.04 Ω. However, the MLP model is found to be oscillatory when R_l reaches to 0.06 Ω, therefore, shows very close approximation of the region of stability with the amount of error, 0.02 Ω.

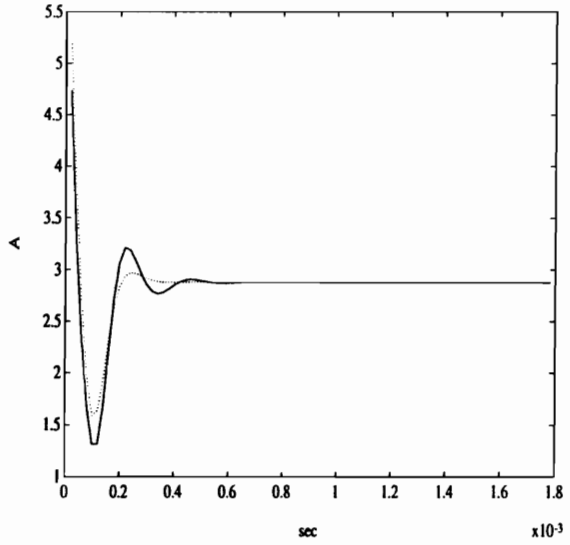


Fig. 33. Input Current (i_{in}):
Identified (dotted), Measured (solid)

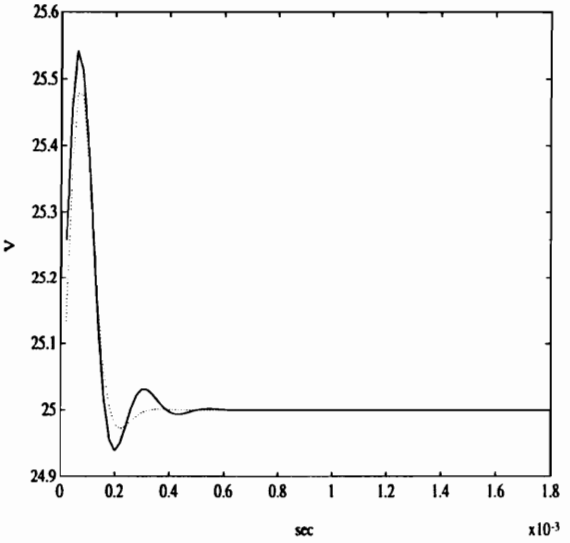


Fig. 34. Output Voltage (v_o):
Identified (dotted), Measured (solid)

Other classical methods of identifying fairly complex nonlinear systems using neural networks is to have a nonlinear squashing function as an output neuron. In this case, the buck converter includes CCM and also DCM, which makes the modeling procedure more complicated. The MLP network commonly uses a sigmoid or hyperbolic tangent activation function. For present purposes the hyperbolic tangent function is used as an output neuron, which responds in a nearly linear fashion to summations between about -2 to +2 and will go into nonlinear range when its input becomes larger. All the procedures are the same as the previous method except using the hyperbolic tangent output function instead of a linear output function.

In order to have minimal network complexity, pruning is done by removing connections whose weights have small values, including connections to neurons in the hidden layer if their contribution is insignificant. Finally, the comparisons between the measured system and the identified system using the MLP of 40 hidden neurons in a single hidden layer is presented. Figs. 35 and 36 show the output voltage waveform in response to both step-up and step-down load changes, respectively. With the above step load change the buck converter undergoes the mode of operation from CCM to DCM and vice versa.

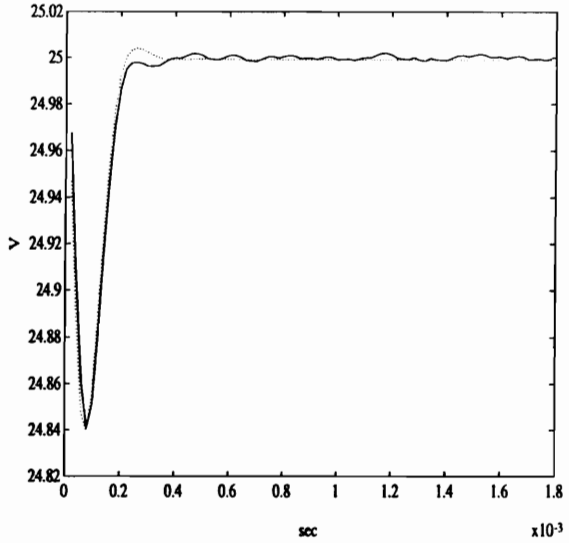


Fig. 35. v_o (Step-Up):
Identified (dotted), Measured (solid)

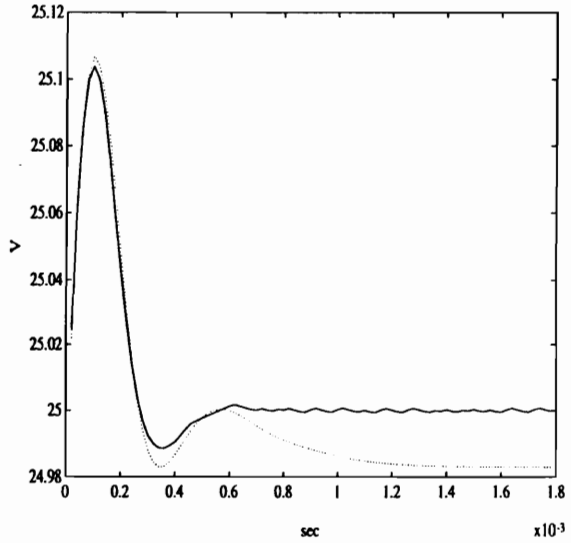


Fig. 36. v_o (Step-Down):
Identified (dotted), Measured (solid)

3.4. Conclusions

As a unified empirical data modeling approach of highly nonlinear power electronics systems, MLPs are proposed for modeling switching converters and resonant converters. In order to get, not only physical insight of system, but also to have a convenient linearized model of the given system at the specific operating point, a small-signal modeling method is introduced using the robust hybrid identification technique combined the classical linear identification method with the back-propagation learning. A boost converter, a SRC and a FMRC are taken as examples to demonstrate the proposed algorithm and the result looks very promising even though the key factor is how to persistently excite the system for showing all kinds of characteristics to be identified. Based on the identified linear model of the power converter an MLP is further trained to capture both the nonlinearities and the dynamics of the system for an approximation of boundaries into unstable region in case of large transients. The estimation of the region of stability is compared with the measurement data and found to be almost identical with negligible error. Due to the extremely long computational time in training MLPs, the neural network approach is sometimes not favored by design engineers for some practical purpose. Nevertheless, once it is trained, it can be used as an emulator in place of the switching converter in a distributed power system for large-signal simulation, which shortens simulation time greatly.

4. Inverted Pendulum Identification

4.1. Identification

4.1.A. State Estimation

In many control problems it is often inconvenient to measure every system state because of various restrictions, but it would be desirable to feed back all the states of the system if it were possible to generate the state variables in some indirect way. Suppose we take a nonlinear system with controllable and observable linear part in a domain of interest. The usual technique for asymptotically stabilizing this nonlinear system is to build a linear state estimator and a controller for the linear part. In the process of designing the controllers, Cheok and Beck [28] generated the state estimates by a delayed-measurement observer which compares with a Luenberger reduced-order observer, but appears to be simpler and more efficient than that of Luenberger since it has only delays. However, their observer depends heavily on the quality of the measurements. Furthermore, they assumed that they already knew which state should be estimated (either discrete integration or differentiation). Therefore, there is no generality applicable to the system unless we are familiar with the internal structure of the system. The reader is referred to Reference [29] for details. In this section contrasting with the delayed-measurement observer, without *a priori* knowledge of the given system, the states will be estimated to the actual shape of the state trajectories by the state estimation method. This novel state estimation method is developed from a selected pseudo-observable form of Chapter 2, providing state information for feedback control.

The design step is to identify the linear model of the plant using a robust hybrid method of multivariable system identification and then, from the information of the identified model, to estimate the state trajectories of the linearized model of the system. This process is exemplified using the sampled data from the given C-T system model to obtain an observable form D-T model, which is subsequently used to generate state trajectories. For example, the following "unknown" C-T state space system of Eqn. (50) is considered.

$$\boldsymbol{R}_c = \begin{bmatrix} \mathbf{A}_c & \boldsymbol{B}_c \\ \mathbf{C}_c & \boldsymbol{D}_c \end{bmatrix} = \left[\begin{array}{ccccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & -4 & -3 & 1 & 4 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 1 & 0 & 0 \\ - & - & - & - & - & + & - & - \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{50}$$

Using the sample interval T = 2 seconds, the representation $\boldsymbol{R}_c$ is discretized into the following D-T ramp-invariant (RI) equivalent system, represented in the partitioned *system matrix* form of the given state space model in Eqn. (51). For comparison with the above D-T ramp-invariant (RI) equivalent model, the ***robust hybrid identification algorithm*** explained in Chapter 2 should be verified using the given C-T system in Eqn. (50).

$$R_{dr}=\begin{bmatrix} A_{dr} & B_{dr} \\ C_{dr} & D_{dr} \end{bmatrix}=\left[\begin{array}{ccccc|ccc} -.100 & .506 & .192 & .148 & 1.100 & 2.084 & .238 & .309 \\ -.767 & -.317 & -.069 & .044 & .767 & 1.235 & .070 & -.034 \\ .274 & -.109 & -.111 & -.113 & -.274 & -.135 & -.104 & -.177 \\ 0 & 0 & 0 & .135 & 0 & 0 & .374 & 0 \\ -.550 & .253 & .096 & .074 & 1.550 & 3.042 & .119 & .154 \\ - & - & - & - & - & + - & - & - \\ 1 & 0 & 0 & 0 & 0 & 1.196 & .035 & .098 \\ 0 & 1 & 0 & 0 & 0 & .394 & .064 & .137 \end{array}\right] \quad (51)$$

As is well known, in order to perform a successful identification, the input signal should be sufficiently long and *sufficiently rich*, i.e. persistently exciting, which is uniformly distributed random signal between -1.5 and 1.5. The resulting "identified D-T" four-matrix model (m = 3, p =2, n = 5) is given by the following system matrix:

$$R_{do}=\begin{bmatrix} A_{do} & B_{do} \\ C_{do} & D_{do} \end{bmatrix}=\left[\begin{array}{ccccc|ccc} 0 & 0 & 1 & 0 & 0 & 2.084 & .238 & .309 \\ 0 & 0 & 0 & 1 & 0 & 1.235 & .070 & -.034 \\ .124 & .058 & .876 & 1.020 & 1.530 & 3.737 & .178 & .088 \\ 0 & 0 & 0 & 0 & 1 & 0.352 & -.090 & -.096 \\ -.030 & -.017 & .030 & -.086 & .281 & .015 & -.037 & -.008 \\ - & - & - & - & - & + - & - & - \\ 1 & 0 & 0 & 0 & 0 & 1.196 & .035 & .098 \\ 0 & 1 & 0 & 0 & 0 & .394 & .064 & .137 \end{array}\right] \quad (52)$$

With the identification procedure used, the representation $\boldsymbol{R}_{d0}$ is in a *pseudo-observable* form (POF). An important aspect is determining and using the most appropriate structural information, e.g. the most numerically stable POF. To keep the problem simple, only 50-rows (data sets) of data were given.

The concept of "generalization" is used to determine the final model. A model is said to *generalize* when its performance over data that has not been used in the identification process is comparable to the response over the "training" data. This property of generalization is strongly dependent on the system structure.

The identification process will be discussed with the assumption that a preliminary analysis has already been made estimating the order of the system to 5. The procedure will be to use the initial data, 30 data pairs, to identify the system parameters from a specified set of indices, then to test the model's performance over the complete data set. The measure of the performance is given by the Frobenius norm of the error matrix, i.e. the square root of the sum of the squares of the entries of the error matrix. The results are compiled in Table 1. Each possible (observability) index set is used to calculate an "identified" model. Each of the entries under the "norm" column in Table 1 has been multiplied by 10^6 and represents the Frobenius norm of the error array from the available data.

TABLE 1. Computational Results

Order	Indices	Norm
5	4,1	19.085
	3,2	3.453
	2,3	1.764
	1,4	15.868

The model selected to represent the system which has the best generalization corresponds to the order-5 system with indices {2,3}. As an admissible set of pseudo-observability indices, $\{n_i\}$ = {2,3}, was selected. The unique set of observability indices of $\boldsymbol{R}_c$ is $\{n_i\}$ = {3,2}. To test the accuracy of the identified system, the eigenvalues of the two different state space representations should be checked. The eigenvalues of the D-T RI equivalent model are {1.0, 0.1353, 0.1353, -0.0563±j0.1231} and those of the identified D-T model are {1.0, 0.1353±j0.0004, -0.0563±j0.1231}. As a matter of fact, the eigenvalues of the original C-T system are {0, -1, -1, -1±j1}. Two outputs and five states of the model to samples of random input signal u(k) shown in Fig. 37 are given in Figs. 38 and 39.

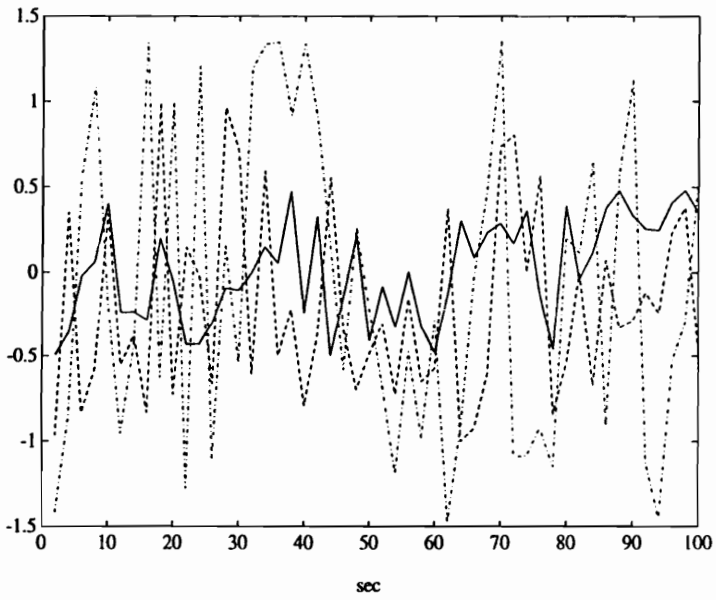


Fig. 37. Input Signals for D-T System: Input #1 (solid), #2 (dashed), #3 (dashdot)

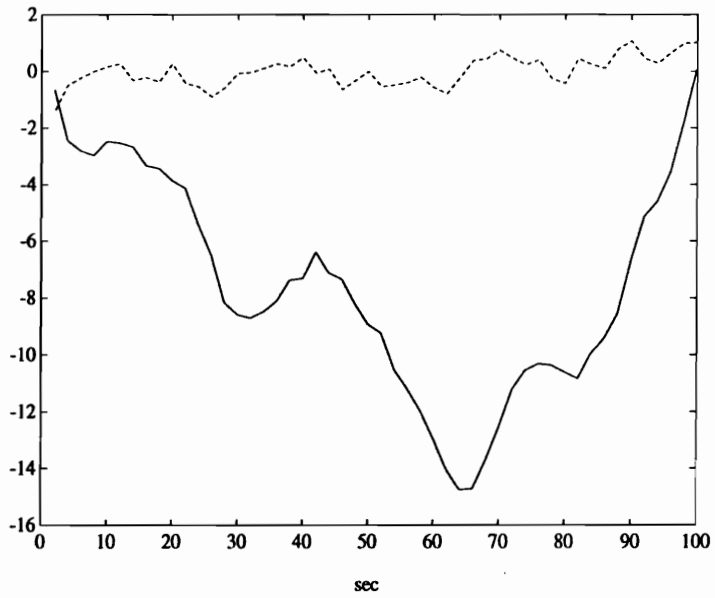


Fig. 38. Output Responses of Identified D-T System:
Output #1 (solid), #2 (dashed)

The responses of the selected model have an error norm of 1.764×10^{-6}. For comparison, the responses of another order-5 candidate model with an error norm of 19.085×10^{-6}, using indices {1,4}, are 10 times less accurate than those of indices {2,3} from Table 1. It is during the last 20-data points that the model reveals the degree of generalization achieved. The error signals between two components of the D-T system's response and that of the D-T system using each indices {2,3} and {1,4} obtained through the identification process are shown in Figs. 40 and 41. Although the results are reasonable, a pole at s = 0 in C-T system caused problems. Note that the errors are becoming larger over "test" data but still bounded using indices {2,3} in Fig. 40. But using indices {1,4}, the errors are becoming rapidly larger particularly toward the end of the time period of Fig. 41, illustrating very poor generalization over the data not used in training.

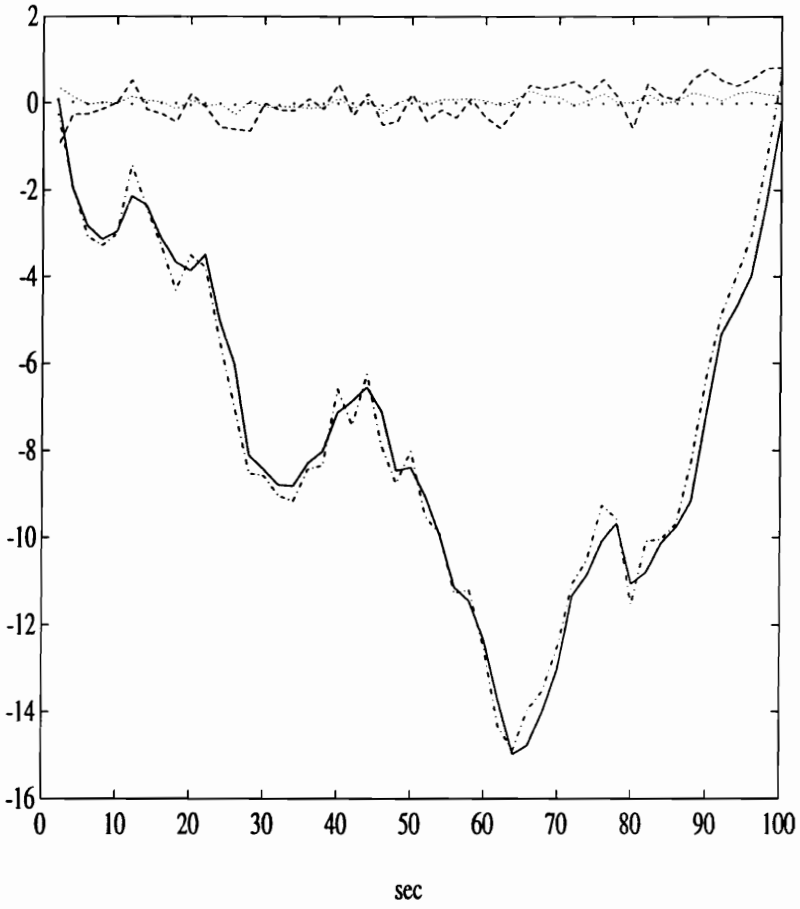


Fig. 39. States of Identified D-T System: State #1 (solid), #2 (dashed), #3 (dashdot) ,#4 (dotted), #5 (point)

No measurement noise was added to the system response. Since the order of the system is estimated to 5, three other states should be estimated to be fed back for control purposes.

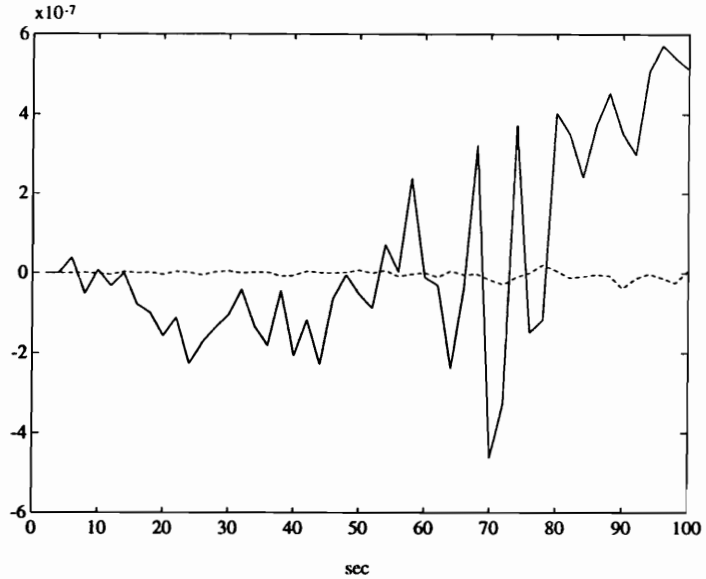


Fig. 40. Error Signals Using Indices{2,3}:
Output #1 (solid), #2 (dashed)

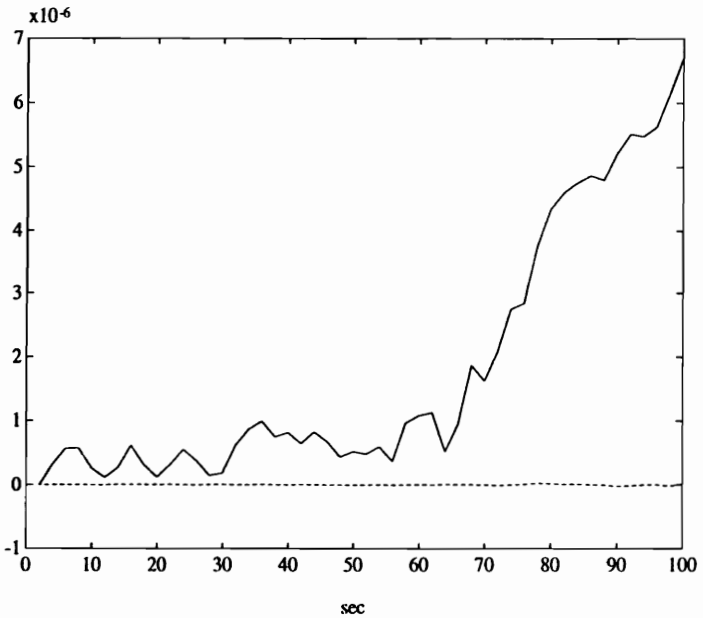


Fig. 41. Error Signals Using Indices{1,4}:
Output #1 (solid), #2 (dashed)

It might be recognized that the trajectories of three states in the upper part of Fig. 39 are similar to that of output #2 and those of the remaining two states in the bottom part of Fig. 39 are close to that of output #1. This "***particular***" form of $\{A_{do}, C_{do}\}$ pair along with the resulting trajectories of the identified D-T model will be used as a basis for estimating the state trajectories of the system.

In order to explain the method of generating the system state trajectories of the C-T system, we assume that only one output (#1) of C-T system of Eqn. (50) is measurable . Since the resulting $\{A_c, C_{c\_output\ \#1}\}$ pair is still observable, the system can be identified using three inputs and one output by the previous identification procedure. The obtained "identified D-T" four-matrix model (m = 3, p = 1, n = 5, $\{n_i\}$ = {5}) is given below:

$$R_{dI} = \begin{bmatrix} A_{d1} & B_{dI} \\ C_{d1} & D_{dI} \end{bmatrix} = \left[\begin{array}{ccccc|ccc} 0 & 1 & 0 & 0 & 0 & 2.084 & .238 & .309 \\ 0 & 0 & 1 & 0 & 0 & 3.738 & .178 & .088 \\ 0 & 0 & 0 & 1 & 0 & 3.987 & .041 & .004 \\ 0 & 0 & 0 & 0 & 1 & 3.996 & .006 & .001 \\ .000 & -.003 & .009 & -.164 & 1.158 & 3.999 & .001 & .000 \\ - & - & - & - & - & + & - & - \\ 1 & 0 & 0 & 0 & 0 & 1.196 & .035 & .098 \end{array}\right] \quad (53)$$

Responses of one output and five states of the identified D-T system to samples of the same random input signal u(k) are given in Figs. 42 and 43. From the form of $\{A_{dI}, C_{dI}\}$ pair it is expected that trajectories of all five states in Fig. 43 are similar to that of the identified output of Fig. 42, i.e. a family of five trajectories. But the eigenvalues of the identified system using only one output are {1.0, 0.1353 ± 0.0011, $-0.0563\pm j0.1231$} which are slightly different from those of the

original system, but are close enough to verify the correctness of the identification procedure.

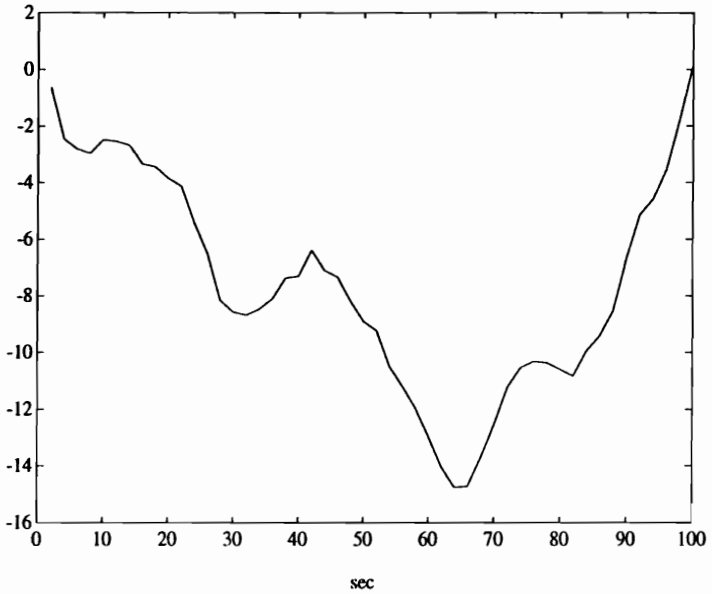


Fig. 42. Output Response
of D-T System

Since the objective of generating the state trajectories of the unmeasured state variables in the real-world system is to feed back the estimated trajectories of the state variables in order to change the dynamics of the system, the estimates should be identified as accurately as possible. Above the linear range, due to the existence of the neglected nonlinearities, the designed observer for the linearized model does not generally show desirable performance. Even a well-designed observer shows overshoot at the initial stage of feedback control, mainly, because of unmeasured estimation errors. Therefore, it would be very beneficial if the state trajectories of the internal states of the system could be identified in software

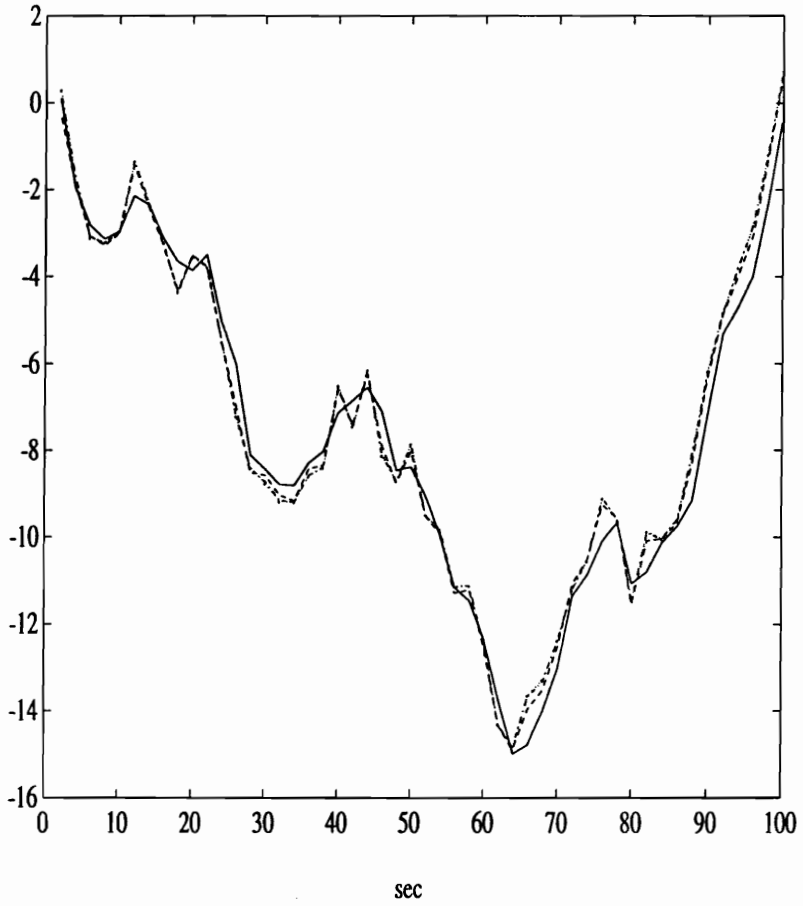


Fig. 43. State #1 (solid), #2 (dashed), #3 (dashdot), #4 (dotted), #5 (point)

without causing unnecessary overshoot at control stage. After closely looking into all the states of the system in Fig. 43, a general method of generating the state trajectories was developed with the following reasoning:

From Eqn. (53), generate one state, $\hat{x}_2(k)$, which is the difference between the present output and one delayed output, namely, y(k) - y(k-1). Then, start the identification procedure of the system including $\hat{x}_2(k)$, where the number of inputs and the order of the system are still the same as Eqn. (53), but the number of outputs is now two. Using the POI (m = 3, p = 2 and n = 5, $\{n_i\} = \{4,1\}$), the

following model was identified:

$$R_{d2} = \begin{bmatrix} A_{d2} & B_{d2} \\ C_{d2} & D_{d2} \end{bmatrix} = \left[\begin{array}{ccccc|ccc} 0 & 0 & 1 & 0 & 0 & 2.084 & .238 & .309 \\ -1 & 0 & 1 & 0 & 0 & .888 & .204 & .210 \\ 0 & 0 & 0 & 1 & 0 & 3.738 & .178 & .088 \\ 0 & 0 & 0 & 0 & 1 & 3.987 & .041 & .004 \\ -.003 & -.000 & .016 & -.200 & 1.187 & 3.996 & .006 & .001 \\ - & - & - & - & - & + & - & - \\ 1 & 0 & 0 & 0 & 0 & 1.196 & .035 & .098 \\ 0 & 1 & 0 & 0 & 0 & 1.196 & .035 & .098 \end{array}\right] \quad (54)$$

Two outputs (output #1 and estimated state $\hat{x}_2(k)$) and five states of the model responding to samples of the previous random input signal u(k) are shown in Figs. 44 and 45.

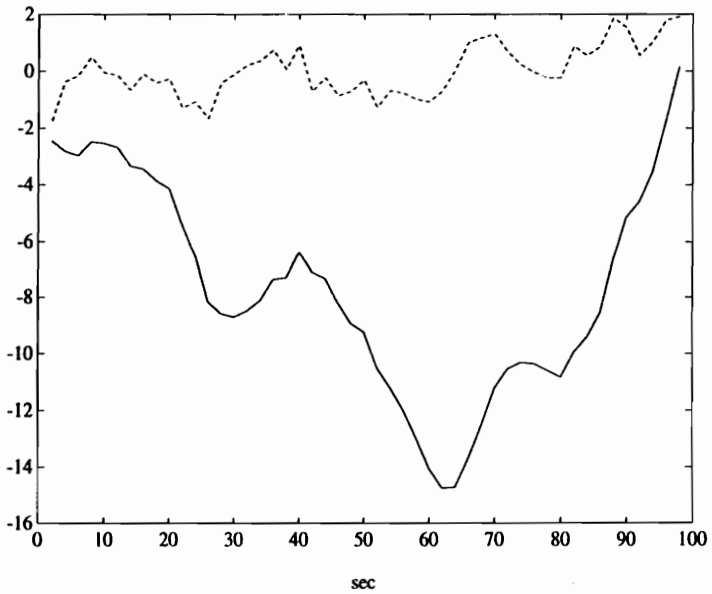


Fig. 44. Output Responses :
Output #1 (solid), State $\hat{x}_2$ (dashed)

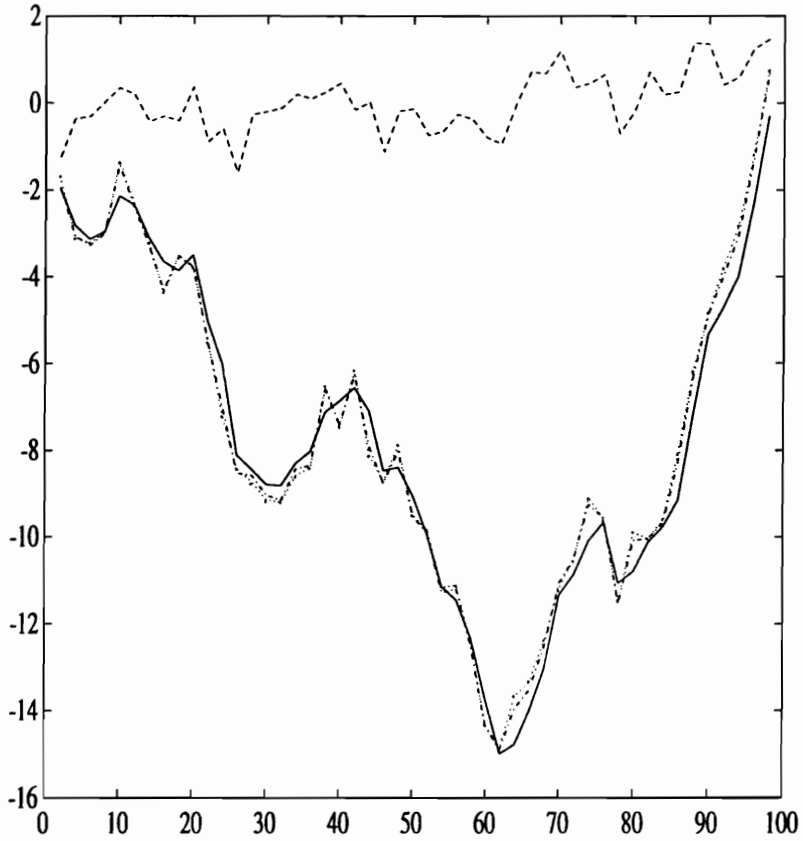


Fig. 45. State #1 (solid), #2 (dashed), #3 (dashdot), #4 (dotted), #5(point)

The identified state is similar to one of the original states of the system in Fig. 39. But, still three other states should be estimated. Since output #1 has all the system information, this method can be expanded to have POI, $\{n_i\} = \{1,1,1,1,1\}$, including all measured outputs and estimated states. The eigenvalues of this identified system are slightly different from the true ones. The resulting eigenvalues of the identified system with one additional identified state are {1.0, $0.1450 \pm j0.0438$, $-0.0513 \pm j0.1300$} which are a little slightly different compared to those of Eqn. (51), but generating close estimates of the state trajectories of the

original system, such as $\hat{x}_3(k)$ and $\hat{x}_4(k)$ shown in the upper part of Fig. 39. These two estimated state variables consist of the difference between one step ahead present estimated state and that of a unit delayed estimated state, namely, $\hat{x}_3(k) = \hat{x}_2(k) - \hat{x}_2(k-1)$ and $\hat{x}_4(k) = \hat{x}_3(k) - \hat{x}_3(k-1)$ in addition to $\hat{x}_2(k)$ (m = 3, **p = 4**, n = 5 and $\{n_i\} = \{2,1,1,1\}$). The resulting system matrix is:

$$R_{d4} = \begin{bmatrix} A_{d4} & B_{d4} \\ C_{d4} & D_{d4} \end{bmatrix} = \left[\begin{array}{ccccc|ccc} 0 & 0 & 0 & 0 & 1 & 2.160 & .221 & .266 \\ -1 & 0 & 0 & 0 & 1 & .878 & .198 & .201 \\ -1 & -1 & 0 & 0 & 1 & -.404 & .175 & .137 \\ -1 & -1 & -1 & 0 & 1 & -1.686 & .152 & .072 \\ -.189 & -.087 & .010 & -.004 & 1.187 & 3.813 & .146 & .055 \\ - & - & - & - & - & + \; - & - & - \\ 1 & 0 & 0 & 0 & 0 & 1.282 & .023 & .064 \\ 0 & 1 & 0 & 0 & 0 & 1.282 & .023 & .064 \\ 0 & 0 & 1 & 0 & 0 & 1.282 & .023 & .064 \\ 0 & 0 & 0 & 1 & 0 & 1.282 & .023 & .064 \end{array}\right] \tag{55}$$

Figs. 46 and 47 show output responses and all 5 states of the modified system including 3 estimated states. The resulting three identified states have similar trajectories to the original states of the system shown in the upper part of Fig. 39.

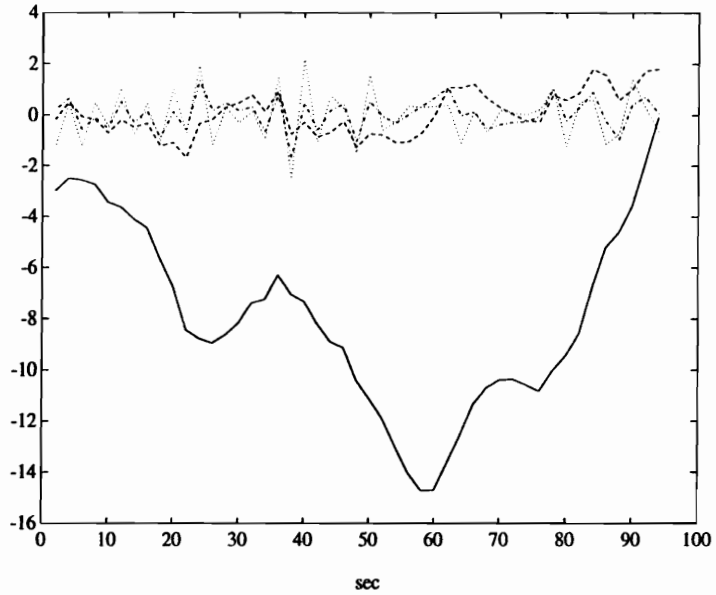


Fig. 46. Output #1 (solid), $\hat{x}_2$ (dashed), $\hat{x}_3$ (dashdot), $\hat{x}_4$ (dotted)

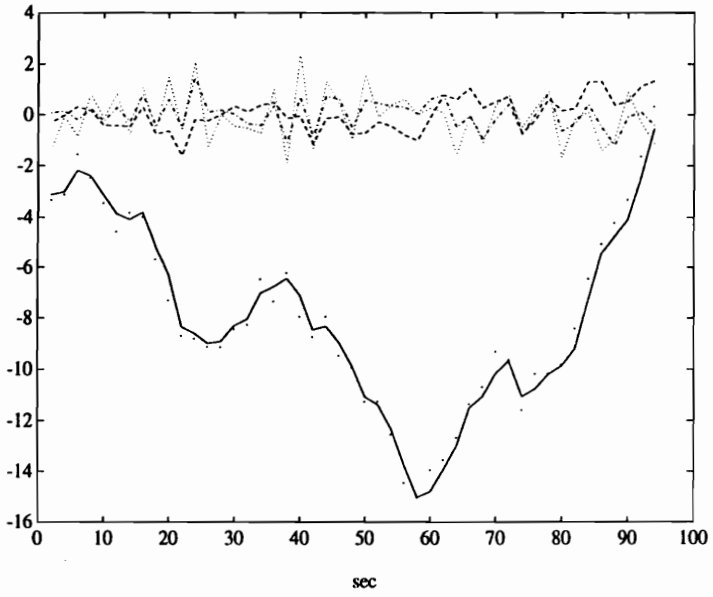


Fig. 47. State #1 (solid), #2 (dashed), #3 (dashdot), #4 (dotted), #5 (point)

After understanding the relationship between POIs and the resulting trajectories of the states to be estimated, the system matrix of Eqn. (52), which has three inputs and two outputs, should be reexamined.

Since it would be easier to generate the state trajectories from two outputs instead of one output for control purposes, the system can be constructed to have POIs, $\{n_i\} = \{1,1,1,1,1\}$ including both measured outputs and three estimated states. The selected POIs of Eqn. (52) is $\{n_i\} = \{2,3\}$, therefore, one of three state trajectories to be identified should come from output #1 and the other two from output #2. Including each delayed outputs, namely, $\hat{x}_3(k) = y_1(k) - y_1(k-1)$, $\hat{x}_4(k) = y_2(k) - y_2(k-1)$ and $\hat{x}_5(k) = \hat{x}_4(k) - \hat{x}_4(k-1) = [y_2(k) - y_2(k-1)] - [y_2(k-1) - y_2(k-2)]$ (m=3, **p=5**, n=5 and $\{n_i\}= 1,1,1,1,1\}$), the following system matrix is obtained:

$$R_{ds}=\begin{bmatrix} A_{ds} & B_{ds} \\ C_{ds} & D_{ds} \end{bmatrix}=\left[\begin{array}{ccccc|ccc} .996 & 2.052 & -.581 & -.356 & -.004 & 2.098 & .205 & .260 \\ -.003 & .677 & -.309 & -.214 & -.007 & 1.251 & .053 & -.043 \\ -.004 & 2.052 & -.581 & -.356 & -.004 & .906 & .167 & .182 \\ -.003 & -.323 & -.309 & -.214 & -.007 & .862 & -.014 & -.171 \\ -.003 & -.323 & -.309 & -1.214 & -.007 & .474 & -.081 & -.298 \\ - & - & - & - & - & + & - & - & - \\ 1 & 0 & 0 & 0 & 0 & 1.192 & .038 & .078 \\ 0 & 1 & 0 & 0 & 0 & .389 & .067 & .128 \\ 0 & 0 & 1 & 0 & 0 & 1.192 & .038 & .078 \\ 0 & 0 & 0 & 1 & 0 & .389 & .067 & .128 \\ 0 & 0 & 0 & 0 & 1 & .389 & .067 & .128 \end{array}\right] \quad (56)$$

Finally, output responses (including three estimated states) and all 5 states of the system are shown in Figs. 48 and 49, respectively.

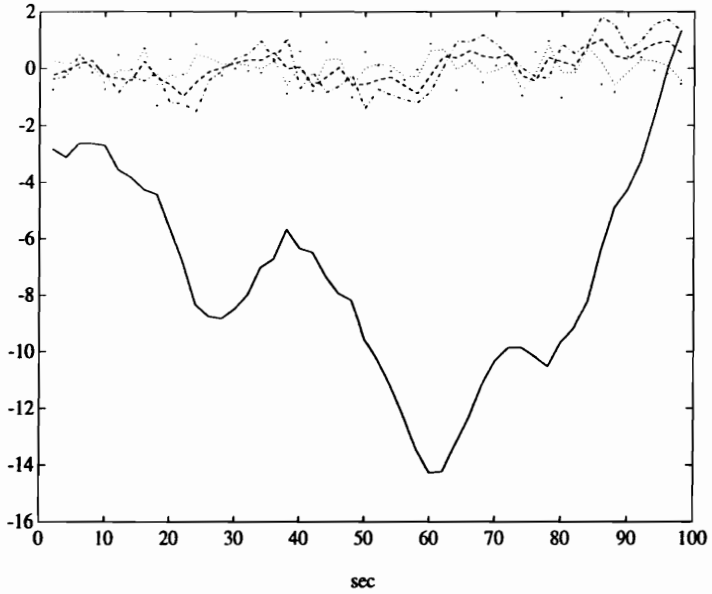


Fig. 48. Output #1 (solid), #2 (dashed), State $\hat{x}_3$ (dashdot), $\hat{x}_4$ (dotted), $\hat{x}_5$ (point)

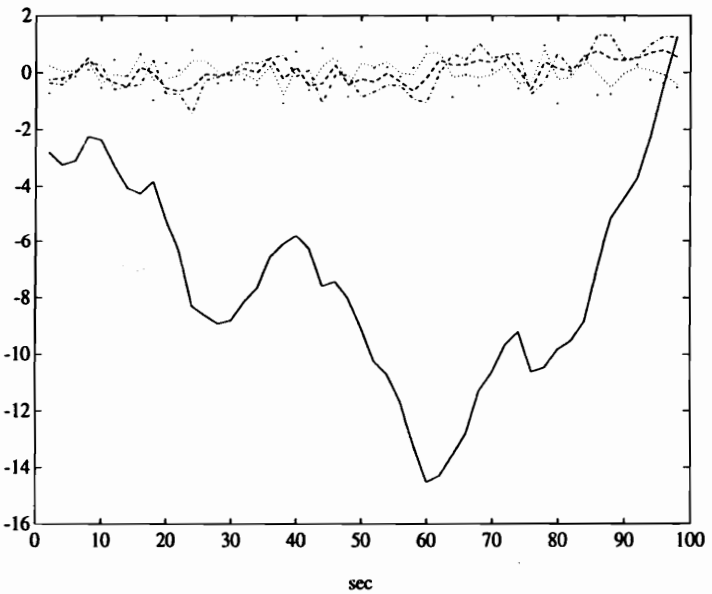


Fig. 49. State #1 (solid), #2 (dashed), #3 (dashdot), #4 (dotted), #5(point)

In the next section a step-by-step procedure is suggested for extracting the estimate of state variables from the identified linear model of the nonlinear system.

4.1.B. State Estimation Algorithm

After determining the system order, n, and the observability index, n_x, one must select an admissible set of (pseudo) observability indices, $n_o = \{n_i\}$, then expand the output array as discussed previously until we have POI, $\{n_i\} = \{1,1, \ldots,1,1\}$, which includes all measured outputs and each delayed output. Using the proposed identification method of Chapter 2, estimate the modified matrices $\mathbf{A}_{rm}$ and $\mathbf{B}_{rm}$. This process can be symbolized as

$$\mathbf{U},\ \mathbf{Y}_{1m},\ \mathbf{Y}_{2m} \rightarrow \mathbf{A}_{rm},\ \mathbf{B}_{rm}$$

which is very similar to the algorithm of Section 2.7 except ***changing the dimension p of the output vector to the estimated order of the system n***. Once $\mathbf{A}_{rm}$ and $\mathbf{B}_{rm}$ have been determined, R_{om} may be found, as formalized in the following steps:

1. Set $\mathbf{I}_n \Rightarrow \quad \mathbf{C}_{om}$

2. Set $\mathbf{S}_a\ \mathbf{A}_{rm} \Rightarrow \mathbf{A}_{om}$

3. Partition $\mathbf{B}_{rm}$ into $n \times m$ blocks:

   $\mathbf{B}_{rm} \Rightarrow [\mathbf{X}_0\ \mathbf{X}_1\ \cdots\ \mathbf{X}_{nx}]$ where each block has m-columns,

   and similarly, partition $\mathbf{A}_{rm}$ into two blocks:

$\mathbf{A}_{rm} \Rightarrow [\mathbf{A}_{r0}\ \mathbf{A}_{r1}]$ where the block $\mathbf{A}_{r0}$ has dimensions $n \times n$.

4. Concatenate $\begin{bmatrix} \mathbf{X}_0 \\ \vdots \\ \mathbf{X}_{nx} \end{bmatrix} \Rightarrow \mathbf{B}_c$, a $(n_x+1)n \times m$ matrix

5. Calculate the $n \times (n_x+1)m$ controllability matrix $\mathbf{Q}_c$ of the pair $\{\mathbf{A}_{om}, \mathbf{S}_a\}$ having n_x+1 blocks, i.e.

$$\mathbf{Q}_c = [\mathbf{S}_a \quad \mathbf{A}_{om}\mathbf{S}_a \quad \cdots \quad (\mathbf{A}_{om})^{n_x}\mathbf{S}_a]$$

6. Set $\mathbf{Q}_c\ \mathbf{B}_c \Rightarrow \mathbf{B}_{om}$
7. Set $\mathbf{C}_{om}\ \mathbf{A}_{om}^{-1}\ \mathbf{B}_{om} - \mathbf{X}_o\ \mathbf{A}_{r0}^{-1} \Rightarrow \mathbf{D}_{om}$
8. Set $\mathbf{y}_{10} - \mathbf{H}_1\ \mathbf{u}_0 \Rightarrow \hat{\mathbf{x}}(0)$

This algorithm can be considered symbolically as $\{\mathbf{u}(k), \mathbf{y}(k)\} \Rightarrow \boldsymbol{A}_{om}, \boldsymbol{B}_{om}, \boldsymbol{C}_{om}, \boldsymbol{D}_{om}, \hat{\boldsymbol{x}}(0)$, which simply applies the relationship between an input/output stream and a corresponding state space description in a selected pseudo-observable form to obtain an estimation of a particular set of state variables for the identified system model.

4.1.C. Linear Part Identification

From Eqn. (2) the linearized model of the inverted pendulum system, assuming that θ and $\dot{\theta}$ are small ($\sin\theta \simeq \theta$, $\cos\theta \simeq 1$), and all product terms involving θ, $\dot{\theta}$ and $\dot{\theta}^2$ are eliminated is given by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-3mg}{4M+m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{6g(M+m)}{l(4M+m)} & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{4}{4M+m} \\ 0 \\ \frac{-6}{l(4M+m)} \end{bmatrix} u(t) \qquad (57)$$

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

where the states are: $x_1 = p$, $x_2 = \dot{p}$; $x_3 = \theta$, $x_4 = \dot{\theta}$; and (measured) outputs, $y_1 = p$ and $y_2 = \theta$. In this inverted pendulum problem, distance p lies between -3 to +3 and force, u goes from -100 to 100 to achieve the control objective for a medium range of initial deflections, θ. A linearized discrete-time inverted pendulum system is extracted by restricting the data collection to a small range of pendulum angle, i.e. between +/-0.3 rad from the system dynamic Eqn. (2) using Euler's forward-difference approximation (the sampling interval, 0.002 sec), which is very close to the discretized version of Eqn. (57). The method used is a conventional approach which exhaustively considers all possible multivariable observable structures to provide that linear model which generalizes best over unused data.

It has been demonstrated that identification of linear MIMO systems can be

greatly enhanced through the investigation of all possible multivariable (observable) structures. In particular, the concept of *generalization*, the ability to perform well on data not used in the identification process, is considered as the principal criterion for comparison between different candidate models. As will be seen below, the preferred method of determining the system order, as well as its linear structure, is based on how well the resulting model "generalizes" over data not used in the identification process.

As a result of the preliminary analaysis based on input/output data, the estimated order of the system lies between 2 and 6, which is fairly dependent on data collection containing a degree of nonlinearity of the system. If data were collected from far beyond the linear region, the sixth order system proved to be more accurate than a lesser order system [23]. According to the procedure described in Chapter 2 the system parameters from a specified set of indices are identified, and then the model's performance over the unused data set in training are tested. The procedure will be to use the initial data, 200 out of 300 data pairs to select best POFs testing over the model's performance over the complete data set. The measure of the performance is given by the Frobenius norm of the error matrix between the supplied output data and the model output. The results are compiled in Table 2. For each system order between 2 and 6 each possible (observability) index set is used to calculate an "identified" model.

As can be seen from Table 2, several models show a very small error norm. The selection is made with a strong consideration of achieving the smallest order of the system which also matches the data well. In this case the model selected to represent the system which has the best generalization corresponds to the order-4 system with indices {2,2}. The responses of the selected model have an error norm of 0.0701 comparable to other higher order models which have similar error norms

and are shown in Figs. 50 and 51. For comparison, the responses of another order-4 candidate model with an error norm of 1.8283, using indices {1,3}, are presented in Figs. 52 and 53. Also those of order-6 candidate model with an error norm of 0.1752, using indices {3,3}, are presented in Figs. 54 and 55. Apparently, selecting higher order forms does not mean a better representation of the system.

TABLE 2. Computational Results

Order	Indices	Norm
2	1,1	7.7104
3	2,1	3.7243
	1,2	1.6951
4	1,3	1.8283
	2,2	0.0701
	3,1	4.3501
5	4,1	1.5868
	3,2	0.2764
	2,3	0.0599
	1,4	1.5868
6	5,1	1.3523
	4,2	5.3902
	3,3	0.1752
	2,4	0.0565
	1,5	1.1634

The performance of other candidate models can be inferred from the norm-column in Table 2. In all six Figs. 50 to 55 both curves represent the measured (solid) and the predicted (dotted). Note that the difference between the data and the model outputs is not discernable in Fig. 50, but can be recognized toward the end of the time period in Fig. 51, which shows the instability of the given system.

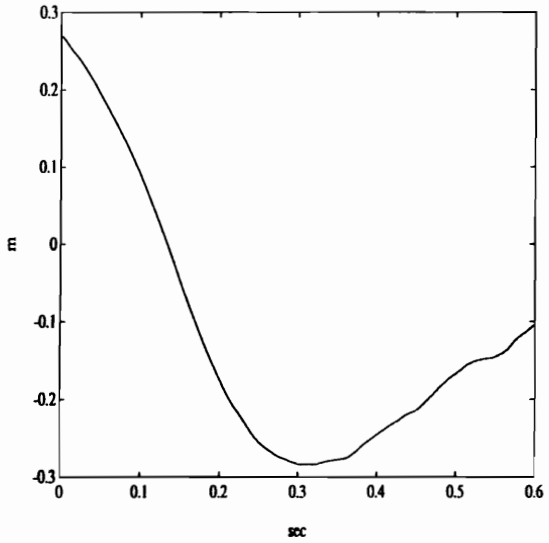


Fig. 50. Linear Model (Position): Indices {2,2}

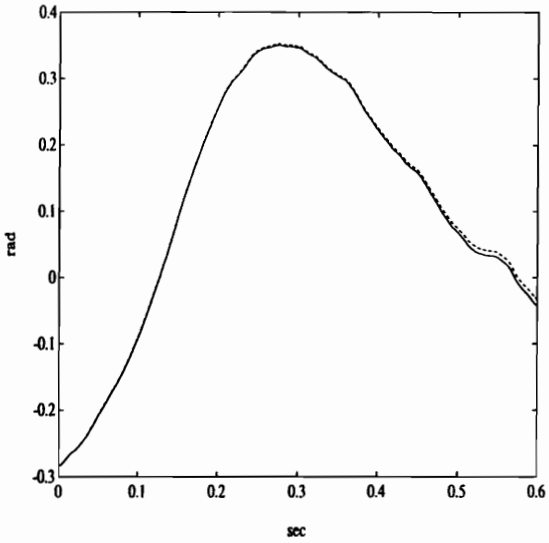


Fig. 51. Linear Model (Angle): Indices {2,2}

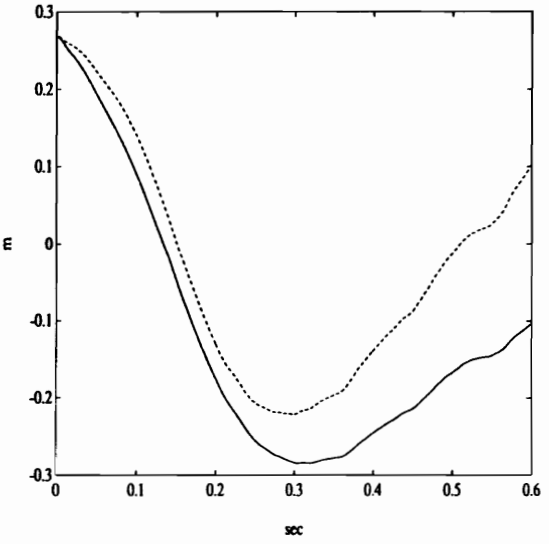


Fig. 52. Linear Model (Position): Indices {1,3}

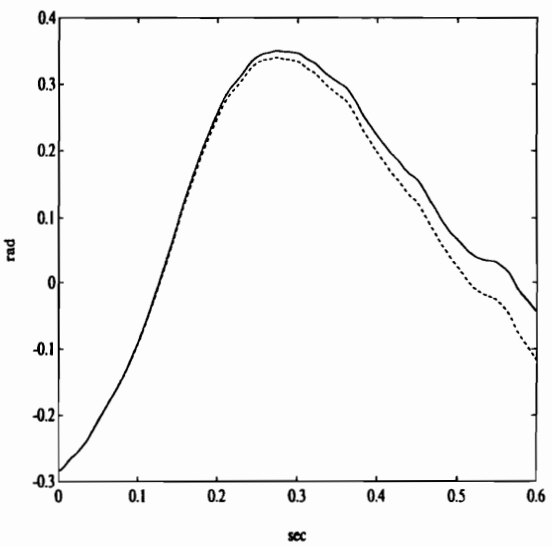


Fig. 53. Linear Model (Angle): Indices {1,3}

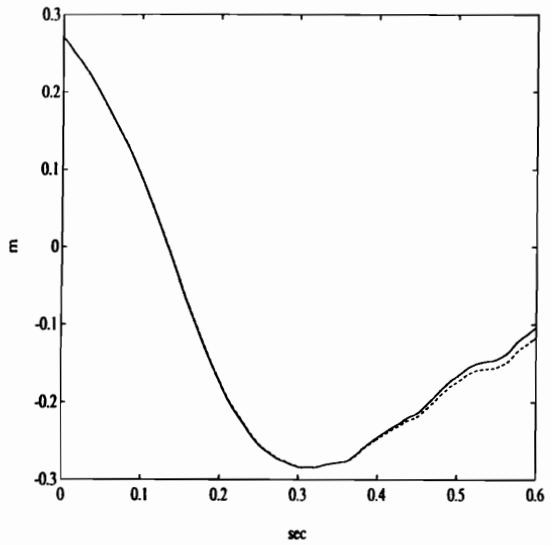


Fig. 54. Linear Model (Position): Indices {3,3}

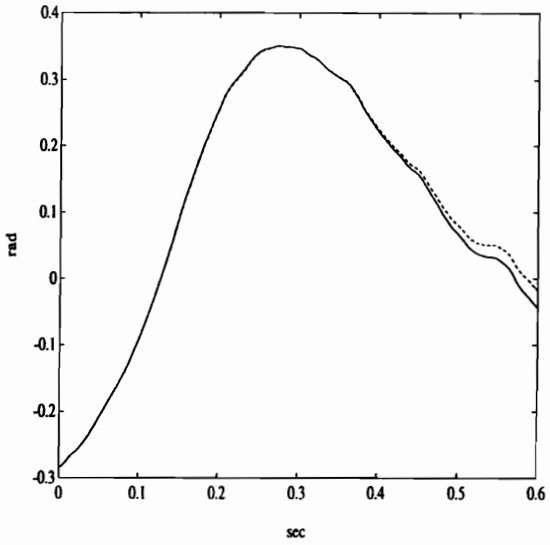


Fig. 55. Linear Model (Angle): Indices {3,3}

It is during the last 100-data points (between 0.4 sec and 0.6 sec) that the model reveals the degree of generalization achieved. Considering the instability of the system and the amount of initial conditions never used in training, those two trajectories are quite acceptable. Again the solid line represents the measured or desired responses and the dotted, the MLP predictions. The selection is made with a strong consideration of achieving the smallest order of the system which also matches the data well. In this case the model selected to represent the system which has the best generalization corresponds to the order-4 system with indices {2,2}.

This system is given below as a state space representation in POF form, along with the identified initial state. First, identified **A/C**, then **B/D** and finally the identified initial state **x**(0) are given:

$$\begin{bmatrix} \mathbf{A} \\ \text{---} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1.0002 & -.0002 & 2.0002 & .0002 \\ .0020 & -.9985 & -.0020 & 1.9986 \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and (58)

$$\begin{bmatrix} \mathbf{B} \\ \text{---} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} 4.369 \\ -8.439 \\ 8.340 \\ -16.106 \\ \text{-----} \\ .872 \\ -1.679 \end{bmatrix} * 1.0E\text{-}06, \quad \mathbf{x}(0) = \begin{bmatrix} .2689 \\ -.2838 \\ .2663 \\ -.2815 \end{bmatrix}$$

The eigenvalues of the identified linear model are {1.0001±j0.0007, 0.9898, 1.0089}, compapring with those of analytical linearized model of the inverted pendulum system, that is, {1.000, 1.000, 1.0080, 0.9921 }.

The ARMA representation of this model is, of course, equivalent to the weights of the (linear) neural network. The {2,2} observability indices were chosen with the resulting linear model:

$$[\mathbf{y}(k+2)] = [\,\mathbf{N} \mid \mathbf{D}\,]\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \text{-----} \\ \mathbf{y}(k) \\ \mathbf{y}(k+1) \end{bmatrix}$$

where

$$\mathbf{N} = \begin{bmatrix} 0.0475 & 0.2625 & 0.0872 \\ -1.0908 & -0.5082 & -0.1679 \end{bmatrix} x\ 1.0E\text{-}05 \tag{59}$$

and

$$\mathbf{D} = \begin{bmatrix} -1.0002 & -.0002 & 2.0002 & .0002 \\ .0020 & -.9985 & -.0020 & 1.9986 \end{bmatrix} \tag{60}$$

4.1.D. Nonlinear Part Identification

From Eqn. (2) the system is seen to be more nonlinear as the angle θ increases. The second stage is therefore to identify the nonlinearity of the system by training the network with a hidden layer in place, but with the linear feedthrough terms fixed. The McCulloch-Pitts neurons are added and the back-propagation algorithm is applied. The nonlinear activation or squashing function in the hidden layer of the network is chosen to be the hyperbolic tangent function. The hyperbolic tangent transfer function is quite similar to the more common sigmoid transfer function in its s-shaped form. However, its output range is -1 to +1, as opposed to the sigmoid range of 0 to 1. The hyperbolic tangent is preferred because of its symmetry about the origin, as well as its larger derivative around the origin which enables the network to converge faster.

Using the results of the previous stage, the linear structure of the order-4 system with indices {2,2} and feedthrough weights of the MLP are determined and fixed. This part of the MLP captures the plant's linear characteristics around its equlibrium point. In order to represent the nonlinearity of the system, new data sets are supplied to the network. At this point the system (which has been stabilized locally with a linear controller) is "persistently excited" as close as possible to the maximum desired deviation from the equilibrium point. In the third training phase hidden layer neurons are added to the network in a sufficient number to permit the capture of the system nonlinearities. In our experiment three neurons were added simultaneously and found to be acceptable. If the number of hidden neurons is overestimated, "pruning" can be used to trim out the excess. It is known that a minimum size network which performs well on training data tends to also do well on test data [56]. In order to achieve a minimal network size, not only are

small weight connections removed, but also existing hidden neurons if their contribution is insignificant. The algorithm of pruning networks is as follows:

1. First, a pool of neurons is added at each time when no significant error reduction has occurred after a predetermined number of training epochs. Since we do not know how many neurons are required to represent the system nonlinearities until a certain number of neurons are added to the network which meet the performance criterion, it is necessary to continue to build a network until we are satisfied with the network's performance. The concept of adding multi-neurons at a time is that if there are very severe nonlinearities found in given system, a pool of neurons can speed up training because many parts of weight space can be explored simultaneously [57]. For future reference, the entire initial network should be stored in memory.

2. After building a network which is complex enough to handle the problem, a specified percentage of weights are pruned at each specific checking point until the same performance criterion as the previous network is acquired. Weights that are disabled might be ones which connect the input-layer to hidden-layer or the hidden-layer to the output-layer.

3. Check through the weights of hidden-layer to output-layer to see if some of them are disabled to specific output neurons, but not to the other ones. In this case, the necessity of these neurons to the entire network can be questioned because they do not contribute to all output neurons to decrease each output error. After recording all the weights of the network again,

disable these neurons one by one temporarily and train the network until the error is small enough to be accepted.

4. Check again through the weights connected from input neurons to neurons in hidden layer. If the weights are disabled more than a specific level; for example, 3 out of 7, disable those neurons one by one according to the number of disabled weights. Since those neurons are not related to all of the inputs, they do not completely contribute to modeling the input-output structure.

5. After recording the weights of the pruned network at a specific percentage level, e.g. 5 percent of the nodes removed, one may increase the level to achieve an even more compact network. Repeat this procedure until no further significant error reduction has occurred at that increased percentage of pruning.

There are two advantages of pruning networks. First, pruning is very useful in multi-output cases, where more complexity exists than with a single output network, and the size of the network is fairly dependent on guesswork, therefore, overestimating the network is typical. Second, according to the development of hardware in recent years, it is becoming easier to implement neural network software into parallel processing chips where the network size is crucial to be cost effective.

In order to achieve the desired accuracy for the cart/pendulum problem, 4 hidden neurons of the single hidden layer result in a reasonably accurate represention the system dynamics out to about 85 degrees of pole deflection. The

final MLP model consists of 7-inputs with bias, 4-hidden layer nodes and 2-output nodes.

Output functions from the identified function are of the form:

$$y = \sum_{i=0}^{7} a_i u_i(k) + \sum_{i=1}^{4} c_i \tanh\left(\sum_{i=0}^{7} b_i u_i(k)\right) \tag{61}$$

Figs. 56 and 57 illustrate the two response signals of the MLP model and those of the linear model to the pseudo-random input signal shown in Fig, 58. Considering the instability of the inverted pendulum system, fairly satisfactory identification of the system for a range of the pole deflection angle of up to 85 degrees was obtained. Even though the cart position was not of concern because it is not critical to the performance of the system, it is identified accurately.

The completed system model can be represented by the structure of Fig. 59, consisting of a linear part and a nonlinear part, all implemented in a feedforward MLP. The technique of establishing the linear part separately with a robust conventional method helps greatly with the training process. Without this first step, it has been found to be difficult to achieve the small number of hidden layer nodes obtained with this method, and virtually impossible to achieve the same accuracy.

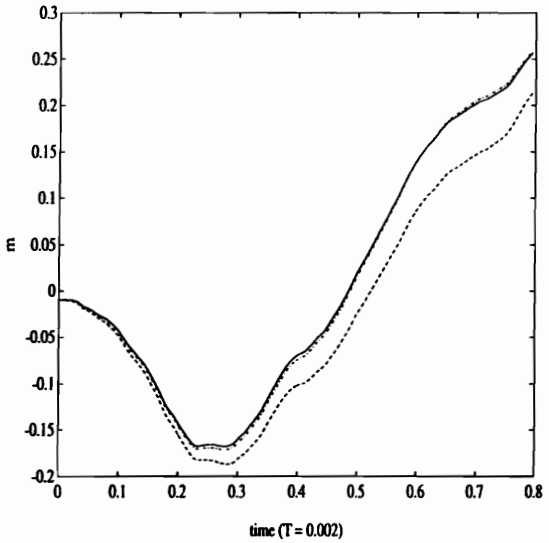


Fig. 56. Position:
Given data (solid), Linear model (dashed), MLP model (dashdot)

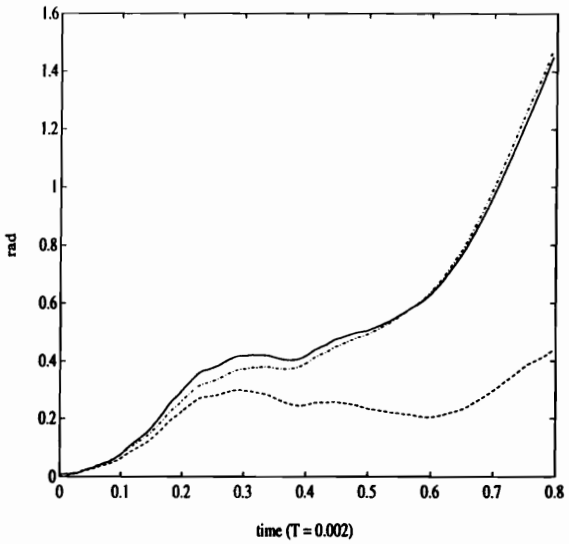


Fig. 57. Angle:
Given data (solid), Linear model (dashed), MLP model (dashdot)

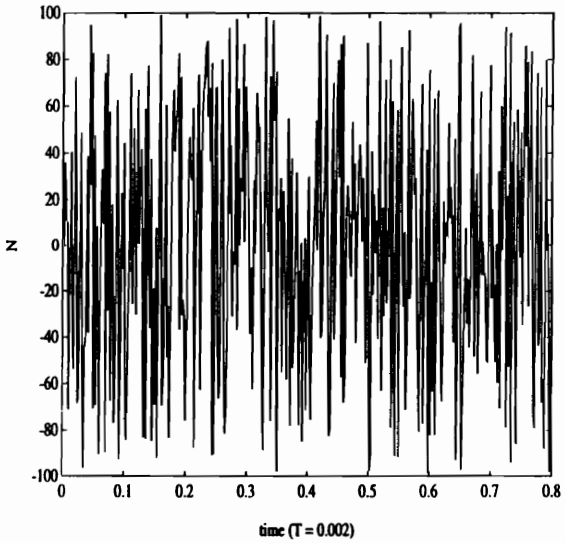


Fig. 58. Input Signal for the D-T System

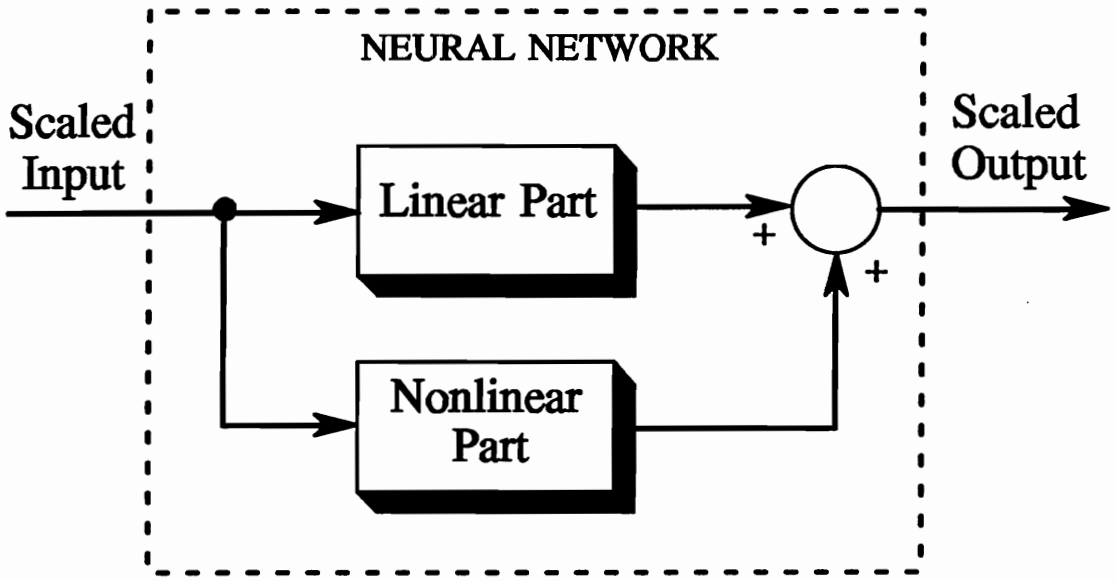


Fig. 59. Block Diagram of MLP

The network inputs and outputs are

$$\text{inputs} = \begin{bmatrix} 1 \\ u(k) \\ u(k+1) \\ u(k+2) \\ y_1(k) \\ y_2(k) \\ y_1(k+1) \\ y_2(k+1) \end{bmatrix}, \quad \text{outputs} = \begin{bmatrix} y_1(k+2) \\ y_2(k+2) \end{bmatrix} \tag{62}$$

Output functions from the identified function of the inverted pendulum system are of the form:

$$\mathbf{a} = \begin{bmatrix} -0.0051001315 & 0.0040995111 \\ 0.0038525667 & -0.00025528629 \\ 0.0021528113 & -0.0074752807 \\ -0.0007713557 & 0.015769219 \\ 0.44584292 & -0.081697509 \\ 0.034205906 & 0.31647062 \\ 0.36140302 & 0.0813438 \\ -0.018705459 & 0.45631856 \end{bmatrix}^T$$

$$\mathbf{b} = \begin{bmatrix} -0.011112717 & -0.041258093 & 0.056903895 & 0.0041546146 \\ 0.020937189 & -0.073869497 & -0.029067401 & -0.068266794 \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 0.025125695 & -0.0095046982 & 0.066035889 & 0.046637584 \\ -0.018313603 & 0.03809898 & -0.017835924 & -0.036910675 \\ -0.055105288 & -0.089030176 & -0.09089753 & 0.0082838954 \\ -0.0482435 & 0.096246056 & 0.03245927 & 0.096631818 \\ -0.039525319 & -0.034028154 & 0.020717261 & 0.018991759 \\ -0.074755631 & -0.095967062 & 0.033297442 & -0.12043883 \\ -0.010378495 & 0.00021409578 & 0.090301216 & -0.007418009 \\ 0.082555152 & -0.039107706 & -0.080727264 & -0.080162108 \end{bmatrix}^T \quad (63)$$

Previously, the "scaling" of the raw data into the MLP was done on the individual inputs for training the MLP without causing initial saturation according to the Eqns. (3) and (4) in Section 2.2. For the 7-inputs, the following scale factors were used,

[0.010039523 0.010039523 0.010039523 14.094272
1.1879871 14.094272 1.1781759]

along with shifts

[0.0038910506 0.0038910506 0.0038910506 -0.061599322
0.055544602 -0.061599322 0.055210775].

After the MLP produces scaled output, the mapping from the MLP output to the inverted pendulum system is "descaled" by:

[0.088688511 1.069437]

with shifts,

[0.0043705218 -0.047366798]

permitting the MLP to be compared directly to the real world inverted pendulum system.

4.2. Design of Neuro-Controller

Considering the design of a nonlinear controller in the presence of model uncertainties such as nonlinearities and time-varying dynamics of the complex system in the noisy environment, not only robust but also highly adaptable control algorithms should be developed. The present controller consists of a state estimator, which is developed from the identified linear model of the system, and a neural net trained from data collected using selected sets of feedback gains which will be established for a predetermined number of operating points of the plant. Therefore, it is necessary to design different control gains according to each operating point chosen selectively. The data is obtained using a switched-linear controller which is constructed for a predetermined number of operating points of the plant in order to develop a neuro-controller at the initial stage of training. The neural net is needed for implementation and interpolation, two inherent and attractive properties of feedforward multilayer networks. Specifically, the inverted pendulum system is divided into two different regions based on angle and position of the cart.

The algorithm we present in the following two sections is fairly "robust" in the sense that the resulting optimal linear state feedback controller, as a role model for the neuro-controller, stabilizes the system against a certain range of deviation at the initial stage and also due to subsequent training of neuro-controller based on a desired performance index the neuro-controller becomes highly "adaptable" because of the capability of neural networks to learn the system's varying dynamics through nonlinear mapping. This last step ensures that the controller goes beyond its previous training and captures the necessary control actions for the nonlinear plant.

When the system is ill-defined or only partially understood and/or, especially, unstable around the desired operating point such as the inverted pendulum problem, an indirect control scheme of Fig. 1 should be the best choice. If data were collected from a real world system instead of a simulated environment, a restricted range of data could be acquired due to the limited performance of the existing linear controller which stabilizes the unstable system around a specific operating point. Since the importance of a critical industrial process is taken into consideration, the devised neuro-controller cannot be installed to the real plant directly at the initial stage. Therefore, the development of a neuro-controller proceeds with the following two steps. At the first step, the identified nonlinear model using input/output data can be used as a simulation tool in combination with the designed controller without risk of experimenting into the real system directly. At the second step, a fine-tuned neuro-controller controls the system in parallel with the previously identified model using a dynamic MLP "on-line" in order to deal with the dynamics of the system.

4.2.A. Step 1 - An Initialization of Neuro-Controller (Switched-Linear Control)

If we can utilize a conventional control technology with *a priori* knowledge of the identified model using one MLP, it will be helpful in designing a more complicated neuro-controller. At this stage, using only the identified linear model of the nonlinear system of Eqn. (58), we can construct a linear optimal controller off line based on a linear quadratic regulator (LQR) method. Even though the LQR problem has been known to have an inherent robustness against a certain range of model uncertainties, it generally cannot cover the entire domain of attraction due to the initial assumption of neglecting the system's nonlinearity. But considering

the importance of the initialization of the neuro-controller, it must be utilized as one tool instead of merely assigning small random numbers as initial weights of the MLP at the initial training stage of the MLP. The discrete time system of Eqn. (58) is *controllable* if it is possible to force the state from any initial state $\mathbf{x}_0$ to an arbitrary " target " state $\mathbf{x}_f$ in a finite number of states. We can use this definition to derive a simple rank calculation to test for the property of controllability in a linear system [58]. Unfortunately, there is no such easy test to apply to a nonlinear system. Therefore, instead of establishing conditions for global controllability we may have to resort local notions. Narendra et al introduced a locally controllable system [22]. From the same concept which we used in designing the nonlinear state estimator, we can check the controllability of the nonlinear system on our domain of attraction. The controllability of the nonlinear system around an equilibrium state can be derived by examining its linearization around the equilibrium point, e.g. $\mathbf{x}_e = \mathbf{0}$. In order to check the stabilization of the nonlinear system around an equilibrium point, the rank calculation was done by checking that the rank of the linearized model's controllability matrix ϑ , n, is 4, where ϑ is $[\ \mathbf{B}_o\ \ \mathbf{A}_o\mathbf{B}_o\ \ \mathbf{A}_o^2\mathbf{B}_o\ \ .\ \ .\ \ .\ \ \mathbf{A}_o^{n-1}\mathbf{B}_o\ ]$. But since all four states are not measurable, a state estimator must be devised first. Using the same concept above in the definition of controllability of the nonlinear system, the observability of the nonlinear system in the domain of attraction was checked and proved to be observable. From the method described in Section 4.1.A an estimated state space model of Eqn. (58) is given below:

$$\begin{bmatrix} \mathbf{A} \\ \text{---} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} 1.0000 & .0000 & 1.0000 & .0000 \\ .0000 & 1.0000 & -.0000 & 1.0000 \\ .0000 & .0001 & 1.0103 & .0087 \\ .0000 & -.0001 & -.0214 & .9822 \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and (64)

$$\begin{bmatrix} \mathbf{B} \\ \text{---} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} 2.532 \\ -4.928 \\ 2.885 \\ -5.606 \\ \text{-----} \\ -.000 \\ .000 \\ 2.532 \\ -4.928 \end{bmatrix} * 1.0E\text{-}06, \quad \hat{\mathbf{x}}(0) = \begin{bmatrix} .2689 \\ -.2839 \\ -.0026 \\ .0024 \end{bmatrix}$$

The eigenvalues of the identified linear model are {1.0014±j0.0093, 0.9891, 1.0006}, which are slightly different to those of the analytical linearized model of the system, due to the lack of information of not using the other original two states, namely, velocity and angular velocity. The responses of the identified model including the estimates of two other states are shown in Figs. 60, 61, 62 and 63.

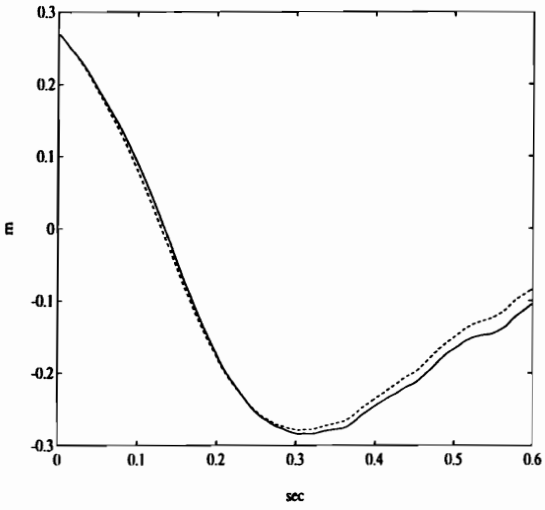


Fig. 60. Position: Measured (solid), Predicted (dashed)

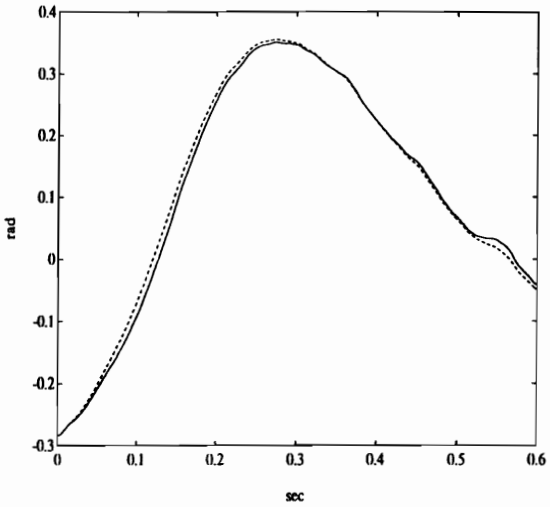


Fig. 61. Angle: Measured (solid), Predicted (dashed)

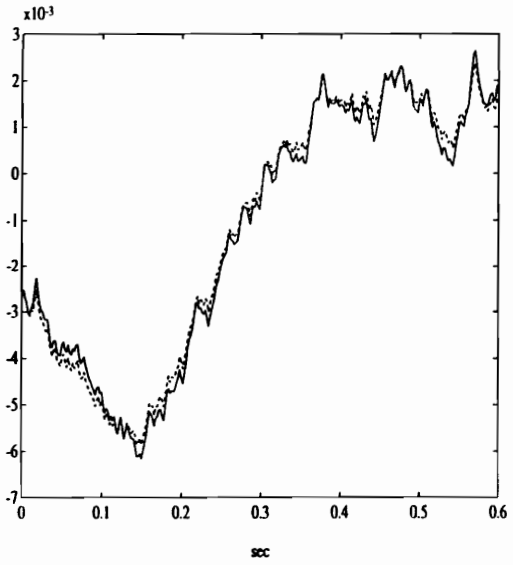


Fig. 62. State #3 : Measured (solid), Predicted (dashed)

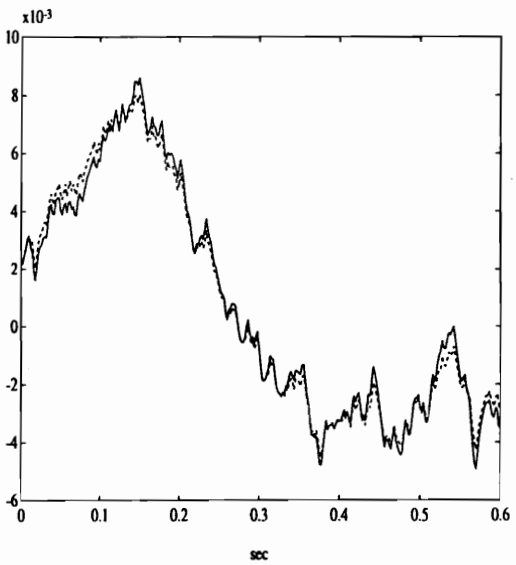


Fig. 63. State #4 : Measured (solid), Predicted (dashed)

Also, for the purpose of comparing with the estimates of the two states, velocity and angular velocity from the plant are shown in Figs. 64 and 65. As the result of the identification process, the estimated states of Figs. 62 and 63 are truely close trajectories of the original two states of Figs. 63 and 64 except scaling.

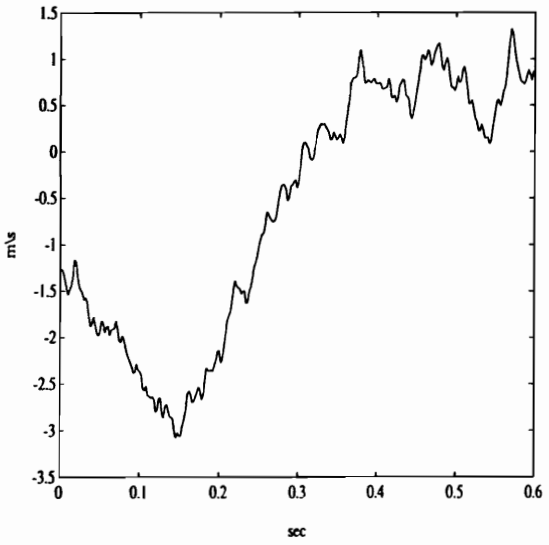


Fig. 64. Measured Velocity

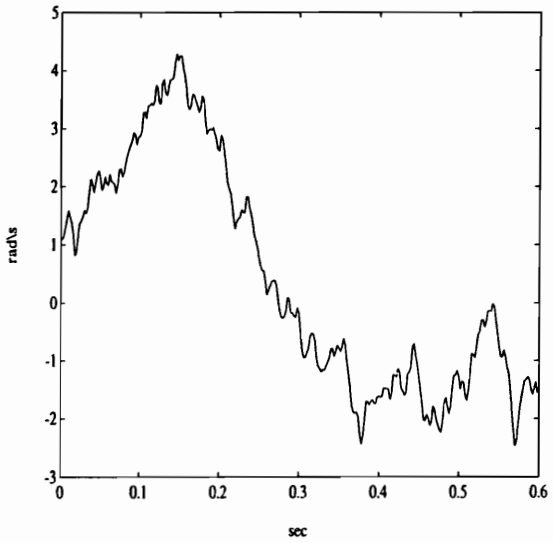


Fig. 65. Measured Angular Velocity

In order to design a linear optimal controller around an equilibrium point the particular method was chosen to formulate an estimated state space model from Eqn. (64), i.e.

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{F}\hat{\mathbf{x}} + \mathbf{G}\mathbf{u} \\ \mathbf{y} &= \mathbf{H}\hat{\mathbf{x}} \end{aligned} \tag{65}$$

and use a Linear Quadratic Regulator (LQR) approach with a cost functional given by

$$J = \frac{1}{2}\int_0^{\infty} (\hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \tag{66}$$

The well known *optimal control* is

$$\mathbf{u}^* = -\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\hat{\mathbf{x}} \tag{67}$$

where **P** is the solution of following *Riccati* equation:

$$\mathbf{PF} + \mathbf{F}^T\mathbf{P} + \mathbf{Q} - \mathbf{PGR}^{-1}\mathbf{G}^T\mathbf{P} = \mathbf{0} \tag{68}$$

In the actual calculation the discrete-time version of the LQR problem was implemented (with a sample interval of T = 0.002 sec.). The result of this phase of design is a constant feedback gain matrix **K** below.

$$\mathbf{K} = [-0.0354 \quad -0.1156 \quad -7.7587 \quad -8.8601] * 10E{+}03 \tag{69}$$

where

$$\mathbf{Q} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 3.0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}, \qquad \boldsymbol{R} = 0.5E{-}03 \tag{70}$$

But with this **K**, it is difficult to cover the entire range of the parameter space. Through the process of optimizing the state feedback controller, it is realized that by optimizing the system for the angle θ, the closed loop system can give excellent performance against large initial angle deflections. However, the settling time for

the position p returning to the origin is large. On the other hand, if the system is optimized for the position p, the cart will quickly return to the origin; but, control performance is only good for small initial angle θ [59]. Given the above observations, the idea of ***switched-linear control,*** also called ***piecewise linear control*** or ***gain scheduling***, will be incorporated. Therefore, it is necessary to design different control gains according to each operating point chosen selectively. Specifically, the inverted pendulum system is divided into two different regions based on angle and position of the cart. Once one state feedback controller **K1** stabilizes the system for large initial deflection θ, another controller **K2** brings the cart to the origin quickly. The switching position is determined when the state $\hat{\mathbf{x}}(k)$ is near zero velocity and very small deflection angle θ. Therefore, a switched-linear controller was designed as following:

Region 1: Large initial angle deflection θ, and small position deviation p,

$$\mathbf{K1} = [-0.0204 \quad -0.0770 \quad -3.8964 \quad -5.7906] * 10E{+}04 \tag{71}$$

where

$$\mathbf{Q} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 30.0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 5.0 \end{bmatrix}, \qquad \boldsymbol{R} = 1.0E{-}02 \tag{72}$$

Region 2: Small initial angle deflection θ, and large position deviation p,

$$\mathbf{K2} = [-0.0127 \quad -0.0193 \quad -2.5924 \quad -1.8877] * 10E{+}04 \tag{73}$$

where

$$\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{R} = 1.0E{-}02 \tag{74}$$

Figs. 66, 67, 68, 69 and 70 show the comparison of simulation results of the closed-loop system using the switched-linear controller and the single (**K1**) linear optimal controller.

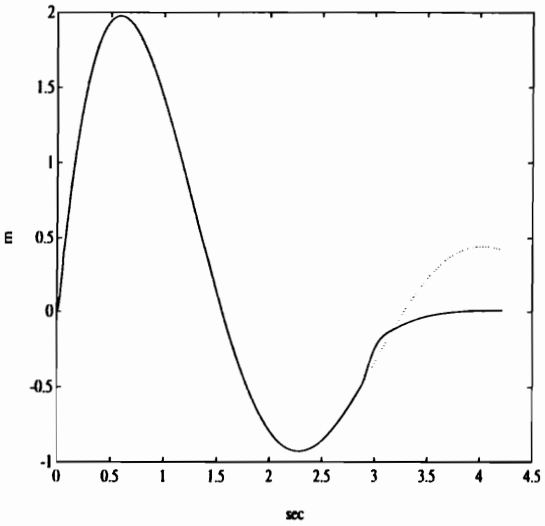


Fig. 66. Position: Switched-Controller (solid), Single-Controller (dotted)

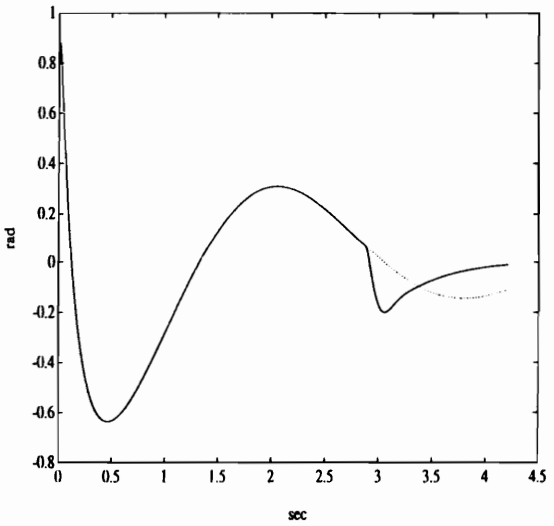


Fig. 67. Angle: Switched-Controller (solid), Single-Controller (dotted)

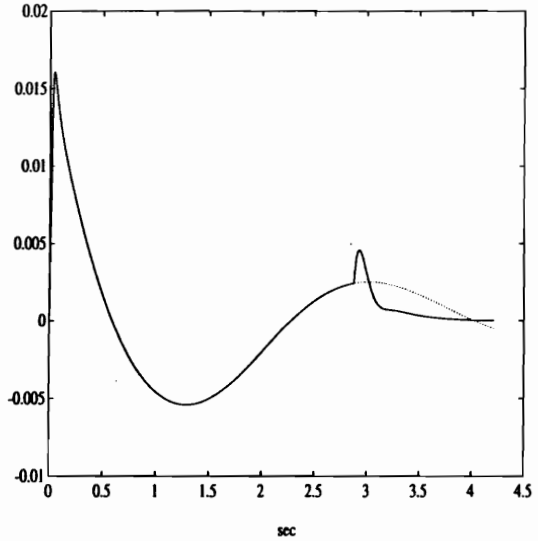


Fig. 68. Estimated State ($\hat{x}_3$): Switched-Controller (solid), Single-Controller (dotted)

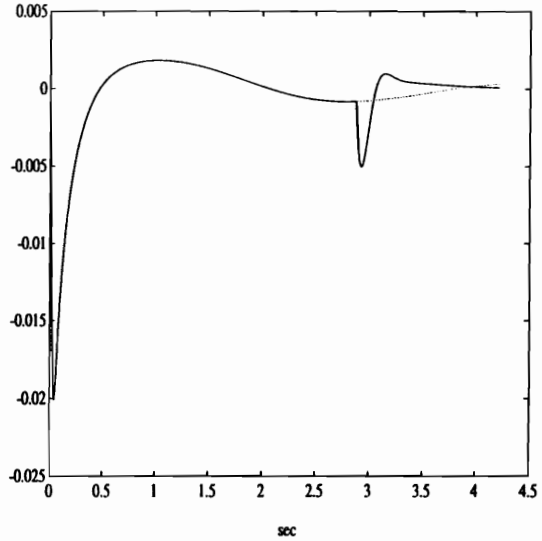


Fig. 69. Estimated State ($\hat{x}_4$): Switched-Controller (solid), Single-Controller (dotted)

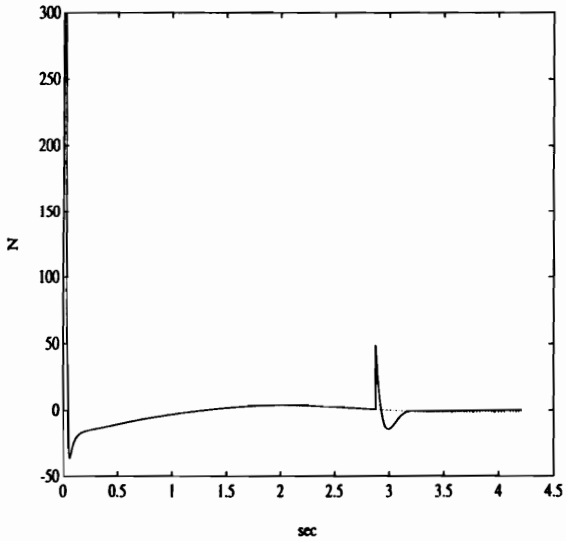


Fig. 70. Force: Switched-Controller (solid), Single-Controller (dotted)

From the above figures, superior characteristics of the switched-linear controller to the single linear controller can be recognized except the spike on the Fig. 70, which shows the switching action. Otherwise, the control objective can be achieved satisfactorily.

As previously discussed, the switched-linear controller serves as an initial role model for the training of the neuro-controller [14]. The completion of this design step is the partial training of the neuro-controller using the setup illustrated in Fig. 71.

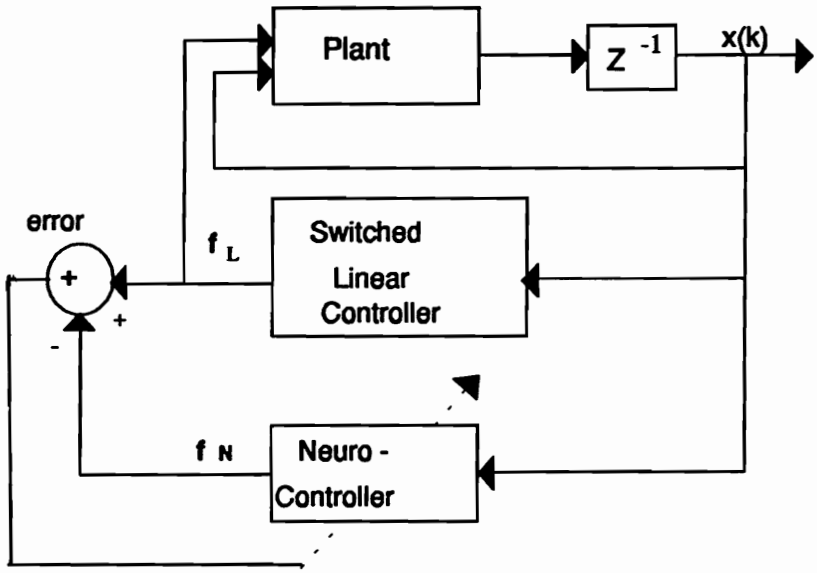


Fig. 71. Structure for Training Neuro-Controller

From this simulation the data will be taken to train the neural network. The components of $\hat{x}$ (measured two outputs and estimated two states) is considered as "inputs" and those of **u**, "outputs" for the training phase. At this stage the neuro-controller learns to mimic the switched-linear controller and interpolates the gains between operating points. After completion of learning, the neuro-controller will be used in place of the switched-linear controller and, therefore, will have a wide range of performance over the entire region of attraction.

The resulting neural network consists of 4-inputs, 10-hidden layer nodes in one hidden layer and 1-output node with fully connected by an array of weights $\mathbf{W}_1$ from inputs to hidden layer, $\mathbf{W}_2$ from hidden layer to output, and an array of "feedthrough" weights $\mathbf{W}_F$ from inputs to output with no bias elements.

$$\mathbf{Y} = \tanh(\mathbf{W}_F \mathbf{U} + \mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{U})) \tag{75}$$

As an output neuron a hyperbolic tangent function was used in contrast with a linear output function in the identifying system. Figs. 72 and 73 show the training results and the resulting errors of the neuro-controller mimicking the switched-linear controller in setup Fig. 71. The performances of the neuro-controller are

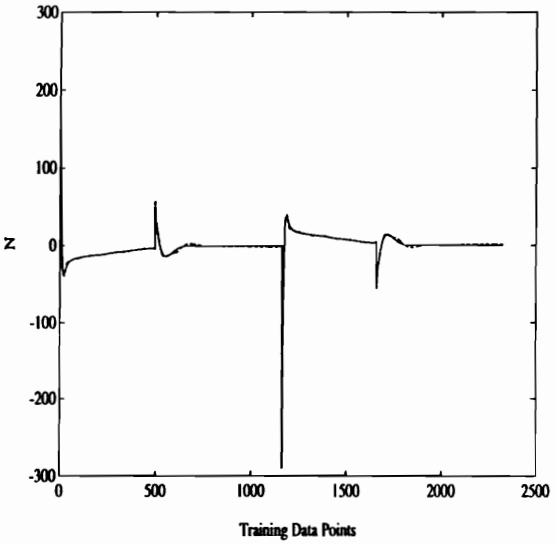


Fig. 72. Neuro-Controller Training Results

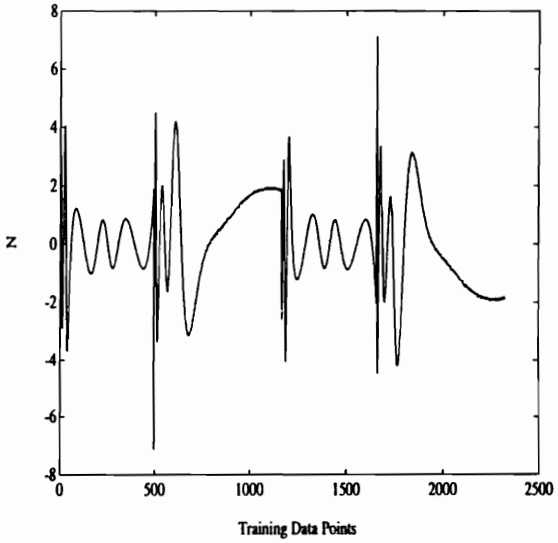


Fig. 73. Neuro-Controller Training Results

shown in Figs. 74, 75 and 76. Abrupt changes due to the switching action are not noticeable in these figures. Especially, the sharp spike shown in Fig. 70 does not exist in Fig. 76 due to the smoothing effect of the neuro-controller.

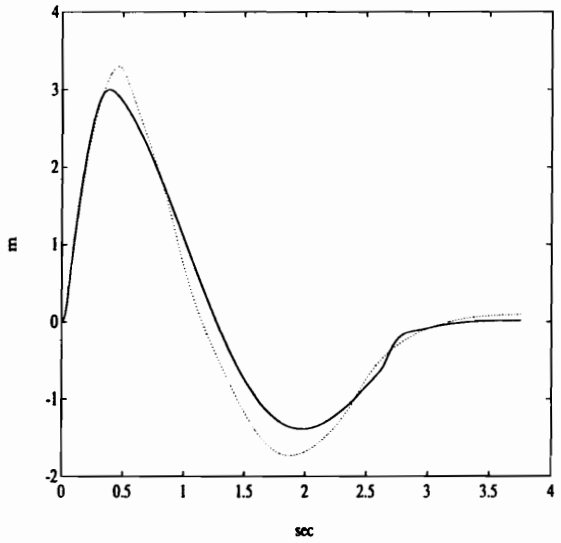


Fig. 74. Position: Switched-Controller (solid), Neuro-Controller (dotted)

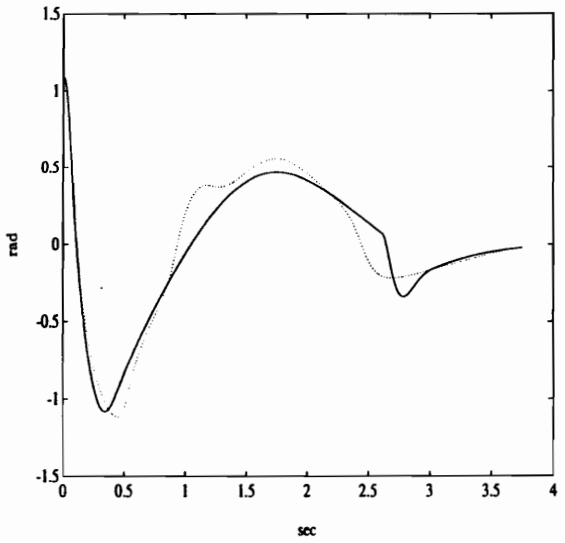


Fig. 75. Angle: Switched-Controller (solid), Neuro-Controller (dotted)

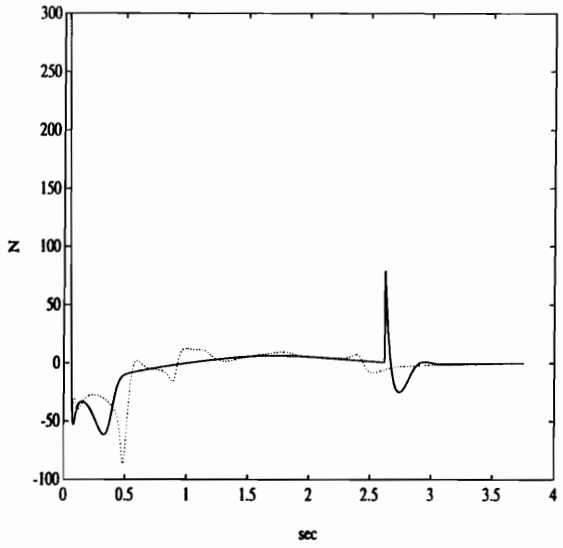


Fig. 76. Force: Switched-Controller (solid), Neuro-Controller (dotted)

A drawback of this neuro-controller is that it cannot deal with the time-varying dynamics of the plant properly due to the fixed weights and, therefore, needs additional training to be adaptable for tracking plant dynamics. Thus, the last step in neuro-controller design, described in the next section further trains the neuro-controller based on a desired performance index as a more *robust adptive nonlinear controller*, thereby eliminating the previous training and adjusting the controller characteristics to stabilize the plant against time-varying dynamics caused by different initial conditions of the unstable system [19]. The resulting neural network controller has the following initial weights with the "scaling" of the 4-inputs and the "descaling" of the output along with no shifts:

SC = [0.50565883 1.1111111 60.957027 48.93086]

DSC = [760.55762]

$$W_1 = \begin{bmatrix} -0.103356 & -0.11608053 & -7.3664923 & 2.8680472 \\ 2.540396 & 3.4299672 & -0.99100953 & 3.6281192 \\ 0.54596907 & 2.0968528 & 3.6568153 & 5.7834744 \\ -0.10546701 & 0.83742344 & 5.1867628 & 0.22185375 \\ -4.4783792 & 1.9706029 & 3.9137454 & 5.109971 \\ 4.0894098 & 1.1906618 & 5.5675988 & 3.2447574 \\ -9.0530691 & 1.9246349 & 3.1605353 & 1.3168296 \\ -3.2654526 & -4.7990246 & 4.6983099 & -4.9487987 \\ 1.7438481 & 2.3845084 & 1.5622174 & 4.0785146 \\ 0.06101219 & -9.4872713 & -0.8210972 & -4.8861198 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} -0.087395087 \\ 0.071130477 \\ 0.069655232 \\ 0.052743338 \\ 0.062222745 \\ 0.067504287 \\ 0.08149299 \\ -0.079884142 \\ 0.080000825 \\ 0.098448247 \end{bmatrix}$$

$$W_F = \begin{bmatrix} 0.15869269 \\ 0.30559608 \\ 0.23306966 \\ 0.41897148 \end{bmatrix}$$

4.2.B. Step 2 - Performance Training

In recent nonlinear control, one of the main trends is borrowing the concept of linear optimal theory in ANNs using not only quadratic errors but also a more general cost function (or called performance criterion) to reduce system output errors. Since ANNs are used to take into account nonlinear effects of the system beyond the conventional linear optimal controller, ANNs broaden the range of control beyond the limited range of using linear optimal control law alone. In designing the above mentioned controllers the most important thing is to develop an efficient training algorithm. Utilizing the highly adaptable characteristics of neural networks, more robust adaptive control algorithms are explained. In order to achieve supervised learning for neuro-controller in this final stage of the design, we can use a traditional performance index given by

$$E = (\mathbf{y} - \mathbf{d})\mathbf{Q}(\mathbf{y} - \mathbf{d})^T = \sum_{j=1}^{2} [y_j(k+1) - d_j]\mathbf{Q}[y_j(k+1) - d_j] \tag{76}$$

where

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, \qquad d = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{77}$$

i.e. **d** is the desired (balanced) position and angle of the system. In this case the standard *back-propagation* learning algorithm should be modified so that the weights, w_i, of the neuro-controller can be updated according to

$$\Delta w_i = -\frac{\partial E}{\partial w_i} \tag{78}$$

At this stage of the design it is known that

$$y_j(k+1) = f_j[\mathbf{x}(k), u(k)] \ , \quad \text{for} \quad j = 1, 2 \tag{79}$$

where analytic expressions are available in Eqn. (61) from the previous identification step. Therefore, it is possible to expand Eqn. (79) as follows:

$$\Delta w_i = -\frac{\partial E}{\partial w_i} = -2\sum_{j=1}^{2} [y_j(k+1) - d_j] \frac{\partial f_j}{\partial u(k)} \frac{\partial u(k)}{\partial w_i} \tag{80}$$

where the last factor is available from the standard *back-propagation* algorithm. In above Eqn. (80), the most difficult problem encountered in all kinds of neural network control problem is to find the partial derivative **f** with respect to input u, that is, the *Jacobian* of the plant, or, in other words, the sensitivity of the plant. Psaltis et al regarded the plant as one of the hidden layers and the above error due to the current states and input is propagated through plant [13]. However, it has been known that a thorough knowledge of the plant is very difficult to achieve over the entire domain of interest. Therefore, so far our research is limited to determining the sensitivity (*Jacobian*) of the plant near around the operating point. Consequentially, this kind of controller has a very limited range of performance even though they are claimed as universal nonlinear controllers.

The complete system, including the nonlinear plant dynamics, the derived state estimator of Section 4.1.A and this neural net controller, was simulated using Turbo C, as shown in Fig. 77.

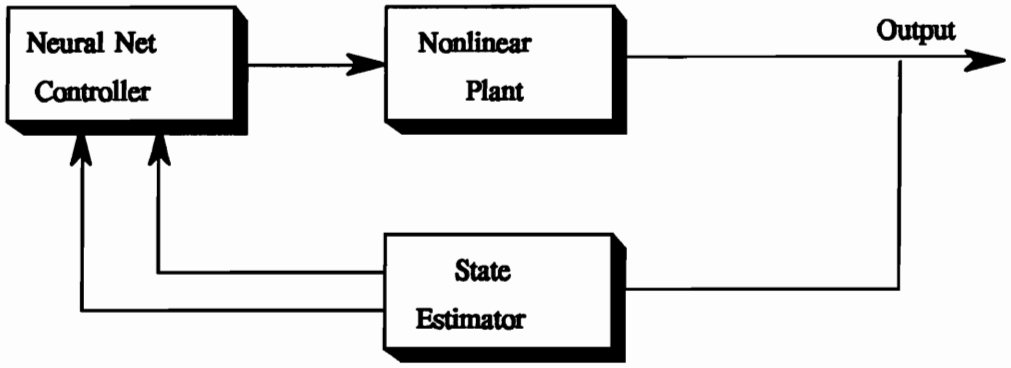


Fig. 77. Diagram for Example Simulation

To demonstrate the performance of the neuro-controller, Figs. 78, 79 and 80 show the position, the angle and the force measurement, respectively. The final stage of training involves minimizing a performance index (cost functional). This process is made possible by the original nonlinear identification which provides the necessary functional derivatives. This final stage is important in improving angle response and minimizing the peak force as can be seen in Figs. 79 and 80. In this manner the final training phase is completed.

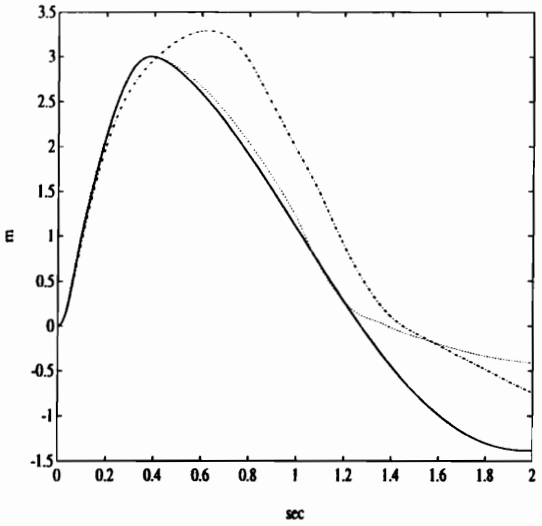


Fig. 78. Position:
Switched-Controller(-), Neuro-Controller without (:) and with a Performance Index Training(-.)

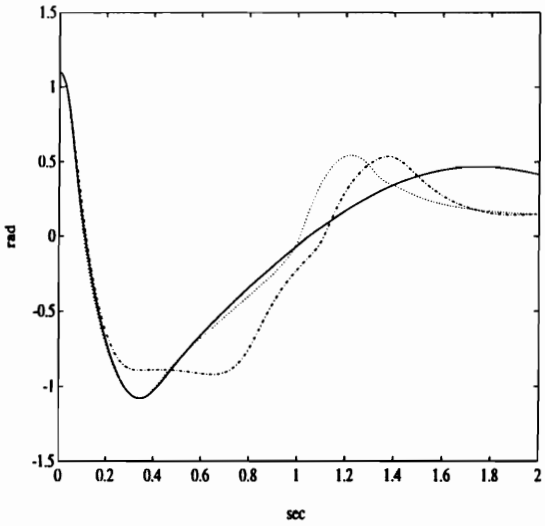


Fig. 79. Angle:
Switched-Controller (-), Neuro-Controller without (:) and with a Performance Index Training(-.)

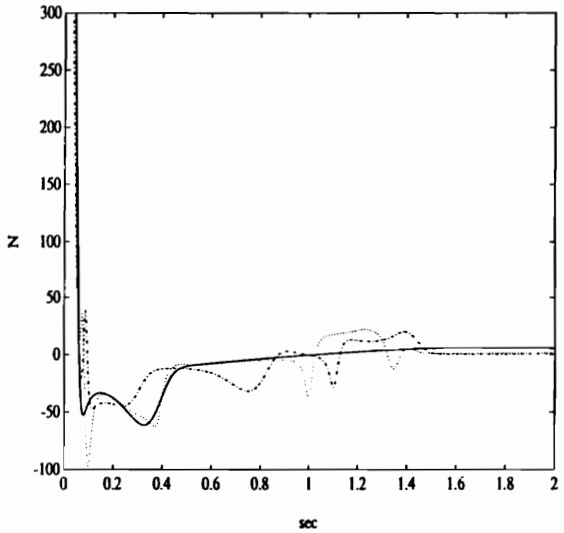


Fig. 80. Force:
Switched-Controller(-), Neuro-Controller without (:) and with a Performance Index Training(-.) ،

5. Conclusions

A three step procedure which consists of the identification of the system, the state estimation based on the identified model and the design of switched-linear controller has been presented for the design of a neuro-controller of complex and/or nonlinear systems with unknown dynamics. For several decades there have been many useful conventional techniques for identification and control of linear systems, but very few which are adequate for identifying and eventually controlling nonlinear systems. Artificial neural nets have the potential of being used to fill this void. The conditions which make the application of neural net technology attractive are: (1) a convenient source of data with which to train the neural net, and (2) either a complex process, or one for which no accurate model exists from conventional methods. However, in addition to the basic concepts of MLP representation, the practical aspects of achieving an accurate nonlinear model which generalizes well and is also of minimal complexity, must be considered. The technique presented here is constructive in nature and emphasizes the utility of "dividing" the problem into linear and nonlinear parts, using one of the well established conventional methods to set the basic structure of the multivariable system. The proposed robust hybrid identification is a unique combination of conventional and connectionist methods. In order to minimize the complexity of the MLP (especially in the MIMO case), the classical system identification technique was used to get necessary information, such as the system order and the best MIMO observability structure, of the quasi-linear system. A failing of the conventional identification technique is that it requires sequential time data, which is not feasible with unstable systems. The back-propagation learning algorithm, on

the other hand, does not require sequential time data, making it suitable for the continued identification process. Since the goal is to understand and control an (unstable) system, using only measured input/output data, the linear part can provide information for a local control law to provide stability around the operating point.

As an application of identifying an unknown plant in power electronics systems, the MLPs are proposed for modeling switching converters and resonant converters. There have been tremendous efforts to construct a general small-signal modeling technique in power electronics system. Especially, due to the evolving new switching converter topologies, more powerful and also convenient modeling approaches are needed. Several approaches have been employed in an attempt to generalize small-signal models for pulse-width modulation (PWM) converters and resonant converters. However, the results often tend to be complicated and difficult to apply. But this proposed modeling approach proved to be simpler and more convenient for small-signal modeling of power electronics systems than any other present modeling method. Boost converter, SRC and FMRC were taken as examples to demonstrate the algorithm with promising results.

Based on the identified small-signal model of the power converter, the MLP is further trained to capture both the nonlinearities and the dynamics of the system for an approximation of boundaries into unstable region in case of large transients. The region of stability was investigated and compared to the measurement data. The result was almost identical with very little error.

From this identified structure, the states of the linear model of the system were estimated to the actual trajectories of the state variables satisfactorily to be

fed back for changing the dynamics of the system. In contrast to a delayed-measurement observer or Luenberger reduced-order observer, without *a priori* knowledge of the given system the states were estimated to the actual trajectory of the state variables by the novel states estimation method. Since the trajectories of the internal states of the system are identified in software without causing unnecessary overshoot at control stage, it is a very beneficial by-product of the proposed identification technique.

The neuro-controller design is based on a switched-linear model which requires the development of a set of linearized models of the original plant along with the corresponding controller gains. This switched-linear controller is used to generate responses which, in turn, are used as training data for the initialization of the neural network controller. As in the training of any feedforward neural net, the selection of the data, e.g. as a trade-off between local and global performance, is critical. This selection is greatly facilitated by the interim "switched controller" since regions of poor training can be addressed specifically. The power of the MLP is then used to capture the nonlinearities of the system for the final purpose of controlling nonlinear characteristic of the system, thereby extending the control range of the system beyond that of the incremental (linear) description. The cart-pendulum system, often used as a benchmark for controller designs, was used to illustrate the identification and control technique in a stochastic environment. Simulations show excellent promise for this approach compared to other methods and, hence, may be suitable for both identifying and controlling critical industrial processes. A key issue is to have a method to train the necessary feedforward nets accurately, which involves the intelligent use of data pre-processing as well as training software that provides visual on-line feed back to the designer.

REFERENCES

[1] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks,* vol. 2, pp. 359-366, 1989.

[2] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics of Control, Signals and Systems,* vol. 2, pp. 303-314, 1989.

[3] S. Chen, S.A. Billings, and P.M. Grant, "Non-linear system identification using neural networks," *Int.J.Control,* vol. 51, no. 6, pp. 1191-1214, 1990.

[4] F.C. Fu and J.B. Farison, "On the Volterra-series functional identification of nonlinear discrete-time systems," *Int.J.Control,* vol. 18, no. 6, pp. 1281-1289, 1973.

[5] S.J. Farlow, Ed., *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, New York: Marcel Dekker, 1984.

[6] J. Park and I.W. Sandberg, "Universal approximation using radial-basis function networks," *Neural Computation,* vol. 3, pp. 246-257, 1991.

[7] M.F. Tenorio and W.T. Lee, "Self-organizing neural nets for the identification problem," *Advances in Neural Information Processing System* 1, (D.Touretzky Ed.), 1989.

[8] H.F. VanLandingham, S. Bingulac and M. Tran, "A comparison of conventional and neural network approaches to system identification," *Control-Theory and Advanced Technology*, vol. 3, no. 1, pp. 77-97, 1993.

[9] K.S. Narendra and K. Parthasarathy, "Neural networks in dynamical systems," *SPIE Intelligent Control and Adaptive Systems* , vol. 1196, 1989.

[10] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. Neural Networks,* vol. 1, pp.

4-27, March 1990.

[11] A.U. Levin and K.S. Narendra, "Control of nonlinear dynamical systems using neural networks: controllability and stabilization,"*IEEE Trans. Neural Networks,* vol. 4, no. 2, pp. 192-206, March 1993.

[12] V.C. Chen and Y.H. Pao, "Learning control with neural networks," *Proc. of IEEE Conf. on Robotics & Automation*, pp. 1448-1453, 1990.

[13] D. Psaltis, A. Sideris and A.A. Yamamura, "A multilayered neural network controller," *IEEE Control System Magazine*, pp. 17-21, April 1988.

[14] A. Guez and J. Selinsky, "A trainable neuromorphic controller," *Journal of Robotic Systems*, vol. 5, no. 4, pp. 363-388, August 1988.

[15] Lt.L.C. Baird III and W.L. Baker, "A connectionist system for nonlinear control," *American Institute of Aeronautics and Astronautics*, pp. 1113-1119, 1990.

[16] Y.L. Gu, "On nonlinear system invertibility and learning approaches by neural networks," *Proc. of American Control Conference*, vol. 3, pp. 3013-3017, 1990.

[17] X. cui and K.g. Shin, "Design of an industrial process controller using neural networks," *Proc. of American Control Conference*, vol. 1 , pp. 508-513, Boston, MA, 1991.

[18] T. Yamada and T. Yabuta, "Nonlinear neural network controller for dynamic system, "*1990 IEEE*, pp. 1244 -1249, 1990.

[19] J.Y.Choi, M.Tran, P.Huynh and H.F.VanLandingham, "Optimal control of nonlinear systems using neural networks," *Proceedings of the Southeastern Symposium on System Theory*, N.C.A.&T. University, Greensboro, NC, 1992.

[20] Y. Iiguni, H. Sakai and H. Tokumara, "A nonlinear regulator design in the

presence of system uncertainties using multilayered neural networks," *IEEE transactions on Neural networks*, vol. 2, no. 4, July 1991.

[21] Denis J.S.R. Bertrand, "Neural network controllers for the X29 aircraft," *IEEE/INNS Int. Conf. on Neural Networks*, Baltimore, 1992.

[22] A.U. Levin and K.S. Narendra, "Stabilization of non-linear dynamical systems using neural networks," *IEEE/INNS Int. Conf. on Neural Networks*, Baltimore, 1992.

[23] J.Y. Choi and H.F. VanLandingham, "Nonlinear system identification using neural networks," *Fifth IFSA World Congress*, vol. 1, pp. 58-61, Seoul, Korea, 1993.

[24] H.F. VanLandingham, J.Y. Choi and M. Tran, "System identification with multilayer perceptrons," *Proceedings of the IEEE Systems and Man and Cybernetics (SMC) Conference*, LeTouquet, France, Oct. 17-20, 1993.

[25] H.F. VanLandingham, J.Y. Choi and S. Bingulac, "A constructive approach for nonlinear system identification using multilayer perceptrons," (to be published) , 1994.

[26] L. R. Hunt and S. Verman, "Observers and controllers for feedback linearizable systems," *Proc. of American Control Conference*, vol. 1, pp. 546-547, Boston MA, June 1991.

[27] J. S. Dhingra, R. L. Moose, H. F. VanLandingham and T. A. Lauzon, "A computationally efficient technique for state estimation of nonlinear systems," *Automatica*, no. 28, vol. 2, pp. 395-399, 1992.

[28] K. C. Cheok and R. R. Beck, "Microprocessor - based state estimators and optimal controllers," *Proc. 23rd Midwest Symp. Circuits and Systems*, Univ. of Toledo, Toledo, OH, pp. 318-324, Aug. 1980.

[29] N. K. Loh, W. Z. Chen and R. R. Beck, "Delayed-measurement observer for

discrete-time linear system," Technical Report TR-80-07-100, School of Engineering, Oakland University, Rochester, MI, July 1980.

[30] A. Barto, R. Sutton and C. Anderson, "Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-13, pp. 834-846, Sep. 1983.

[31] C. Anderson, " Learning to Control an Inverted Pendulum Using Neural Networks", *IEEE Control System Magazine*, pp. 31-37, April 1989.

[32] T. Troudet and W. Merrill,"Neuromorphic learning of continuous-valued mappings from noise-corrupted data,"*IEEE transactions on Neural networks*, vol. 2, no. 2, pp. 294-301, Mar. 1991.

[33] S. Bingulac and H.F. VanLandingham, *Algorithms for Computer-Aided Design of Multivariable Control Systems*, Marcel Dekker, Inc., New York NY, 1993.

[34] J.L. McClelland and D.E. Rumelhart, *Parallel Distributed Processing*. **1**, MIT Press, Cambridge, MA, 1986.

[35] W.S. McCulloch and W.H. Pitts, "A Logical Calculus of the Ideas Imminent in Nervous Activity," *Bulletin of Mathematical Biophysics*, **5**, 115-133, 1943.

[36] A.S. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modeling," Technical Report, Los Alamos National Laboratories, Los Alamos NM, 1987.

[37] J. Villiers and E. Barnard,"Backpropagation neural nets with one and two hidden layers,"*IEEE transactions on Neural networks*, vol. 4, no. 1, pp. 136-141, Jan. 1991.

[38] H.F. VanLandingham, R.L. Moose, A. Tsoukkas, M. Tran and J.Y. Choi, "Neural network controller design for nonlinear stochastic systems", (to be

published), 1993.

[39] D.G. Luenberger,"Canonical forms for linear multivariable systems,"*IEEE Trans. on Automatic Control*, AC-12, pp. 290- 293, 1967.

[40] C.-T. Chen, *Linear System Theory and Design*, Holt, Rinehart and Winston, Inc., New York, NY, 1984.

[41] T. Kailath, *Linear Systems*, Prentice-Hall Pub. Co., Englewood Cliffs, NJ, 1980.

[42] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[43] S. Bingulac and R. V. Krtolica, "On Admissibility of Pseudo-Observability Indices," *IEEE Trans. on Automatic Control*, **AC-32**, 920-922,1987.

[44] B. Widrow and S.D. Sterns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1985.

[45] G.W. Wester and R.D. Middlebrook, "Low-frequency characterization of switched dc-to-dc converters," *IEEE Power Processing and Electronics Specialist Conf. Record*, pp. 9-20, 1972.

[46] R.D. Middlebrook and S. Cuk, "A general unified approach to modeling switching converter power stages, " *IEEE Power Electronics Specialist Conf. Record*, pp. 18-34, 1976.

[47] F.C. Lee, R.P. Iwens, Y. Yu and J.E. Triner, "Generalized computer-aided discrete-time modeling and analysis of dc-dc converters," *IEEE Trans. Industr. Electron. Contr. Instrum.*, vol. IECI-26, no. 5, pp. 58-69, 1979.

[48] G.C. Verghese, M.E. Flbuluk and J.G. Kassaksian, "A general approach to sample- data modeling for power electronic circuits," *IEEE Trans. Power Electronics*, vol. 1, no. 2, 1986.

[49] J.O. Groves, "Small-signal analysis using harmonic balance methods," *IEEE*

Power Electronics Specialist Conf. Record, pp. 74-79, 1991.

[50] E.X. Yang, F.C. Lee and M.M. Jovanoic, "Small-signal modeling of power electronic circuits using extended describing function technique," *Proc. Virginia Power Electronics Center (VPEC) Conf.*, vol. 9, pp. 167-178, 1991.

[51] H.F. VanLandingham, M. Tran and J.Y. Choi, "Multivariable process identification," *Proceedings of the IEEE Systems and Man and Cybernetics (SMC) Conference*, LeTouquet, France, Oct. 17-20, 1993.

[52] R.W. Ericson, S. Cuk and R.D. Middlebrook, "Large-Signal Modeling and Analysis of Switching Regulators", *IEEE Power Electronics*, pp. 24-248, 1982.

[53] V. Vorperian, "Simplified analysis of PWM converters using the model of the PWM switch: Part I and II," *IEEE Trans. Aerospace and Electronics Systems*, vol. 26, no. 3, pp. 490-505, 1990.

[54] V. Vorperian and S. Cuk, "Small-signal analysis of resonant converters," *IEEE Power Electronics Specialist Conf. Record*, pp. 269-282, 1983.

[55] E.X. Yang, F.C. Lee and M.M. Jovanoic, "Small-signal modeling of series and parallel resonant converters," *Proc. IEEE Power Electronics Conf.*, pp.941-948,1992.

[56] J. Sietsma and R.J.F. Dow,"Neural Net Pruning-Why and How," *Proceedings of the IEEE Int. Conf. on Neural Networks*, vol.1, pp 325-334, 1988.

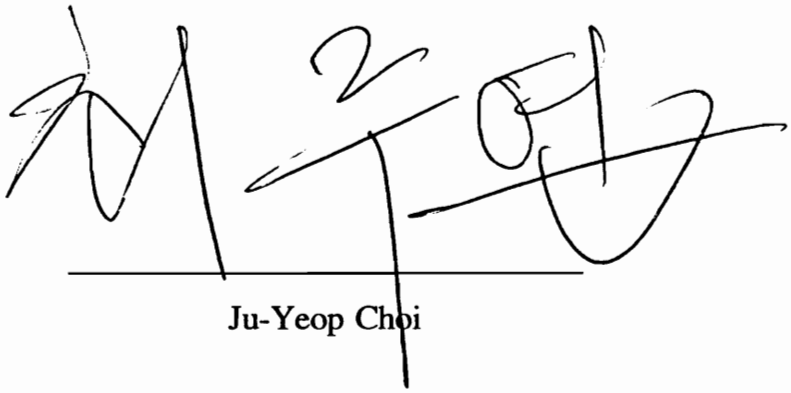
[57] S.E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture, "*Advances in Neural Information Processing System 2*, (D. Touretzky, Ed.), pp. 524-532, 1990.

[58] H. VanLandingham, *Introduction to Digital Control Systems*, Macmillan Publishing Company, New York, 1985.

[59] P. Huynh and M. Tran, "A survey of inverted pendulum system using linear optimal control and neural network control", VPI&SU Elec. Eng. Tech. Report, Dec. 1991.

Vita

Ju-Yeop Choi was born in Yeo-Soo, Korea in Feb. 11, 1961. He received a Bachelor's degree from Seoul National University, Seoul , Korea in 1983, a Master's degree from University of Texas at Arlington, Texas, U.S.A. in 1990, both in Electrical Engineering. His research interests are in the areas of design and implementation of neural networks for pattern recognition, MIMO controllers, nonlinear controllers and also fuzzy logic controllers for nonlinear system.



Ju-Yeop Choi