Improved Gray-Box Modeling of Electric Drives using Neural Networks

By
Yamilka Isabel Báez Rivera

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2003

Approved by:

_____	_____
Shawn D. Hunt, PhD. Member, Graduate Committee	Date
_____	_____
Krishnaswami Venkatesan, PhD. Member, Graduate Committee	Date
_____	_____
Miguel Vélez Reyes, PhD. Chairperson, Graduate Committee	Date
_____	_____
José Luis Cruz Rivera, PhD. Chairperson of the Department	Date
_____	_____
Lionel R. Orama Exclusa, PhD Representative of Graduate Studies	Date

Abstract

Electric drives are used in many industrial and commercial applications. High performance control of electric drives requires the accurate modeling of the motor and mechanical load. In many industrial applications, it is desirable that the electric drive has the capability of self-tuning controller parameters to be able to drive different mechanical loads. One way to achieve this flexibility is by direct identification of the drive and mechanical load. Modeling and identification of Electric drive coupled to a load can be a challenging task. This research investigates the use of gray box models to identify electric drive systems connected to an unknown load.

In the proposed model, the electrical subsystem of the machine is modeled using physical principles while the mechanical subsystem is modeled using a black box model based on neural networks. A two-stage identification approach that separates electrical subsystem parameter estimation from mechanical subsystem identification is presented. At each stage the parameters are estimated using the linear least squares approach. Simulation results are presented to demonstrate the feasibility of the approach.

Resumen

Los sistemas de accionamiento eléctrico se utilizan en varias aplicaciones industriales y comerciales. El control de alto rendimiento de estos sistemas requiere un alto grado de precisión en el modelaje del motor y la carga mecánica. En muchas aplicaciones industriales, es deseable que el sistema de accionamiento contenga la capacidad the ajustar automaticamente los parametros del controlador, para que sea capáz de manejar diferented cargas mecánicas. Una manera de obtener esta flexibilidad es por identificación directa del sistema de accionamiento eléctrico y de la carga mecánica. El modelaje e identificación de sistemas de accionamiento acomplados a cargas mecánicas puede ser un gran reto. Este trabajo investiga el uso de modelos de caja gris para identificar el sistema de accionamiento eléctrico conectado a una carga desconocida.

En el modelo propuesto, el subsistema eléctrico the la máquina es modelado usando principios físicos mientras el subsistema mecánico es modelado usando modelos de caja negra basados en redes neurales. La técnica presentada en este trabajo es la técnica de identificación the dos etapas que separa la estimación de parámetros eléctricos de la identificación de parámetros mecánicos. En cada etapa los parámetros son estimados usando la técnica de los cuadrados mínimos. Resultados de simulación son presentados para demostrar la viabilidad de esta técnica.

Acknowledgements

I would like to thank Professor Miguel Vélez Reyes for being the best professor and advisor. Thanks Professor for your time, your patience and the opportunity for being one of your graduate students.

I want to thank Professor Shawn Hunt and Professor Krishnaswami Venkatesan for serving as member of my graduate committee. I want to thank the faculty and staff of the Electrical and Computer Engineering Department who's in one form or another helps me, especially to Ms. Sandra Montalvo, for being my counselor and friend.

I want to thank the Center of Power Electronics Systems (CPES) for the opportunity the center bring to my student and personal life, specially I want to thank Ms. Claribel Lorenzo for all your help, for being the right hand of the center.

I would like to thank my graduate fellows, my friends Jose Miguel Ortiz, Carlos Andres Gonzalez, Sandra Liliana Ordónez and specially my best friend Leila S. Rodriguez. Thanks for the good times.

I want to thank my mom, my dad, my sister and my little brother for being my support, for made me be the being I am. I am always being great fully for that.

I want to thank my husband Bienve, for be with me all the time, for give me your support, for being there when I need you most, for giving me your love and always give me the hope to live and survive every day. I love you.

Last but certainly not least, I would like to thank God for give me the opportunity to live every day.

This work was supported primarily the ERC Program of the National Science Foundation under Award Number EEC-9731677. additional support came from NSF grant ECS-9702860.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Electric drives are used in many industrial and commercial applications. High performance control of electric drives requires the accurate modeling of the motor and mechanical load. In many industrial applications, it is desirable that the electric drive has the capability of self-tuning controller parameters to be able to drive different mechanical loads. One way to achieve this flexibility is by direct identification of the drive and mechanical load. Modeling and identification of Electric drive coupled to a load can be a challenging task. This research investigates the use of gray box models to identify electric drive systems connected to an unknown load.

In gray box modeling, the model is structured in two parts, a physical model (or white box) component that models the known part of the system and a black box component for the unknown part. In our system, we know the model of the electric subsystem of the drive, but we don't know the mechanical load model. The use of gray-box for modeling the electric drive system is a good alternative since it takes advantage of available system knowledge while leaving enough flexibility to deal with the unknown mechanical load.

The use of neural-network-based gray-box model for dc motor drives was investigated in [1]. However, no structure optimization was performed resulting in models that had a significantly large number of parameters when compared to physical models and used a nonlinear least squares approach for parameter estimation. The large number of parameters turned the parameter estimation problem into an Ill-conditioned problem where network parameters are very sensitive to noise in the data and training algorithms converged quite slowly. In [1], the network structure is fixed and deals with ill-conditioned using the Levenberg-Marquardt training algorithm to regularize the network training. This work explored a different identification approach where used a radial basis function neural network for the black box resulting in a two-stage linear least squares method for the parameter estimation. To optimize the network structure, orthogonal least squares pruning techniques are used. With this approach, a lower dimension model with better-conditioned parameters improving their interpolation and extrapolation capabilities was obtained.

1.1 Research Motivation

The motivation of this work is directly related to the Center of Power Electronics Systems. The industrial process has several components, as shown in Figure 1. An important objective in the center is integrated drives for process control, improving reliability and reduce the system complexity, size and cost using several steps. These

steps are the integrated design approach, sensorless control and adding system intelligence by self-commissioning, tuning and diagnostics.

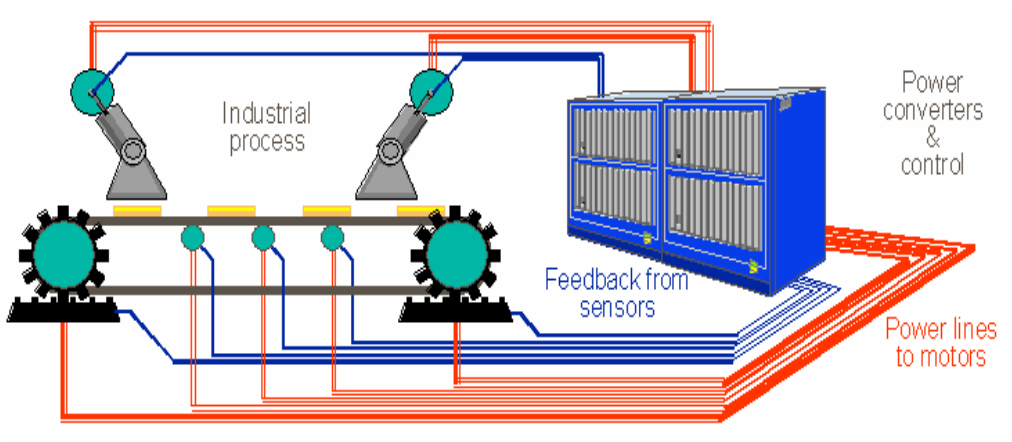

Figure 1 Basic Industrial Process

1.2 Objectives

The main objective of this work was to develop an automated methodology for the identification of models that can be used to tune the controller of electric drive systems for a wide range of mechanical loads.

The specific objectives of this work were:

- To develop an identification algorithm that optimized the model structure as well as the estimation of the parameters for a motor drive.

- To explore the use of radial basis function neural networks to model the unknown mechanical load instead of the multilayer perceptron used in [1].
- Study capabilities of gray box modeling techniques for identification of electric drives.
- To validate the approach.

1.3 Contribution

One of the contributions of this work is the development of a two-stage linear least squares parameter estimation process. This method avoids the difficulties associated with the nonlinear least squares method to the parameter estimation. The second contribution was the development and generalization of box models for mechanical load. With this work we developed methodology for the identification of models that can be used to make self-commissioning of electric drive systems driving mechanical loads. The last contribution of this work was implemented pruning method by orthogonal linear least square to reduce model complexity during the identification process.

1.4 Outline

This thesis is organized as follows. In Chapter 2, the literature review is presented. The Basic Concepts in Electric Drive Systems, System Identification, Gray

Box Models, Artificial Neural Networks, Parameter Estimation, Model Complexity and Pruning are also discussed. Chapter 3 presents the Two - stage parameter estimation method. Simulation results for identification and validation are presented in Chapter 4. Chapter 5 presents conclusions and recommendations for future work.

Chapter 2

BACKGROUND

This Chapter presents basic concepts in electric drives and drive commissioning, the use of the gray box model for the system identification, and the use of neural networks for the black box modeling. Parameter estimation methods, model complexity and pruning techniques are also discussed here.

2.1 Basic Concepts in Electric Drives

About fifty percent of all electricity is used in electric drives. An electric drive can be defined as a system that converts electrical energy to mechanical energy (in motoring) or vice versa (in regenerative braking) for running various process such as pumps, air compressors, disc drives, robots, etc. Electrical drives are an integral part of many industrial applications. A combination of a prime mover, transmission equipment and mechanical working load is what we call electric drive system. More specifically, the electric drive system is composed of an electric machine, sensors, controller, a power electronics power-processing unit and the mechanical load. A block diagram of a drive system is presented in Figure 2.

In an electric drive system, the power processing unit delivers appropriate form of voltage and frequency to the motor, the motor is used for speed or position control applications, and the controller with the help of the sensors, controls the motor and power

converter to meet the load requirements. The objective of the electric drive systems is to make an efficient electric to the mechanic power conversion.

A reason to use electric drive system is the advantages that they offer. Some of these advantages are the availability of drives over a wide range of power, the lower noise level, the high efficiency, there are available in a variety of design rating to meet different types of loads, drives operate in first, second or fourth quadrants, which offer a variety of applications, reliability and versatility.

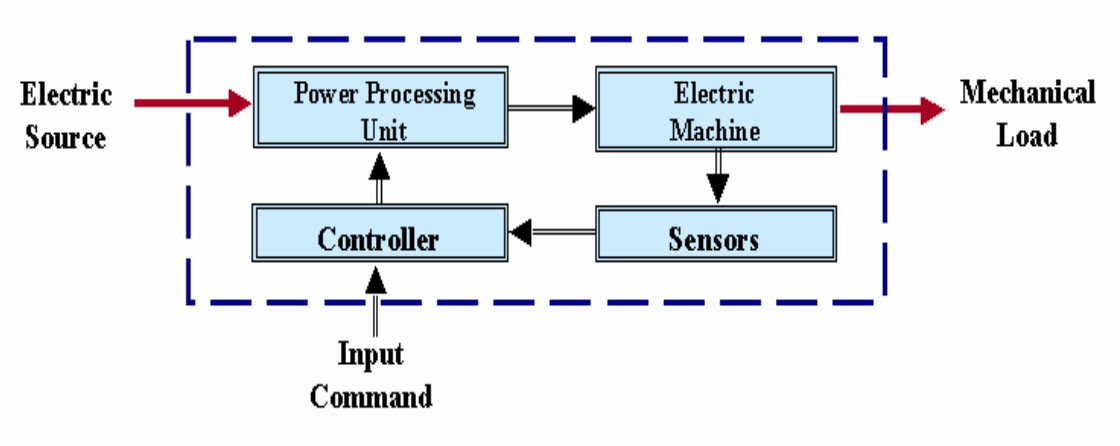

Figure 2 Basic Electric Drive System Schematic

2.1.1 DC Motor Drives

DC motor drives are very versatile for purposes of speed control. Fast response and smooth speed control is possible by varying the armature voltage and or field current. The converter technology is well established and the power converter is simple and inexpensive. A special feature of dc motor drives is that it is possible to connect the field and armature windings in several ways so as to achieve a variety of torque-speed

characteristics. By properly adjusting the relative field of the dc motor, a speed-torque curve with desired speed regulation may be obtained. The speed-torque characteristic of an electric motor drive is very significant since it sets the application of the motor [3].

The principal reference for this work is "Gray-Box Modeling of Electric Machines Using Neural Networks" [1]. Like in [1], the capabilities of the gray-box modeling approach for electric drive modeling are studied. For this purpose, an identification of a simulated drive system was performed. The simulated system was a permanent magnet DC motor driving a nonlinear load. For simulation purposes, we used a permanent magnet DC Motor with the following characteristics: 1hp, 220 volts and 550 rpm. Figure 3 presents the schematic of the system.

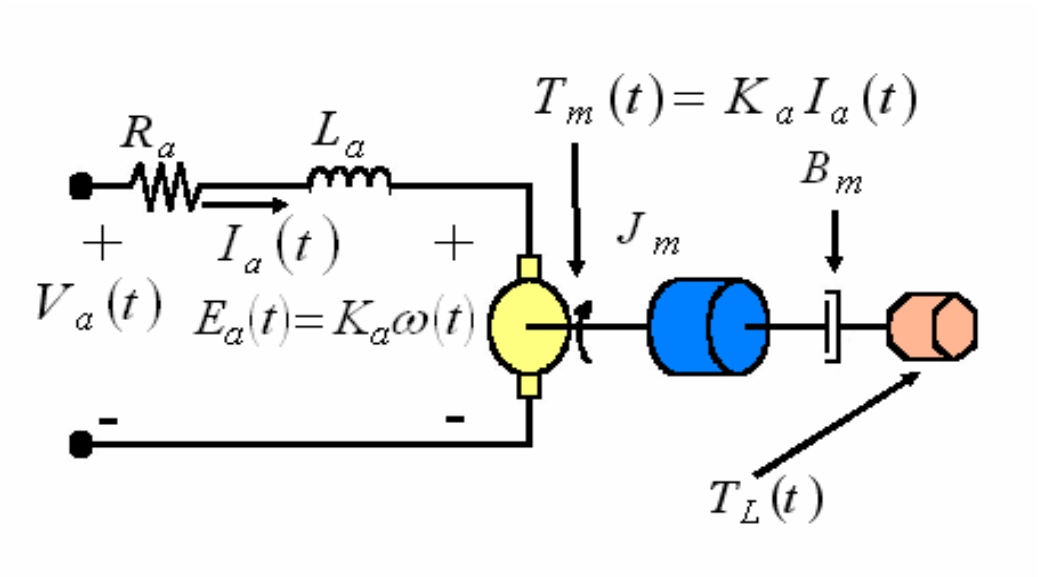

Figure 3 Schematic for DC Motor and Load System

The general equations that model the electrical and mechanical parts of the system are the following:

$$L_a \frac{di_a(t)}{dt} = V_a(t) - R_a i_a(t) - K_a w(t) \quad \textbf{(2. 1)}$$

$$J_m \frac{dw(t)}{dt} = t_{em}(t) - B_m w(t) - t_L(w(t)) \quad \textbf{(2. 2)}$$

$$t_{em}(t) = K_a i_a(t) \quad \textbf{(2. 3)}$$

Where: L_a is the armature inductance, R_a is armature resistance, $V_a(t)$ is the input voltage, K_a is the armature constant, J_m is the combined load and motor inertia of the motor, B_m is the damping coefficient, $w(t)$ is the motor speed, $i_a(t)$ is the armature current and t_L is the load. The numerical values for the parameters used in the simulations are shown in Table 1.

Table 1 Parameters of the DC Motor Drive.

Parameter	Value
J_m	0.06 Kg.m^2
K_a	3.475 Nm.A^{-1}
R_a	7.56 Ω
L_a	0.055 H
B_m	0.03475

Here we consider two mechanical loads. These loads are simples and can be found in multiple mechanical applications as in pumps, variable speed drives, etc. The load parameters in both cases are such that the load torque equals rated torque at rated

speed. The inertia of the system always results in the same value. The load equations and their parameter values are presented in Table 2.

Table 2 Torque Loads and their parameter values.

Case	Torque Equation	Parameter Values
Nonlinear Friction	$t_l(t) = sign(w(t))\left(m\arctan\left(\frac{a}{abs(w(t))} \right) + b \right)$	μ=1N.m α=1Rad/s β=3N.m
Fan Load	$t_l(t) = m sign(w(t)) w^2(t)$	μ=1N.m

2.1.2 Mechanical Modeling for Electric Drives

For modeling mechanical systems for electric Drives, some prior knowledge has been used. Beineke [4] present a method for commissioning the speed and position control system of an electric drive. In this case, the system includes identification of the mechanical load. This is an easy example to understand the concept. Beineke identify the mechanical system assuming it behaves like a one mass system; then use extracted characteristics features from acquired data to determine the validity of the system when we compare with a two mass model for the system. The features used in this case for structure selection are based on knowledge of the ideal response of the system when it is assume to be a one mass or two masses.

In [5], Beineke develops models for nonlinear loads on electric drives system. In this case, the authors integrated radial basis function networks in a physics based model for the load. The physical model estimates position and speed, measured current and the

radial basis function estimates friction. This work is a good reference because the authors implement the work in an actual system, which produces a good performance in the parameter estimation task. The deal in this work is the modeling of the mechanical part. The representation of the basis model is:

$$\dot{x} = Ax + Bu + NL(x,u) \quad \textbf{(2. 4)}$$

Where $NL(x,u)$ is the nonlinear representation of the equation (2.4).

Only some works have made the use of gray-box modeling approach for electromechanical systems. For example, in [6] a neural network is used in gray-box model of a rotating arm unknown friction torque component driven by an induction motor described by angular displacement, velocity, effective inertia, friction, motor gain, input voltage of the motor and the parameters associated to the friction characteristic function. In this work, the authors used white-box for the physics of the model and used black box for the unknown friction characteristic. The unknown part is modeled by an artificial neural network, which corresponds to the mechanical part of the system.

2.2 Commissioning of Electric Drives

As result of the fast development in automation technology, the demand for drives has been increasing. It is important for the drives to be able to overcome the influence of load variations and keep the performance of the overall system unchanged. An important problem in drive systems is controller tuning prior to system operation. Drives controller tuning is needed to ensure that the drive system will meet the system performance requirements. Drive commissioning is the tuning of system parameters before it is put to

operation. During this process, different test are applied to the drive system to calibrate the drive controller [7]. The steps during drive commissioning are:

- Initial setting of necessary control parameters
- Identification of electrical and mechanical parameters
- Selection of controller
- Tuning of control parameters

The commissioning process is usually performed by a trained technician or field engineer, which involved costly and time consuming. Self-commissioning has been proposed as a solution to the problem.

Self-commissioning is the automation of the commissioning process. Some of the benefits with self-commissioning is that it facilitates system installation and assures proper drive tuning before the system is fully operational and achieve significant improvements in reliability of drive systems by increasing the intelligence of their control systems. Some of the issues is self-commissioning are the load identification and the controller tuning. The load identification can be performed by direct or indirect form. The direct form estimates the parameters directly for the load and the indirect form set the parameters in the controller. For this process to be applicable to a wide range of mechanical loads the model structure needs to be quite flexible while still incorporating the physical knowledge about the electric motor. The solution presented here is based on gray box models. The proposed load identification methodology is applied to the identification of a permanent magnet DC motor drive system driving an unknown static load.

2.3 System Identification

A model is a very useful and compact way to summarize the knowledge about a process. Mathematical model building can be based on physical laws that govern the system. In most cases, it is not possible to make a complete model of the system only from physical knowledge. Some parameters must be determined from experimental data. This approach is called system identification, in other words, system identification is the experimental approach to process modeling. A complete example of this system identification process is presented in [8, 9]. System identification process includes the following steps:

- Experimental planning
- Selection of model structures
- Criteria
- Parameter estimation
- Model validation

For the selection of model structure generally, load model consist of static and dynamic load model. When formulating and identification problem, a criterion is postulated to give a measure of how well a model fits the experimental data. For example, one criterion recommended by Astrom and Wittenmark [9] was the principle of least squares, linear least square is explained in more details is section 2.5.1. The parameter estimation problem can be formulated as and optimization problem, where the best model is one that best fits the data according to the given criterion. The model

validation is useful to determine such factors as step responses, impulse response and prediction errors.

The key problem in system identification is finding a suitable model structure in which a good model is to be found. In this work with the electric machines, the electric subsystem is fully understood, however, the model of the mechanical load could be uncertainly and we don't know the equations that describe the system load exactly. To solve this problem, we are going to model the unknown load using black box models.

The modeling approaches for dynamic systems are physical-based modeling, empirical models and the combination of both [10,11].

The physical model known as white box structure is based on the physical laws governing the phenomena. The system is perfectly known and the entire system can be constructed from physical approach. This model required a good knowledge of the system structure and parameters. However, this structure is difficult to derive for many physical systems. Good extrapolation performance can be found using this structure.

The empirical models known as black box models are based on available measurements of the system output observations, in this case there is no need for good insight of the system. One advantage of this structure is the flexible model structure. The drawbacks of the black box structure are the lack of physical meaning requires intensive study of the model quality and the poor extrapolation performance.

The gray box model represents the tradeoff between the white box and the black box models integrating some aspects from both. The structure of gray box relies strongly on prior knowledge (white box) and the model parameters are mainly determined by

measurement data (black box). It is the case when some physical knowledge is available but several components remain to be determined from observed data. The structure of gray box combines the flexibility of black box structure with the physical insight of white box structures. With the gray box model we have partial knowledge of the model structure.

The gray box model used here is presented in equation (2.5). This equation has two parts: the known part, which is the electrical equation of the system $(h(x(t)))$ and the unknown part, which is the mechanical load $(g(x(t)))$. The black box model represents the unknown part, which is based on artificial neural networks.

$$\dot{\mathbf{x}} = \mathbf{h}(x(t), \mathbf{u}(t)) + \mathbf{g}(x(t), \mathbf{u}(t)) \qquad \textbf{(2. 5)}$$

Where $\mathbf{h}(x(t), \mathbf{u}(t))$ is white box and $\mathbf{g}(x(t), \mathbf{u}(t))$ is black box

2.4 Artificial Neural Networks

An artificial neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure. The artificial neural network was motivated originally by the biological structures in the brains of humans and animals. These structures are extremely powerful for such task as information processing, learning, and adaptation. The most important characteristics of the neural networks are [10]: large number of simple units, the highly parallel units, strongly connected units, robustness against the failure of single units, and learning from data.

Artificial Neural Networks have proven to be valuable tool in exploiting ideas common to nonlinear dynamics and system identification [12,13]. It is the power of neural networks to represent non-linear mappings and hence to model nonlinear systems which is the feature of to be most readily exploited in the area of nonlinear controllers [14]. In addition the potential benefit of the artificial neural networks in the systems is to seek solutions to their complicated problems which is a great advantageous because it will adjust its parameters to reproduce the system in an input-output sense and an important advantageous in our case is because they do not require physical descriptions. Neural Networks are potential model structures candidates for electric drive system identification as shown in [1]. In some cases the appropriate neural network model set could reach a very large sized, therefore the size of the parameter set would become even larger. This is a problem intrinsically related to the approximation problem using neural networks.

The next Figure 4 presents the block diagram for the actual electric drive system. We have in the diagram all the constant and parameters that compound the equations for the system. In this case we use in the toque $t_L(v(t))$, which can be substitute for the equation of the mechanical load under study, fan load. But one of the objectives of this work was the use of neural networks for the load parameter estimation. In this case we used instead of the toque equations the neural network algorithm. This substitution is presented in Figure 5, in it can observe two biggest divisions, the first division is the physical known part of the system, which was compound of the electrical parameters and

the second division of the block diagram represent the black box part of the system included the neural network.

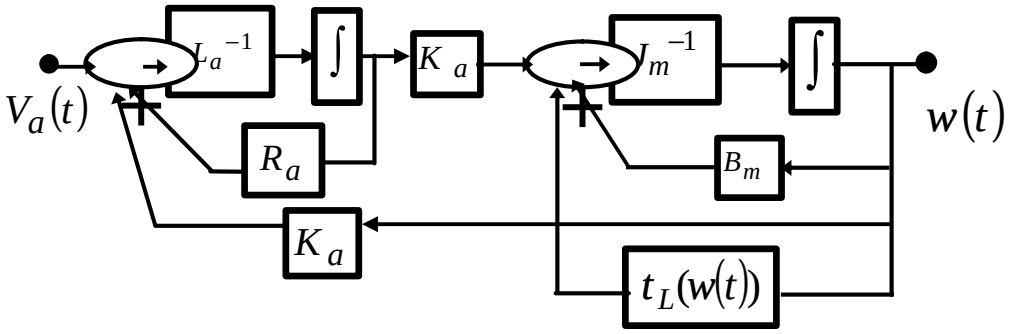

Figure 4 Actual Drive System Block Diagram

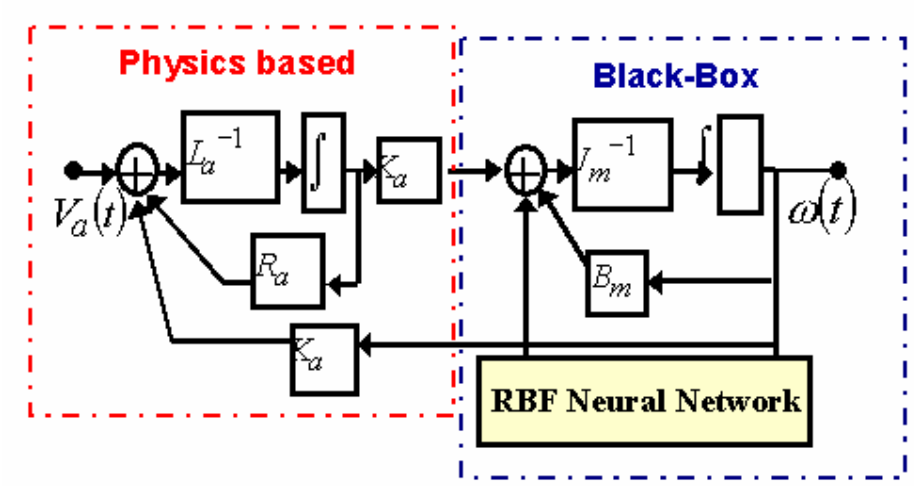

Figure 5 Gray – Box Model Block Diagram

Another possible black box structure of the system is derived from the real physical model. It is based on the fact that some models can be not identifiable. There are multiple combinations of neural network parameters and viscous friction constant

resulting in the same load characteristics. For this reason, another model structure is form by the elimination of the viscous parameter, as shown in Figure 6.

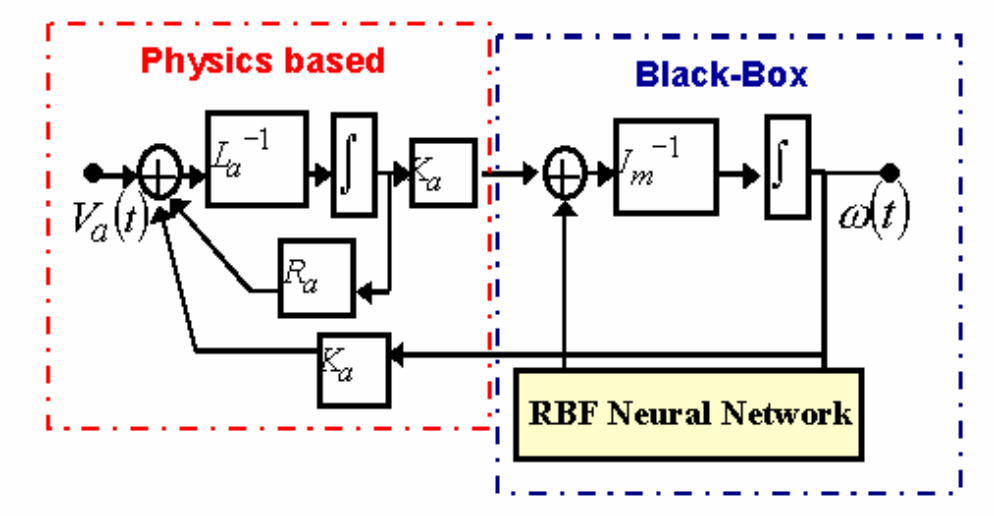

Figure 6 Gray – Box Model Block Diagram without viscous friction constant

2.4.1 Multilayer Perceptron (MLP)

The multiplayer perceptron is a feedforward neural network with hidden layers. MLP is one of the most widely used neural network architectures for function approximation. One of the attractive feature of the being fitted data is flexible. In MLP, the approximation function is determined by fitting the available data, which involves constructing an approximation function *f(x)* capable of mapping a collection of input vectors to a set of associated output vectors. When we work in absence of noise, we hope to achieve an exact fit. Example of this neural network is the sigmoid functions.

2.4.2 Sigmoid Functions

The sigmoid function was the activation function used in [1]. The sigmoid function has some important characteristics, is bounded, it is monotonically increasing

and it is continuous and smooth without gasps and corners. One example of the sigmoid type function is the logistic function given by:

$$g(u) = \frac{1}{1 + \frac{1}{e^{u}}} \qquad \textbf{(2. 6)}$$

This logistic function was the type of neural network activation function used by [1]. In this work, a single layer radial basis function was used. The objective to using a different artificial neural network was to simplify the training and pruning of the network.

2.4.3 Radial Basis Function

The radial basis function neural network was used to model the unknown static mechanical load of the drive system. The radial basis function artificial neural network utilizes the radial construction mechanism presented in Figure 7. The output of the radial basis function is given by:

$$\hat{y} = \sum_{i=0}^{M} w_i \mathbf{F}_i \left(\left\| \underline{u} - \underline{c}_i \right\| \right) \qquad \textbf{(2. 7)}$$

In Figure 6, the hidden layer nodes use the Gaussian Function as the basis function. The mathematical representation of the radial basis function neural network is given by:

$$y = f(x) = \sum_{i=1}^{N} e^{\left(-\frac{1}{2} \frac{(x(t) - c_i)^2}{s_i^2} \right)} \qquad \textbf{(2. 8)}$$

Where x is the input data, C_i - is the center of the ith node and s_i - is the "variance" of the ith node.

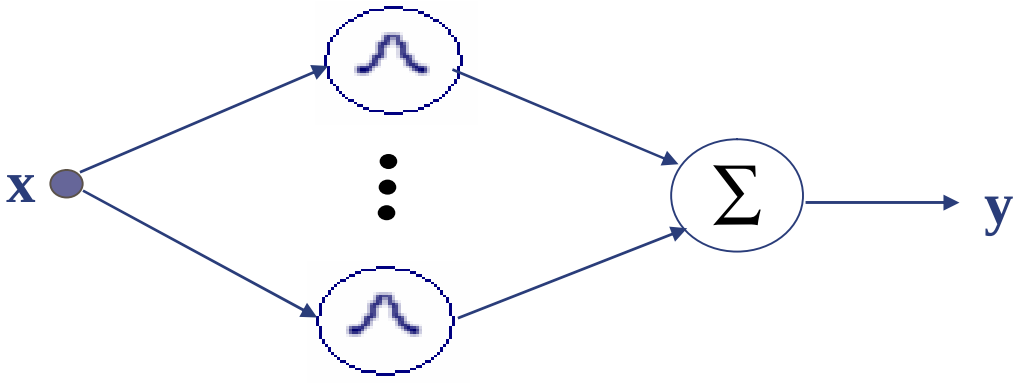

Figure 7 Basic Structure of a Radial Basis Function Neural Network

A radial basis function has three types of parameters

- Output layer weights = are linear parameters and they determine the amplitude of the basis function.
- Centers = are nonlinear parameters of the hidden layer neurons. They determine the position of the RBF.
- Standard deviation = are nonlinear parameters of the hidden layer neurons. They determine the width of the RBF.

Radial basis functions are attracting a great deal of interest due to their rapid training, faster learning, generality and simplicity [15]. RBF train rapidly without local minima problems and approximate any continuous function with arbitrary accuracy. Radial Basis function neural networks are powerful techniques with a definite range of applicability; they greatly accelerate the development and evaluation process. Their rapid training makes them suitable for situations where on-line learning is necessary. The linearity in the radial basis function parameters implies that their values can be computed

using standard least-squares techniques, which is faster than the gradients methods used to solve sigmoid values.

Some drawbacks of the radial basis functions are: the priori information needed to locate the centers and set the variances, and the principal drawback is that the number of basis functions increases exponentially with the dimension of the input space [16].

2.5 Parameter Estimation

Parameter estimation is of primary importance in many areas of process modeling, both for on-line applications such as real time optimization and for off-line applications. The objective is to determine estimates of model parameters that provide the best fit to measured data, generally based on some type of least squares criterion. In general case, this is a nonlinear optimization problem. Because of the high dimensionality of the parameter vector in neural network, th eparameter estimation is an ill-conditioned problem where network parameters are very sensitive to noise in the data and training algorithms converge quite slowly to the estimate. In [1], they fixed the network structure and dealt with Ill-conditioning problem using the Levenberg-Marquardt training algorithm to regularize the estimates. By using a radial basis function, we deal with the ill – conditioning by eliminating network nodes using pruning. So we can optimize the network structure in addition to compute optimal parameters estimates.

2.5.1 Linear Least Square Problem

Least square is a method based on the minimization of the sum of the squares of the error. The goal of linear least square method is to find the model output $\hat{y}$ that the best approximates the process output y, with the minimal sum of squared error value. In vector/matrix notation the model output can be written as

$$\hat{\mathbf{?}} = \arg\min_q \left\| \mathbf{y} - \hat{\mathbf{y}} \right\|_2^2 \quad \textbf{(2. 9)}$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{?}} \quad \textbf{(2. 10)}$$

Where the optimal solution to this problem is given by

$$\hat{\mathbf{?}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \quad \textbf{(2. 11)}$$

The difference $\mathbf{y}$ and $\hat{\mathbf{y}}$ is called the residual. Ideally the residual should be zero, but, in practice, the examination of the residuals can reveal many details about the estimation quality.

Based on [17] Linear Least Squares has several attractive features for system identification:

- Large errors are heavily penalized
- Linear least square estimates can be obtained by straight forward matrix algebra
- Linear least square criterion is related to statistical variance and the properties of the solution can be analyzed according to statistical criteria

2.5.2 Nonlinear Least Square Problem

Physical parameter estimates, network weights and bias are computed by minimizing a cost function. The nonlinear least squares parameter estimation problem can be formulated as [1]:

$$\hat{\mathbf{?}} = \arg\min_q S(\mathbf{?}) \quad \textbf{(2. 12)}$$

Where

$$S(\mathbf{?}) = \sum_{t=1}^{N} [\, f_i(q) - y_i \,]^2 = \left\| \mathbf{f}(q) - \mathbf{y} \right\|^2 \quad \textbf{(2. 13)}$$

The two most popular algorithms for the nonlinear parameter estimation problem are the Gauss-Newton and Levenberg-Marquardt method, described in more detail next.

2.5.3 Gauss-Newton

The Gauss-Newton is an iterative algorithm.

$$\hat{q}^{(R)} = \hat{q}^{(R-1)} + h_{R-1}\, p_R \quad \textbf{(2. 14)}$$

$$p_R = \arg\min \left\| \mathbf{J}^{(R+1)} - r^{(R)} \right\| \quad \textbf{(2. 15)}$$

Where

$$\mathbf{J}^{(R)} = \mathbf{J}(\hat{q}^{(R)}) = \left. \frac{\partial f}{\partial q} \right|_{q=\hat{q}^{(R)}} \quad \textbf{(2. 16)}$$

$$\mathbf{J}^{(R)} = \begin{bmatrix} \dfrac{\partial f(t_1)}{\partial q_1} & \cdots & \dfrac{\partial f(t_1)}{\partial q_n} \\ \vdots & & \vdots \\ \dfrac{\partial f(t_N)}{\partial q_1} & \cdots & \dfrac{\partial f(t_N)}{\partial q_n} \end{bmatrix} \quad \textbf{(2. 17)}$$

and $$r^{(R)} = f\left(\hat{q}^R\right) - y \quad \textbf{(2. 18)}$$

The problem with this method occurs if the matrix $\mathbf{J}_p$ which is poorly conditioned or even singular. This can happen when we have a large number of parameters as in neural network training.

2.5.4 Levenberg-Marquardt

The Levenberg-Marquardt is an extension of the Gauss-Newton method algorithm, where the search direction P_R is computed by solving the linear least squares problem.

$$P_R = \arg\min\left\|\mathbf{J}_p - r\right\| + l_R^2\left\|p\right\| \quad \textbf{(2. 19)}$$

Where $\mathbf{J}_p$ is the Jacobian matrix and l_R is the regularization parameter, which solves the problem of a poorly conditioned matrix. The advantages of this method are basically the same of the Gauss-Newton method, however, due to modified search direction, this algorithm is more robust.

2.6 Model Complexity

The model complexity is related to the number of parameters that the estimated model possesses. A model becomes more complex if additional parameters are added and it becomes simpler if some parameters are removed [10]. Model complexity also represents the flexibility of the model. For a good performance in a wide range of operation, a model should not be too simple, because it would not be capable of capturing

the process behavior with a reasonable degree of accuracy. On the other hand, a model should not be too complex because it would possess too many parameters to be estimated with the available finite data set.

The characteristic of the process, the amount and quality of the available data, the prior knowledge and the type of model imply an optimal model complexity. Each additional parameter makes the model more flexible, however, it makes it harder to accurately estimate the optimal parameter values. The technique used in this work to optimize the model complexity was pruning.

2.7 Pruning

When working with neural networks to solve real problems, we require the use of highly structured networks of large size. Based on [18], the issue is minimizing the network size and yet achieved good approximation. Pruning is a generic term for all kinds of neural networks training techniques that decrease the network complexity by removing parameters or nodes. It is a method that decreases the capacity of the model in order to limit over-fitting of the system modeling [19]. The most important reason for performing this structure optimization was the possibility that the number of possible models and parameters was significantly reduced. Pruning works iteratively discarding those nodes and parameters that have little influence and then re-estimate the values of the remaining ones. Some of the pruning techniques are the followings:

- Clustering – weight decay method and weight elimination method
- Threshold pruning

- Hessian matrix of error surface – which includes optimal brain damage and optimal brain surgeon methods
- Orthogonal Least Squares

Based on [20] an effective way to prune an artificial neural network is to employ a clustering algorithm on weight matrices to perform an appropriate reduction. This algorithm picks out the distinctive subsets of weights embedded in a high dimensional space and selects lower dimensional weight matrices. The pruning algorithm presented in [20] applies a metric distance method to find the best way partition that the metric difference among the weights in a cluster is less that the error provided by the user, which eliminates the irrelevant links and redundant nodes from the trained neural network.

Other way to make pruning presented in [11,20] is threshold pruning. The threshold pruning is applied to eliminate redundant links and possibly input or hidden nodes from the trained neural network architecture while preserving accuracy of the original network. The idea is that links with low weights are not decisive for a neuron's activation function and they are not contributing new information in the network, consequently these low weights links are not retained and they are labeled as irrelevant. After pruning, the network maintains high accuracy but lower dimension once all the redundant links and nodes are eliminated.

The idea of the method using the Hessian matrix of error surface is to use information of the second order derivatives in order to make a trade-off between network complexity and training error performance. The optimal brain damage (OBD) identifies a set of parameter whose deletion from the neural network will cause the least increase in

the value of the cost function. Optimal brain surgeon (OBS) simplifies the computation making as an assumption that the Hessian matrix is a diagonal matrix.

The orthogonal least squares method calculates the individual contribution to the desired output from each basis vector. In other words, the algorithm selects the basis functions that produce a better fit of the model.

The use of any of this method can assure better generalization, fewer training examples and improves the speed of learning of the neural network. Also eliminates the ill – conditioning problem, which is related to the number of parameters in the estimation.

2.8 Conclusions

This Chapter provided the background on gray box modeling, DC drives systems and parameter estimation used in this work.

Chapter 3

PARAMETER ESTIMATION: TWO STAGE METHOD

Parameter estimation is a common problem in many areas of process modeling, both in on-line applications such as real time optimization and in off-line applications such as the modeling of electric drives loads. The goal was to determine values of model parameters that provide the best fit to data. This chapter presents the derivation of the two-stage linear least squares method for drive identification.

3.1 Gray – Box Model Structure

After knowing and understanding the complete mathematical model for the dc drive and load, a function for the load modeling was determined, assumed to be unknown. The selection of artificial neural networks for modeling the load was not in an intuitive way. The use of artificial neural networks was the model structure use on previous work [1], for which we conclude that the use of artificial neural networks was a good step in the parameter estimation technique tacking into account the purpose of this work. Radial basis function was selected for used as artificial neural networks. Equation (3.1) and equation (3.2) presents the mathematical expression for the gaussian radial basis function neural network introduced in Chapter 2.

$$NN(x) = \sum_{i=1}^{N} \mathbf{a}_i \mathbf{F}_i\left(x, c_i, s_i\right) \quad \textbf{(3. 1)}$$

$$\mathbf{F}\left(x.c, s\right) = e^{\left(-\frac{1}{2}\frac{(w(t)-c)^2}{s^2}\right)} \quad \textbf{(3. 2)}$$

The motor torque is known to be a linear function of the armature current. Furthermore it is known that friction and fan load, common characteristics of mechanical systems are nonlinear functions of speed. These facts were used to further specify the mechanical subsystem model as in equation (3.3). For this work, the gray box model included the terms in the load speed state function and the radial basis function artificial neural network. The complete mathematical model for the mechanical equation of the system including the neural network is presented in equation (3.4).

$$\frac{dw(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{B_m}{J_m} w(t) - \frac{1}{J_m} t_L\left(w(t)\right) \quad \textbf{(3. 3)}$$

$$\frac{dw(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{B_m}{J_m} w(t) - \frac{1}{J_m}\left(\sum_{i=1}^{N} a_i e^{\left(-\frac{1}{2}\frac{(w(t)_i-c_i)^2}{s_i^2}\right)}\right) \quad \textbf{(3. 4)}$$

Another possible gray box structure of the system was derived from the physical model [1]. It is based on the fact the models based in the structure of equation (3.3) may not be identifiable. That is, multiple combinations of neural network parameters and loads constants, resulting in the same load characteristics are possible. The second model structure under considerations was:

$$\frac{dw(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{1}{J_m} t_L\left(w(t)\right) \quad \textbf{(3. 5)}$$

$$\frac{dw(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{1}{J_m}\left(\sum_{i=1}^{N} \mathbf{a}_i e^{\left(-\frac{1}{2} \frac{(w(t)_i - c)^2}{s^2} \right)} \right) \quad \textbf{(3. 6)}$$

For training, the system was excited with a pre-selected voltage signal as shown in Figure 8. The parameters of the neural network, centers and standard deviations were selected a priori.

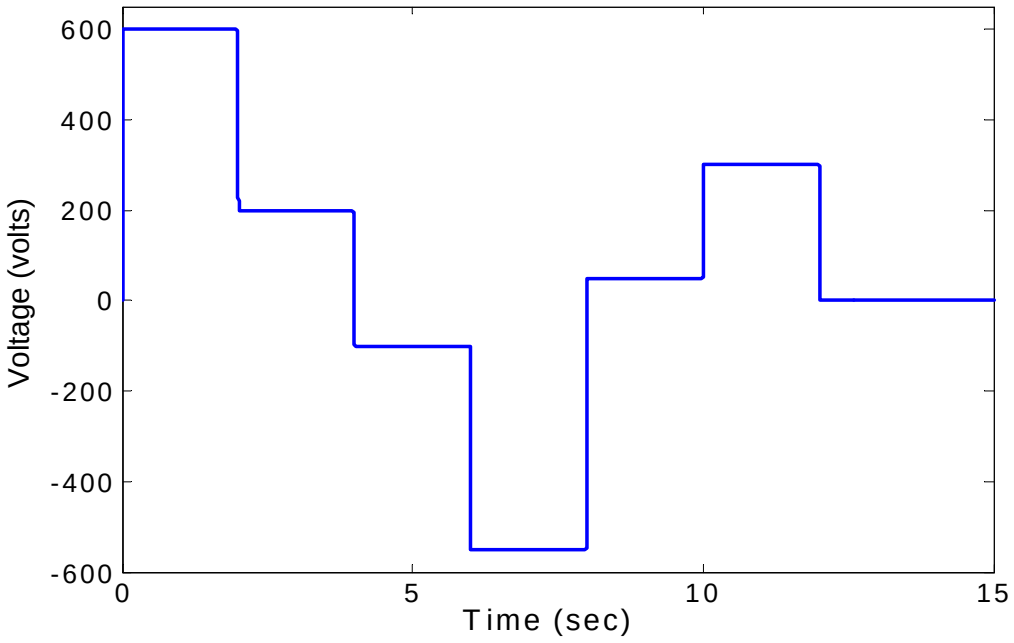

Figure 8 Excitation voltage used for the training of the model

3.2 Linear Regression Models for Drive Identification

In our model we can group model parameters into the electrical parameters q_e and the mechanical parameters q_m. The mechanical ones include the parameters of the

artificial neural network. For parameter estimation, two-stage method was used. This method estimated the parameters in two independent steps, first the electrical parameters are estimated and second the mechanical parameters are estimated. This method made the estimation process straightforward, because it reduces the problem from a complex non-linear system toward two simple estimation problems. After simplifying the problem, solved it by the use of linear least squares to estimate both groups of parameters, electrical and mechanical. The most important advantage of the used of the two-stage method was avoiding the used of the Gauss Newton method, which is computationally complex and time consuming.

The method was used with two different model structures. One with the viscosity parameter B_m and the other no viscosity term is explicitly present.

Case 1: Mechanical Model without explicit viscosity term.

The model studied here is described by the following equation where no viscosity term B_m appears explicitly in the mechanical equation.

$$\frac{di_a(t)}{dt} = \frac{1}{L_a} Va(t) - \frac{R_a}{L_a} i_a(t) - \frac{K_a}{L_a} w(t) \qquad \textbf{(3. 7)}$$

$$\frac{dw(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{1}{J_m} T_L(w(t)) \qquad \textbf{(3. 8)}$$

The electrical equations can be arrange into the linear regression form as follows

$$\mathbf{y}_e(t) = \mathbf{x}_e^T(t)\mathbf{?}_e \qquad \textbf{(3. 9)}$$

Where the linear regression model for which $\hat{q}_e$ is computed using linear least squares is presented in equation below.

$$\mathbf{y}_e(t) = \frac{di_a(t)}{dt} \quad \textbf{(3. 10)}$$

$$\mathbf{x}^T{}_e(t) = \begin{bmatrix} V_a & -i_a & -w(t) \end{bmatrix} \quad \textbf{(3. 11)}$$

$$\hat{\boldsymbol{\theta}}_e = \begin{bmatrix} q_{e1} & q_{e2} & q_{e3} \end{bmatrix}^T \quad \textbf{(3. 12)}$$

The actual electrical parameter estimations can be obtained from q_{e1}, q_{e2} and q_{e3} as follows

$$\hat{L}_a = \frac{1}{\hat{q}_{e1}} \quad \textbf{(3. 13)}$$

$$\hat{R}_a = \frac{\hat{q}_{e2}}{\hat{q}_{e1}} \quad \textbf{(3. 14)}$$

$$\hat{K}_a = \frac{\hat{q}_{e3}}{\hat{q}_{e1}} \quad \textbf{(3. 15)}$$

A linear regression model can be obtained with the mechanical equation (3.16), if the electromagnetic torque T_{em} is as shown in equation (3.17)

$$J_m \frac{dw(t)}{dt} + \mathbf{F}(w(t))\mathbf{a} = T_{em} \quad \textbf{(3. 16)}$$

$$\mathbf{y}_m(t) = \mathbf{x}^T{}_m(t)\boldsymbol{\theta}_m \quad \textbf{(3. 17)}$$

Where

$$\mathbf{y}_m(t) = T_{em} = K_a i_a \quad \textbf{(3. 18)}$$

$$\mathbf{x}^T{}_m(t) = \left[\frac{dw(t)}{dt} \quad \mathbf{F}\left(w(t)\right) \right] \tag{3. 19}$$

$$\mathbf{?}_m = \left[J_m \quad \mathbf{a} \right]^T \tag{3. 20}$$

Since T_{em} is not available, we estimate it using the estimated $\hat{K}_a$ obtained from the electric regression model.

Case 2: Mechanical Model with explicit viscosity

The mechanical model used for this second case is

$$\frac{dw(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{B_m}{J_m} w(t) - \frac{1}{J_m} T_L\left(w(t)\right) \tag{3. 21}$$

By the substitution of $\hat{T}_{em} = \hat{K}_a i_a$ and the artificial neural network for the load $T_L\left(w\left(t\right)\right) = \mathbf{F}\left(w\left(t\right)\right)\mathbf{a}$ in previous equation (3.21), we obtained equation (3.22)

$$J_m \frac{dw\left(t\right)}{dt} + B_m w\left(t\right) + \mathbf{F}\left(w\left(t\right)\right)\mathbf{a} = T_{em} \tag{3. 22}$$

The corresponding linear regression model is given by

$$\mathbf{y}_m(t) = T_{em}(t) = \hat{K}_a i_a(t) \tag{3. 23}$$

$$\mathbf{x}_m^T(t) = \left[\frac{dw(t)}{dt} \quad w(t) \quad \mathbf{F}\left(w(t)\right) \right] \tag{3. 24}$$

$$\mathbf{?}_m = \left[q_{m4} \quad q_{m5} \quad q_{m6} \right]^T \tag{3. 25}$$

Parameter estimation is done using the linear least squares method as described previously where K_a is substituted by $\hat{K}_a$.

3.3 State Variable Filters: Avoiding Differentiation

The most obvious problem with the two-stage algorithm is the differentiation of the measured signals, $w(t)$ and $i_a(t)$. The acceleration is a high bandwidth signal, which in practice is difficult to measure. Therefore, this quantity is obtained by estimation techniques, such as observers, direct differentiation and state variable filters [21, 22].

- Observer - the response is faster than the system response, but include more parameters to the system, add additional and add a delay to the system.
- Direct Differentiation (forward of backward) – From typical velocity signal derived from a pulse encoder output produces noisy estimates and amplifies inaccuracies to a level that cannot be accepted in practical high performance applications.
- State Variable Filter – this estimation is relatively simple, involves only one gain and produce good performance of the results but its more sensitive to noise than the observer. Some pre-filtering techniques could be applied to restore the corrupted velocity signal in order to facilitate the estimation of the acceleration.

To avoid differentiation in this work, state variable filters were used. The basic idea was to use filters reducing noise effects from the process signal by combining filtering and derivations [23]. It is important to keep in mind that the input – output process was applied to all the measures, current, voltage and velocity; in other words, the same type of state variable filter, filters all the measurements. Figure 9 shows the basic

block diagram of the state variable filter. This filter is a simple one that is composed of one integer, one gain and unitary feedback that keeps tracking of the original signal with a minimum error.

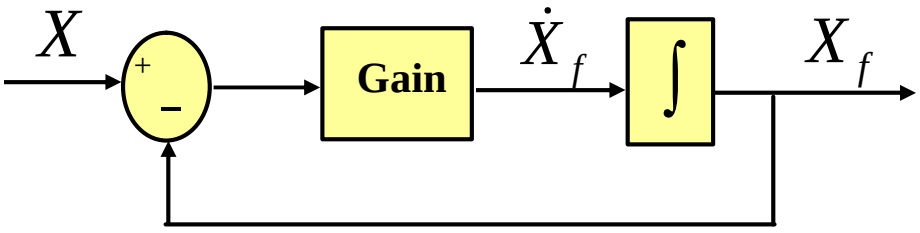

Figure 9 State Variable Filter

As example of the results of used state variable filters was presented in Figure 10, which showed the original and filtered velocities.

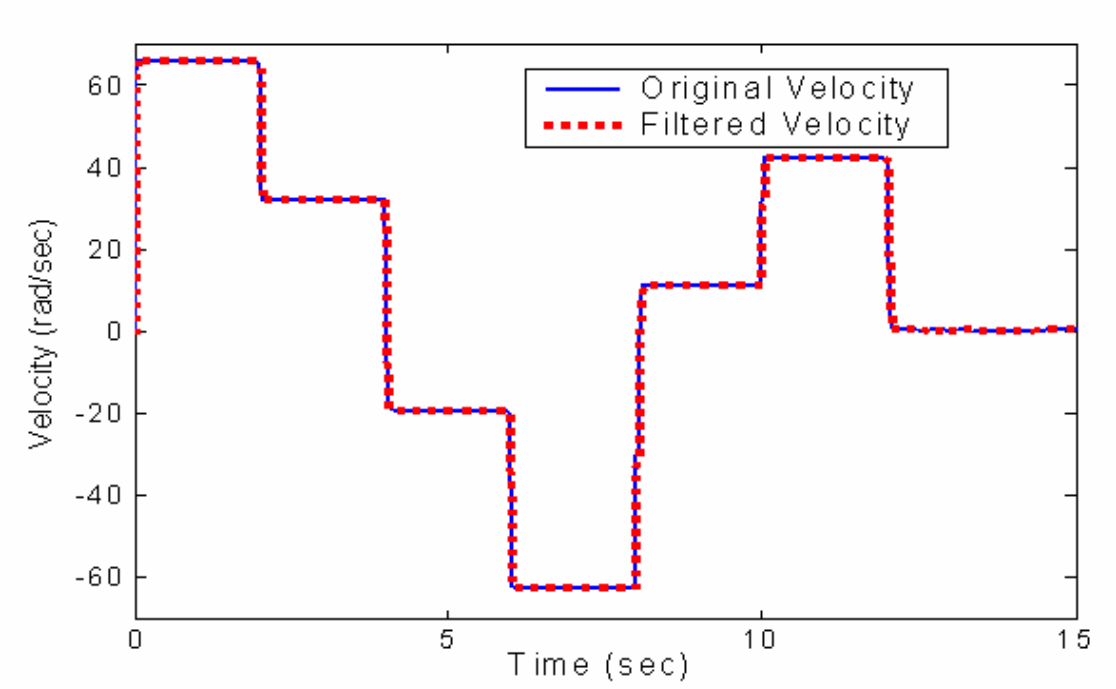

Figure 10 Original velocity vs. filtered velocity

Figure 11 present how the state variable filters are used in algorithm implementation. Using these filters, the original values, the filtered values and their derivatives values was obtained.

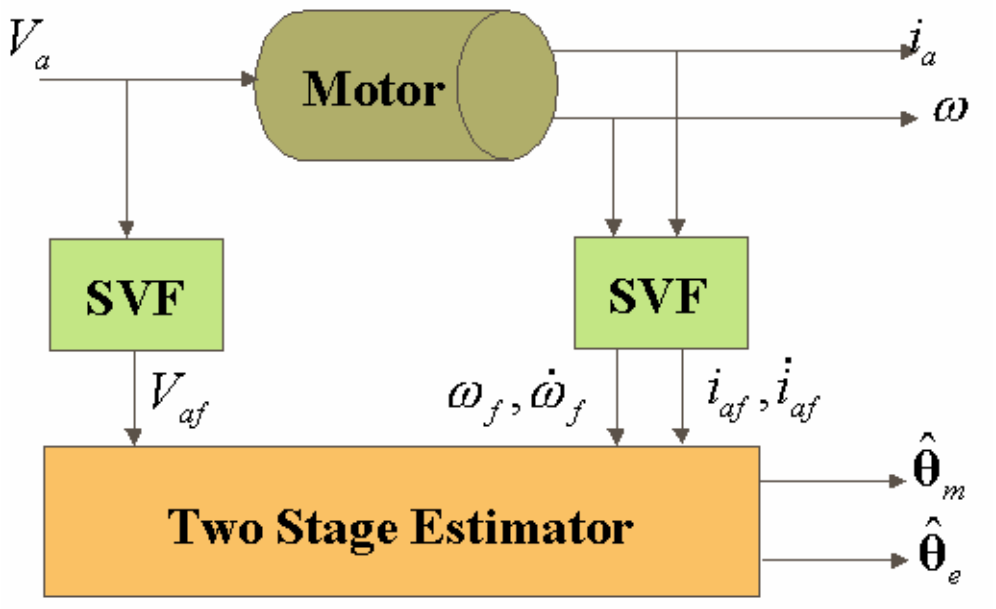

Figure 11 Avoiding differentiation: State Variable Filters Process

3.4 Two-Stage Algorithm

The two-stage algorithm is as follows

Stage 1: Compute $\hat{q}_e$ solving the linear least squares problem.

$$\hat{\boldsymbol{\theta}}_e = \arg\min_{q_e} \left\| \mathbf{y}_e - \mathbf{X}_e \boldsymbol{\theta}_e \right\|_2^2 \quad \textbf{(3. 26)}$$

$$\hat{\boldsymbol{\theta}}_e = \left(\mathbf{X}_e^T \mathbf{X}_e \right)^{-1} \mathbf{X}_e^T \mathbf{y}_e \quad \textbf{(3. 27)}$$

Compute the electrical parameter estimates as follows

$$\hat{L}_a = \frac{1}{\hat{q}_{e1}},\ \hat{R}_a = \frac{\hat{q}_{e2}}{\hat{q}_{e1}},\ \hat{K}_a = \frac{\hat{q}_{e3}}{\hat{q}_{e1}} \qquad \textbf{(3. 28)}$$

Stage 2: Compute the electromagnetic torque T_{em} using

$$\hat{T}_{em} = \hat{K}_a i_a \qquad \textbf{(3. 29)}$$

Compute $\hat{q}_m$ solving the linear least squares problem

$$\hat{\mathbf{?}}_m = \arg\min_{q_m} \left\| \mathbf{y}_m - \mathbf{X}_m \mathbf{?}_m \right\|_2^2 \qquad \textbf{(3. 30)}$$

$$\hat{\mathbf{?}}_m = \left(\mathbf{X}_m^T \mathbf{X}_m \right)^{-1} \mathbf{X}_m^T \mathbf{y}_m \qquad \textbf{(3. 31)}$$

3.5 Pruning using Orthogonal Least Squares

To explain the orthogonal least squares algorithm, let start by defining the linear model with equation (3.32)

$$\mathbf{y} = \mathbf{F}\,\mathbf{x} + e \qquad \textbf{(3. 32)}$$

The orthogonal least squares method is based on the transformation of the regression matrix $\mathbf{F}$, to a set of orthogonal basis vectors. This transformation would make possible the assessment of individual contributions of each basis vector.

The fist step is the QR decomposition of the regression matrix $\mathbf{F}$ into:

$$\mathbf{F} = \mathbf{QR} \qquad \textbf{(3. 33)}$$

Where R is an upper triangular matrix and Q is a matrix with orthogonal columns.

The second step is rearrange the equation (3.33) into (3.34), in where b is a transformed parameter vector satisfying $\mathbf{Rx} = \mathbf{b}$

$$\mathbf{y} = \mathbf{Qb} + e \tag{3. 34}$$

The solution of equation (3.34) is given by the least square represented in equations (3.35) and (3.36).

$$\mathbf{b} = \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T\mathbf{y} \tag{3. 35}$$

$$\mathbf{b}_i = \frac{\mathbf{q}_i^T\mathbf{y}}{\mathbf{q}_i^T\mathbf{q}_i} \tag{3. 36}$$

It is important to note that the previous equation is the part of the output variance described by the regressor q_i. Therefore the most influential regressors are chosen iteratively until a point is reach where adding a regressor contains no useful information. Finally the value of the regresors is estimated.

3.6 Full Identification Algorithm

The diagram in Figure 12 presented the full identification process algorithm. The first step was the selection of the model structure. The used of the radial basis function neural network for the black box was selected. The second step was the used of two-stage method to make the parameter estimation. The electrical parameters are separated from the mechanical parameters and neural network parameters into two different groups. First the estimation of the electrical parameter using linear least squares and then the mechanical parameters and neural networks parameters estimated by linear least squares method. The next step was the verification of the training data fitting. If the fitting was

good continue to the next step, on the other hand, if the fitting was not to good, go back to the model structure and estimate the parameters again. The next step was the pruning of the system to reduce the model complexity. If the system needed pruning go back again to the model structure, on the other hand if the system do not need pruning continue to system validation. Validation used the new values from the estimation with a new validation data in were the corroboration of the performance of the system was made. If the results of the validations were not satisfactory, go back to the model selection step and start again.

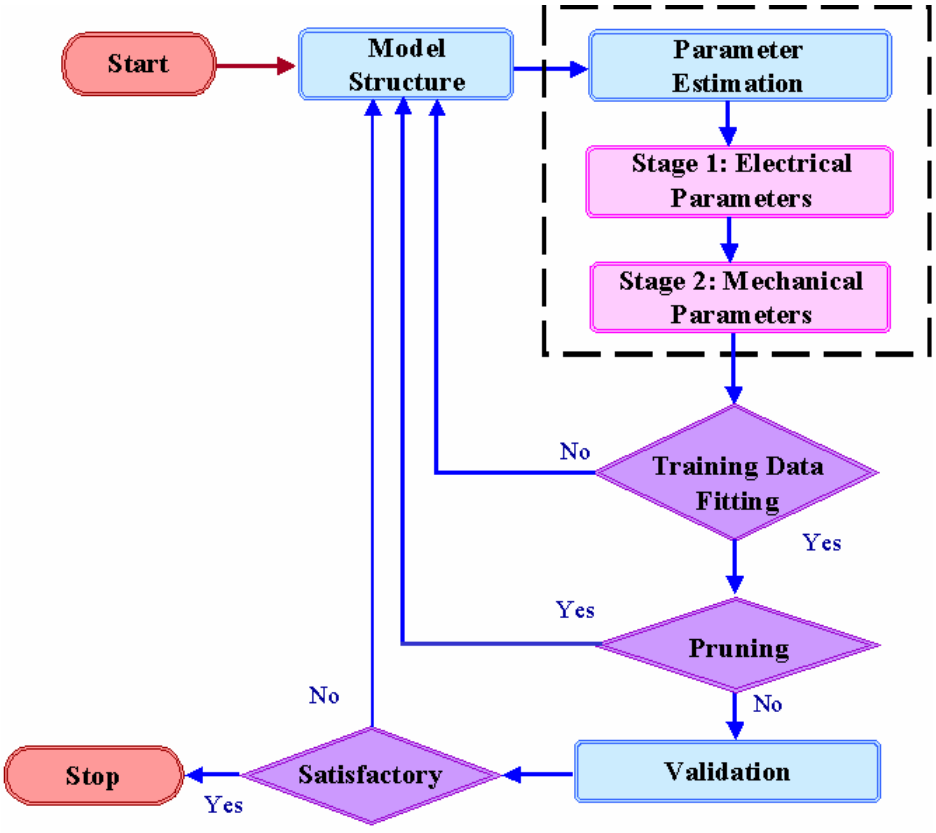

Figure 12 Full Identification Algorithm

3.7 Conclusions

This chapter presented the identification process step by step. The two-stage linear least squares method results in a simple technique for parameter estimation. The state variable filter is a useful way to avoid the differentiation without adding noise to the system, producing good results.

Chapter 4

SIMULATION RESULTS

This Chapter presents the results of the algorithm implementation for the identification of the system. The results are presented for two different models. The first model is without implicit viscosity parameter and the second is with the implicit viscous parameter. The results are presented with and without noise for the fan load. Also the validation results are presented. Finally, the pruning results by orthogonal least squares.

4.1 Initialization

Before present the estimation results is important to know how the simulation data was generated. It is important to remark that here is assumed that the system may be operated over the entire nominal operation range. In real implementation, this could be another source of ill-conditioning since it may be the case that operation constraints will limit how much excitation could be given to the system. This is a major concern in the system identification process and even mote here since artificial neural networks are function approximators for which this capability depends on hoe representative of the real system is the data available for the parameter estimation phase [1].

An important part of the identification is the training of the neural network in order to reduce the error. The training input of the model is performed using Matlab,

which is an armature voltage curve. The training armature voltage was selected a priori. This training armature voltage is shown in Figure 13. The measure signals are the current and the velocity. The parameter values of the permanent magnet DC Motor was obtained by [1], and they are presented in Table 1, Chapter 2. There is some result noiseless free and results with noise, which is a gaussian noise of 0.1. For the initial model, 121 points for radial basis function centers were selected. This center distribution and the variances were selected a priori.

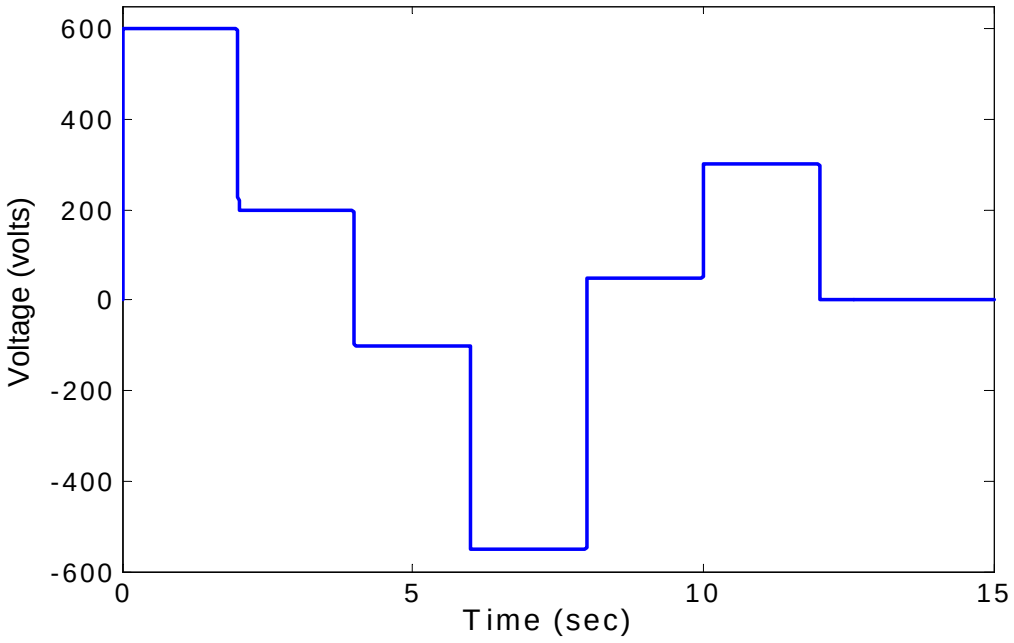

Figure 13 Excitation voltage used for the training of the model

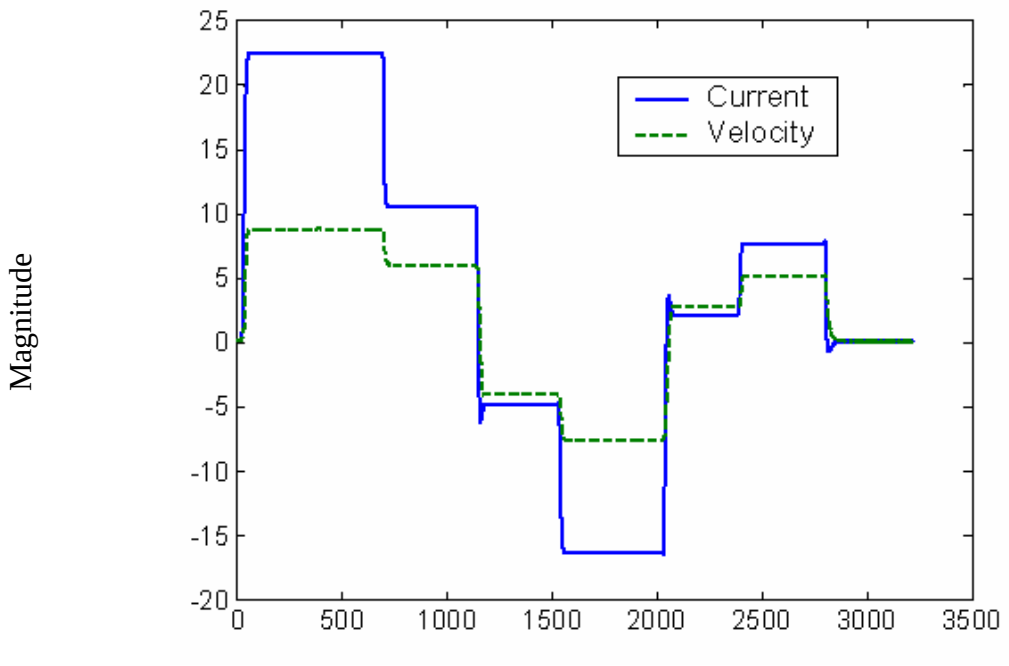

Figure 14 Current and Velocity Data

4.2 Mechanical Model Without Viscosity Parameter

Table 3 shows the estimation results for the noise free case.

Table 3 Parameter Estimation Results: Without Implicit Viscosity Parameter

Parameter	Real Value	Estimated Value	Error
R_a	7.56	7.5601	~0.0 %
L_a	0.055	0.055	0.0 %
K_a	3.475	3.475	0.0 %
J_m	0.06	0.056446	5.92 %

The following figures present results of the model identification. Figure 15 shows the estimated load torque characteristics. As we can see in Figure 15 the performance of the parameters estimation using the radial basis function artificial neural network in the black box for the identification process produces a good performance of the system. Figure 16, shows the estimation error between the real torque values versus the identified torque. In the error in torque figure, can be observed that the greater error its at the ends of the response. It may be seen that the approximation of the respective load characteristics by the artificial neural network is very good inside the nominal operations, but it start to lose performance outside this range.

Figure 17, compares the measured and estimated current, while Figure 18 shows the estimation error for the real current values and the estimated ones. Figure 17, clearly shows that both graphs are overlay, which means that good fitting performance was obtained by the identification. As shown in Figure 18, the maximum error between the real value current and the results of the current with the estimated parameters was approximately 1.5 ampere, but the mean of the error was very small, near to cero. Basically, the error occurs in the transition of the steps in the current curve.

Figures 19 presents the results of the real velocity values versus the estimated velocity, while Figure 20 presents the error of both velocities. Figure 19 present the performance of the system with the estimate parameters. Figure 20 presents that the velocity error was very small too. So the error in torque estimation does not have a significant effect on the speed estimate.

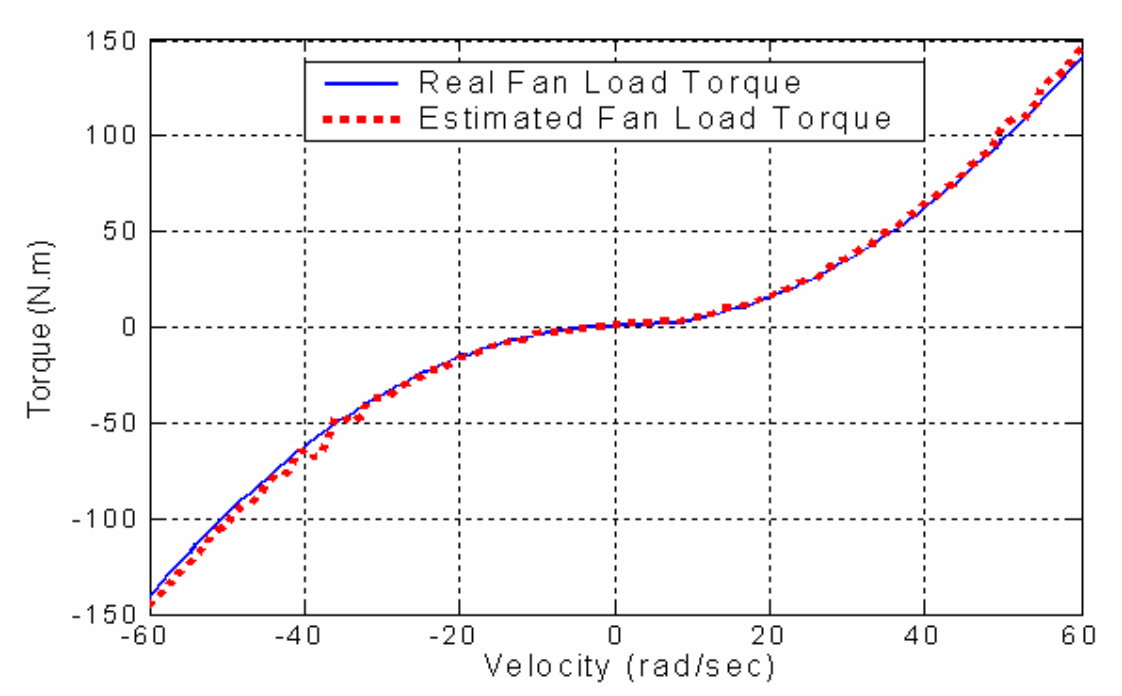

Figure 15 Real Fan Load Torque vs. Estimated Fan Load Torque

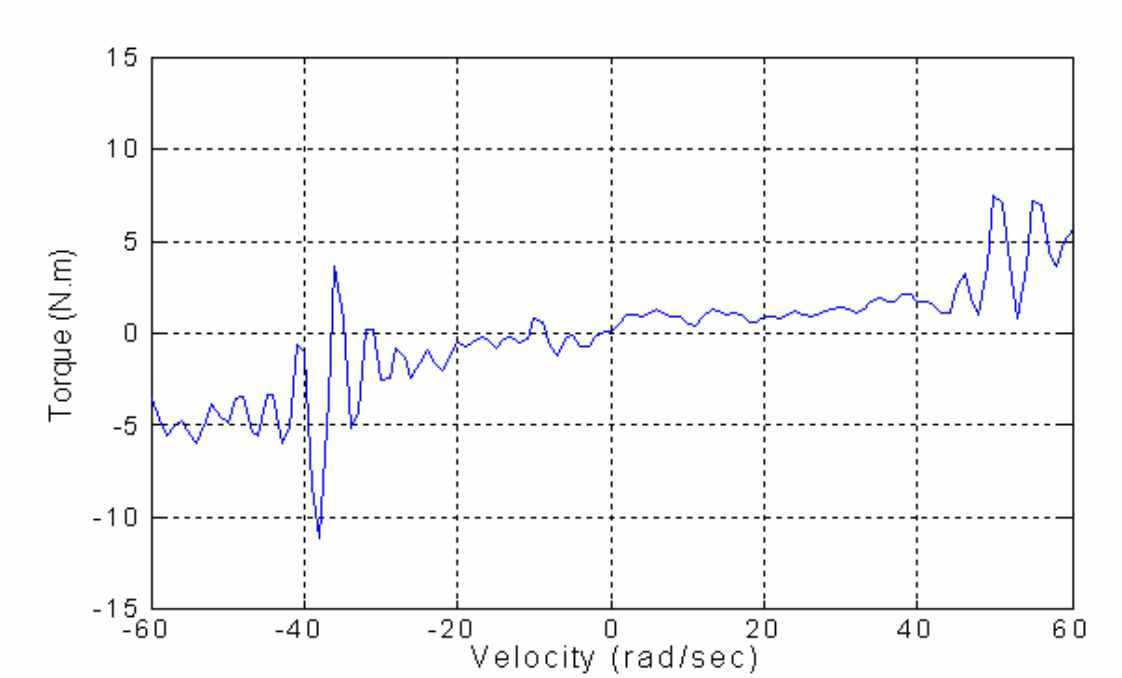

Figure 16 Torque error for the identified model

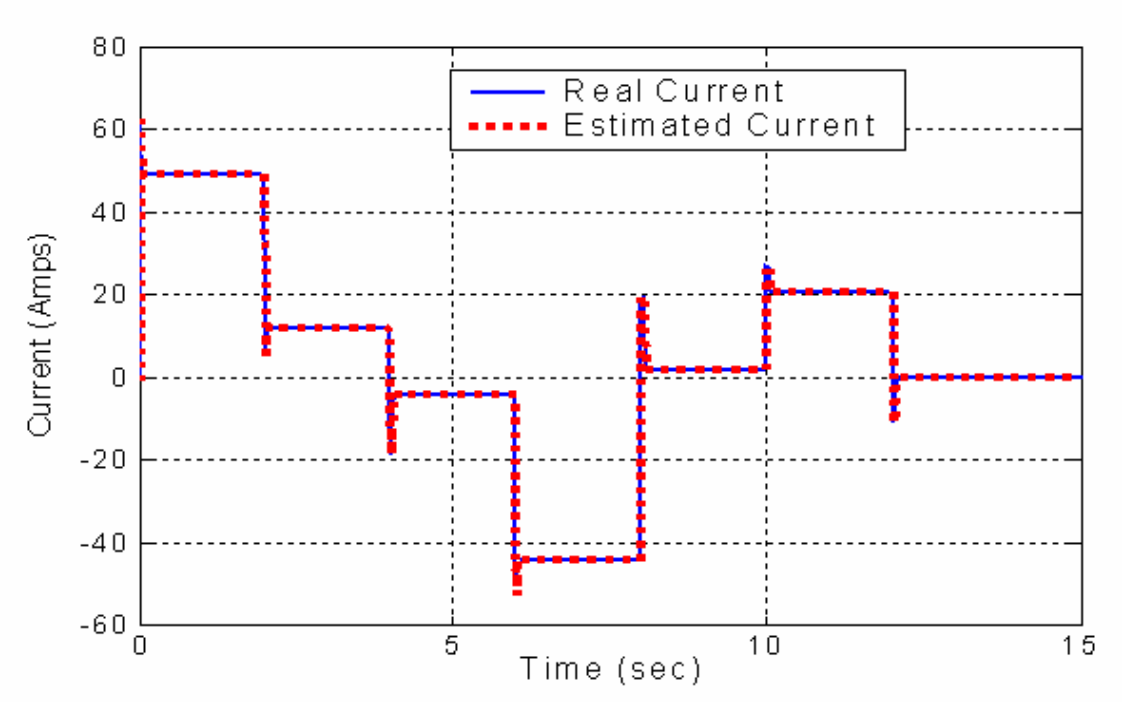

Figure 17 Real Current vs. Estimated Current

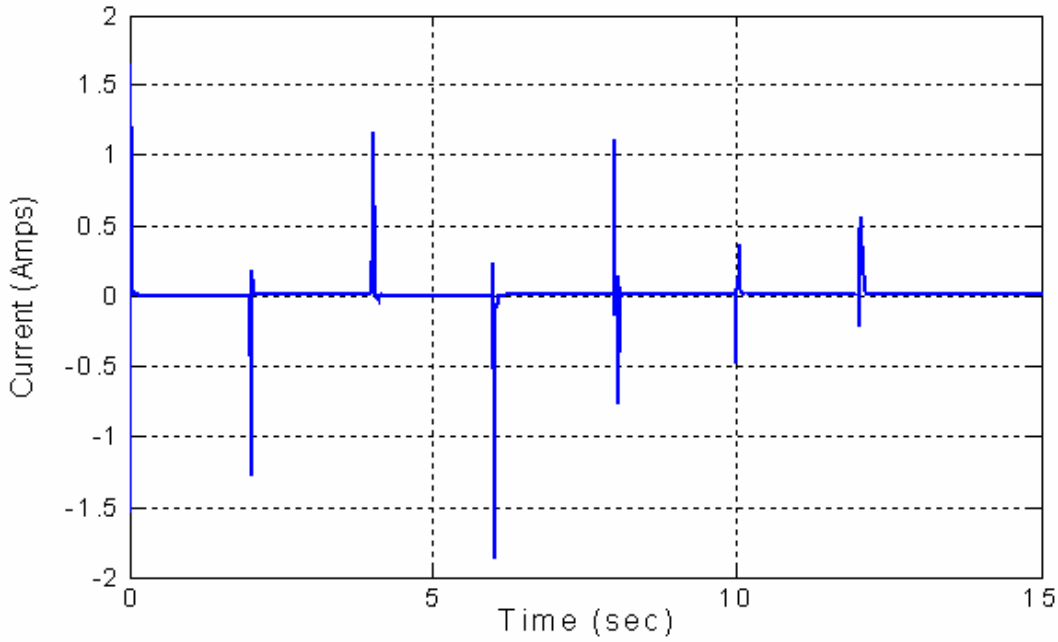

Figure 18 Current error for the identified model

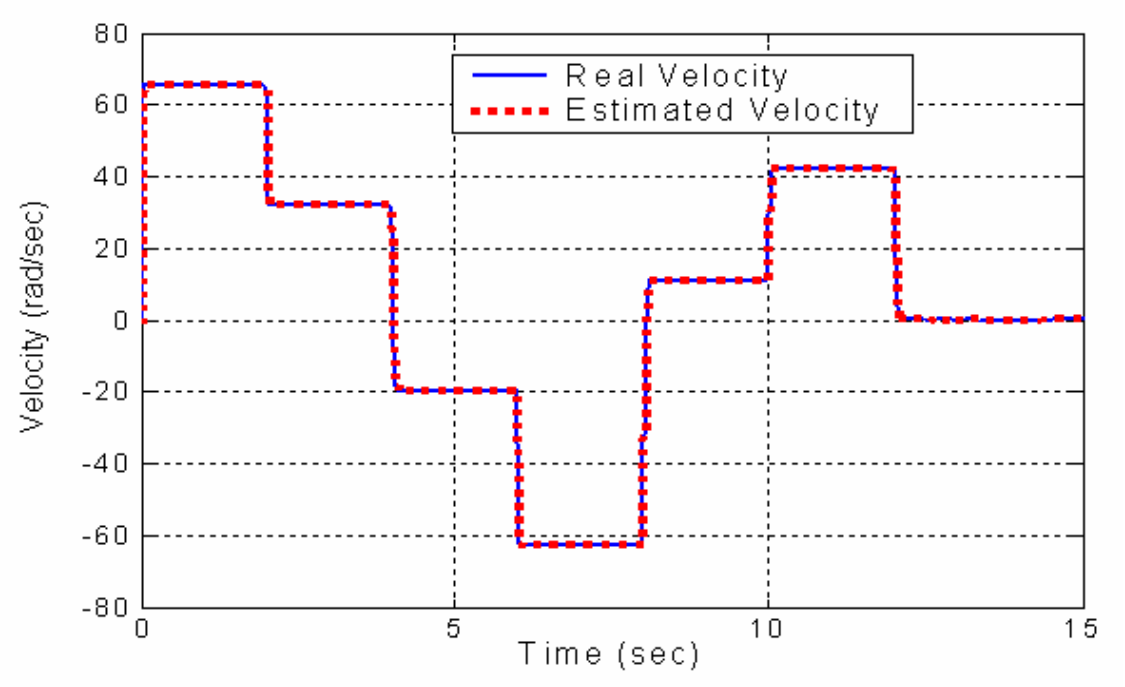

Figure 19 Real Velocity vs. Estimated Velocity

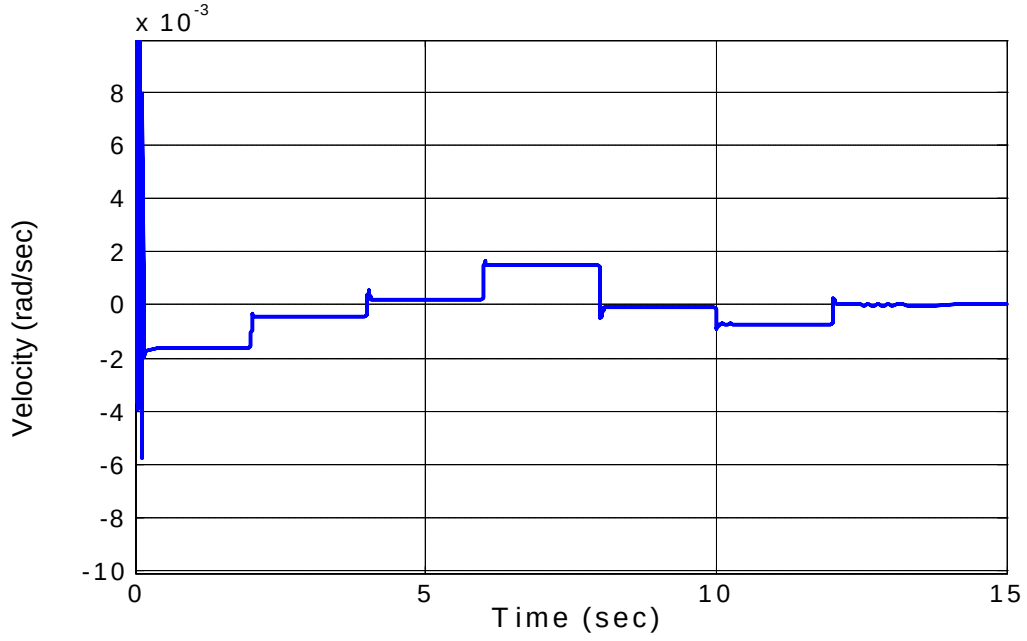

Figure 20 Velocity error for the identified model

We can conclude that the identified model has a satisfactory performance on the training data. We still need to evaluate noise effects and validate the model, as we shall see later in this chapter.

4.3 Mechanical Model With Explicit Viscosity Parameter

The estimates of the physical parameters for the model based on equations (3.7) and (3.8) are shown in Table 4. These results are for noise free data.

Table 4 Parameter Estimation Results with second case

Parameter	Real Value	Estimated Value	Error
R_a	7.56	7.5601	0.001 %
L_a	0.055	0.055	0.0 %
K_a	3.475	3.475	0.0 %
J_m	0.06	0.059937	0.105 %
B_m	0.03475	0.5252	93.3 %

We obtain good estimates for the electrical parameters and for the load inertia parameter value. On the other hand, the estimate of B_m has a very large error. Based on [1], some models may not be identifiable, that is, maybe multiple combinations of the artificial neural networks and viscous friction constant resulting in the same load characteristics. One reason for that error in the estimation of the viscosity parameter used in second case study, possibly is the fact that this model have more degrees of freedom and this could introduce a bias in the estimate of the parameter. Another possibility of

the error presented in torque but not in the currents and voltages in case 2 can be the role of the neural networks. Which means that maybe the artificial neural network in the gray-box model makes the arrangements that the parameter estimation failed and recovered in the current and voltage responses.

Figure 21 present the real torque values versus the torque results for the estimated values for mechanical model with explicit viscosity parameter. Figure 22 presents the error between both values, real versus estimated. As presented in the next figures the error increase when the estimated value for the viscosity parameter was used.

Figure 23, compares the measured and estimated current, while Figure 24 shows the estimation error for the real current values and the estimated ones. Figure 23, clearly shows that both graphs are overlay, which means that good fitting performance was obtained by the identification. As shown in Figure 24, the maximum error between the real value of the current and the results of the current with the estimated parameters was approximately 1.8 ampere, but the mean of the error was very small, near to cero. Basically, the error occurs in the transition of the steps in the current curve. With this small current value error we can conclude that the error in the viscosity parameter do not have a significant effect in the current of the estimate model.

Figures 25 presents the results of the real velocity values versus the estimated velocity, while Figure 26 presents the error of both velocities. Figure 25 present the performance of the system with the estimate parameters. Figure 26 presents that the velocity error was very small too. So the error in torque estimation for the viscosity parameter does not have a significant effect on the speed estimate.

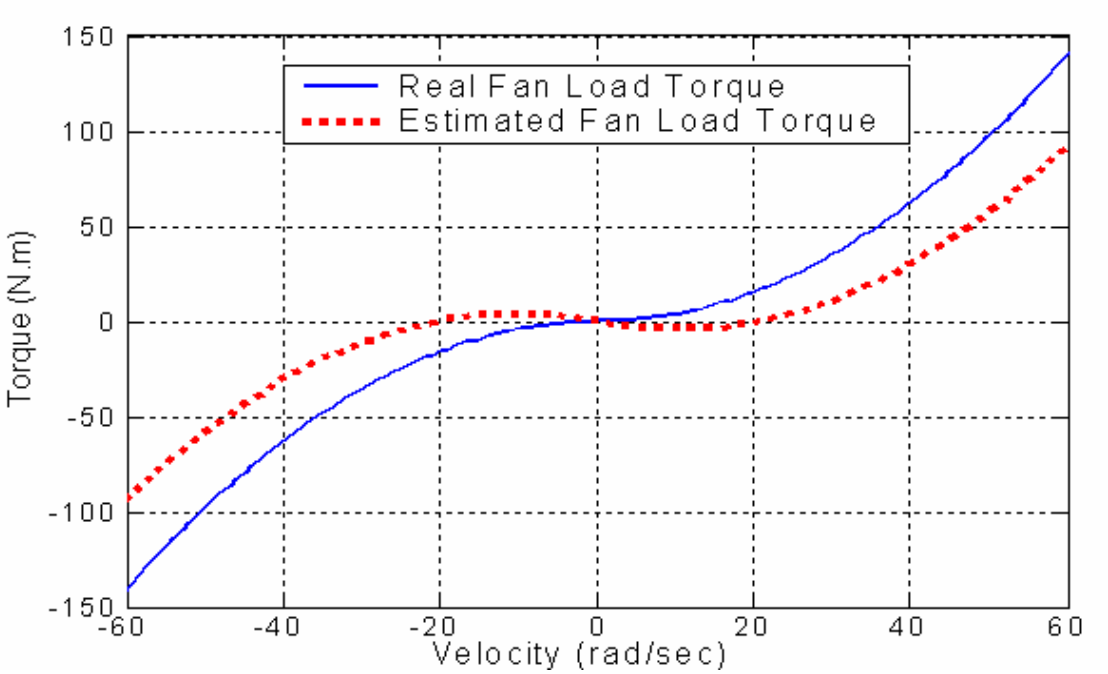

Figure 21 Real Torque vs. Estimated Torque

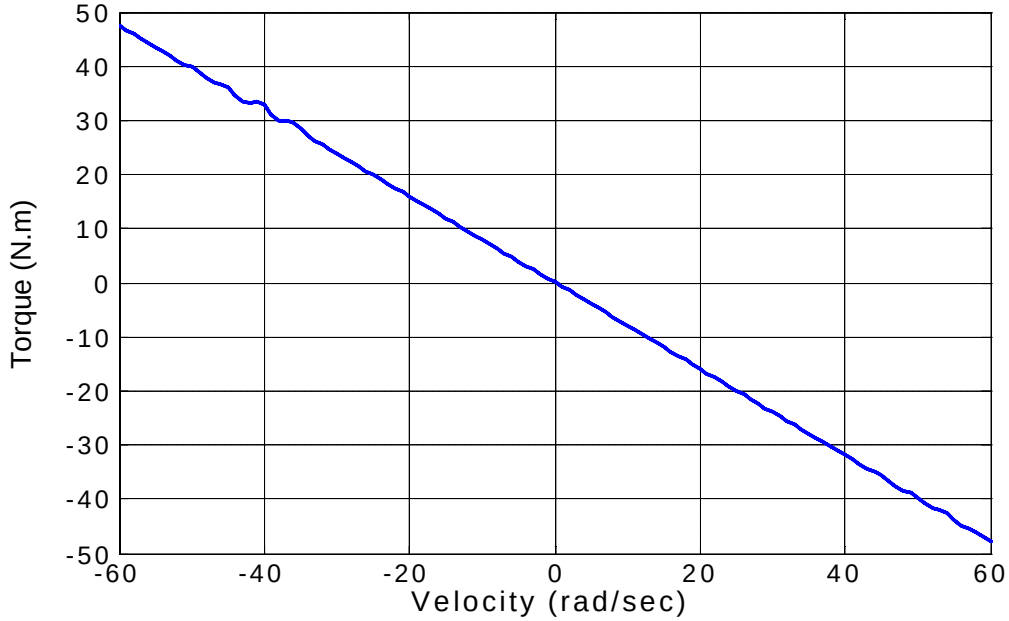

Figure 22 Torque error for identified model

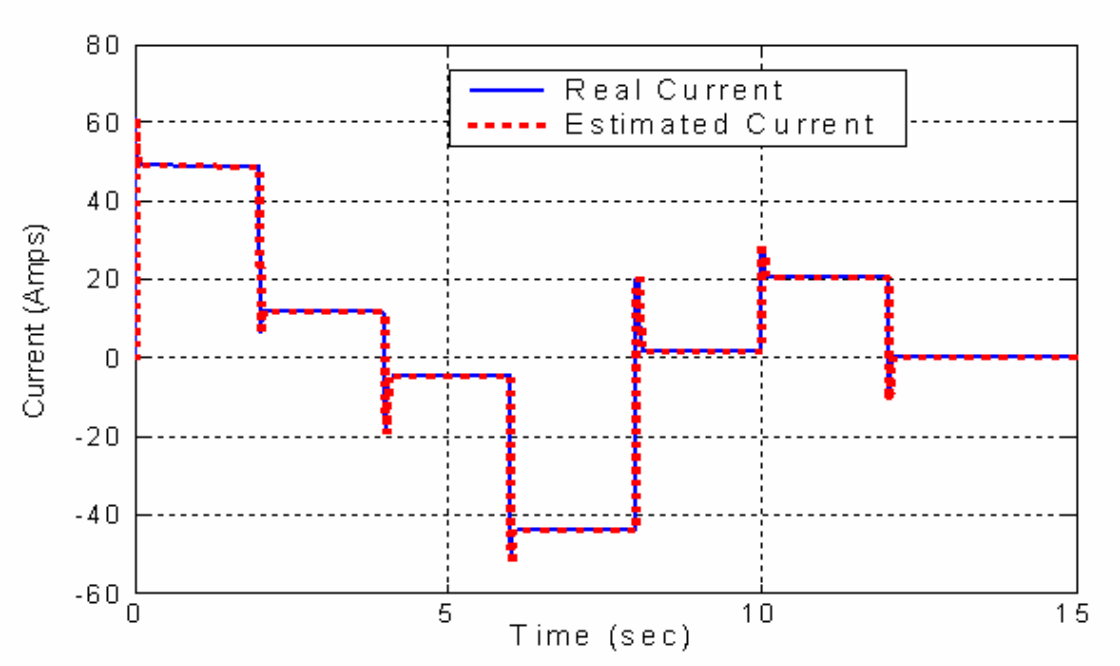

Figure 23 Real Current vs. Estimated Current

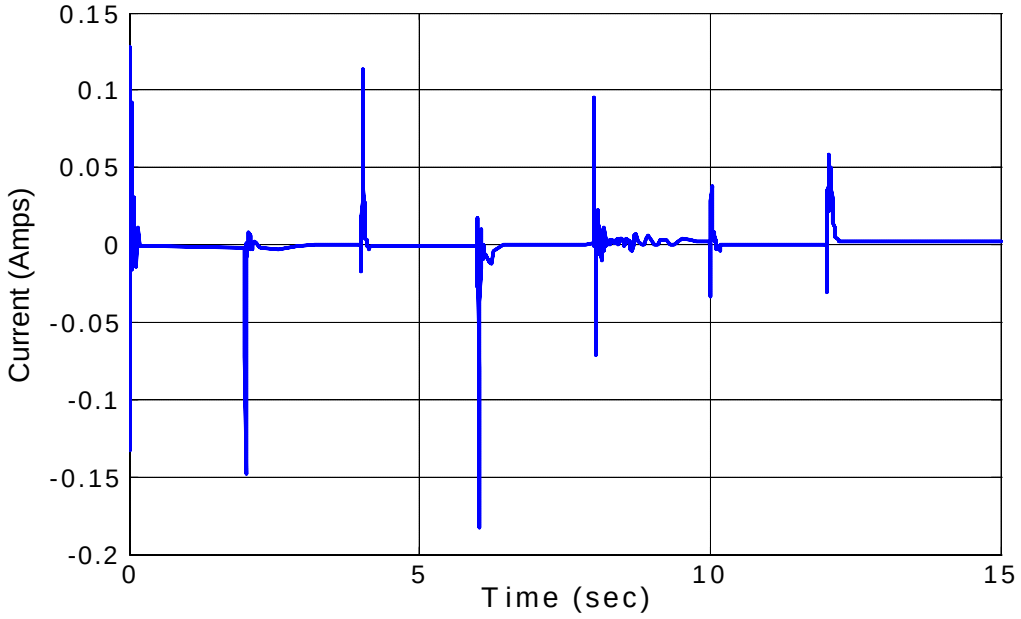

Figure 24 Current error for the identification model

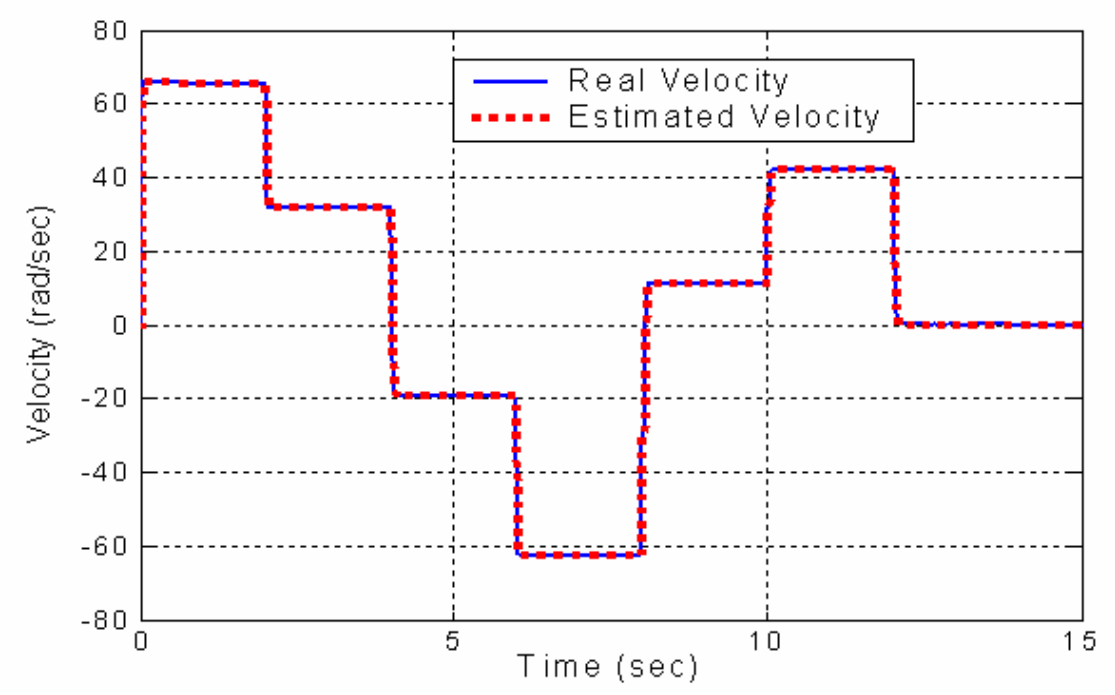

Figure 25 Real Velocity vs. Estimated Velocity

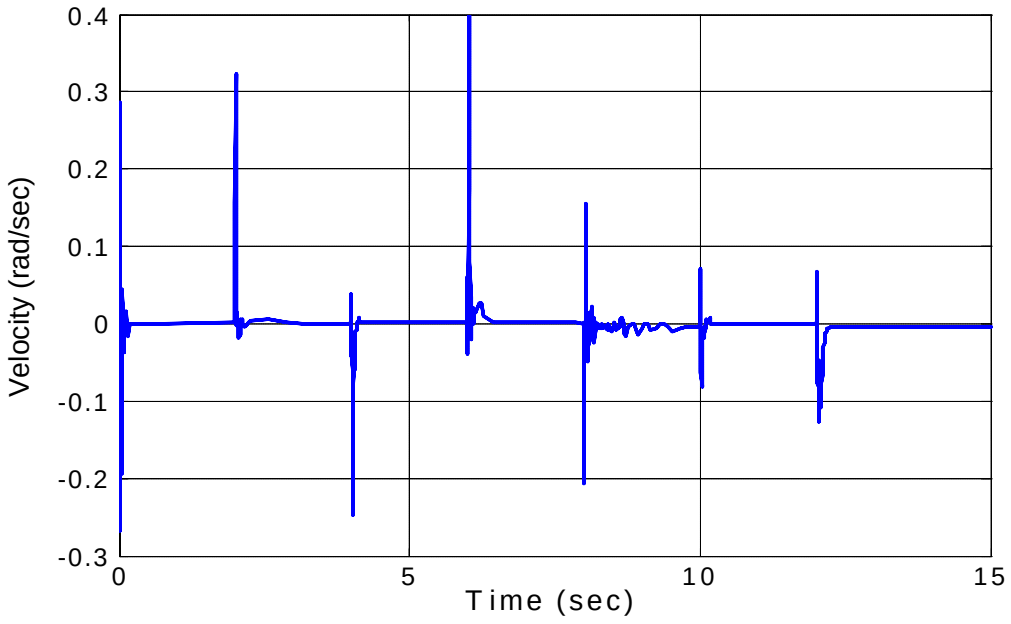

Figure 26 Velocity error for the identification model

Generally we obtain a good performance of the system using this mechanical model structure with explicit viscosity parameter for the currents and velocities estimated models. However, the estimated value for the viscous parameter has a big estimation error. Taking as base, the current and velocities results, we can conclude that we do not need the viscosity friction term to obtain good output results.

4.4 Validation

The simplest way to evaluate the quality of a model is to train it on a training data set and evaluate its performance of a different data set [10]. If the amount of available data is huge this causes no difficulties and is the most straightforward approach. However, we have to take care that both, training data and validation data, are representative and cover all considered operating regimes of the process. If the training set lacks data from some regimes, the model cannot be expected to perform well in these regimes. On the other hand, if important data is missing in the validation set the evaluation of the model performance becomes untrustworthy.

For the validation test, a different input armature voltage was used and shown in Figure 27. The voltage range is similar to the training input presented in Figure. In the validation process the identified model is fed with the validation voltage and it response is compared to that of the real system.

As conclude in previous section, the viscosity term B_m is not necessary to obtain good identification results. Based on this reason, we make the validation for the case

where the mechanical model does not have the explicit term of the viscosity. The results for the current and the velocity outputs for the validation results are presented in Figure 28 respectively.

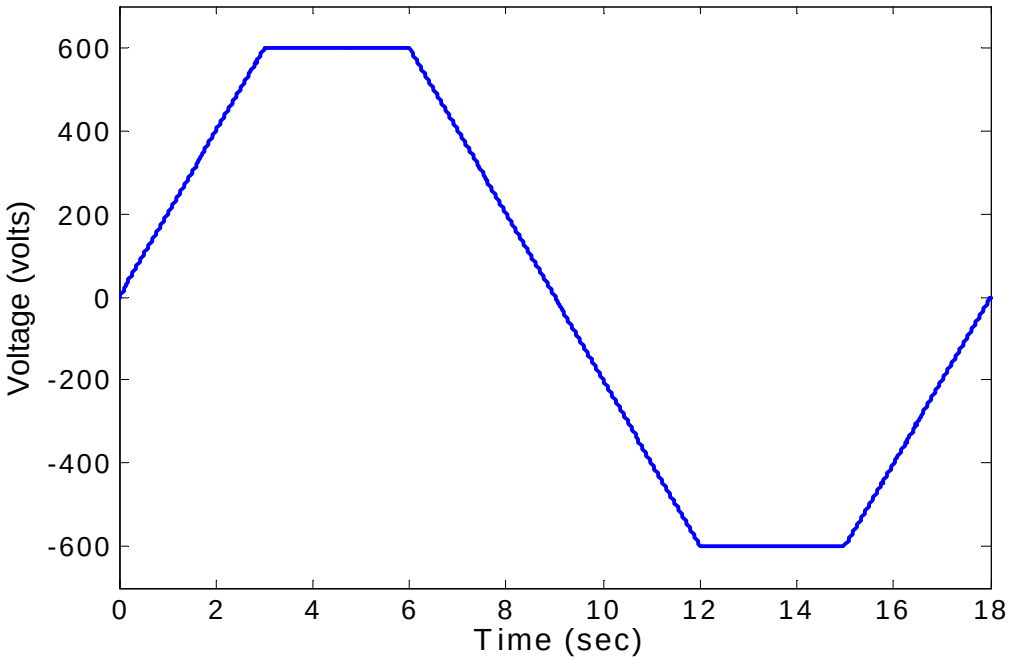

Figure 27 Excitation voltage used for the validation of the model

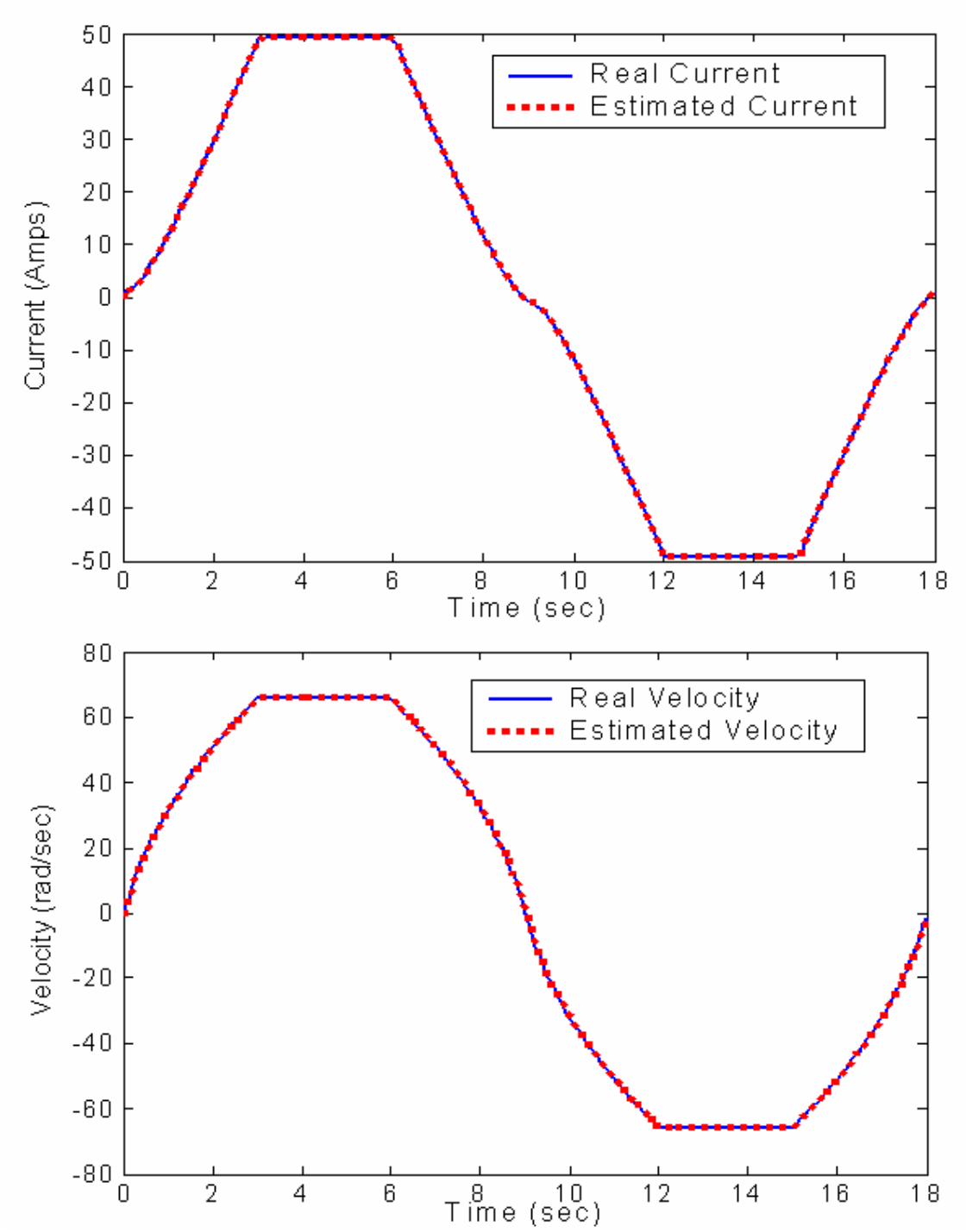

Figure 28 Validation of the System Performance for the case when the viscosity term is not present: (a) current response and (b) velocity response

We can conclude that perform validation to the model is a good practice. The validation results for the current and velocity plots presents the good performance of the estimated values, which produces good results with different validation data set.

4.5 Pruning

The problem that presents parameter estimation was the quantity of parameters that we need to make the load model a good one. The principal issue to motivate pruning is the number of parameter, which is indicative of over fitting and redundant parameters, and at the same time results in ill-conditioned parameters. Another issue is the bias/variance tradeoff, shown in Figure 29 from [10].

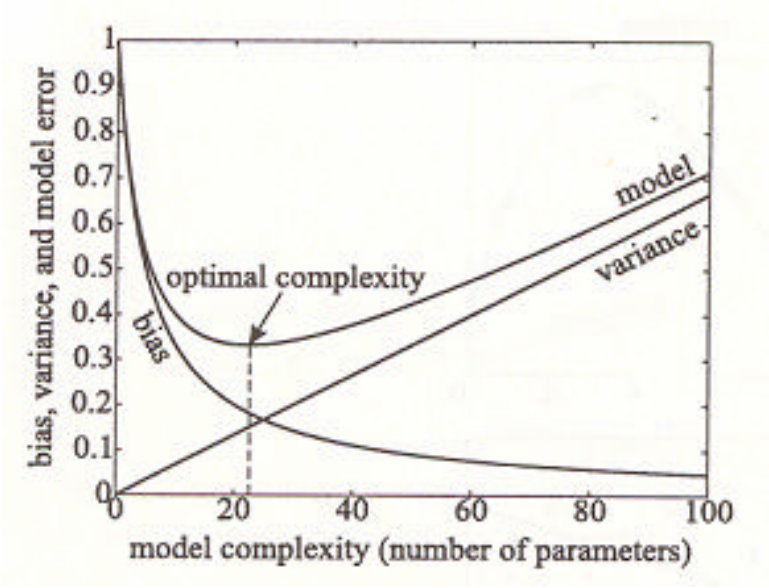

Figure 29 Bias/variance tradeoff (from [10]).

Figure 29 presents the effect of the bias and variance error on the model error. A very simple model has a high bias but a low variance error, while a very complex model

has a low bias but high variance error. A simple model can be improved by the incorporation of additional parameters, because the increase in the variance error is compensating by the decrease the bias error. On the other hand, the model that is too complex can be improved by discarding parameters, because the increase in the bias error is compensated by the decrease in the variance error. Somewhere between both models, simple and complex, lies the optimal model complexity, which do not have a lot of parameters but at the same time has a lower variance with a reasonable value of bias [10].

Orthogonal least square was the method implemented to avoid over-fitting and ill-conditioned parameters of the system modeling. This method involves the transformation of the set of regressors into a set of orthogonal basis vectors. The regression matrix can be decomposed in a triangular matrix and in a matrix with orthogonal columns. The space spanned by the set of orthogonal basis vectors is the same of this new orthogonal columns matrix. With this transformation of the parameters vector we satisfy the solution. The advantage of this method is that the algorithm makes it possible to calculate the individual contribution to the desired output variance from each basis vector. In other words, the algorithm selects the basis functions that produce a better fit of the model. In our case, we expect that the orthogonal least squares method make a considerably reduction on the dimension of the neural network.

Using orthogonal least squares, the number of nodes was reduced from one hundred twenty-one basis functions to fifty basis functions for both cases. We start reducing the basis function to the half and verify the current and velocity responses of the reduced model. After this first reduction, we started to reduce the model slowly to see

the effects of the basis function reduction in the current and velocity responses. During the reduction process, can be note that there is a range of basis function that does not affect the current and velocity response. For example, if the reduction was made from 121 basis functions to 55, produce the same results as if the model was reduced form 121 to 57 basis functions. With this information, the value of final basis function reduction model was selected. Also taking care of the parameter estimation error.

The estimates of the physical parameters for the model after model reduction by orthogonal least squares are shown in Table 5. These results are for noise free data.

Table 5 Parameter estimation results after pruning

Parameters	Real Value	Estimated Value	% error
Ka	3.475	3.50	0.72
Ra	7.56	7.6	0.53
La	0.055	0.054	1.81
Jm	0.06	0.056	6.67

Figure 30 shows the torque results after model reduction. The error obtained in this reduction was approximately 11%. Figure 31 present the performance of the velocity with the pruned model and Figure 32 present the performance of the current after model reduction. The maximum error occurs at the ends of the model. This can be clearly observed in Figure 32, were a difference between original current value and pruned current value is notable.

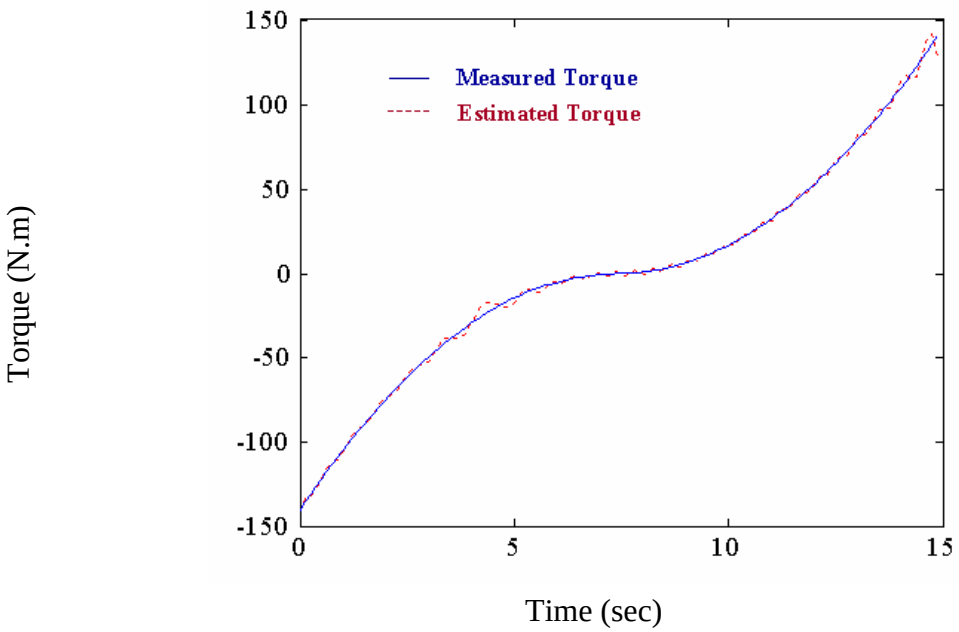

Figure 30 Torque results after pruning.

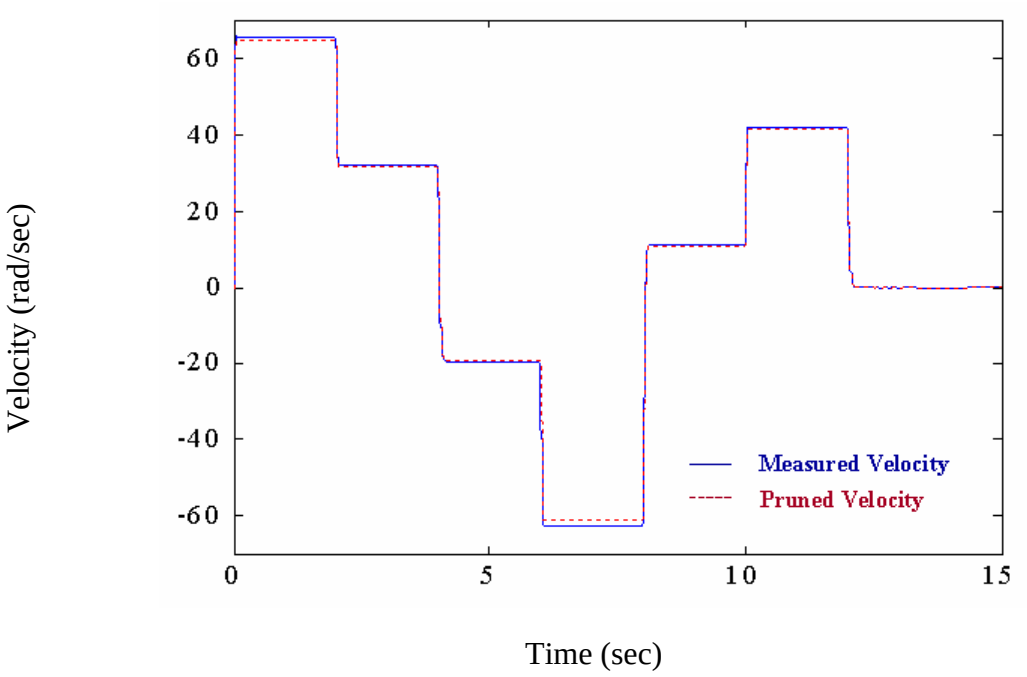

Figure 31 Pruned Velocity Results.

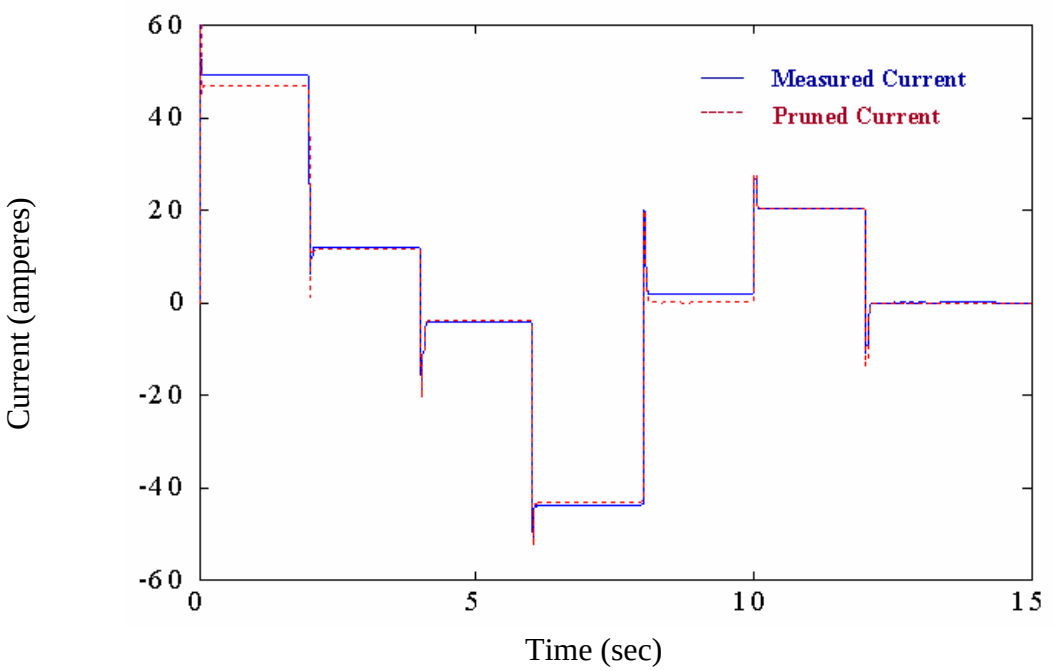

Figure 32 Pruned Current Results.

There are some issues when parameter estimation is performed. One of these issues is the tradeoff, in where to obtain an optimized model, the balance between number of parameters, bias and variance need to be found. Pruning is a nice way to make model reduction. Orthogonal least square is a good pruning technique for reduces the model complexity.

4.6 Mechanical Model With Implicit Viscosity Parameter: Results with noise.

Table 6 shows the estimation results with the gaussian noise of 0.1.

Table 6 Parameter Estimation Results: Without Implicit Viscosity Parameter

Parameter	Real Value	Estimated Value	Error
R_a	7.56	7.5616	0.02%
L_a	0.055	0.055	0.0%
K_a	3.475	3.4741	0.03%
J_m	0.06	0.0564	6.0%

The following figures present results of the model identification. Figure 33 shows the estimated load torque characteristics. As we can see in Figure 33 the performance of the parameters estimation using the radial basis function artificial neural network in the black box for the identification process produces a good performance of the system. Figure 34, shows the estimation error between the real torque values versus the identified torque. In the error in torque figure, can be observed that the greater error its at the ends of the response. It may be seen that the approximation of the respective load characteristics by the artificial neural network is very good inside the nominal operations, but it start to lose performance outside this range.

Figure 35, compares the measured and estimated current, while Figure 36 shows the estimation error for the real current values and the estimated ones. Figure 35, clearly

shows that both graphs are overlay, which means that good fitting performance was obtained by the identification. As shown in Figure 36, the maximum error between the real value current and the results of the current with the estimated parameters was approximately 1.5 ampere, but the mean of the error was very small, near to cero. Basically, the error occurs in the transition of the steps in the current curve.

Figures 37 presents the results of the real velocity values versus the estimated velocity, while Figure 38 presents the error of both velocities. Figure 37 present the performance of the system with the estimate parameters. Figure 38 presents that the velocity error was very small too. So the error in torque estimation does not have a significant effect on the speed estimate. If we compare the results without noise and these results, it possible observed that a good performance of the system was obtained.

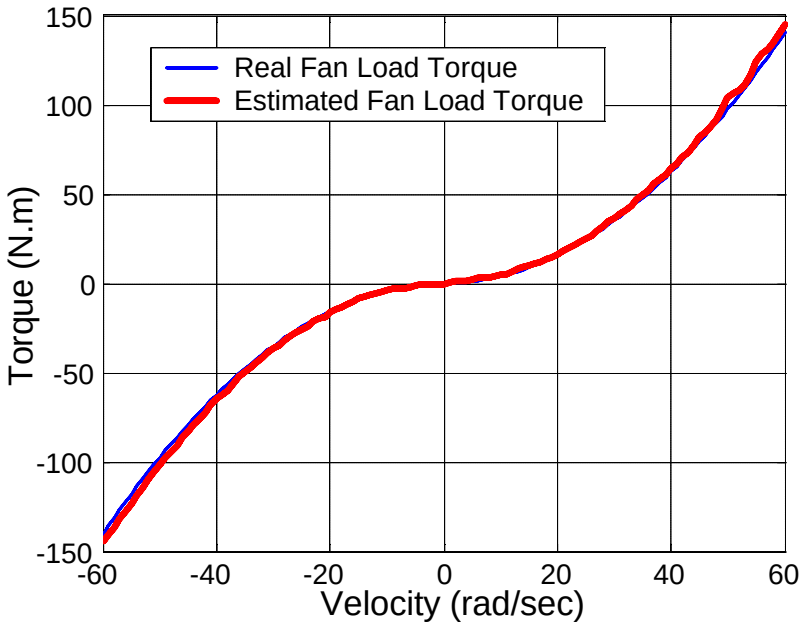

Figure 33 Real Fan Load Torque vs. Estimated Fan Load Torque (Noise System)

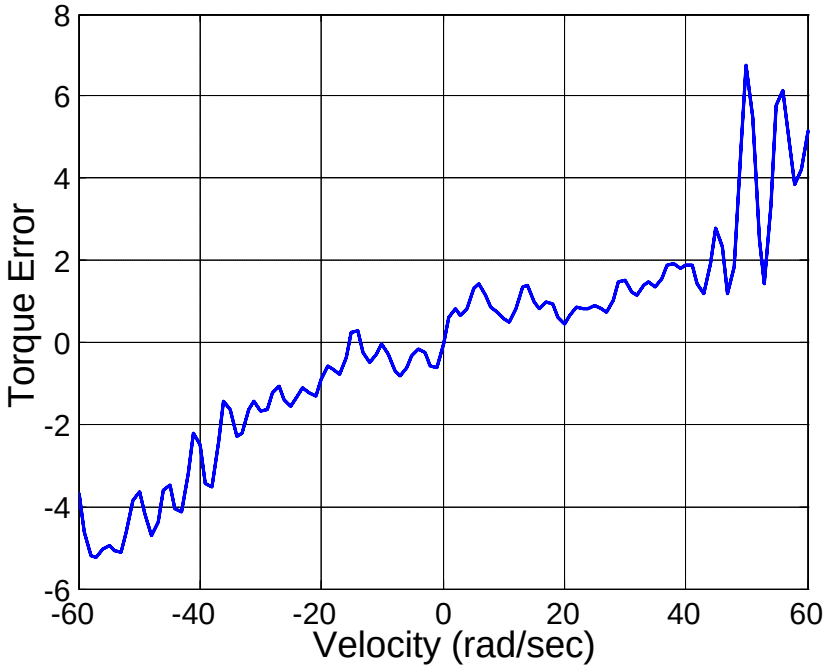

Figure 34 Torque error for the identified model (Noise System)

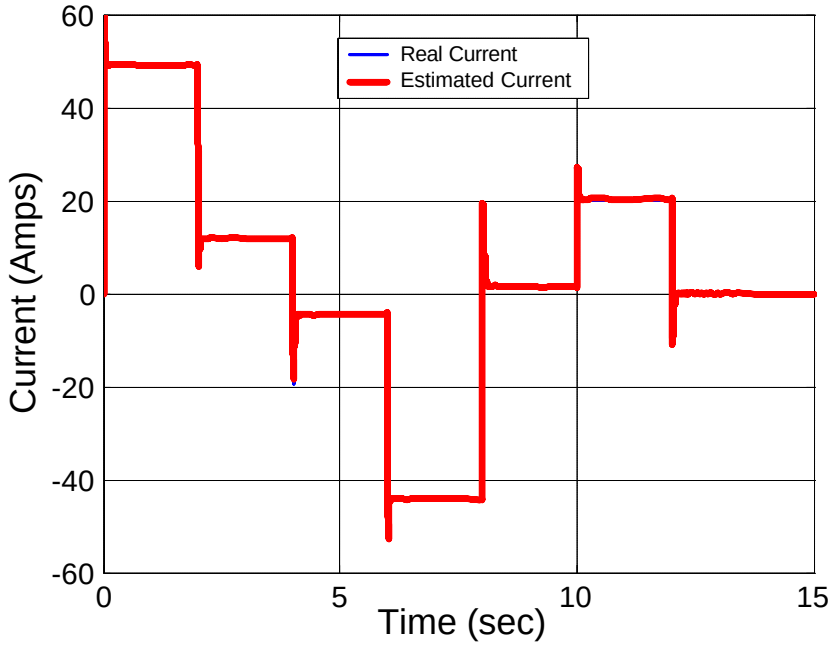

Figure 35 Real Current vs. Estimated Current (Noise System)

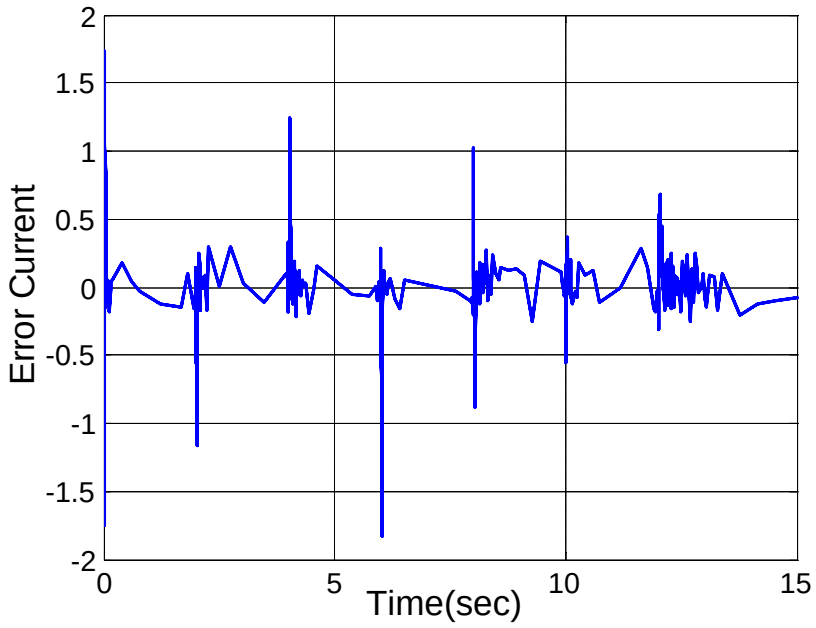

Figure 36 Current error for the identified model (Noise System)

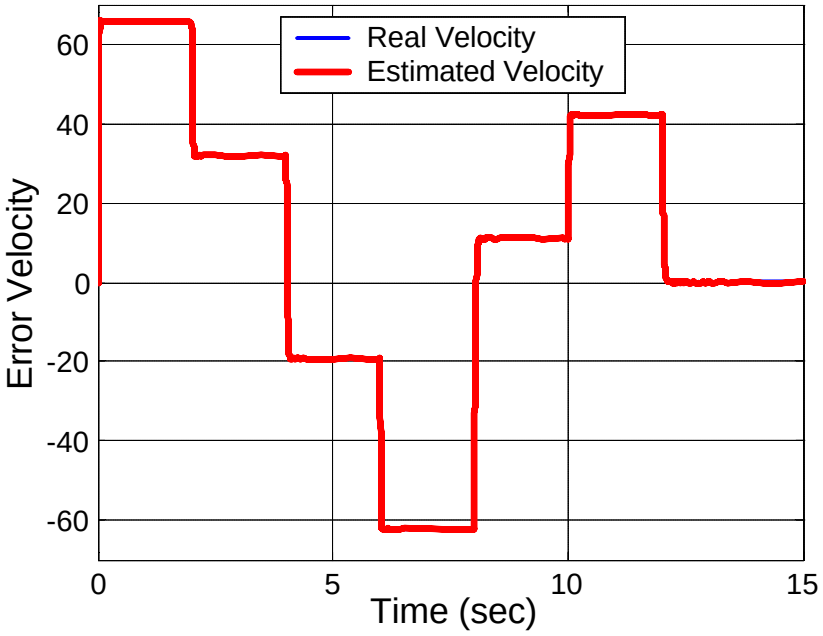

Figure 37 Real Velocity vs. Estimated Velocity (Noise System)

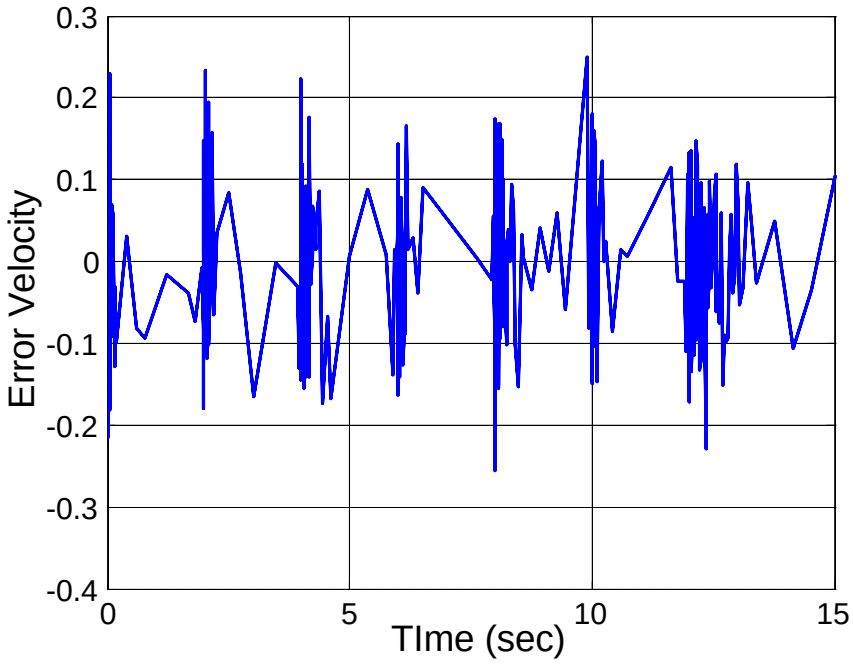

Figure 38 Velocity error for the identified model (Noise System)

For the validation test, a different input armature voltage was used and shown in Figure 27. In the validation process the identified model is fed with the validation voltage and it response is compared to that of the real system.

The results for the current and the velocity outputs for the validation results are presented in Figure 39 respectively. For the Figure 39 (a) it is possible to note that the difference between the results of the current without noise is the part between 12 to 15 seconds. The rest of the plot is overlay in the real system plot.

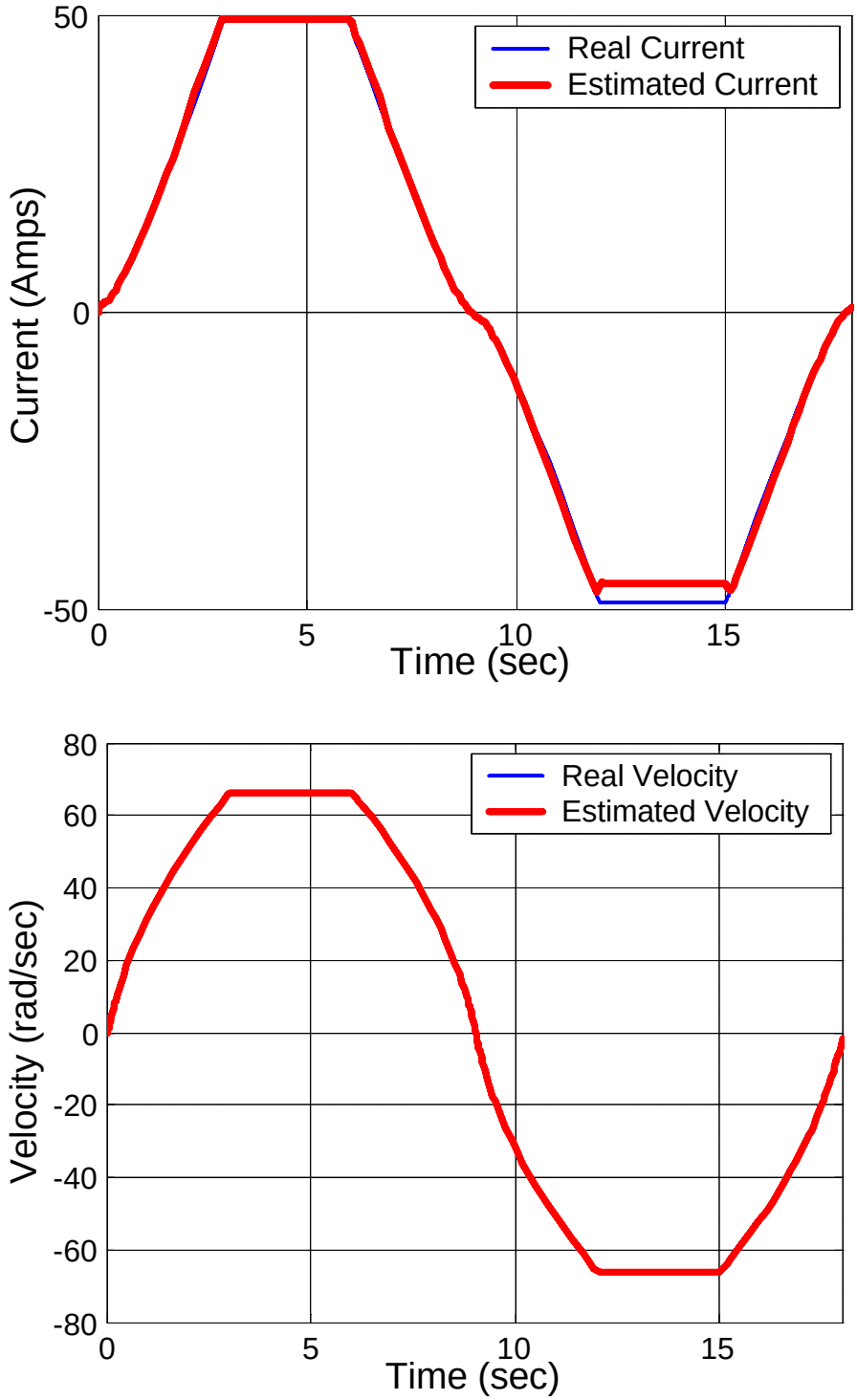

Figure 39 Validation of the System Performance for the case when the viscosity term is not present; system with noise: (a) current response and (b) velocity response

Using orthogonal least squares, the number of nodes was reduced from one hundred twenty-one basis functions to fifty basis functions. The estimates of the physical parameters for the model after model reduction by orthogonal least squares are shown in Table 7. These results are for gaussian noise of 0.1.

Table 7 Parameter estimation results after pruning

Parameter	Real Value	Estimated Value	Error
R_a	7.56	7.5614	0.01%
L_a	0.055	0.055	0.0%
K_a	3.475	3.4738	0.03%
J_m	0.06	0.0561	6.5%

Figure 40 shows the torque results after model reduction. Figure 41 present the performance of the velocity with the pruned model and Figure 42 present the performance of the current after model reduction. The maximum error occurs at the ends of the model. This can be clearly observed in Figure 42, were a difference between original current value and pruned current value is notable.

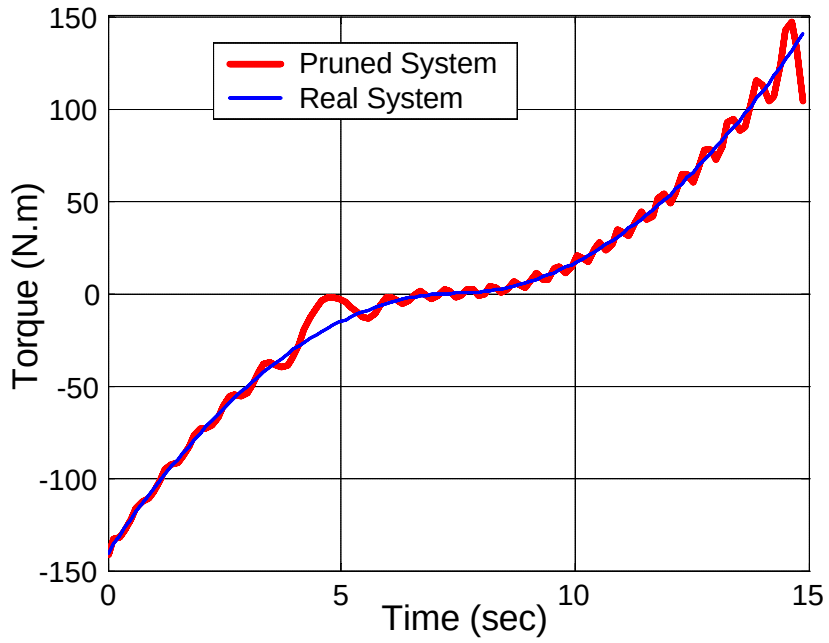

Figure 40 Torque results after pruning (Noise System).

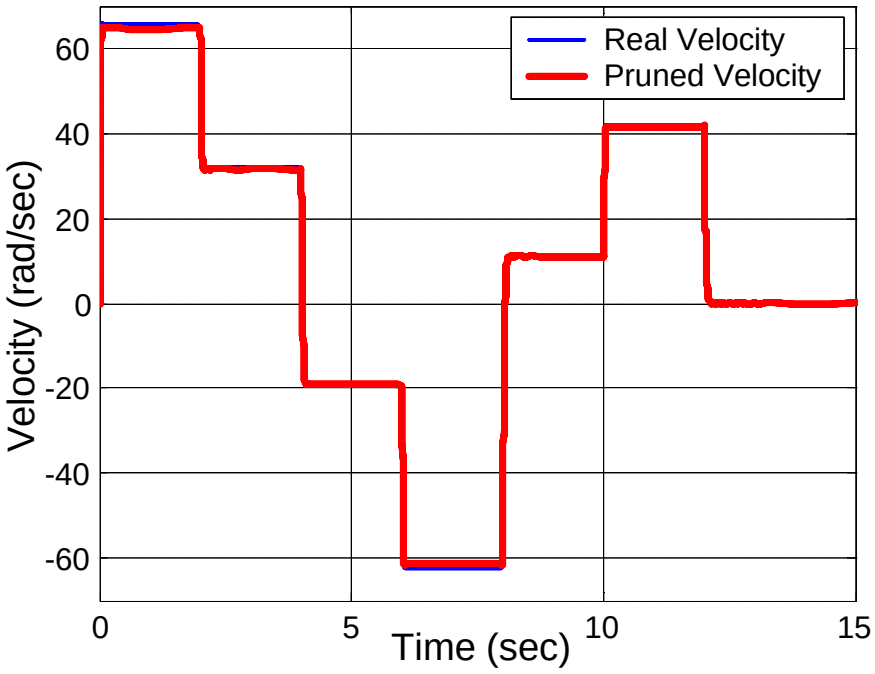

Figure 41 Pruned Velocity Results(Noise System).

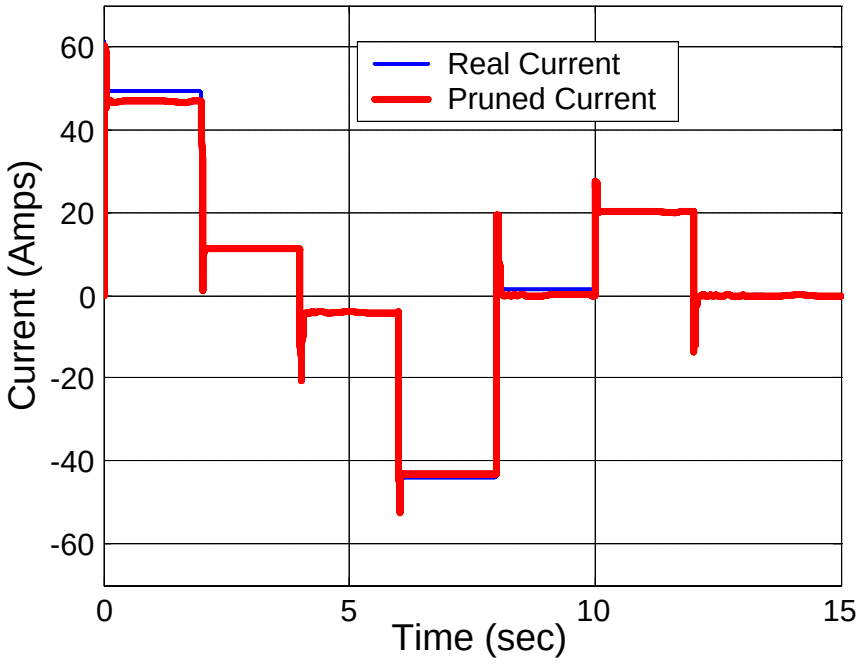

Figure 42 Pruned Current Results (Noise System).

4.7 Conclusions

Simulation results showed that a good identification of electrical parameters and inertia is feasible. Simulation results also showed that viscous damping coefficient is not observable which result in large estimation error, but the model predicted accurately the total mechanical load. We conclude that the best structure for the mechanical model does not have an explicit term for viscous damping. Validation results showed good performance for the identified model. Orthogonal Least Squares is a good method to perform the model complexity reduction, which produce good results with less basis functions. The reduced order model gave reasonable good performance.

Chapter 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The purpose of this work was the study of a modeling methodology that can be used in a self-commissioning scheme for an electric drive. The modeling methodology needed to be flexible enough to allow the drive to be capable of handling a wide range of loads. We also wanted a simple enough method such that it could be adopted by industry. The developed approach meets both constrains.

The two-stage method results in good performance of our parameter estimation results with relatively lower computational requirements that other methods based on multilayer perceptron. The importance of this two-stage method is that it can be used to solve the parameter estimation problem in any type of motor, without losing the effectiveness and low computational time. It is important to point to the fact that the electrical parameters could be estimated with small errors and they do not depend on the mechanical load or the selected mechanical model structure.

Simulations results show potential benefits of the method for modeling and identification of electric drives systems. In general, we achieve good estimation of physical parameters. We can conclude that physical meaning of estimates is still achievable.

The artificial neural networks based on radial basis functions did good approximations of the load characteristics within the training data range and validation data range.

The use of state variable filters to avoid the common problem of differentiation results in acceptable performance for the estimator.

The model validation is necessary to prove the effectiveness of our model. However, we have to take care of the characteristics of the validation data, which need to be inside the same range of training data so the system work in the same range as it was trained for.

Pruning help to obtain models with reduced complexity that performed well.

We can conclude that this methodology is viable if there is enough information regarding the operation of the load.

5.2 Future Work

This research work let a number of questions that should be addressed in future work.

Some future lines of work could be:

- Application of the approach to dynamics loads.
- Incorporation of load identification into self-commissioning drive systems.

- Expand the use of the parameter estimation process, two-stage method, to another types of motors, for example the induction motor.
- Recursive implementation for on-line identification.
- Experimental validation. Using a similar set-up as shown below.

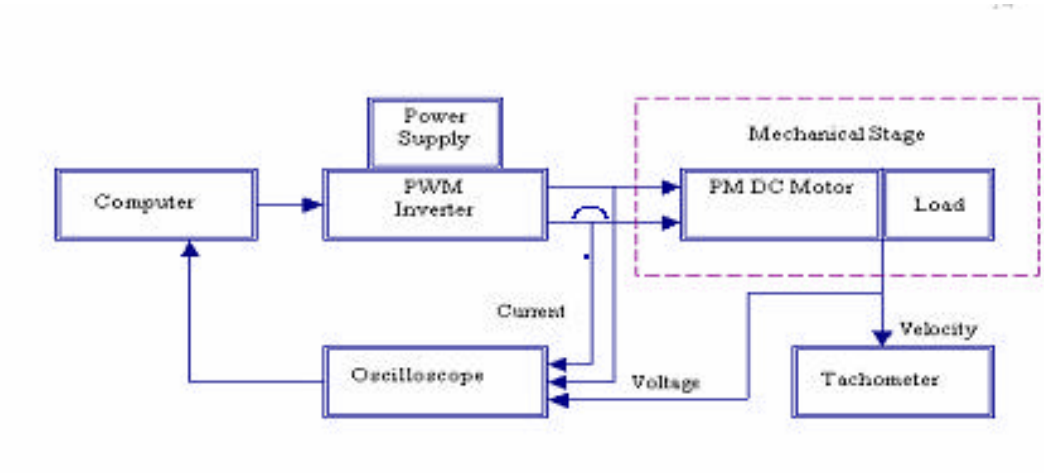

Figure 43 Proposed Experimental Set-up

References

[1] R. Rivera-Sampayo, *Gray-box modeling of electric machines using neural networks;* Master Thesis, University of Puerto Rico, Mayagüez, PR, 2001.

[2] R. Krishnan, *Electric Motor Drives: Modeling, Analysis and Control*; Prentice Hall, New Jersey, 2001

[3] V. Subrahmayan, *Electric Drives: Concepts and applications*; Mc Graw Hill Co., New York, 1994.

[4] S.Beineke, F. Schutte, H. Wertz, H. Grotstollen. *"Comparison of parameter identification schemes for self-commissioning drive control of nonlinear two-mass system". In Proceedings of IEEE, IAS Annual Meeting*, Volume 1, pages 493-500, 1997.

[5] S. Beineke, f. Schutte, J. Groststollen. *"Online identification of nonlinear mechanics using extended kalman filter with basis function networks". Proceedings of the Industrial Electronics Conference*, pages 316-321, 1997.

[6] R. H. A. Hensen, G. Z. Angelis, M. J. G. Vanda Molgraft, A. G. Jager, J. J. Kok; *"Grey-box modeling of friction: an experimental case study"; European Journal of Control*, pp.258-267, 2000.

[7] Y. Báez-Rivera, M. Vélez-Reyes; *"Improved gray box modeling of electric drives using neural networks", North American Power Symposium Proceedings (NAPS)*, pp.182-267, 2003.

[8] K. Tae-Eung, L. Jong-Pil, J. Pyeong Shik, N. Sang Cheon, K. Jung-Hoon, L. Jae Yoon; *"Load characteristics identification using artificial neural network and transient stability analysis"; Proceedings of EMPD'98 International Conference on Energy Management and Power Delivery*, Vol. 1 pp.329-334, 1998

[9] K. J. Astrom, B. Wittenmark; *Computer Controlled Systems Theory and Design*; Prentice Hall, Upper Saddle River, New Jersey, 1997

[10] O. Nelles; *Nonlinear System Identification from Classical Approaches to Neural Networks and Fuzzy Models*; Springer, Germany, 2001

[11] J. Sjöberg, A. Benveniste, Q. Zhang, B. Delyon, L. Ljung, P. Giorennec, A. Juditsky, H. Hjalmarsson; *"Nonlinear black-box modeling in system identification: A unified overview"*; *Automática* vol. 31 No. 12, pp.1691-1724, 1995

[12] T. Bohlin; *"A case study of gray-box modeling identification"*; *Automática* vol. 30 No. 2, pp.307-318, 1994

[13] R. Rico-Martínez, J. S. Anderson, I. G. Kevrekidis; *"Continuous-time nonlinear signal processing: A neural network based approach for gray-box identification"*; *Proceedings of the 1994 IEEE Workshop of Neural Networks for Signal Processing*, pp.596-605, 1994

[14] G. Horvath, R. Dunay; *"Modeling of nonlinear dynamic systems by using neural networks"*; *Proceedings of IEEE International Symposium on Industrial Electronics*, Vol. 1, 1996

[15] P. D. Wasserman; *Advance Method in Neural Computing*; Van Nostrand Reinhold, New York, 1993

[16] K.S. Narendra, E.G. Kraft, L.H. Ungar, *Workshop on Neural Networks in Control Systems, American Control Conference*, Boston, Massachusetts 1991.

[17] R. Johansson, *System Modeling Identification*; Prentice Hall, New Jersey, 1993.

[18] S. Haykin, *Neural Networks: A comprehensive Foundation*, Macmillan Publishing Co., New Jersey, 1994.

[19] C. Goutte; *"On the use of a pruning prior for neural networks"*; *Proceedings of the 1996 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing*, 1996

[20] R. Nayak, *A methodology for rule extraction from artificial neural networks*., PhD Dissertation, University of Technology, Australia, 1999.

[21] E.Ho, P.C. Sen, *"Control dynamics of speed drive systems using sliding mode controllers with integral compensation." IEEE Industry Applications*, 1991 pags. 883-892.

[22] X.Z. Gao, S. Valiviita, S.J. Ovaska, J.Q. Zhang, *"Neural networks-based approach to the acquisition of acceleration from noisy velocity signal", IEEE Instrumentation and Measurement Conference*. 1998 pags. 935-940

[23] Geiger, G. *"Monitoring of an electrical driven pump using continuous-time parameter estimation methods", IFAC Identification and System Parameter Estimation Conference*, Washington D.C. 1982.

[24] J. Tafur, *Adaptive feedback linearization controller for a DC Shunt Motor;* Master Thesis, University of Puerto Rico, Mayagüez, PR, 1995.

Appendix

APPENDIX 1 - Translations from Neural Network into System Identification Language

The terminology used in this work follows the standard system identification and optimization literature rather than the neural network language. The following expressions are often used in the literature related to this work [10].

Table 8 Translation from neural network into system identification.

Neural Network Terminology	System Identification Terminology
Mapping or Approximation	Regression
Classification	Discriminant analysis
Neural Network	Model
Neuron	Basis function
Weight	Parameter
Bias or Threshold	Offset or intercept
Hidden Layer	Set of basis functions
Input Layer	Set of inputs
Input	Independent variable
Output	Predicted value
Error	Residual
Learning or Training	Estimation or optimization
Generalization	Interpolation or extrapolation
Over fitting or Over training	High Variance error
Under fitting or Under training	High Bias Error
Error bar	Confidence Interval
Online Learning	Sample Adaptation
Offline Learning	Batch Adaptation

APPENDIX 2 - Pruning by Singular Value decomposition.

The simplicity of the two-stage method is due to the fact that by fixing the radial basis function centers and variances, the estimation of the radial basis function amplitudes $\mathbf{a}_i$ is a linear problem. Here we investigate full nonlinear estimation where the centers and variances are also estimated. Nonlinear estimation and pruning allows for models with significantly smaller dimension than those obtained by orthogonal least squares.

Fan Load Static Model

After the model structure is performance we study the behavior of the fan load system. The physical parameters estimates and radial basis function network weights are computed by least square and nonlinear least square for the linear and nonlinear parameters respectively.

This model implemented produce a high performance in the estimates with a low estimation error; as we can see in Figures 44 our system has a good performance in the estimate of the fan load curve. Figure 45 present the error between real torque values and estimated torque value in which we have a lower estimation error value, which indicates our good performance.

A disadvantage of this model was the large number of parameters, which may introduce a parameter redundancy, over-fitting and as a result ill-conditioned system. To

avoid over-fitting and ill-conditioned of the system modeling we used singular value decomposition for pruning of the system.

In pruning method, to examine the parameter redundancy we examine the Jacobian matrix, which is the matrix of the parameters sensitivity. The rank of this matrix gives an idea of the number of parameters and basis functions that can be estimated [1]. Figure 46 shows a plot of the Jacobian singular values of the fan load estimation. Since there are 93 parameters (31 radial basis functions) in our model, in which the matrix has many small singular values, this is indicative of over fitting and redundant parameters, which results in ill - conditioning.

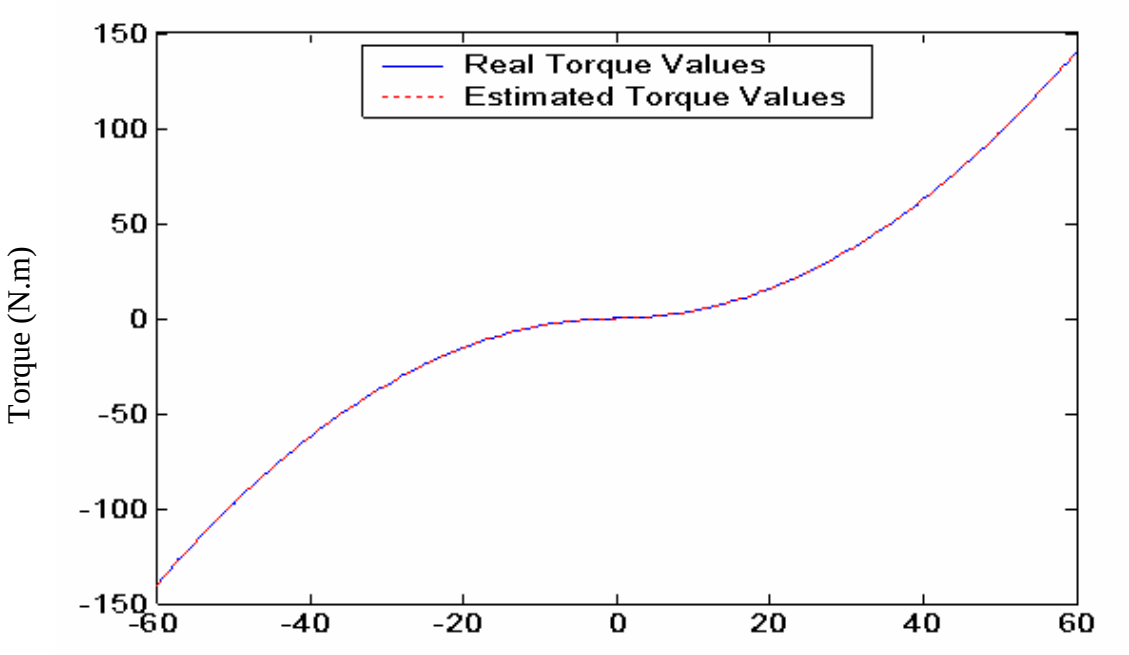

Figure 44 Real Torque vs. Estimated Torque of Fan Load

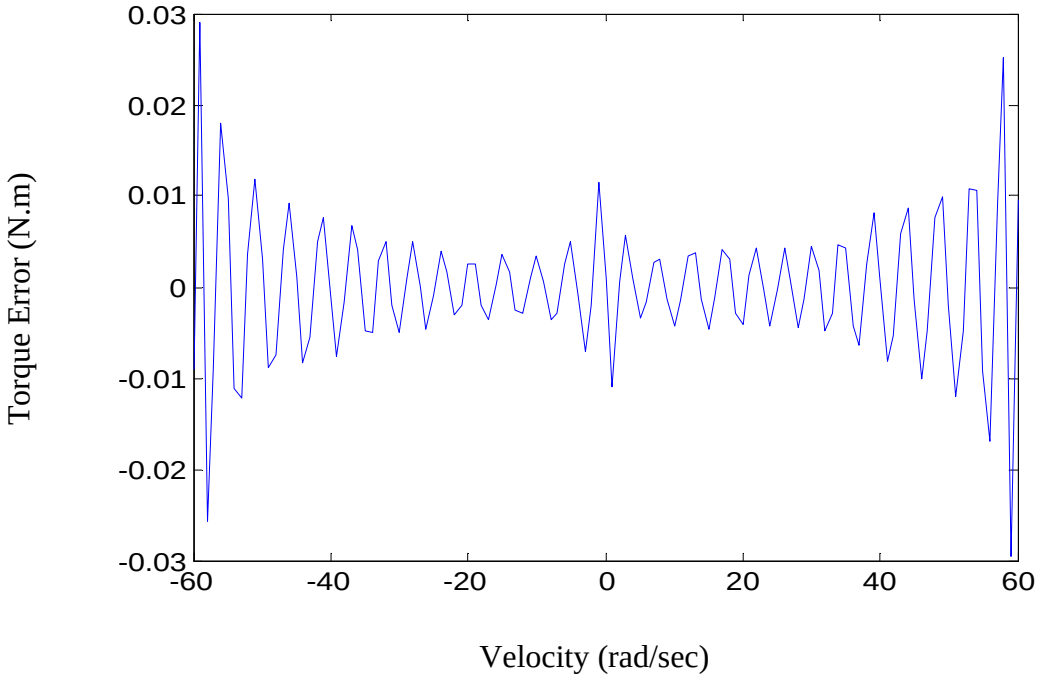

Figure 45 Error Between Real Torque and Estimated Torque.

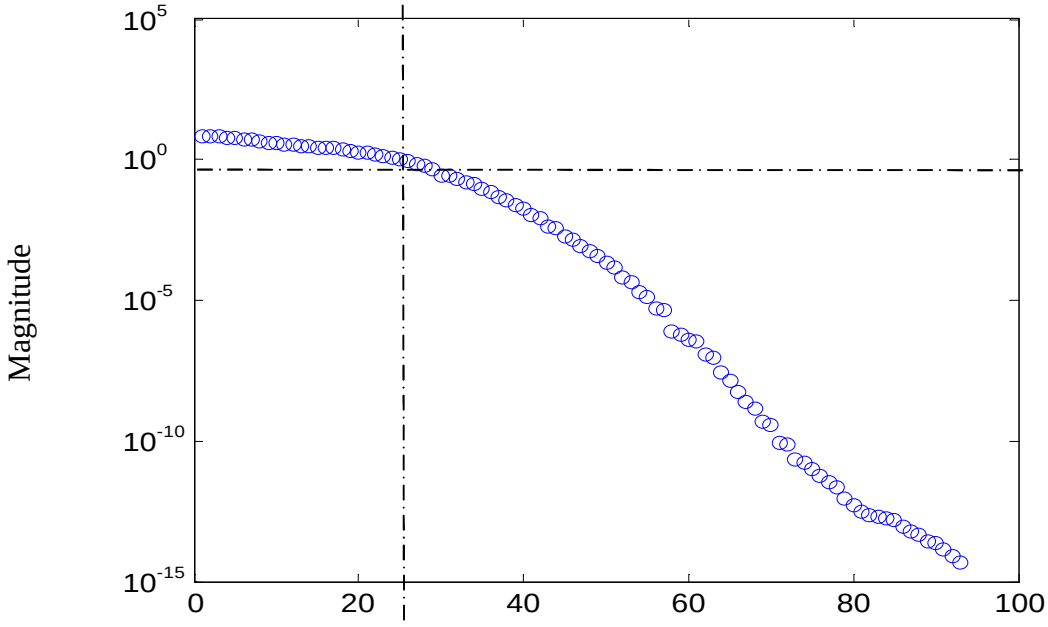

Figure 46 Singular Values of Fan Load with full parameters

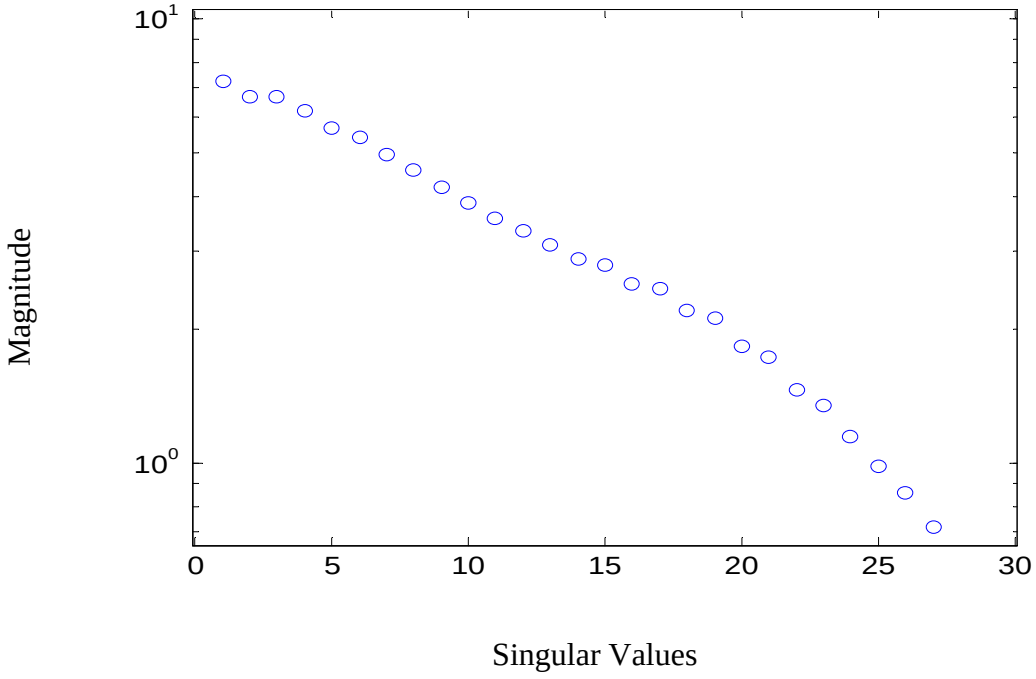

Figure 47 Singular Value Decomposition after parameter reduction.

As we can see in Figure 46 we have a threshold marked. This value is obtained by $\left(\frac{s_{\min}}{s_{\max}}\right)$ of S matrix. In this case this threshold is approximated 0.44 in magnitude, which means approximately 27 parameters, or 9 radial basis functions (Figure 47). This was our break point, this value was used as a clue of how many radial basis functions we need to make a good estimate of the friction load without redundancy, over-fitting and eliminating ill-conditioning results.

After the parameter reduction from 33 radial basis functions to 9 radial basis functions the parameters were estimated. The results of this estimation are presented in Figure 48, which presents the results for the fan load with eight basis functions. As we can see, we obtain a good fitting.

Figure 49 presented the error between the real fan load values and the estimated fan load values. We can compare the error in Figure 49 and Figure 45 that the error increase, but at least is too small to make a big difference in the fan load parameters estimation and by the way in the system performance. The big difference occurs when comparing nonlinear method error with the two-stage error. The error in the nonlinear method is less than the other, however the computational requirements for the nonlinear method are more than the requirements of the two-stage.

The use of the singular values decomposition method for pruning our model produces excellent results. If we compare the simulation results of both methods (nonlinear and two-stage) we obtain good performances. However, in the use of singular value decomposition we are dealing with nonlinear least squares, which make our identification process, time consuming. For this reason we think that in practice the two-stage method will find more acceptance.

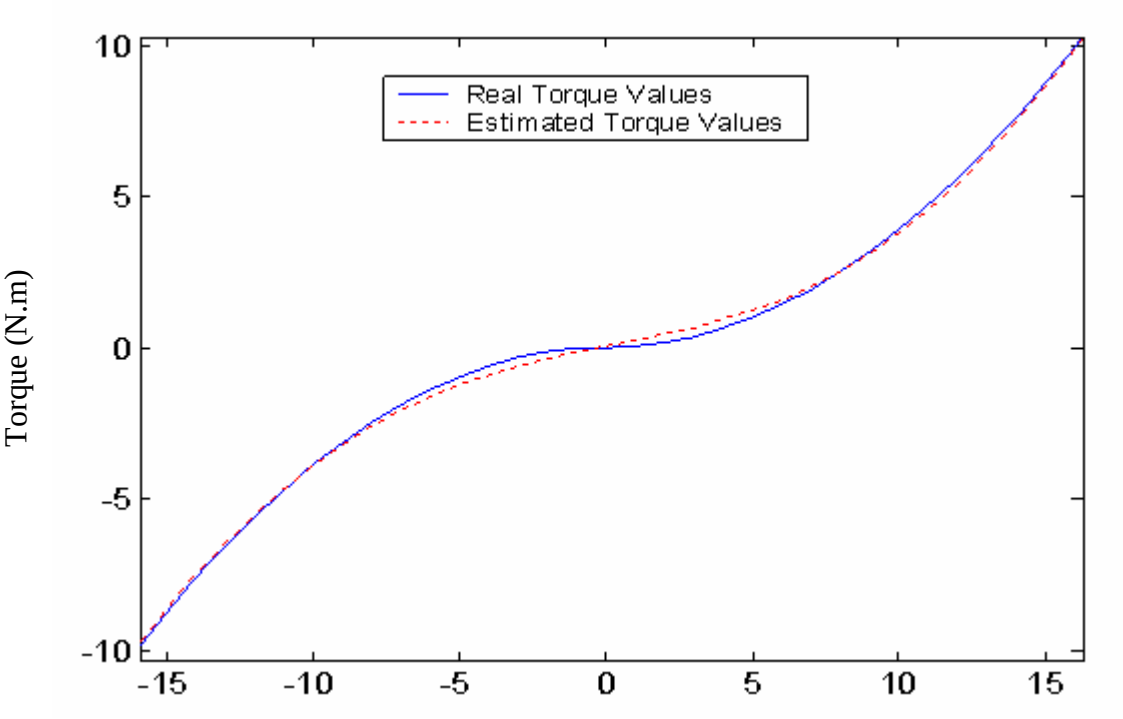

Figure 48 Real Torque vs. Estimated Torque after Pruning.

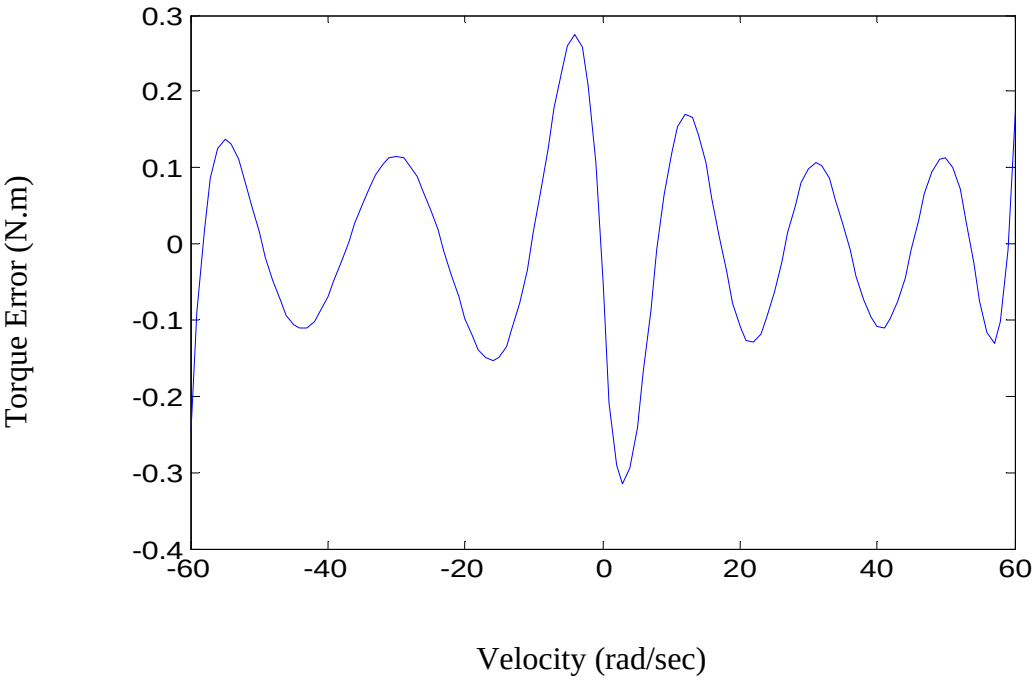

Figure 49 Error Between Real Torque and Estimated Torque after Pruning.

Friction Load Static Model

With this model we follow the same methodology that we used for the fan load identification. We model the system with all the parameters then make the reduction by singular value decomposition for pruning the system.

In Figure 50, the simulation results for the identification of the friction load parameters are presented. As Figure 51 shows, the error between estimated and actual torque is small.

Here we have the same problem that with the fan load model, due to the large number of parameters. As shown in Figure 52, there is like 93 parameters in the identification process. Based on the singular values, the number of basis functions was reduced from thirty-one basis functions to eleven as shown in Figure 53.

The parameters of the reduced order model were estimated using nonlinear least squares. As shown in Figure 54, the estimated torque was close to the real torque and the error was small. See Figure 55.

Comparing the results for the nonlinear method and the two-stage linear least squares method we can conclude that the best results are from the nonlinear method. However, the complexity of the algorithm and the time that consume are two biggest drawbacks of this method. The two-stage method was a simple one, with an easy algorithm, no time consuming and at the same time produces good results.

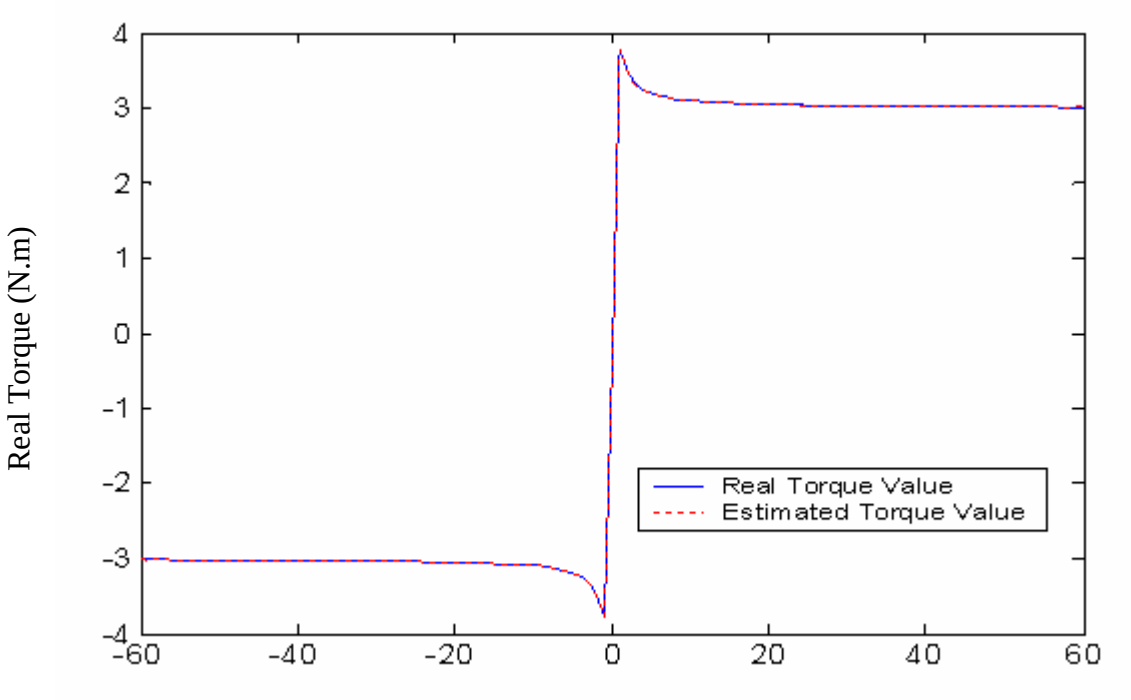

Figure 50 Real Torque vs. Estimated Torque of Friction Load.

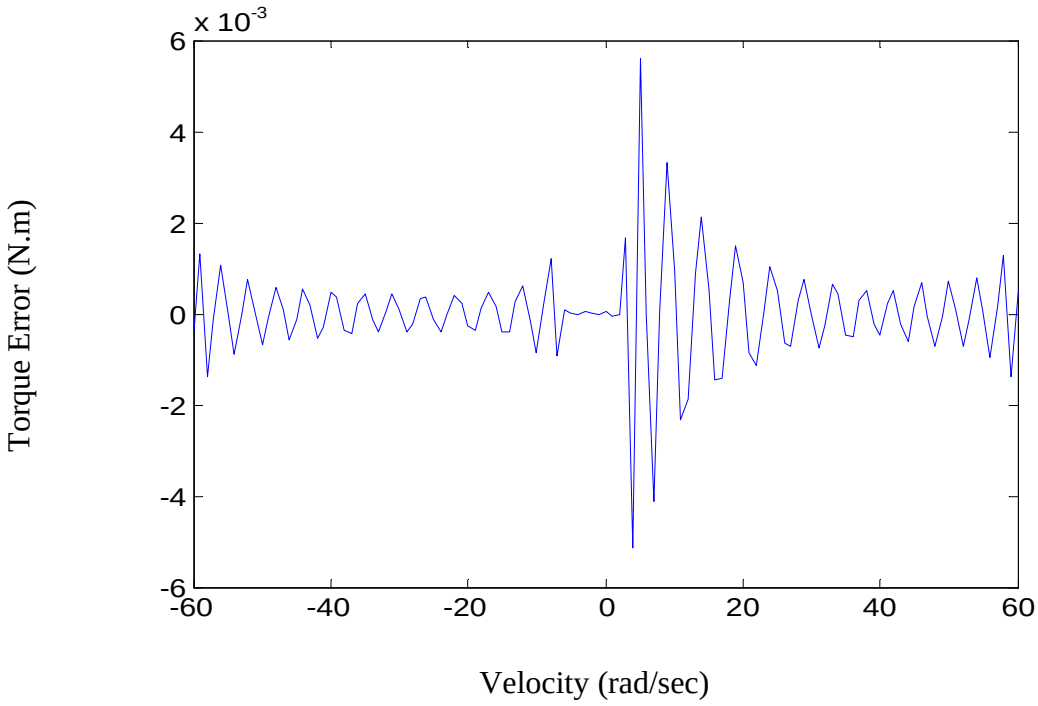

Figure 51 Error Between Real Torque and Estimated Torque of Friction Load.

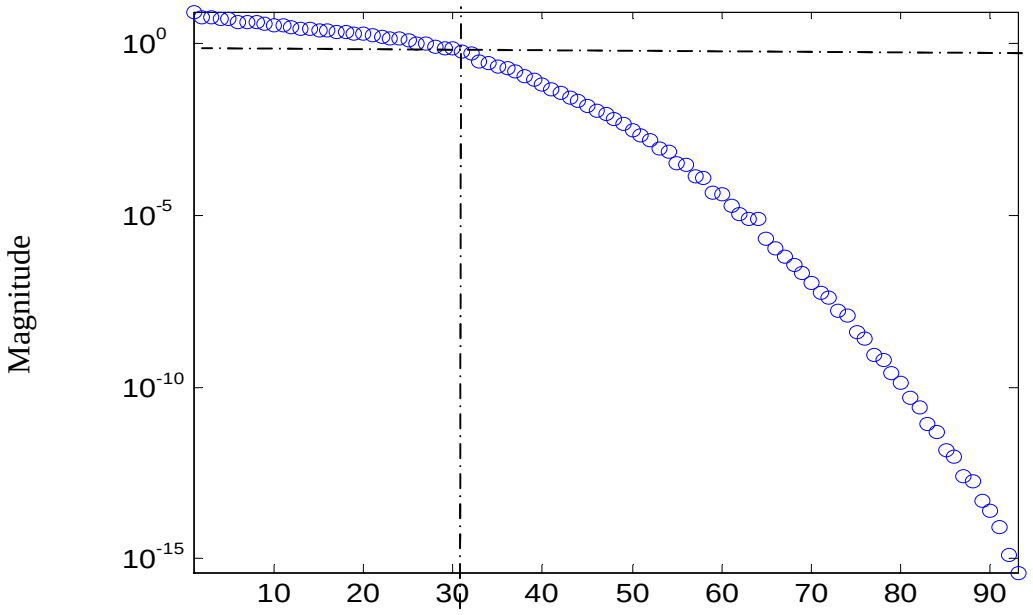

Figure 52 Singular Values for Friction Load with full parameters.

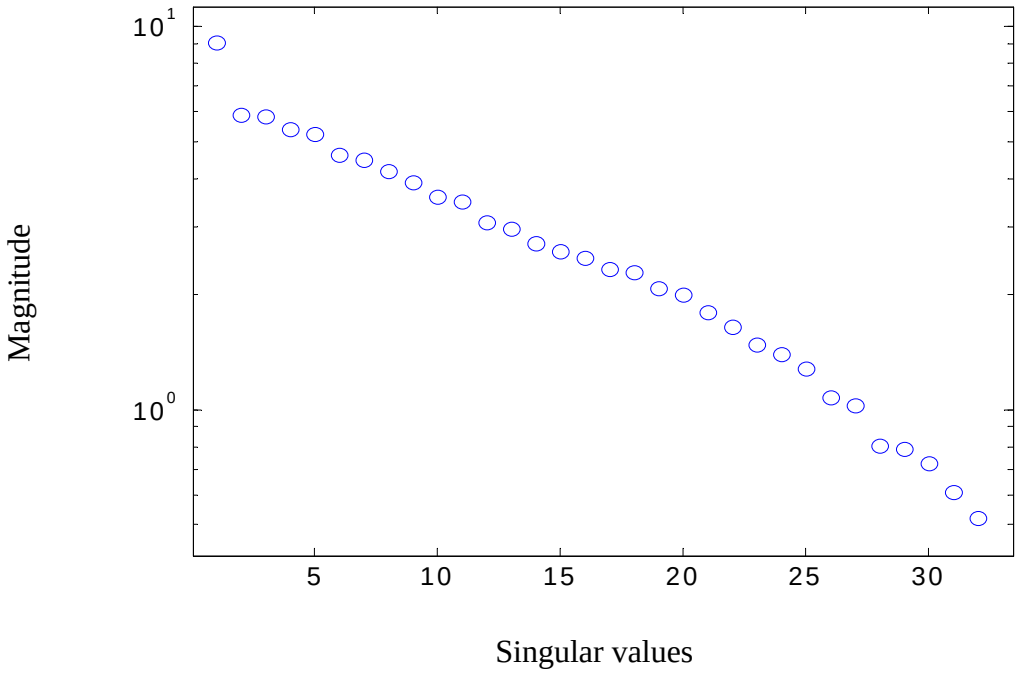

Figure 53 Singular Values after Pruning.

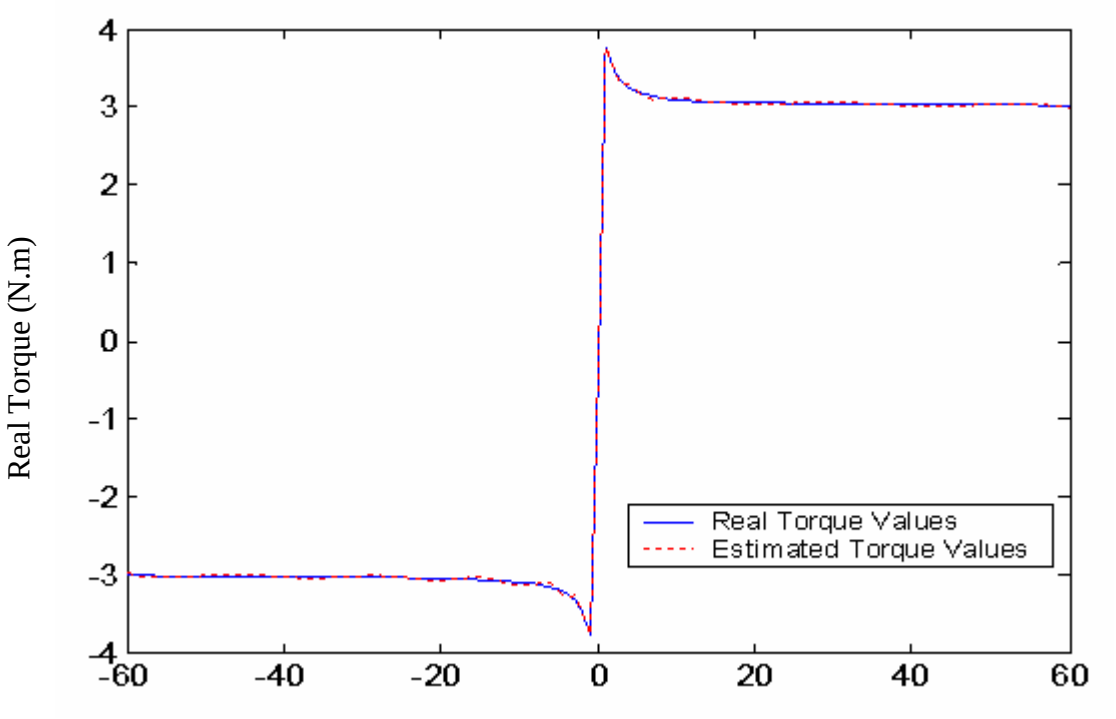

Figure 54 Real Torque vs. Estimated Torque of Friction Load after Pruning.

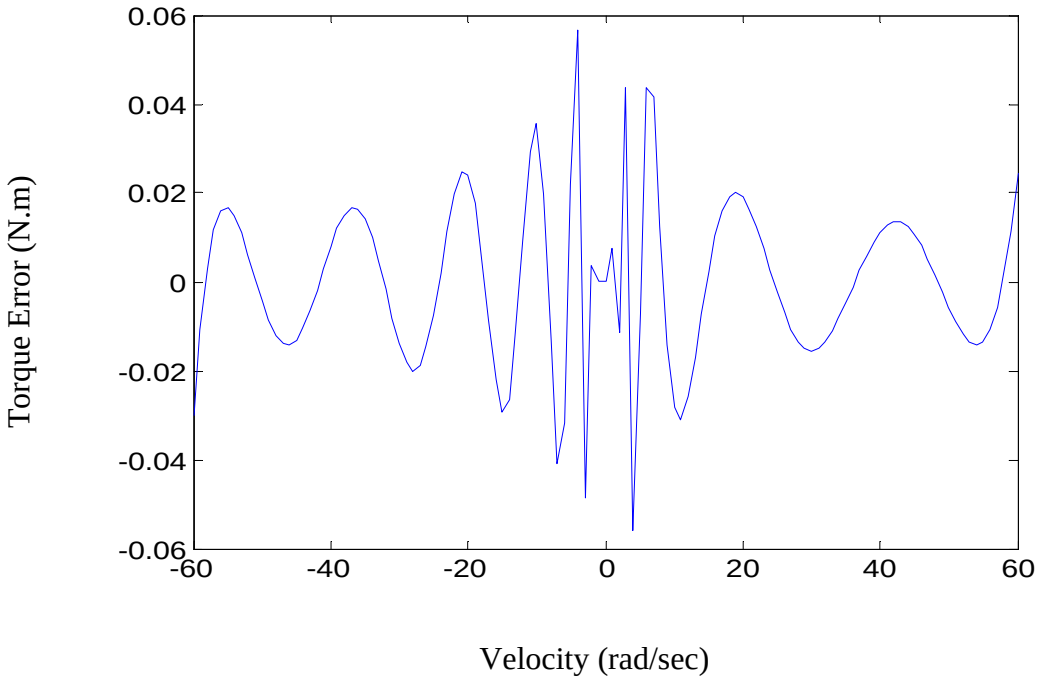

Figure 55 Error Between Real Torque and Estimated Torque after Pruning.

APPENDIX 3 – Experimental set-up.

The control goal is to track a reference speed under unknown parameters and load disturbances. The stability of internal dynamics is assured by analysis of zero dynamics. A nonlinear observer for load torque with linear error dynamics was also developed to give a load torque estimate to the adaptive controller.

Motor control applications make extensive use of DC motor due to their relative simplicity and achieved high performance with precise control. Their outstanding advantages lie in flexibility and versatility in positioned systems and speed regulation.

The control goal is that the motor speed should track the speed reference under unknown parameter and load disturbances.

Parameter Estimation of the physical motor

In this section, experimental methods are presented for the determination of the parameters of the DC motor model.

The resistance R_a was measured by the voltmeter-ammeter method [24]. R_a can be measured by simply measuring the resistance across the motor terminals. Turn off the power and calculate >4, then take the average. The armature constant K_a was determined by measuring the armature terminal voltage in open circuit when the motor is operated as a DC generator. The self-inductance L_a was measured by the transient method [24]. The voltage across a resistor in series with the winding that is proportional to the current response is recorded on a storage oscilloscope. The self inductance was

calculated from the measurement the time constant (t) assuming that the setting time (t_s) and the time constant are related by t_s=4t, then the self inductance is given by $L_a = R_e \frac{t_s}{4}$ where R_e is the equivalent resistance of the RL circuit.

The friction coefficient can be determined by using de mechanical equation presented in chapter 3, equation (3.3) in steady state,

where $\frac{dw}{dt} = 0 \quad \frac{di}{dt} = 0$ and $T_L = 0 \;\; T_m = Bw$, $T_{em} = k_b i_a$, $k_b i_a = Bw \therefore B = \frac{k_b i_a}{w}$

The motor rotor inertia (J_m) is determined by the retardation test speed versus time characteristic due to switching off the motor after steady state is reached. Using equation (3.3) and considering $T_L = 0$ and $T_{em} = 0$, results in $J \frac{dw}{dt} + Bw(t) = 0$ and the solution of the above linear differential equation give

$$w(t) = w(0)\exp\left(-Bt/J\right) = w(0)\exp\left(-t/t\right)$$

where $t = \frac{J}{B}$ is the mechanical time constant of the motor. The ? vs. time curve was registered.

Drive system Experimental set-up

The block diagram of the experimental set-up consists of the control plant, the DC motor, the analog and digital I/O board and the Driver (PWM).

The amplifier to drive the DC motor is a Copley 421 PWM servo amplifier, which provides a 5Amperes continuous, 10Amperes peak at switching frequency of 25kHz. The

servo amplifier is configured to work as a dc voltage amplifier with gain of 10 as shown in Figure 56. Since the voltage applied to the motor is only positive, the positive from 0 to 10V is used and the output voltage of the DC amplifier range from 0Volts to 100Volts average. The power input to the driver is 135Volts at 6Amperes.

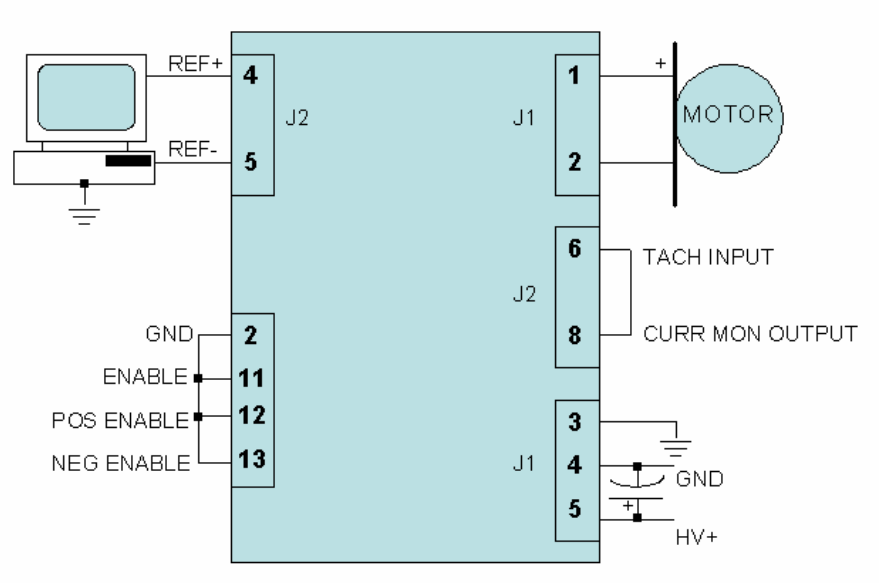

Figure 56 Servo amplifier configuration