

GRAY BOX MODELING OF ELECTRIC DRIVES USING RADIAL BASIS FUNCTIONS: AN EXPERIMENTAL CASE

by

Angel A. Aquino Lugo

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
2006

Approved by:

Efraín O'Neill Carrillo, PhD
Member, Graduate Committee

Date

Agustín Irizarry Rivera, PhD
Member, Graduate Committee

Date

Miguel Vélez Reyes, PhD
President, Graduate Committee

Date

Francisco Maldonado Fortunet, PhD
Representative of Graduate Studies

Date

Isidoro Couvertier, PhD
Chairperson of the Department

Date

ABSTRACT

Modeling of electric drives coupled to complex mechanical loads may be a challenging task. Since the electric drives are commonly used in the industry, for many applications is desirable to have the capability of self-tuning the controller parameters to drive different mechanical load. A possible solution consists of correctly identify the drive and the mechanical load. For that reason, gray-box modeling using neural networks is presented as a possible solution for the identification of the mechanical loads and the drive system. In the proposed gray-box modeling, the drive system is divided into the known part governed by the physical laws, which in our case was the electrical subsystem, and an unknown part, which in our case was the mechanical subsystem. This type of approach is known as a two stage identification process. At each stage, the parameters are estimated using the method of linear and recursive linear least squares. To validate the effectiveness of this approach, simulations and experiments were perform and their results are presented in this work.

RESUMEN

El modelar los sistemas de accionamiento eléctricos acoplados a cargas mecánicas complejas puede ser una tarea muy retadora. Ya que los sistemas de accionamientos eléctricos son comúnmente usados en la industria; para muchas aplicaciones es deseable poder tener la capacidad de poder auto ajustar los parámetros del controlador para así poder manejar diferentes cargas mecánicas. Una posible solución consiste en identificar correctamente el sistema de accionamiento eléctrico y la carga mecánica. Por esa razón es que el modelo de caja gris usando redes neurales es presentado como una posible solución para la identificación de cargas mecánicas y de sistemas de accionamientos eléctricos. En el propuesto modelo de cajas grises, el sistema de accionamiento eléctrico es dividido en una parte conocida que es gobernada por las leyes físicas, que en nuestro caso era el subsistema eléctrico, y en una parte desconocida, que en nuestro caso era el subsistema mecánico. A este tipo de enfoque se le conoce como el método de identificación en dos etapas. En cada etapa, los parámetros son estimados usando el método de los cuadrados mínimos en su forma lineal y recursiva. Para validar la efectividad de este enfoque se realizaron simulaciones y experimentos y sus resultados son presentados en este trabajo.

ACKNOWLEDGEMENTS

First, I would like to thank Professor Miguel Vélez Reyes for all his support and guidance during the time I had work with him. Thanks for letting me work under your supervision and for being one of your students.

I would like to thank Dr. Agustín Irizarry Rivera and Dr. Efraín O’Neill Carrillo for serving as members of my graduate committee.

I want to thanks my family, for always giving support and always believe in me no matter what I do with my life. Thanks for always giving me support and cheers, especially in difficult situations.

I would like to thank my graduate fellow students, for always giving me reason to laugh and to hang out at the Electrical and Computer Graduate Students Office when I had stress.

I would like to thank my girlfriend Linda, for always giving me support, for always giving me love. For supporting me and understanding me when I was working late in the lab, especially during the time mounting the test bed implementation. You know that the person who I am right now is mostly in part because of you and you don’t know how thankful to God I am because you entered my life at the precise moment I needed you. I love you.

Finally, thanks to all of the people who directly or indirectly had help through my graduate period because the little things that someone does one day if you see it in a totality had influenced my life; so to all of you thanks.

This work was supported primarily by the ERC Program of the National Science Foundation under Award Number EEC-9731677. Partial support also received from NSF Grant ECS-9702860.

TABLE OF CONTENTS

| | |
|---|-----------|
| Abstract | ii |
| Resumen | iii |
| Acknowledgements | iv |
| Table of Content | v |
| List of Tables | viii |
| List of Figures | ix |
| Chapter 1 | 1 |
| Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Objectives | 2 |
| 1.3 Contribution | 2 |
| 1.4 Thesis Outline | 3 |
| Chapter 2 | 4 |
| Background | 4 |
| 2.1 Concepts in Motor Drives | 4 |
| 2.1.1 DC Motor Drives | 5 |
| 2.1.2 Mechanical Load Identification of Electric Drives | 7 |
| 2.2 System Identification | 9 |
| 2.3 Self Commissioning of Electric Drives | 11 |
| 2.4 Radial Basis Function | 13 |
| 2.5 Parameter Estimation | 18 |
| 2.5.1 Linear Least Squares | 18 |
| 2.5.2 Recursive Linear Least Squares | 20 |
| 2.6 State Variables Filters | 22 |
| Chapter 3 | 24 |
| Two Stage Identification | 24 |

| | | |
|------------------|--|-----------|
| 3.1 | Gray-Box Model | 24 |
| 3.2 | Linear Regression Model for Electrical and Mechanical Subsystems | 25 |
| 3.3 | Recursive Algorithm for Electrical and Mechanical Subsystems | 28 |
| 3.4 | Identification Process Algorithm | 30 |
| Chapter 4 | | 32 |
| | Simulation Results | 32 |
| 4.1 | Initialization | 32 |
| 4.2 | Simulations Results for Batch Two-Stage Method | 33 |
| 4.3 | Validation Results for the Batch Two-Stage Method | 38 |
| 4.4 | Simulation Results for Batch Two-Stage Method: Noise Case | 42 |
| 4.5 | Validation Results for the Batch Two-Stage Method: Noise Case | 47 |
| 4.6 | Simulations Results for Recursive Two Stage Method | 50 |
| 4.7 | Validation Results for Recursive Two-Stage Method | 56 |
| 4.8 | Simulation Results Recursive Two-Stage Method: Noise Case | 59 |
| 4.9 | Validation Results Recursive Two-Stage Method: Noise Case | 66 |
| Chapter 5 | | 70 |
| | Experimental Setup | 70 |
| 5.1 | Experimental Setup Overview | 70 |
| 5.1.1 | DSPACE Data Acquisition Board | 71 |
| 5.1.2 | Signal Conditioning for Data Acquisition | 73 |
| 5.2 | Permanent Magnet DC Motor | 79 |
| 5.3 | Mechanical Load | 81 |
| 5.4 | Experimental Initialization | 82 |
| 5.5 | Experimental Results from Batch Two-Stage Method | 84 |
| 5.6 | Validation Results from the Batch Two-Stage Method | 89 |
| 5.7 | Experimental Results for the Recursive Two-Stage Method | 92 |
| 5.8 | Validation Results for Recursive Two-Stage Method Experiment | 99 |
| 5.9 | Experimental Results Batch Two-Stage Method with Fixed R_a | 102 |

| | | |
|------------------|---|-----|
| 5.10 | Validation Results Batch Two-Stage Method with Fixed R_a | 109 |
| 5.11 | Results for the Recursive Two-Stage Method with Fixed R_a | 111 |
| 5.12 | Validation Results for Recursive for the Two-Stage Method with Fixed R_a | 118 |
| Chapter 6 | | 121 |
| | Feedback Linearization | 121 |
| 6.1 | Feedback Linearization Theory | 121 |
| 6.1.1 | Lie Derivatives Notation | 121 |
| 6.1.2 | Feedback Linearization for Nonlinear Systems | 123 |
| 6.1.3 | Feedback Linearization for Gray-Box Modeling of Electric Drives | 126 |
| 6.2 | Feedback Linearization Simulation Results | 132 |
| Chapter 7 | | 137 |
| | Conclusions and Future Work | 137 |
| 7.1 | Conclusions | 137 |
| 7.2 | Future Work | 139 |
| References | | 140 |
| Appendix | | 142 |
| | Appendix A - Matlab File for the Experiment Batch Least Square Case | 142 |
| | Appendix B – Matlab File for the Experiment Recursive Least Square Case | 147 |

LIST OF TABLES

| | | |
|-----------|---|-----|
| Table 2.1 | Parameter of the DC Motor Drive | 7 |
| Table 2.2 | Case Study Load Torque | 7 |
| Table 4.1 | Parameter Estimates from Simulation for the Batch Two Stage Method .. | 33 |
| Table 4.2 | Parameter Estimates from Simulation for the Batch Two Stage Method: Noise Case | 42 |
| Table 4.3 | Parameter Estimates from for Recursive Two Stage Method | 50 |
| Table 4.4 | Parameter Estimates from for Recursive Two Stage Method: Noise Case .. | 59 |
| Table 5.1 | Parameter Estimates from Experiment for the Batch Two Stage Method .. | 84 |
| Table 5.2 | Parameter Estimates from Experiment for Recursive Two Stage Method .. | 92 |
| Table 5.3 | Parameter Estimates from Experiment for the Batch Two Stage Method with Fixed R_a | 104 |
| Table 5.4 | Parameter Estimates from Experiment for the Recursive Two Stage Method with Fixed R_a | 112 |

LIST OF FIGURES

| | | |
|-------------|--|----|
| Figure 2.1 | Electric Drive System | 4 |
| Figure 2.2 | Schematic for a DC Motor and Load System | 6 |
| Figure 2.3 | Actual Motor Drive System | 15 |
| Figure 2.4 | Gray-Box Model Structure | 15 |
| Figure 2.5 | Structure of a Radial Basis Function Neural Network | 16 |
| Figure 2.6 | First Order State Variable Filter | 23 |
| Figure 3.1 | Two Stage Algorithm Summary for both Identification Cases | 28 |
| Figure 3.2 | Diagram for the Identification Process Algorithm | 31 |
| Figure 4.1 | Input Voltage used for Training the Model | 33 |
| Figure 4.2 | Real and Estimated Fan Load Torque Characteristic | 34 |
| Figure 4.3 | Estimated and Real Fan Load Torque Estimation Error | 35 |
| Figure 4.4 | Real and Estimated Current | 36 |
| Figure 4.5 | Real and Estimated Current Error | 36 |
| Figure 4.6 | Real and Estimated Speed | 37 |
| Figure 4.7 | Real and Estimated Error | 38 |
| Figure 4.8 | Input Armature Voltage used for Validation | 39 |
| Figure 4.9 | Real and Estimated Current Validation from Simulations | 40 |
| Figure 4.10 | Current Estimation Error: Validation | 40 |
| Figure 4.11 | Real and Estimated Speed Validation from Simulation | 41 |
| Figure 4.12 | Validation Error from Real and Estimated Speed | 41 |
| Figure 4.13 | Real and Estimated Fan Load Torque Characteristic with Noise | 43 |
| Figure 4.14 | Estimated and Real Fan Load Torque Estimation Error with Noise | 43 |
| Figure 4.15 | Real and Estimated Current with Noise | 44 |
| Figure 4.16 | Current Estimation Error with Noise | 45 |
| Figure 4.17 | Real and Estimated Speed with Noise | 46 |

| | | |
|-------------|---|----|
| Figure 4.18 | Speed Estimation Error with Noise | 46 |
| Figure 4.19 | Real and Estimated Current Validation from Simulations with Noise .. | 47 |
| Figure 4.20 | Current Estimation Error: Validation | 48 |
| Figure 4.21 | Real and Estimated Speed Validation from Simulation | 49 |
| Figure 4.22 | Validation Error from Real and Estimated Speed | 49 |
| Figure 4.23 | Real and Estimated Electrical Parameters from Recursive Simulation .. | 51 |
| Figure 4.24 | Real and Estimated Mechanical Parameters from Recursive Simulation .. | 51 |
| Figure 4.25 | Real and Estimated Fan Load Torque: Recursive Estimator | 52 |
| Figure 4.26 | Real and Estimated Fan Load Torque Error from Recursive Simulation .. | 53 |
| Figure 4.27 | Real and Estimated Current from Recursive Simulation | 54 |
| Figure 4.28 | Real and Estimated Current Error from Recursive Simulation | 54 |
| Figure 4.29 | Real and Estimated Speed from Recursive Simulation | 55 |
| Figure 4.30 | Speed Error from Recursive Estimator | 56 |
| Figure 4.31 | Real and Estimated Current Validation from Recursive Simulation | 57 |
| Figure 4.32 | Validation Error from Real and Estimated Current from Recursive Simulations | 57 |
| Figure 4.33 | Real and Estimated Speed Validation from recursive Simulation | 58 |
| Figure 4.34 | Errors between Real and Estimated Speed for Recursive Estimation : Validation | 58 |
| Figure 4.35 | Real and Estimated Electrical Parameters: Recursive Estimator with Noise .. | 60 |
| Figure 4.36 | Real and Estimated Mechanical Parameter: Recursive Estimator | 61 |
| Figure 4.37 | Real and Estimated Fan Load Torque: Recursive Estimator with Noise .. | 62 |
| Figure 4.38 | Real and Estimated Fan Load Torque Error from Recursive Simulation .. | 62 |
| Figure 4.39 | Real and Estimated Current from Recursive Simulation: Noise Case .. | 63 |
| Figure 4.40 | Real and Estimated Current Error from Recursive Simulation | 64 |
| Figure 4.41 | Real and Estimated Speed from Recursive Simulation | 65 |
| Figure 4.42 | Speed Error from Recursive Estimator | 65 |
| Figure 4.43 | Real and Estimated Current Validation Recursive Estimator Noise | 67 |

| | | |
|-------------|---|----|
| Figure 4.44 | Validation Error from Real and Estimated Current from Recursive Simulations | 67 |
| Figure 4.45 | Real and Estimated Speed Validation Recursive Estimator Noise | 68 |
| Figure 4.46 | Errors between Real and Estimated Speed for Recursive Estimation : Validation | 69 |
| Figure 5.1 | Experimental Setup | 70 |
| Figure 5.2 | dSPACE Matlab Simulink Block Diagram | 73 |
| Figure 5.3 | Current Measurement Circuit | 76 |
| Figure 5.4 | Voltage Measurement Circuit | 77 |
| Figure 5.5 | Speed Measurement Circuit | 79 |
| Figure 5.6 | Load Torque for the Fan used in this Work | 82 |
| Figure 5.7 | Input Voltage used for Training the Experimental Model | 83 |
| Figure 5.8 | Real and Estimated Fan Load Torque..... | 85 |
| Figure 5.9 | Estimated and Real Fan Load Torque Error | 85 |
| Figure 5.10 | Real and Estimated Current from Experiment | 86 |
| Figure 5.11 | Real and Estimated Current Error from Experiment | 87 |
| Figure 5.12 | Real and Estimated Speed from Experiment | 88 |
| Figure 5.13 | Real and Estimated Speed Error from Experiment | 88 |
| Figure 5.14 | Input Voltage used for Validation | 89 |
| Figure 5.15 | Real and Estimated Current Validation from Experiment | 90 |
| Figure 5.16 | Validation Error from Real and Estimated Current | 90 |
| Figure 5.17 | Real and Estimated Speed Validation from Experiment | 91 |
| Figure 5.18 | Validation Error from Real and Estimated Speed | 92 |
| Figure 5.19 | Real and Estimated Electrical Parameters from Recursive Experiment .. | 94 |
| Figure 5.20 | Real and Estimated Mechanical Parameters from Recursive Experiment .. | 94 |
| Figure 5.21 | Real and Estimated Fan Load Torque from Recursive Experiment | 95 |
| Figure 5.22 | Real and Estimated Fan Load Torque Error from Recursive Experiment .. | 95 |
| Figure 5.23 | Real and Estimated Current from Recursive Experiment | 96 |

| | | |
|-------------|--|----------|
| Figure 5.24 | Real and Estimated Current Error from Recursive Experiment |97 |
| Figure 5.25 | Real and Estimated Speed from Recursive Experiment |98 |
| Figure 5.26 | Real and Estimated Speed Error from Recursive Experiment |98 |
| Figure 5.27 | Real and Estimated Current Validation from Recursive Experiment |99 |
| Figure 5.28 | Validation Error from Real and Estimated Current from Recursive Experiments |100 |
| Figure 5.29 | Real and Estimated Validation from Recursive Experiment |101 |
| Figure 5.30 | Validation Errors from Real and Estimated Speed from Recursive Experiments |101 |
| Figure 5.31 | Real and Estimated Fan Load Torque with Fixed Ra |105 |
| Figure 5.32 | Estimated and Real Fan Load Torque Error with Fixed Ra |106 |
| Figure 5.33 | Real and Estimated Current from Experiment with Fixed Ra |107 |
| Figure 5.34 | Real and Estimated Current Error from Experiment with Fixed Ra |107 |
| Figure 5.35 | Real and Estimated Speed from Experiment with Fixed Ra |108 |
| Figure 5.36 | Real and Estimated Speed Error from Experiment with Fixed Ra |108 |
| Figure 5.37 | Real and Estimated Current Validation from Experiment with Fixed Ra | 109 |
| Figure 5.38 | Validation Error from Real and Estimated Current with Fixed Ra |110 |
| Figure 5.39 | Real and Estimated Speed Validation from Experiment with Fixed Ra | 110 |
| Figure 5.40 | Validation Error from Real and Estimated Speed with Fixed Ra |111 |
| Figure 5.41 | Real and Estimated Electrical Parameters from Recursive Experiment with Fixed Ra |113 |
| Figure 5.42 | Real and Estimated Mechanical Parameters from recursive Experiment with Fixed Ra |113 |
| Figure 5.43 | Real and Estimated Fan Load Torque from Recursive Experiment with Fixed Ra |115 |
| Figure 5.44 | Real and Estimated Load Torque Error from Recursive Experiment with Fixed Ra |115 |
| Figure 5.45 | Real and Estimated Current from Recursive Experiment with Fixed Ra | 116 |

| | | |
|-------------|--|-----|
| Figure 5.46 | Real and Estimated Currents Errors from Recursive Experiments with Fixed Ra | 116 |
| Figure 5.47 | Real and Estimated Speed from Recursive Experiment with Fixed Ra | 117 |
| Figure 5.48 | Real and Estimated Speeds Errors from Recursive Experiments with Fixed Ra | 117 |
| Figure 5.49 | Real and Estimated Current Validation from Recursive Experiment with Fixed Ra | 119 |
| Figure 5.50 | Validation Error from Real and Estimated Current from Recursive Experiment with Fixed Ra | 119 |
| Figure 5.51 | Real and Estimated Speed Validation from Recursive Experiment with Fixed Ra | 120 |
| Figure 5.52 | Validation Error from Real and Estimated Speed from Recursive Experiment with Fixed Ra | 120 |
| Figure 6.1 | Feedback Linearization with the Motor Drive | 130 |
| Figure 6.2 | Linearize System with Double Integrator | 130 |
| Figure 6.3 | Feedback Linearization Diagram with Recursive Linear Least Squares | 131 |
| Figure 6.4 | Matlab Simulink Model for the Feedback Linearization of the System | 132 |
| Figure 6.5 | Simulated Motor Speed and Double Integrator Output | 133 |
| Figure 6.6 | Speed Output and Reference Speed | 134 |
| Figure 6.7 | Simulated Motor Speed and Double Integrator Output with Noise | 135 |
| Figure 6.8 | Speed Output and Reference Speed with Speed | 135 |

CHAPTER 1

INTRODUCTION

1.1 Motivation

Since a fast development in automation technology is occurring in our times, an urgent demand for high performance motor drives has been increasing. To meet these high performance requirements, it has become necessary to develop control schemes that can overcome the influence of varying motor's parameters, the influence of nonlinear friction, and especially the influence of load variation such as inertia, viscous friction and unknown loads. Commissioning is one of the most common answers to the tuning of high performance motor drives. Self commissioning automates the tuning of a control system for a specified motor and load system. The tuning of the controller requires accurate models of the motor drive system. In the design of a typical motor drive, the dynamics of electric machines are well understood and accurate models for control already exists. On other hand, the mechanical loads depend on the application and modeling might be a complex task.

Here we propose to study gray-box modeling as an approach to develop identification methods for drive systems that can be used for a wide range of mechanical loads in a self-commissioning scheme. In our case, we know the model of the electric subsystem of the drive, but we don't know the exact model for the mechanical load. Gray-box modeling is a good alternative for modeling electric drives because it takes advantage of the available system knowledge while leaving flexibility to attend the mechanical load.

1.2 Objectives

The main objective of this work is to develop an automated methodology for the identification of mechanical loads models that can be used for tuning the controller of electric drive systems.

The specific objectives of this work are:

- To develop an identification algorithm for a motor drive system using a gray-box model.
- To implement a control scheme with a recursive identification algorithm for DC motor drive.
- To validate the proposed scheme using simulations and experiments.

1.3 Contribution

The contribution of this work is the development and validation of a two-stage linear least square parameter estimation method for the identification of a DC permanent magnet motor drive. This method separates the system in two subsystems, the physical and empirical part. The physical part is the electrical and mechanical model of the motor drive without the mechanical load. This part is also known as a white-box model. The empirical part or black-box is the mechanical load of the system. This system is represented with a parallel gray-box model:

$$\dot{\mathbf{x}}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1.1)$$

where: $\mathbf{h}(x(t), \mathbf{u}(t))$ is the white box model and $\mathbf{g}(x(t), \mathbf{u}(t))$ is the black box model. This model is called a gray-box model because is a tradeoff between the two individual subsystem models. In our work, \mathbf{h} will be associated with the electrical subsystem while \mathbf{g} will be associated with the mechanical subsystem.

Previous works [1-7] in the field of motor control and load identification, the load separation from the known physical system into an empirical model had been presented. In this work, we not only separate and estimate the mechanical load with a black-box model, but also we are estimating the physical model with linear least squares. This kind of method is the main contributions of this work, because it could be easily implemented on a self-commissioning scheme of a motor drive.

1.4 Thesis Outline

The thesis is organized as follows. Chapter 2 presents the literature review. In Chapter 3, we present the Two-Stage parameter estimation method. Chapter 4 presents simulation and validation results. Chapter 5 presents experimental and validation results. Chapter 6 conclusions and future work.

CHAPTER 2

BACKGROUND

This chapter presents basic concept of motor drives and gray box models using neural networks. Also the concepts of linear and recursive parameter estimation as well as the control scheme used in this work are discussed.

2.1 Concepts in Motor Drives

About 65 % of the total electric energy produced in the US is consumed by electric motor drives. An electric drive converts electrical energy to mechanical (rotational or kinetic) energy for many applications in the industry and almost everywhere around us. Electric drives are an integral part of many industries. Motors drives are responsible for running large pumps, air compressors, and elevators and in small applications such the hard disk drive from a computer. The combination of an electric motor, a controller, the sensors, a power electronics unit and the mechanical load is what is called electric drives (Figure 2.1).

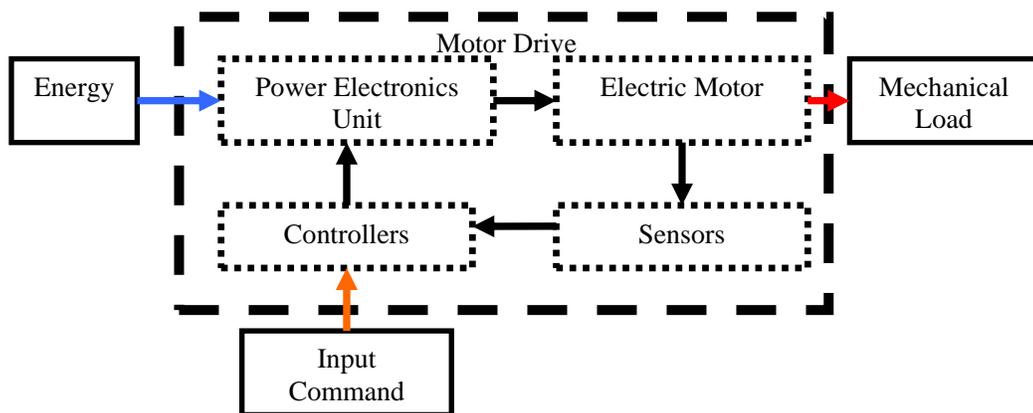


Figure 2.1 Electric Drive System Block Diagram

Since electric drives are being used in almost everything that needs to be moved or driven is very important to have an efficient drive on a particular system. In order to have an efficient electric drives, all of its components need to be properly designed. The controller in conjunction with the power electronics units will produce the necessary input signal to the electric motor to drive the mechanical load to meet the required specifications. Significant attention is paid to the operation and the efficiency of the system. This is the reason why on this work we present a methodology to obtain a good model for high performance drives.

2.1.1 DC Motor Drives

In [1,8], the capability of the gray-box approach for motor drive modeling was studied. Their work showed identification of a simulated drive system. The simulated system was a permanent magnet Direct Current (PMDC) motor driving a nonlinear static load. The PMDC motor is used in the test-bed experiment of this work. The nameplate data of the DC motor used are the following:

- Horse Power: ½ HP
- Voltage: 90 V
- Current: 5.2 A
- Rated Speed: 2500 RPM
- Wound: Permanent Magnet
- Manufacturer: General Electric Company

This nameplate data was used in the simulations and to evaluate experimental results.

In Figure 2.2, a schematic of the simulated system is presented.

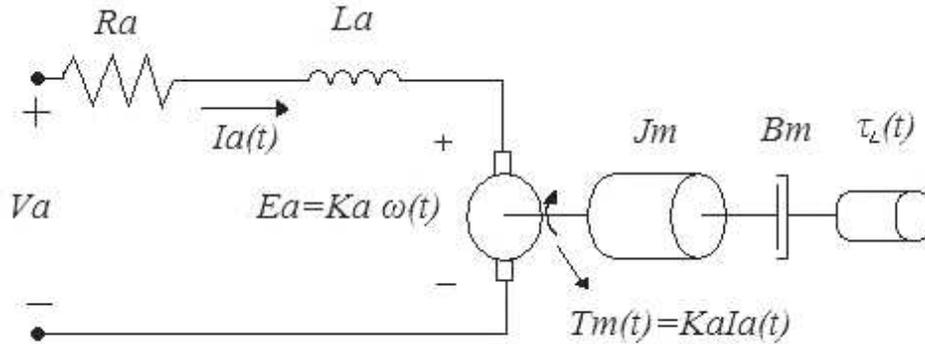


Figure 2.2 Schematic for a DC Motor and Load System

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

$$L_a \frac{di_a(t)}{dt} = V_a(t) - R_a i_a(t) - K_a \omega(t) \quad (2.1)$$

$$J_m \frac{d\omega(t)}{dt} = \tau_{em}(t) - B_m \omega(t) - \tau_L(\omega(t)) \quad (2.2)$$

$$\tau_{em}(t) = K_a i_a(t) \quad (2.3)$$

where: L_a is the armature inductance, R_a is the armature resistance, $V_a(t)$ is the input voltage, K_a is the induced emf constant, J_m is the combined load and motor inertia of the motor, B_m is the damping coefficient, $\omega(t)$ is the rotor speed, $i_a(t)$ is the armature current, and τ_L is the speed depend load torque. Table 2.1 summarizes the physical parameters values used for the simulations and the experiments. These parameters were obtained using the procedure described in Chapter 4.

Table 2.1 Parameters of the DC Motor Drive

| Parameter | Value |
|-----------|----------------------------|
| J_m | 0.002387 Kg.m ² |
| K_a | 0.3409 Nm.A ⁻¹ |
| R_a | 1.587 Ω |
| L_a | 0.409 H |
| B_m | 0.00086 |

In this work, we also consider a nonlinear static mechanical load. The basic load to be used on the work is a fan. In real life this load is a 16 inch blade with 26 degrees of pitch fan. The fan torque load had the following equation:

$$\tau_F = \mu \text{sign}(\omega(t)) \omega^2(t) \quad (2.4)$$

where μ (Nm*s²) is a constant value and the function is set so that the direction of the fan torque always opposes the direction of motion [2]. The fan load equation and its parameter values are presented in Table 2.

Table 2.2 Case Study Load Torque

| Load Case | Torque Equation | Parameter Value |
|-----------|---|--------------------|
| Fan Load | $\tau_L(\omega) = \mu \text{sign}(\omega(t)) \omega^2(t)$ | $\mu=0.000063$ N-m |

2.1.2 Mechanical Load Identification for Electric Drives

Identification of a mechanical load using prior physical knowledge has been extensively studied. In [3], a method for commissioning the speed and position control system of an electric drive is presented. It includes the identification of the nonlinear mechanical load on the system. The identification of the mechanical system was

implemented assuming it behaves like a one mass or two mass, then using extracted characteristics features from acquired data to determine which system to use. The key features used in this case for structure selection are based on knowledge of ideal response of the system when it is assumed to be one mass or two.

In [4], the authors develop models for nonlinear mechanical loads in electric drive system. Radial basis function networks are integrated with a physics-based model for the load. An EKF is used to estimate position and speed, and the friction as a nonlinear load. Implementation on a real system is presented. The basic feature of this work is the modeling of the mechanical part. The following model representation was used:

$$\dot{x}(t) = Ax + Bu + NL(x, u) \quad (2.5)$$

where $NL(x, u)$ is the nonlinear representation of the mechanical friction. The work found that EKF integrated with the radial basis function is a powerful algorithm for online parameter and state estimation. Our work use a similar model structure as equation (2.5).

The work on [5] describes experiments whose objective is to estimate dynamic friction without relying on measured friction force or a structured friction model. The method addresses friction characterization in the presence of variable and uncontrollable factors; including, wear, environmental conditions, lubricant condition, and normal force variation. The method also used the EKF to determine friction force without relying on a priori friction model. The method relies on accurate dynamic modeling of the system and measured motion to extract the unknown friction force. A very similar concept as the one presented in this thesis. This is the importance of this work because friction force is often

difficult or expensive to measure on-line; estimation can provide data from which locally valid dynamic models can be constructed. The work also presented that their estimate can be used for friction-based diagnostics by monitoring friction force variations during various operating conditions and, also friction force estimation can be used for friction compensation.

Finally, on [6], two gray-box models are presented combining a white-box with a black-box model. The authors used Neural Networks and a polytopic model as black-box models that are capable of identifying friction characteristics that are left unexplained by the first principles modeling (white-box model). They also introduced an experimental case-study where both gray-box models are applied to identify a rotating arm subject to friction. The authors performed an experiment where the parameters of the two proposed models are estimated with experimental data obtained from the rotating arm. Later a closed-loop nonlinear state feedback friction compensation, which is a control loop for the friction, was used to verify if the proposed method was effective for an on-line application.

2.2 System Identification: Gray-Box Modeling

Models help us understand and describe our knowledge of a system in a mathematical sense. For that reason, modeling of physical system on the basis of physical principles is widely used and implemented. This implementation of physical laws that are governing the physical systems provides a mathematical model that requires the estimation of key parameters. System identification is concern on the determination of a system, on the basis of input and output data samples. The estimation task is to determine at each instant time a

suitable estimate of the parameters which, once specified, completely describe or characterize the system.

For the case of gray-box modeling, the system model is partitioned into two components. One component is based on the knowledge of physical principles that govern the system dynamics (white-box). That is the case of the electric subsystem in the motor drive. In this electric subsystem all the parameters, the signals and the model equations are available. This white box models required a good knowledge of the system in order to represent the physical system correctly.

When the structure of a system is not known or the data available is insufficient to use a physical modeling, a black box or empirical model is used. A black-box model tries to estimate both the functional form of relations between parameters and all the unknown variables affecting the physical equations of the white-box model. If there is no priori information of the model, we would try to use functions as general as possible to cover all different models. An often used approach for black-box models are neural networks. The problem with using large set of functions to describe a system is that estimating accurately the parameters becomes increasingly difficult when the amount of parameters increases. In our case, the unknown mechanical load is approximated with a radial basis function neural network. The problems of estimating all of the parameters of the neural network are partially solved with the assumptions made when implementing the radial basis function. This topic is described in the Section 4.1.

A gray-box model represents the tradeoff between the white box and the black box models [8]. The gray-box models structure relies on prior knowledge and the model parameters are mainly determined using measured data. That is the case when a part of the system can be represented with the prior knowledge of physical principles but another part remains to be determined from observed data. In other words, with gray-box models we have partial knowledge of the model structure that is to be identified.

The parallel gray box model is given by:

$$\dot{\mathbf{x}}(t) = \mathbf{h}(x(t), \mathbf{u}(t)) + \mathbf{g}(x(t), \mathbf{u}(t)) \quad (2.6)$$

where: $\mathbf{h}(x(t), \mathbf{u}(t))$ is the white box model and $\mathbf{g}(x(t), \mathbf{u}(t))$ is the black box model. In our work, \mathbf{h} will be associated with the electrical subsystem while \mathbf{g} will be associated with the mechanical subsystem.

2.3 Self Commissioning of Electric Drives

Due to the fast development in automation technology, the urgent demand for high performance electrical drives has been increasing [8]. To meet these high performance drives requirements, it has become necessary to develop controllers that can overcome the influence of nonlinear friction, the influence of changing motor's parameters and especially the influence of load variation, to keep the performance of the overall drive system unchanged. An important problem in drive systems is the controller tuning prior to system operation or commissioning. In a conventional motor drive, commissioning is usually

performed using analytical design and experimental tests based on given or measured motor parameters. The steps during the drive commissioning are the following [7]:

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

Self-commissioning is the automation of the commissioning process. This process is usually performed by a trained technician or engineer, which involve costs and time. With self-commissioning, the system itself determines the electrical parameters of the machine during the commissioning and sets the control parameters accordingly [9]. Some of the issues in self-commissioning are the load identification and the controller tuning. In this work a solution for load identification is presented based on a gray box model.

The advances in computers made possible the creation of sophisticated control algorithms. The auto-tuning self commissioning (ATSC) control of an electric drive becomes a new issue in the design of a universal drive for various applications with unknown motor parameters and load dynamics. The two basic requirements of this ATSC are the self commissioning and auto-tuning. The self commissioning has been explained but the auto tuning concerns the automatic tuning of the control parameters when the motor drive is operating to achieve a satisfactory drive performance. This kind of tuning eliminates the necessity of constantly tuning the control schemes when the loads on an electric drive changes. In [7], Beineke presents a computer aided motor drive commissioning tool.

Basically they design a software tool which permits the identification of the mechanical system and the selection commissioning of several speed and position controllers for a two-mass systems with mechanical imperfections such as friction and backlash. They design a tool that was easy to use for the industry commissioning personnel, which is one of the objectives of developing auto-tuning self-commissioning control schemes for electric drives.

2.4 Radial Basis Function Neural Networks

Radial Basis Functions are a type of Artificial Neural Network. The Artificial Neural Networks (ANN) models are based on our present understanding of biological nervous systems. One of the most important attributes of the neural networks is the ability to generalize. What this mean is the ability to successfully interpret data which it has not previously been encountered and provide a sensible result on the data. Naturally, there are limits to the generalizing ability of the networks and it is essential for the network to have been trained on information or data which is closely related to that on which the network is expected to generalize. In the generalization, it is better to consider the network as an interpolator within the multidimensional space it was trained.

Artificial neural networks have high computation rates provided by their basic features:

- High parallelism on the approach

- The ability to distribute memory over a large number of components within the network
- Learning features from the training data.

They are useful for approximating nonlinear mapping and for extracting characteristics from noise corrupted signals in random environment [10]. The type of ANN used on [8] to emulate load behavior is a layered feedforward network. In this type of network, there can be many hidden layers in between the input and the output but every unit must send its output to layers higher than its own and must receive its input from layer lower than its own.

In this work, a certain type of neural network was used to approximate the load torque characteristic of a motor drive system. Figure 2.3 presents the block diagram of the actual motor drive system. In this case we used the torque $\tau_L(\omega(t))$ to represent a static load like. One of the objective of this work was the used of artificial neural network (black-box) to model the load. For that reason we used an artificial neural network instead of the fan load torque equations. This substitution is presented in Figure 2.4. On this figure the system is separated in two divisions, the physical known part of the system, which is compound of the electrical parameters, and the black box part which is composed by some parameters and the neural networks.

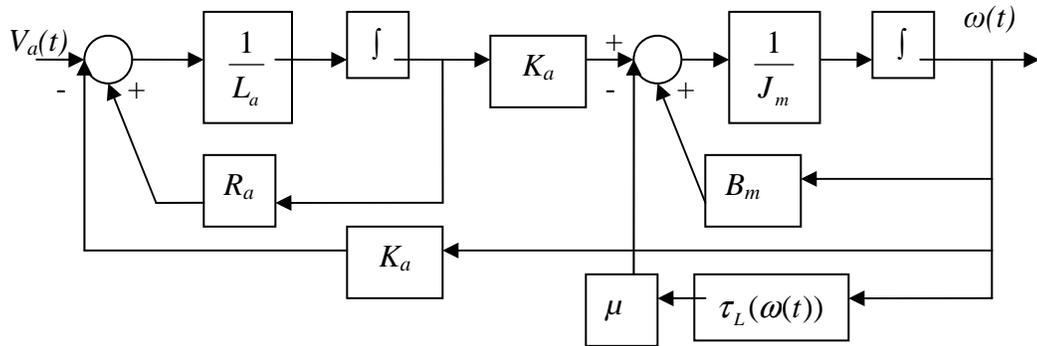


Figure 2.3 Actual Motor Drive System

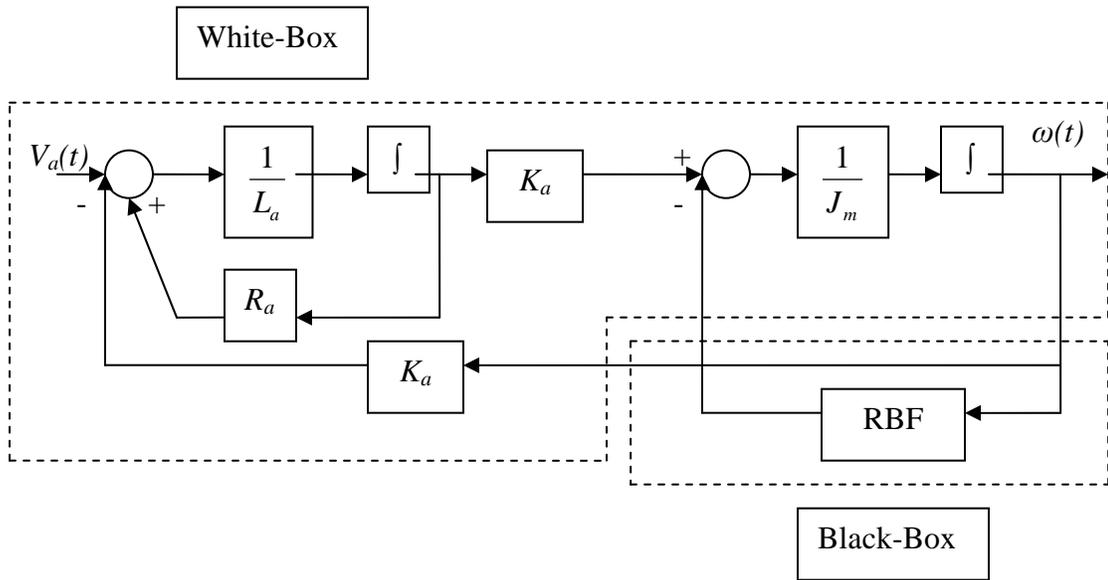


Figure 2.4 Gray-Box Model Structure

In this work, as in [8], the radial basis function (RBF) neural network was used to model the unknown load τ_L on the mechanical equation (2.7). The RBF uses the radial construction presented in Figure 2.5.

$$J_m \frac{d\omega(t)}{dt} = \tau_{em}(t) - B_m \omega(t) - \tau_L(\omega(t)) \quad (2.7)$$

The output of the radial basis function neural network is given by:

$$\hat{y} = \sum_{i=1}^N \alpha_i F_i(\|u - c_i\|) \quad (2.8)$$

where $F(\cdot)$ is the basis function. For the Gaussian case, (2.8) changes to:

$$y = f(x) = \sum_{i=1}^N \alpha_i e^{-\left(\frac{(x(t)-c_i)^2}{2\sigma_i^2}\right)} \quad (2.9)$$

where: $x(t)$ is the input data, α_i is the weight of the function, c_i is the center of the i^{th} node and σ_i is the variance of the i^{th} node.

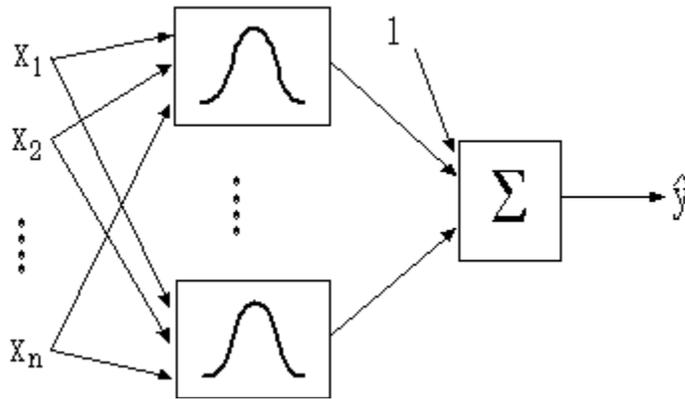


Figure 2.5 Structure of a Radial Basis Function Neural Network

The radial basis function has three types of parameters:

- Output layer weights, α_i : that are linear parameters and they determine the amplitude of the basis function.
- Centers, c_i : are nonlinear parameters of the hidden layer neurons, and they determine the position or location of the RBF as well as the weights α_i .
- Standard deviation, σ_i^2 : are the nonlinear parameters of the hidden layer neurons, and they determine the width of the RBF.

In an RBF the i th input variable is linked to the i th RBF via the center component c_i . The i th output component of the network is related to the i th RBF via the weight α_i . Solving for the weights is equivalent to solving a system of P linear equations with N unknowns.

Here much attention is given to the RBF neural networks because in this work the black-box model representation of the load is based on this type of network. The main attraction of RBF is that the model parameters may be calculated using linear methods. This assumes that the locations, or centers, of the approximating functions are predetermined or selected in priori and fixed. In this situation, the error function is quadratic in the model parameters, a fact that has the effect of computational effort simplification. There is a global minimum which may be attained by solving a least squares problem or adaptively using a descent method [11]. Another major attraction of RBF is their flexibility. The basis function may be chosen to be either local or global, and they may or may not incorporate shape parameters, which can be tuned to reflect the nature of the data. Moreover, RBF train rapidly without local minima problems and approximate any continuous function with accuracy. Their rapid training makes them suitable for situations where on-line learning is necessary

which is going to be the case of this work. More detailed information on Radial Basis Function neural network is presented on [6 and 11].

2.5 Parameter Estimation

All of the presented models have parameters that need to be estimated. Generally the model parameters are estimated using the least square method. The technique of least squares was developed independently by A.M. Legendre and Carl Friedrich Gauss in 1806 and 1797 respectively. The objective of the least squares method is to find estimates of the model parameters that best fit to measured data points. Since the data may include errors in measurement or experiment inaccuracies, we do not require the model to fit through all the data points. Instead, we require the model to provide an optimal approximation in the sense that the sum of squares of errors between the values of the data points and the corresponding estimates of the model are minimized [12]. For linear models, the least squares parameter estimates can be found easily, but for nonlinear models the estimation of parameters becomes a more difficult problem to solve. In that case, the use of iterative or recursive methods to compute estimates of the parameters is needed. In our case, we take advantage of the nonlinear model structure to solve the problem as a sequence of linear least square problems.

2.5.1 Linear Least Squares

The least square method finds the parameters such that the model output \hat{y} that best approximate the measured output y by minimizing the sum of squared errors. For the linear case,

$$y(t) = \mathbf{a}^T(t)\boldsymbol{\theta} + e(t) \quad (2.10)$$

t is discrete time index, $\mathbf{a}(t)$ is the $m \times n$ regressor vector, $y(t)$ represents the available measurement, and $\boldsymbol{\theta}$ is a vector of the parameters to be estimated. For $m > n$, and $\text{Rank}(\mathbf{a}(t)) = n$, in general, there will be no solution to (2.10), so in the linear least squares method the parameters are optimizing by minimizing the square norm of the error $\mathbf{e} = \mathbf{A}_i\boldsymbol{\theta} - \mathbf{y}_i$ as follows:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^n} \|\mathbf{A}_i\boldsymbol{\theta} - \mathbf{y}_i\|_2 \quad (2.11)$$

where $\|\cdot\|_2$ is the Euclidian norm, $\mathbf{y}_i = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix}$ and $\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_1^T(t) \\ \vdots \\ \mathbf{a}_i^T(t) \end{bmatrix}$

The solution to (2.11) is given by:

$$\hat{\boldsymbol{\theta}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \quad (2.12)$$

The linear least squares have several important features for system identification [13]. For example, with least squares large errors are heavily penalized. Also the linear least squares estimates can be obtained by straightforward matrix algebra. Finally in many cases the linear least squares criterion is related to statistical variance and the properties of the solution can be analyzed according to statistical criteria.

2.5.2 Recursive Least Square Algorithm

The purpose of this work is to provide a tool for on-line identification of the electric drive. For that reason we used the recursive Newton-Raphson algorithm applied to scalar linear predictor with a least squares identification criterion..

The Newton-Raphson algorithm [14] generates a sequence of parameter values θ_i , given a starting value $\theta_i(N)$, by means of the recursion:

$$\theta_i(N) = \theta_i(N-1) - \left[\frac{\partial^2 \mathbf{V}}{\partial \theta^2}(\theta_i(N-1)) \right]^{-1} \frac{\partial \mathbf{V}^T}{\partial \theta}(\theta_i(N-1)) \quad (2.13)$$

Here the row vector $\partial \mathbf{V} / \partial \theta$ denotes, as usual, the gradient of \mathbf{V} . $\partial^2 \mathbf{V} / \partial \theta^2$ is the matrix of second partial derivatives ($\partial^2 \mathbf{V} / \partial \theta_i \theta_j$) or the Hessian of \mathbf{V} . Now modifying the Newton-Raphson algorithm applied to scalar predictor models when a least squares identification criterion

$$\mathbf{V}(\alpha) = \frac{1}{N} \|\mathbf{e}_i(\theta_i(N))\|^2 = \frac{1}{N} \sum_{i=1}^N \mathbf{e}_i^2(\theta_i(N)) \quad (2.14)$$

is adopted. Here $\mathbf{e}_i(\theta(N))$ are the prediction errors associated to the model:

$$\mathbf{e}_i(\theta(N)) = \mathbf{y}_i - \mathbf{a}_i(N)^T \boldsymbol{\theta}_i(N), \quad (2.15)$$

where a is the model input, y is the model output and θ are the model parameters.

Given data $\mathbf{y}_i(N)$, $\mathbf{a}_i(N-1)$ and an estimate, $\boldsymbol{\theta}_i(N-1)$, the algorithm supplies an updated estimate, $\boldsymbol{\theta}_i(N)$, according to the rule:

$$\boldsymbol{\theta}_i(N) = \boldsymbol{\theta}_i(N-1) - \alpha \mathbf{H}_i(\boldsymbol{\theta}_i(N-1))^{-1} \frac{\partial \mathbf{V}_i^T(N)}{\partial \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i(N-1)) \quad (2.16)$$

where the gradient of \mathbf{V}_N with respect to the parameter $\boldsymbol{\theta}$ is given by:

$$\frac{\partial \mathbf{V}_i(N)}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{2}{N} \sum_{i=1}^N e_i(\boldsymbol{\theta}(N)) \mathbf{J}_i(\boldsymbol{\theta}(N)) \quad (2.17)$$

In this expression, the row vector \mathbf{J}_i is defined by the equation:

$$\mathbf{J}_i(N) = -\frac{\partial f_k}{\partial \boldsymbol{\theta}}(y_i(N-1), \boldsymbol{\theta}_i(N-1)), \quad (2.18)$$

$\mathbf{H}_i(\boldsymbol{\theta}(N))$, is an approximation to the Hessian of $\mathbf{J}_i(N)$ at $\boldsymbol{\theta}$ given by:

$$\mathbf{H}_i(\boldsymbol{\theta}(N)) = \frac{2}{N} \sum_{i=1}^N \mathbf{J}_i^T(\boldsymbol{\theta}(N)) \mathbf{J}_i(\boldsymbol{\theta}(N)) \quad (2.19)$$

and α is a suitable positive number acting as the forgetting factor.

The introduction of \mathbf{J}_i provides us with a convenient new approximation, \mathbf{R}_i , to the Hessian, where:

$$\mathbf{R}_i(\boldsymbol{\theta}(N)) = \frac{1}{N} \sum_{i=1}^N \mathbf{J}_i^T(\boldsymbol{\theta}(N)) \mathbf{J}_i(\boldsymbol{\theta}(N)) \quad (2.20)$$

Now we have to apply the matrix inversion lemma presented in [14] and it results in a recursive equations for $\mathbf{P}_i(N) = (N\mathbf{R}_i(N))^{-1}$, namely:

$$\mathbf{P}_i(N) = [I - (1 + \mathbf{J}_i(N)\mathbf{P}_i(N-1)\mathbf{J}_i^T(N))^{-1} \mathbf{P}_i(N-1)\mathbf{J}_i^T(N)\mathbf{J}_i(N)]\mathbf{P}_i(N-1) \quad (2.21)$$

After the corresponding approximations and derivations we come with a new updating formula (2.16) in terms of \mathbf{P}_N :

$$\boldsymbol{\theta}_i(N) = \boldsymbol{\theta}_i(N-1) - \alpha \mathbf{P}_i(N) \mathbf{J}_i^T(N) e_i(N) \quad (2.22)$$

The equations (2.15), (2.18), (2.21) and (2.22) define the recursive algorithm.

2.6 State Variable Filters

On the equations established in the Section 2.1, the derivative of the armature currents and rotor speed are required. This introduces a problem because the direct differentiation of analog or digital quantities can amplify the noise of the measure signals.

One possible solution for this problem consists of the use of state variable filters (SVF). The state-variable filter method is based on the block diagram representation used in the so-called phase-variable description of linear systems that use the outputs of a chain of cascaded integrators as state variables [15].

The transfer function of the system of interest is given by:

$$H(s) = \frac{K}{s^n + a_{n-1}s + \dots + a_1s + a_0} \quad (2.23)$$

where n is the order of the highest desired derivative. The state variable filter is realized in the controllable canonical form in order to obtain the necessary derivative terms.

The figure 2.6 shows a block diagram of a first order state variable filter. This filter was implemented in our work because we only needed the first derivative of the armature current and the rotor speed. Notice that the filter variable and its derivative are available. In the processing, we use the filtered variables instead of the real ones.

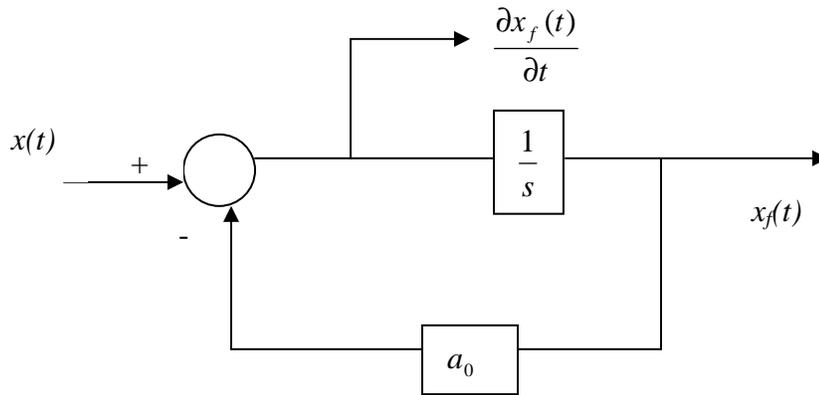


Figure 2.6 First Order State Variable Filter

The most difficult part in the design of such filters is the selection of the filter bandwidth. The optimal bandwidth is a compromise between the noise rejection properties of the filter and the quality of the input signals to the estimation algorithm. Filters of this kind can be easily designed using techniques like Bessel, Butterworth or Chebyshev.

CHAPTER 3

TWO STAGE PARAMETER ESTIMATION

Parameter estimation is one of the main issues in system identification. The goal is to find the parameters that best describes or characterize the data used for the system model. On this chapter a linear least square and a recursive linear least square two stage parameter estimation ARE presented.

3.1 Gray-Box Model

In Chapter 2, we mentioned that the torque load of the permanent magnet DC motor is going to be assumed unknown and is going to be approximate with a neural network. This decision was made base on the results presented on [1] and [8], where the same issue were addressed. In those previous works, neural networks were implemented to approximate the load. In our case, we used Radial Basis Functions (RBF) as the neural network for the load. The following equations describe the RBF structure used on this work.

$$RBFNN = \sum_{i=1}^N \alpha_i F_i(x, c_i, \sigma_i) \quad (3.1)$$

where:

$$F_i = e^{\left(\frac{(\omega(t)-c_i)^2}{2\sigma_i^2} \right)} \quad (3.2)$$

The load used in this work was a fan. The fan torque is a nonlinear function of speed. Knowing this matter, we substitute the Radial Basis Function (3.1) in the mechanical

equation (3.3) of the DC motor. This new mathematical model for the mechanical equation of the motor is now given by

$$\frac{d\omega(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{B_m}{J_m} \omega(t) - \frac{1}{J_m} \tau_L(\omega(t)) \quad (3.3)$$

$$\frac{d\omega(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{B_m}{J_m} \omega(t) - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(-\frac{(\omega(t)-c_i)^2}{2\sigma_i^2} \right)} \right) \quad (3.4)$$

In this work, we decide not to estimate the viscous friction factor B_m . In [8], they conclude that the error without the viscous friction where lower than the case when the viscous friction was included. They concluded that the radial basis function was absorbing the effect of the viscous friction factor. Following their recommendations, we eliminated the viscous friction from equation (3.4) resulting in

$$\frac{d\omega(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(-\frac{(\omega(t)-c_i)^2}{2\sigma_i^2} \right)} \right) \quad (3.5)$$

for the mechanical load.

Now we have the complete mathematical model for the gray-box model of the permanent magnet DC motor.

3.2 Linear Regression Model for Electrical and Mechanical Subsystems

Here we show two linear regressor models derived from (3.3) and (3.4) used for the design of the estimator. These models made the estimation process easy to solve, because it reduces the problem from a complex nonlinear one into two linear simple estimation problems [8]. The most important advantage of this method is avoiding the use of the iterative Gauss Newton method which is computationally complex and time consuming. Instead the problem is simplified to solve it by the use of linear least squares which is less complex and than the Gauss Newton method.

From equation (3.3), the electrical equation can be arranged into the linear regression form as follows:

$$\mathbf{y}_e(t) = \mathbf{a}_e(t)^T \boldsymbol{\theta}_e \quad (3.6)$$

where:

$$\mathbf{a}_e(t) = \begin{bmatrix} v_a & -i_a & -\omega \end{bmatrix} \quad (3.7)$$

$$\hat{\boldsymbol{\theta}}_e = \begin{bmatrix} \theta_{e1} & \theta_{e2} & \theta_{e3} \end{bmatrix}^T \quad (3.8)$$

and

$$\mathbf{y}_e(t) = \frac{di_a(t)}{dt} \quad (3.9)$$

The electrical parameters are obtained as follows:

$$\hat{L}_a = \frac{1}{\hat{\theta}_{e1}} \quad (3.10)$$

$$\hat{R}_a = \frac{\hat{\theta}_{e2}}{\hat{\theta}_{e1}} \quad (3.11)$$

$$\hat{K}_a = \frac{\hat{\theta}_{e3}}{\hat{\theta}_{e1}} \quad (3.12)$$

Now considering the mechanical equation (3.13) without the explicit viscosity B_m , it can be arranged as follows:

$$J_m \frac{d\omega(t)}{dt} + \sum_{i=0}^N \alpha_i e^{\left(\frac{(\omega(t)-c_i)^2}{2\sigma_i^2} \right)} = \tau_m(t) \quad (3.13)$$

The corresponding linear regression model is given:

$$\mathbf{y}_m = \mathbf{a}_m(t)^T \boldsymbol{\theta}_m \quad (3.14)$$

where

$$\mathbf{a}_m(t) = \left[\frac{d\omega}{dt} \quad F_1(\omega) \cdots F_N(\omega) \right] \quad (3.15)$$

$$\hat{\boldsymbol{\theta}}_m = \left[\hat{\theta}_{m1} \quad \hat{\theta}_{m2} \cdots \hat{\theta}_{m162} \right]^T \quad (3.16)$$

and

$$\mathbf{y}_m = \tau_m(t) = \hat{K}_a i_a(t) \quad (3.17)$$

The mechanical parameters are estimated as follows:

$$\hat{J}_m = \hat{\theta}_{m1} \quad (3.18)$$

$$[\alpha_i \cdots \alpha_{161}]^T = [\hat{\theta}_{m2} \cdots \hat{\theta}_{m162}] \quad (3.19)$$

As is presented in (3.17) the electromagnetic torque is estimated using the estimated \hat{K}_a obtained from the electrical regression model.

The Figure 3.1 summarizes the two-stage algorithm implemented on this work.

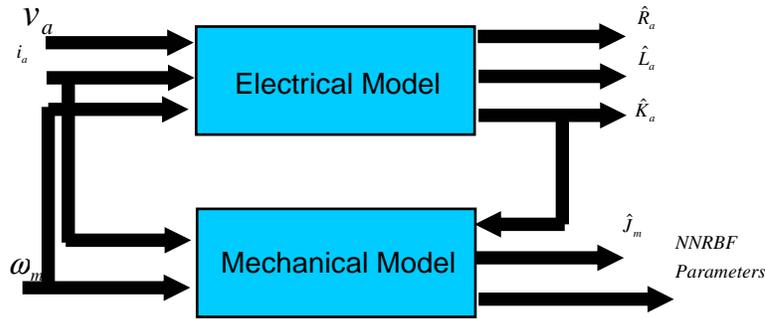


Figure 3.1 Two Stage Algorithm Summary for both Identification Cases

3.3 Recursive Algorithm for Electrical and Mechanical Subsystems

The recursive algorithm implemented on this work is a two-stage method similar to the batch case. The gray-box model introduces many parameters because of the radial basis function used to estimate the nonlinear load. Separating the algorithm in electrical parameters θ_e and mechanical parameters θ_m made the algorithm faster to converge than in the case with all of the parameter being estimated recursively at the same time. This technique is similar to the separable nonlinear least squares introduced in [16]. In that work, they estimate a Hammerstein model separating the parameters. The Hammerstein model consists of a static nonlinear block that is parameterized by a feed-forward or radial basis function neural network in cascade with a linear block that is parameterized by a linear ARX. They separated the nonlinear parameters of the static neural network and the linear parameters of the ARX model.

Now implementing the two-stage recursive method in the Newton-Raphson algorithm we first arranged the updating equation (2.22) for the electrical parameters as follows:

$$\boldsymbol{\theta}_i(N) = \boldsymbol{\theta}_i(N-1) - \alpha \mathbf{P}_i(N) \mathbf{J}_i^T(N) e_i(N) \quad (3.20)$$

$$\boldsymbol{\theta}_e(N) = \boldsymbol{\theta}_e(N-1) - \alpha \mathbf{P}_e(N) \mathbf{J}_e^T(N) e_e(N) \quad (3.21)$$

where P_{eN} , J_{eN} and r_{eN} are the recursive Hessian approximation, Jacobian and prediction errors respectively for the electrical parameters,

and

$$\hat{\boldsymbol{\theta}}_e = \begin{bmatrix} \theta_{e1} & \theta_{e2} & \theta_{e3} \end{bmatrix}^T \quad (3.22)$$

The original electrical parameters are estimated as follows:

$$\hat{L}_a = \frac{1}{\hat{\theta}_{e1}}, \quad \hat{R}_a = \frac{\hat{\theta}_{e2}}{\hat{\theta}_{e1}} \quad \text{and} \quad \hat{K}_a = \frac{\hat{\theta}_{e3}}{\hat{\theta}_{e1}} \quad (3.23)$$

Similarly, the updating equation (2.22) can be arranged for the mechanical parameters as follows:

$$\boldsymbol{\theta}_m(N) = \boldsymbol{\theta}_m(N-1) - \alpha \mathbf{P}_m(N) \mathbf{J}_m^T(N) e_m(N) \quad (3.24)$$

where $P_m(N)$, $J_m(N)$ and $e_m(N)$ are the recursive Hessian approximation, Jacobian and prediction errors respectively for the mechanical parameters,

and

$$\hat{\boldsymbol{\theta}}_m = \begin{bmatrix} \theta_{m1} & \theta_{m2} \end{bmatrix}^T \quad (3.25)$$

The original mechanical parameters are estimated as follows:

$$\hat{J}_m = \hat{\theta}_{m1} \quad \text{and} \quad [\alpha_i \cdots \alpha_{161}]^T = [\hat{\theta}_{m2} \cdots \hat{\theta}_{m162}] \quad (3.26)$$

\hat{J}_m in this case is the motor inertia mechanical parameter not the Jacobian matrix of the recursive algorithm equations. This is the recursive algorithm presented for this work. The results of the simulations and the experiments are presented on Chapter 5 of this work.

3.4 Identification Process Algorithm

Figure 3.2 presents the identification process used in this work. In the first step we began with the selection of the model structure. In our case, radial basis functions were used to model the mechanical load as a black-box. In the next step, we proceed to estimate the parameters. For this part a simulation and an experiment were performed. The two-stage method was used to estimate the parameters with two different algorithms. In the first algorithm the system parameters are estimated using linear least squares and in the second algorithm they were estimated with a recursive linear least squares. In both algorithms simulations were done by substituting the estimated parameters into the model and running it with the same input as the real system. The simulations results were compare with the real results. No the next step would be the validation of the results. For this step, we used a different input signals as the one used for the parameter estimation step. Then we perform simulations by substituting into the model the estimated parameters from the parameter estimation step. Then we run the model with the new input signals from the validation step to corroborate if the estimated parameters are working with a different set of input signals. If this validation is correct then we stop the process, but if it is not correct then we return to the model structure step to verify or change the model.

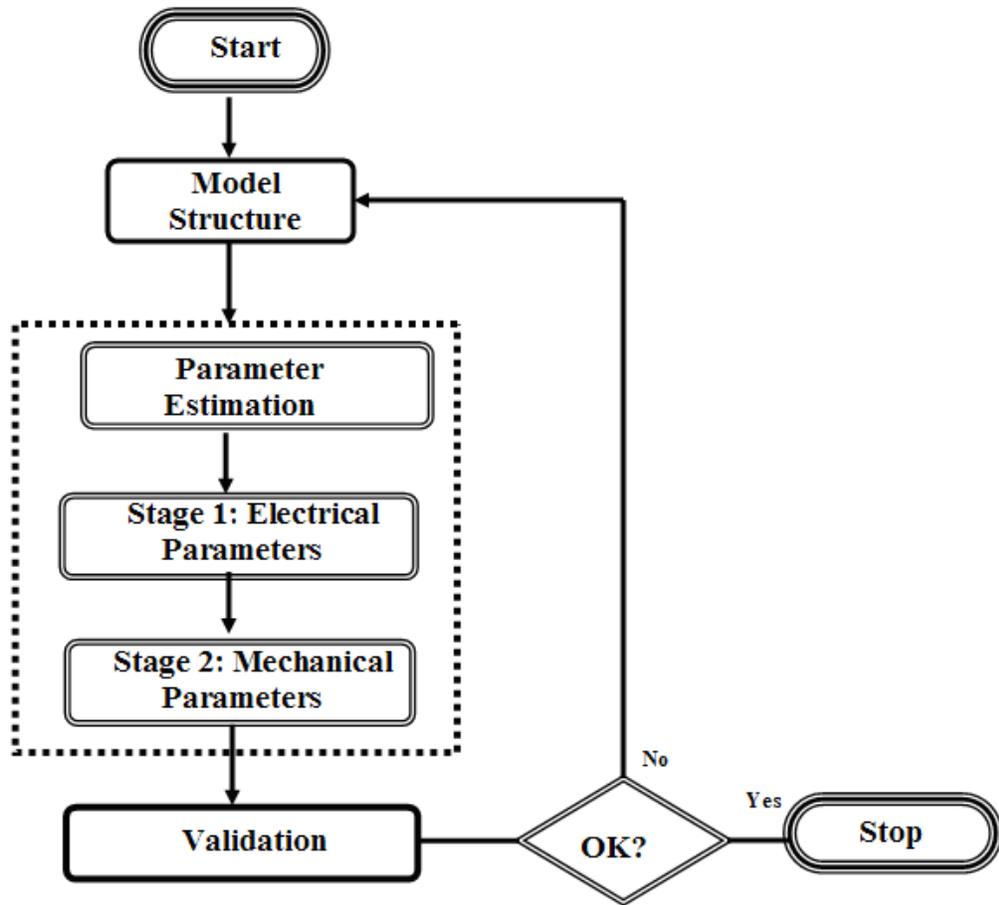


Figure 3.2 Diagram for the Identification Process Algorithm

CHAPTER 4

SIMULATION RESULTS

This chapter presents the simulation results for the two proposed algorithms for parameter estimation. The validation is also presented on this chapter for all of the algorithms. The experimental result are presented in the next chapter.

4.1 Initialization

Before any simulation was performed, it was very important to initialize the data and the radial basis neural network. First of all it is very important to assume that the motor is going to be operated over the entire nominal operating range. In real life, this would limit the operating range in which you could operate the motor and could introduce errors to the system identification process. Another important remark is using the neural network for modeling the load.

The input data used for training the model was an armature voltage (V_a) signal that would result in a speed response within a range of 0 rad/s, to approximately, 150 rad/s. For the radial basis function model, 161 points were selected priori. These 161 points simulated the speed from 0 rad/s to 160 rad/s were selected as the center of the NN. The variance was also selected in priori and fixed to a value of 4. The armature voltage used for this system is presented on Figure 4.1.

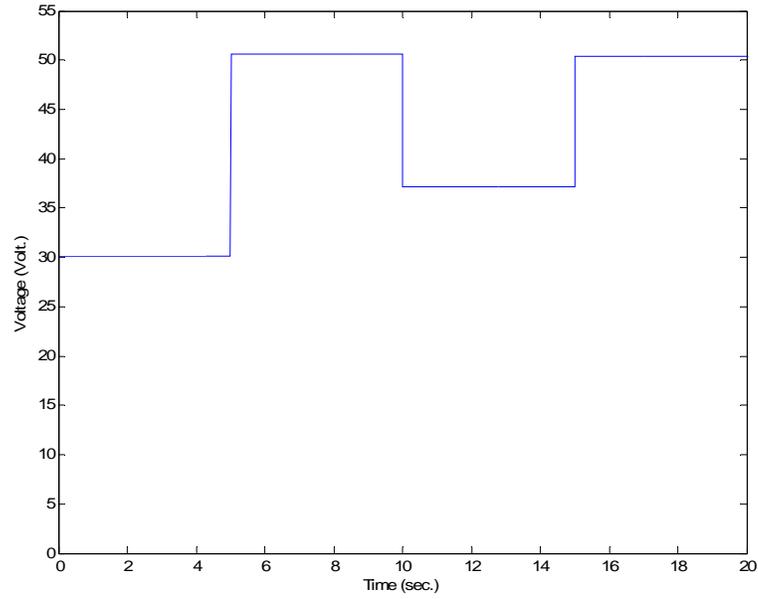


Figure 4.1 Input Voltage used for training the model

4.2 Simulation Results for Batch Two Stage Method

Table 4.1 show the estimation results for the noise free case.

Table 4.1 Parameter Estimates from Simulation for the Batch Two State Method

| Parameters | Real Value | Estimated value | %Error |
|-------------------|-------------------|------------------------|---------------|
| L_a | 0.4094 | 0.4094 | 0 % |
| R_a | 1.587 | 1.587 | 0 % |
| K_a | 0.3409 | 0.3409 | 0 % |
| J_m | 0.002387 | 0.002387 | 0 % |

The parameter estimates are presented in Table 4.1. These simulation parameters are noise free. From the results it could be seen that the estimates have no error.

The Figure 4.2 presents the load torque estimate in comparison with the real load torque vs. speed characteristic. From the Figure 4.3, it could be seen that the error is small. The only big difference occurs when the velocity reaches about 155 rad/s. This happens because the radial basis was trained with a speed range between 0 to 150 rad/s and also the speed input to the network was about a maximum of 140 rad/s (Figure 4.6). These results demonstrate that the approximation of the load characteristic is very good inside the training range, but not outside. As mentioned before the NN are good interpolators but not extrapolators.

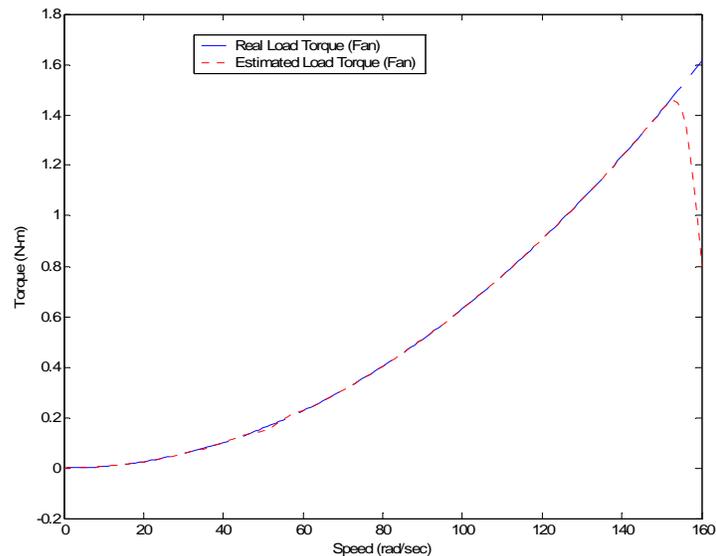


Figure 4.2 Real and Estimate Fan Load Torque Characteristic

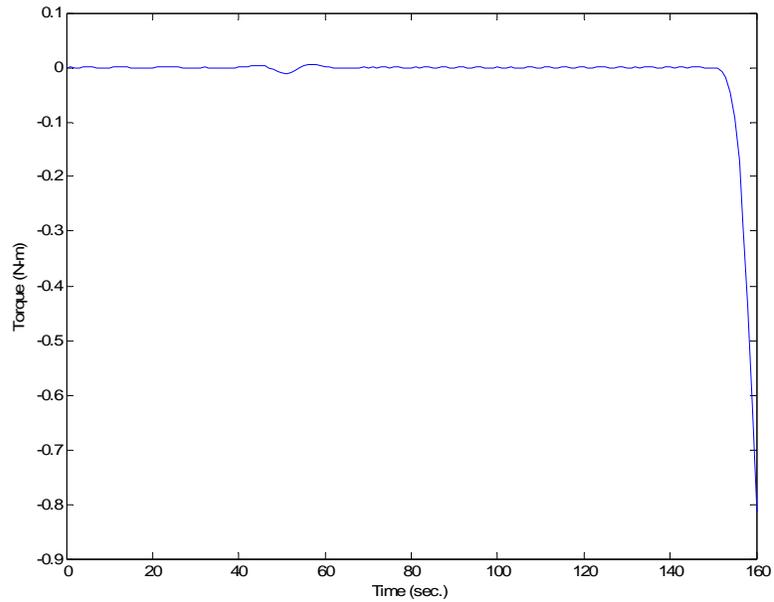


Figure 4.3 Estimated and Real Fan Load Torque Characteristic Estimation Error

Figure 4.4 compares the estimated and real current from the simulations. It could be seen from the figure that the curves are almost the same. Furthermore from Figure 4.5 it could be seen that the error is in the order of 1×10^{-7} A in the highest error which is on the transition from one step to the other during the curve. The rest of the curve shows good agreement. These demonstrate the good performance of the identification model for the noise free case as expected.

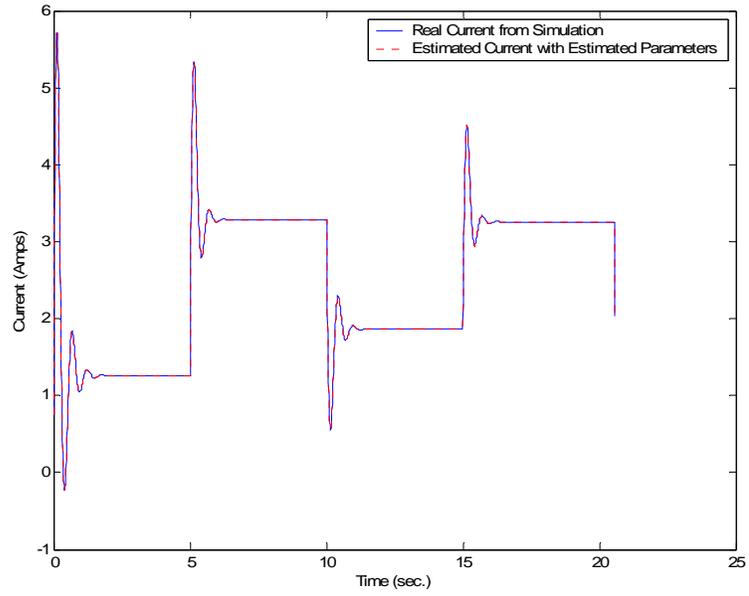


Figure 4.4 Real and Estimated Current

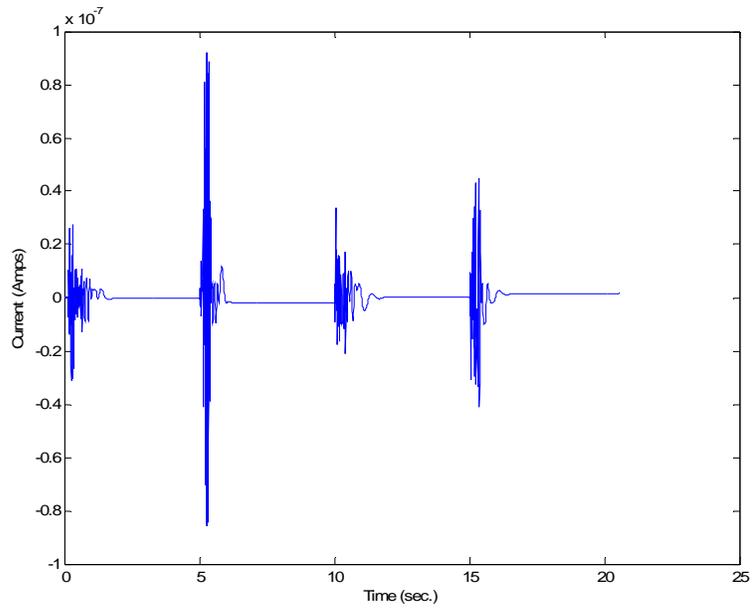


Figure 4.5 Current Estimation Error

Figure 4.6 presents the real and estimated speed. In this figure the estimated speed is behaving almost identical as the real speed from the simulation. Figure 4.7 presents the speed estimation error. The highest error magnitude is in the order of 1×10^{-12} rad/s, which is negligible. Again, as in the armature current, this error occurs in the transition between steps on the speed curve. This demonstrates that the parameter estimation algorithm works in ideal conditions.

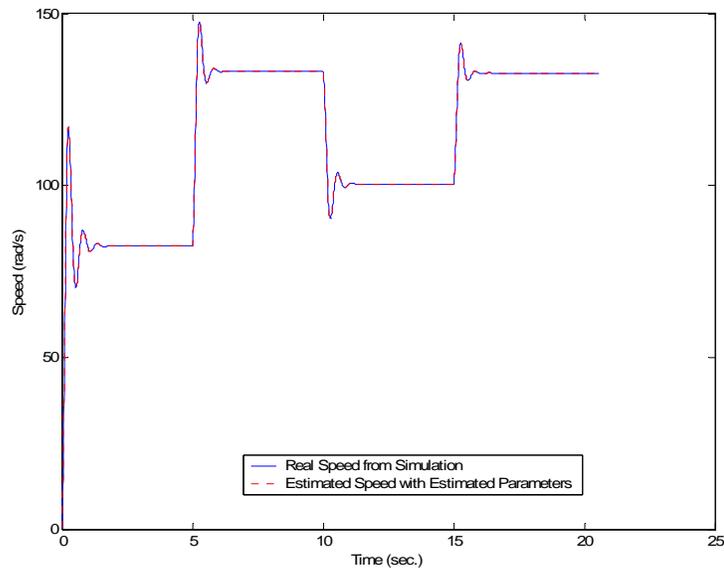


Figure 4.6 Real and Estimated Speed

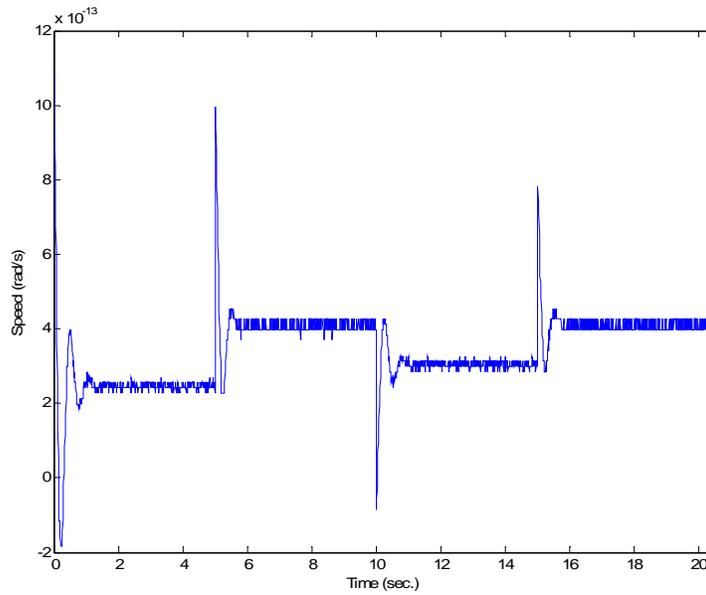


Figure 4.7 Speed Estimation Error

4.3 Validation Results for the Batch Two Stage Method

The validation process performed in this work was training the model with one set of data and using another set of data to validate the model. For validation, the second data set was selected inside the identification speed range. This action was implemented to make sure that the model performs correctly without adding some new errors because of not operating the model inside the training data range.

For the validation test, a new voltage input waveform was selected for this purpose. This new voltage input is presented in Figure 4.8.

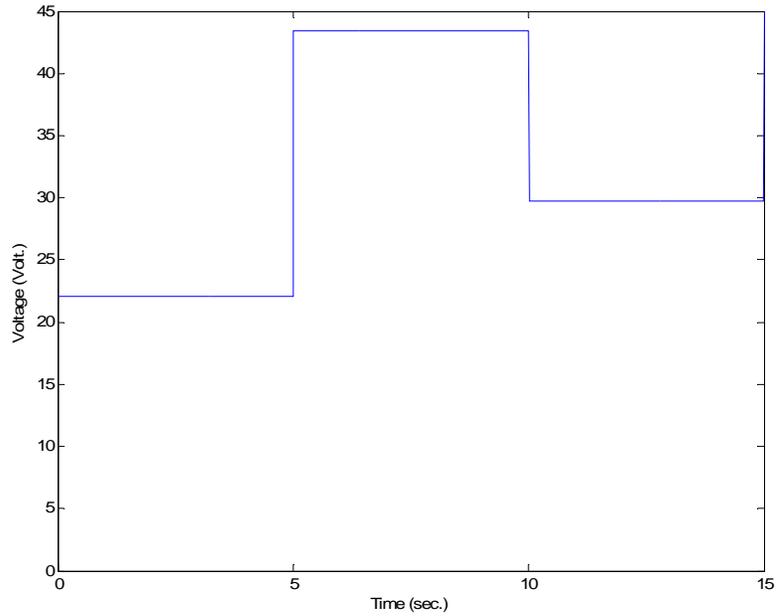


Figure 4.8 Input Armature Voltage used for Validation

The real and validation current is presented on Figure 4.9. Here the validation current curve behaves almost as the real simulated current. The error between the validation and real current is presented on Figure 4.10. The error has an average of almost zero with the exception of the initial transient of the step. This transient error was between -0.03 Amperes and 0.02 Amperes, which is less than 1 % of the maximum current.

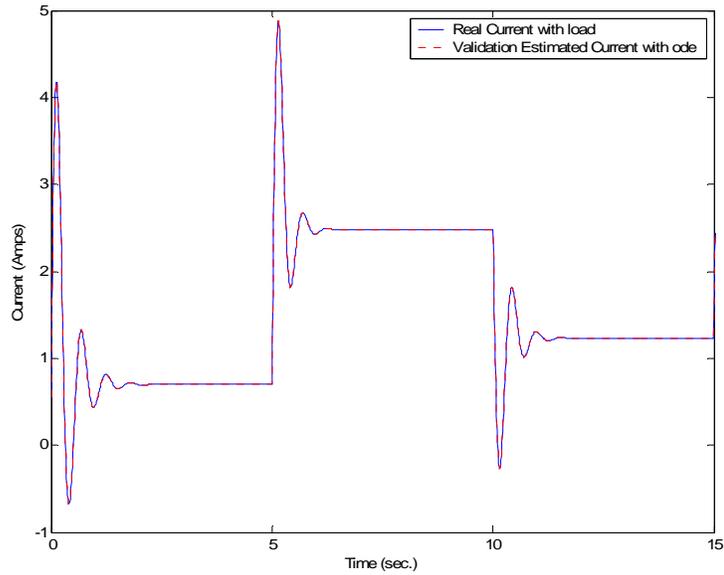


Figure 4.9 Real and Estimated Current Validation from Simulations

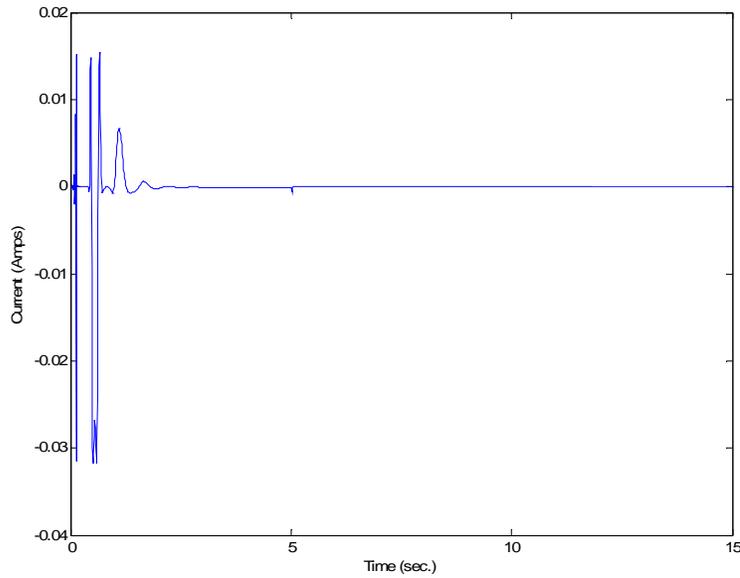


Figure 4.10 Current Estimation Error: Validation

The Figure 4.11 presents the validation and real simulated speed. Here both curves are almost identical. Moreover the Figure 4.12, that presents the error between the validation

and real simulated speed, shows that the error has an average of zero rad/s and with the highest error on the steps inside the curve.

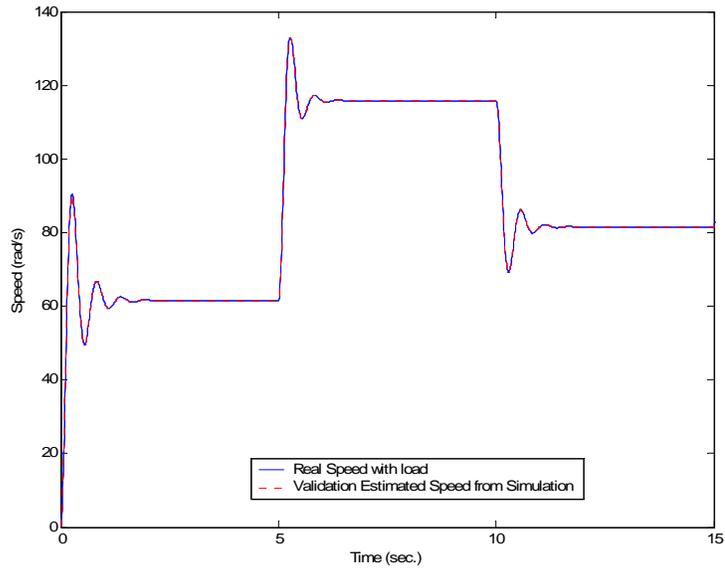


Figure 4.11 Real and Estimated Speed Validation from Simulations

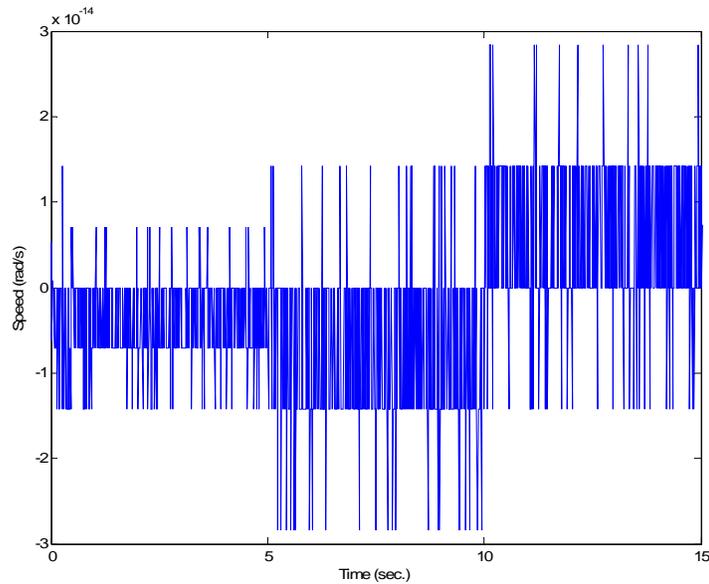


Figure 4.12 Validation Error from Real and Estimated Speed

4.4 Simulation Results for Batch Two Stage Method: Noise Case

Table 4.2 show the estimation results with Gaussian Noise of 0.1 dB.

Table 4.2 Parameter Estimates from Simulation for the Batch Two State Method

| Parameters | Real Value | Estimated value | %Error |
|------------|------------|-----------------|---------|
| L_a | 0.4094 | 0.4094 | 0 % |
| R_a | 1.587 | 1.586 | 0.06 % |
| K_a | 0.3409 | 0.34085 | 0.014 % |
| J_m | 0.002387 | 0.002385 | 0.083 % |

The parameter estimates are presented in Table 4.2. From the results it could be seen that the estimates have no error.

The Figure 4.13 presents the load torque estimate in comparison with the real load torque vs. speed characteristic. From the Figure 4.14, it could be seen that the error is small. The only big difference occurs when the velocity reaches about 155 rad/s. This happens because the radial basis was trained with a speed range between 0 to 150 rad/s and also the speed input to the network was about a maximum of 140 rad/s (Figure 4.17). These results demonstrate that the approximation of the load characteristic is very good inside the training range, but not outside.

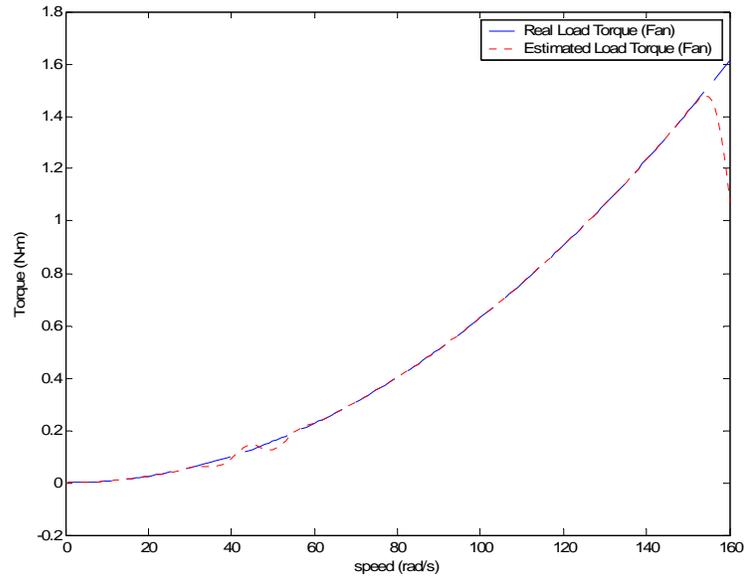


Figure 4.13 Real and Estimate Fan Load Torque Characteristic with Noise

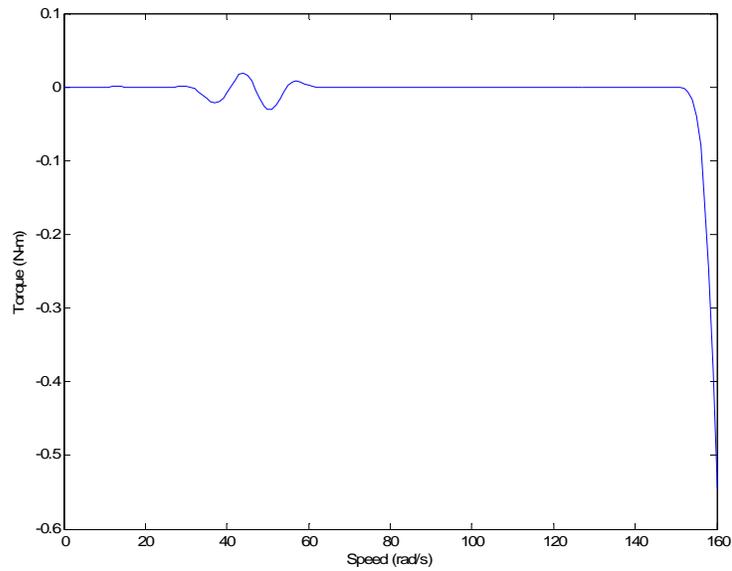


Figure 4.14 Estimated and Real Fan Load Torque Characteristic Estimation Error

Figure 4.15 compares the estimated and real current from the simulations. It could be seen from the figure that the curves are almost the same. Furthermore from Figure 4.16 it

could be seen that the error is in the order of 2.5×10^{-7} A in the highest error which is on the transition from one step to the other during the curve. The rest of the curve shows good agreement. These demonstrate the good performance of the identification model for the noise free case as expected.

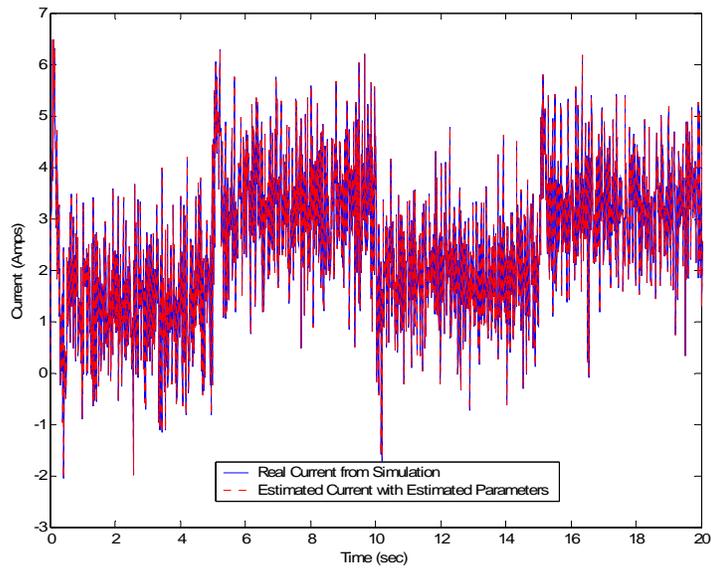


Figure 4.15 Real and Estimated Current with Noise

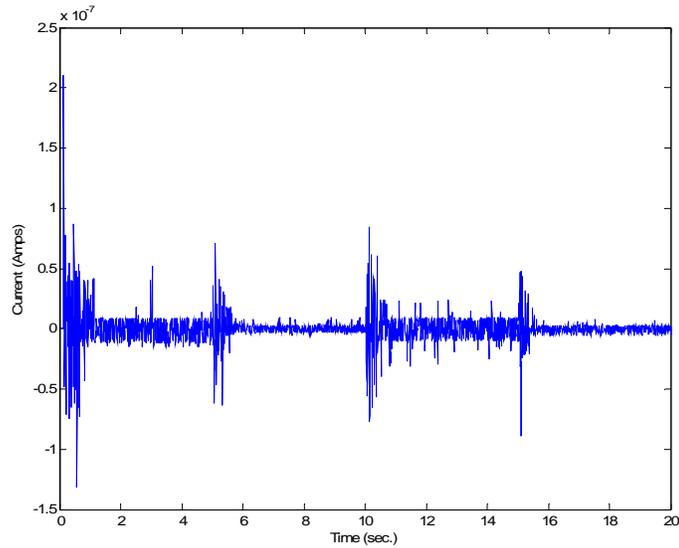


Figure 4.16 Current Estimation Error with Noise

Figure 4.17 presents the real and estimated speed. In this figure the estimated speed is behaving almost identical as the real speed from the simulation. Figure 4.18 presents the speed estimation error. The highest error magnitude is in the order of 3×10^{-13} rad/s, which is negligible. Again, as in the armature current, this error occurs in the transition between steps on the speed curve.

This results compare with the results without noise demonstrate the good performance of the two-stage method.

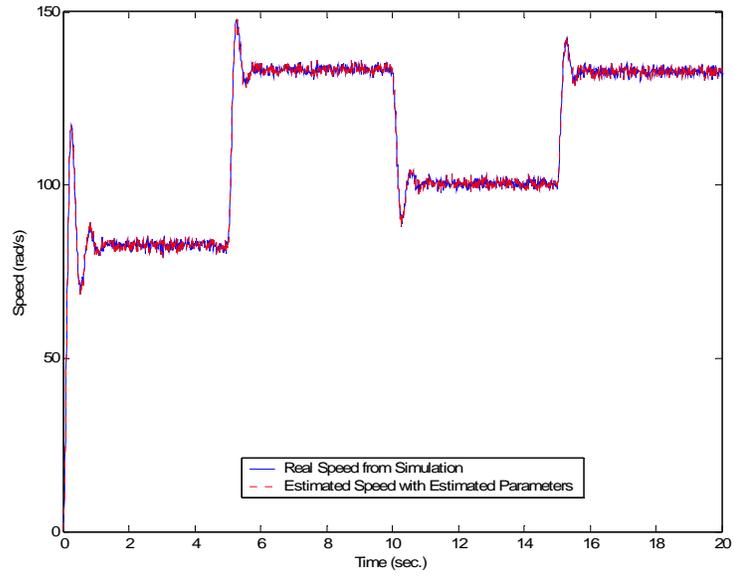


Figure 4.17 Real and Estimated Speed with Noise

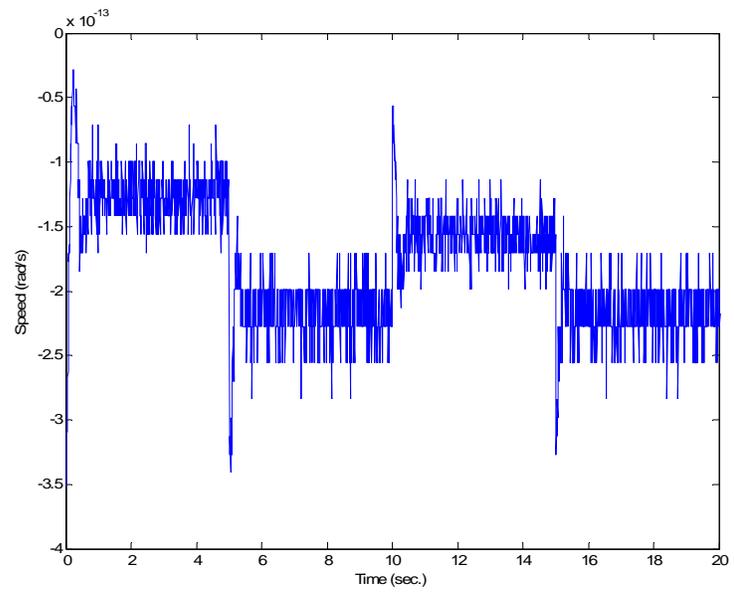


Figure 4.18 Speed Estimation Error with Noise

4.5 Validation Results for the Batch Two Stage Method: Noise Case

For the validation test, a new voltage input waveform presented in Figure 4.8 was used for this purpose. The real and validation current is presented on Figure 4.19. Here the validation current curve behaves almost as the real simulated current. The error between the validation and real current is presented on Figure 4.20. The error has an average of almost zero with the exception of the initial transient of the step. This transient error was between -0.03 Amperes and 0.005 Amperes, which is less than 1 % of the maximum current.

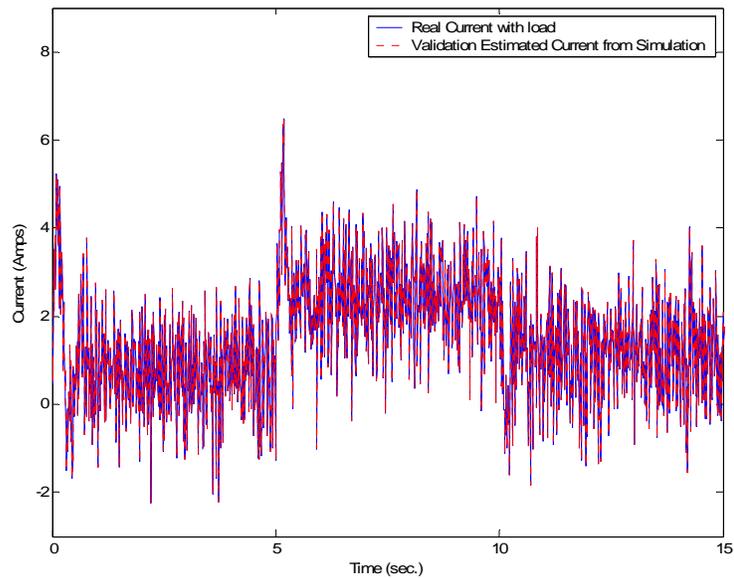


Figure 4.19 Real and Estimated Current Validation from Simulations with Noise

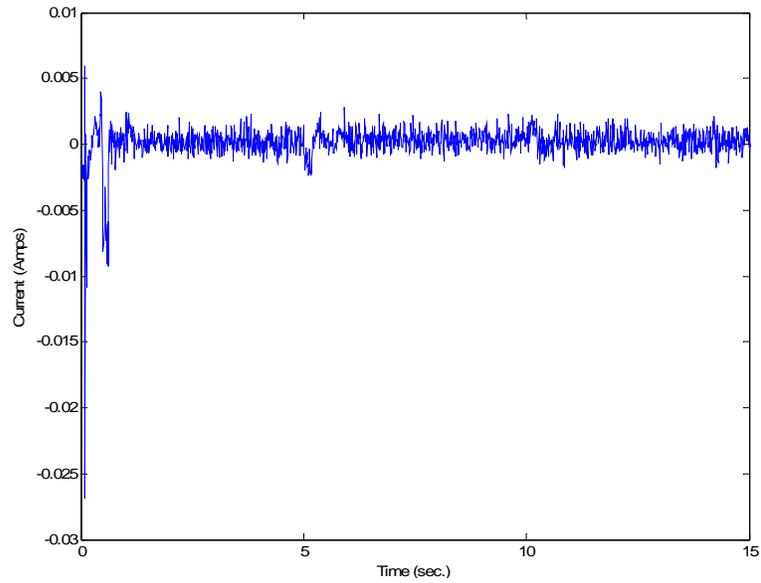


Figure 4.20 Current Estimation Error: Validation

The Figure 4.21 presents the validation and real simulated speed. Here both curves are almost identical. Moreover the Figure 4.22, that presents the error between the validation and real simulated speed, shows that the error has an average of approximately 0.02 rad/s and with the highest error on the steps inside the curve.

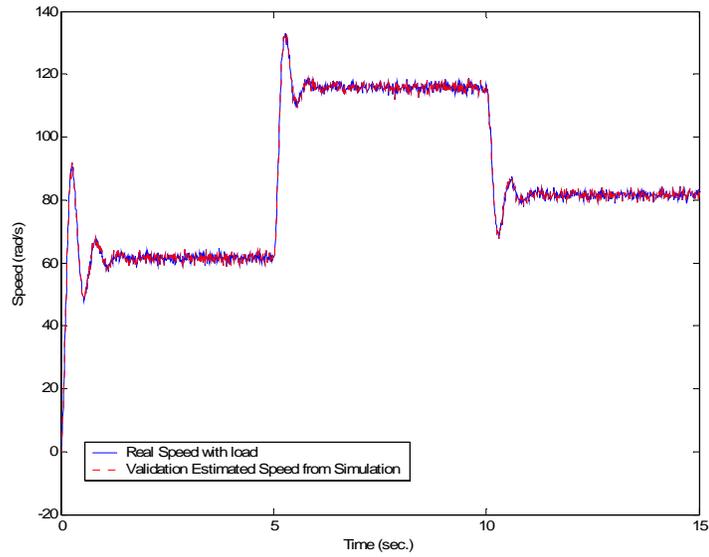


Figure 4.21 Real and Estimated Speed Validation from Simulations with Noise

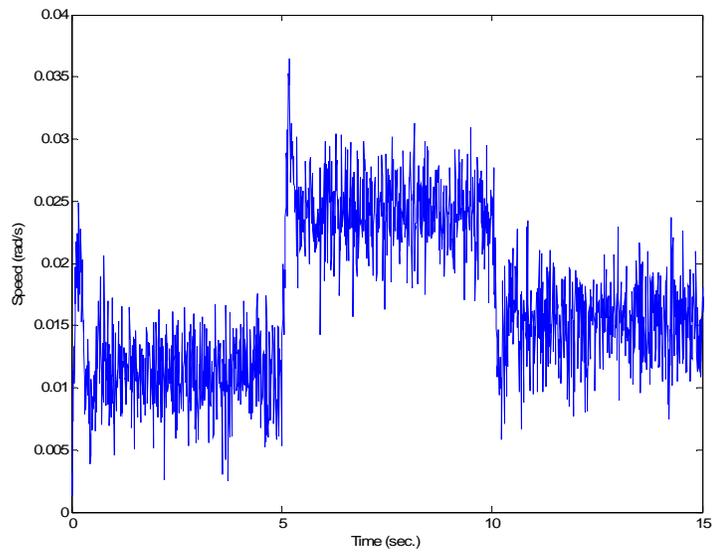


Figure 4.22 Validation Error from Real and Estimated Speed

4.6 Simulation Results for Recursive Two Stage Method

Table 4.3 Parameter Estimates for the Recursive Two Stage Method using Simulated Data

| Parameters | Real Value | Estimated value | %Error |
|------------|------------|-----------------|--------|
| L_a | 0.4094 | 0.4279 | 4.52 % |
| R_a | 1.587 | 1.564 | 1.44 % |
| K_a | 0.3409 | 0.3423 | 0.41 % |
| J_m | 0.002387 | 0.002357 | 1.26 % |

The parameter estimates using the recursive method are presented in Table 4.3. These parameters are estimated using the input voltage in Figure 4.1. From the results it could be seen that all the estimates have errors less than 5 %, being the highest error of 4.52% on the armature resistance (R_a). Figure 4.23 show the time response of the electrical parameter estimates. On this figure the response of the linear regression model parameters are presented. The blue line is R_a/L_a , the Red line is $1/L_a$, and the green line is K_a/L_a . It could be seen that the estimated parameters converge to a value near the nominal parameters values (dashed lines). These nominal values are 3.876 for the R_a/L_a line, 2.4425 for the $1/L_a$ line and 0.8326 for the K_a/L_a line. In steady state, the estimation errors were low as the Table 4.3 shows. The Figure 4.24 shows the estimates of K_a/J_m . The parameter almost converges to the nominal value of 142.815 (Blue dotted line). The estimated parameter is the dotted blue line. The error is almost zero as could be seen on Table 4.3.

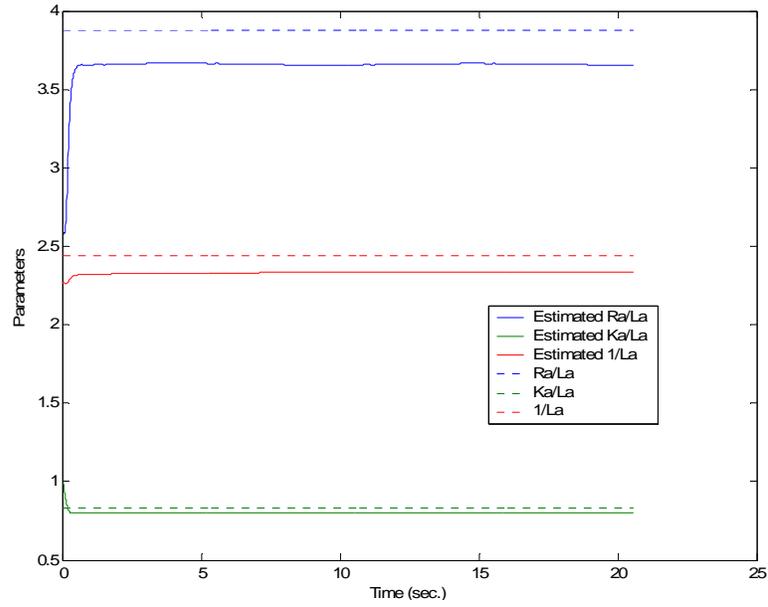


Figure 4.23 Real and Estimated Electrical Parameters: Recursive Estimator

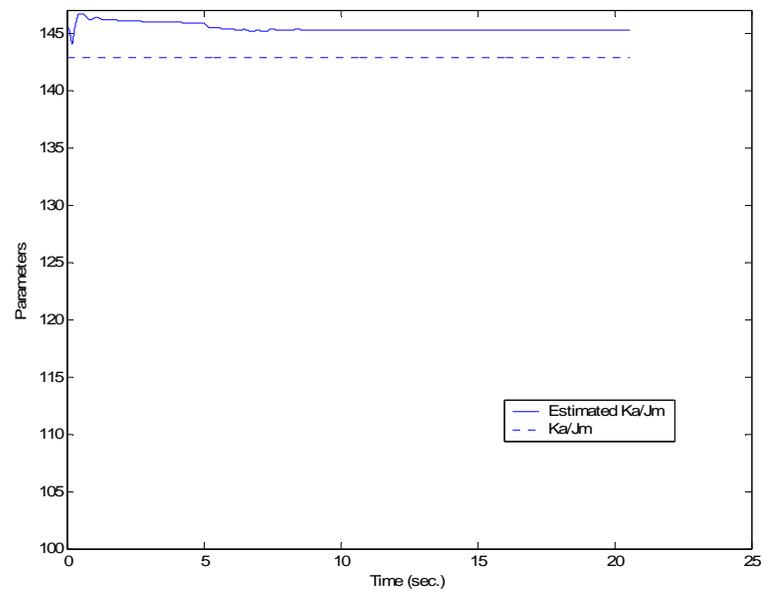


Figure 4.24 Real and Estimated Mechanical Parameters: Recursive Estimator

Figure 4.25 presents a comparison between the load torque characteristic estimated with the recursive algorithm and the real one. From Figure 4.26, it could be seen that the error is small, but is higher than in the batch case presented in Section 4.2. The largest difference occurs when the velocity reaches about 155 rad/s. This is because the neural network was trained for a velocity range between 0 to 150 rad/s. These results demonstrate that the approximation of the load characteristic is very good inside the training range.

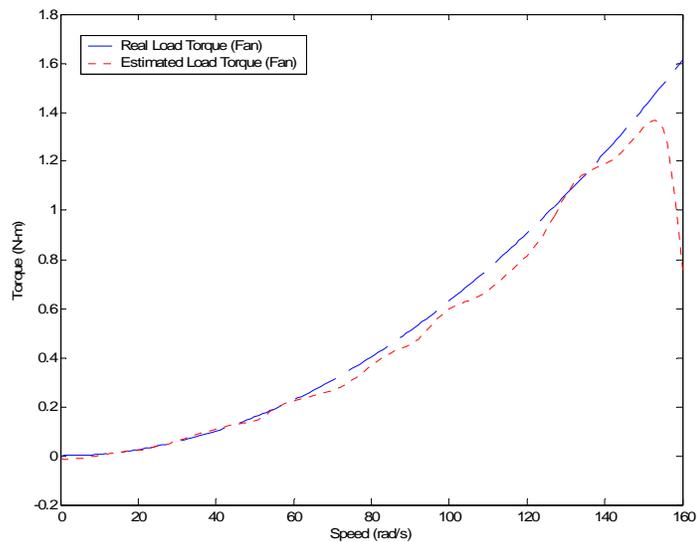


Figure 4.25 Real and Estimate Fan Load Torque: Recursive Estimator

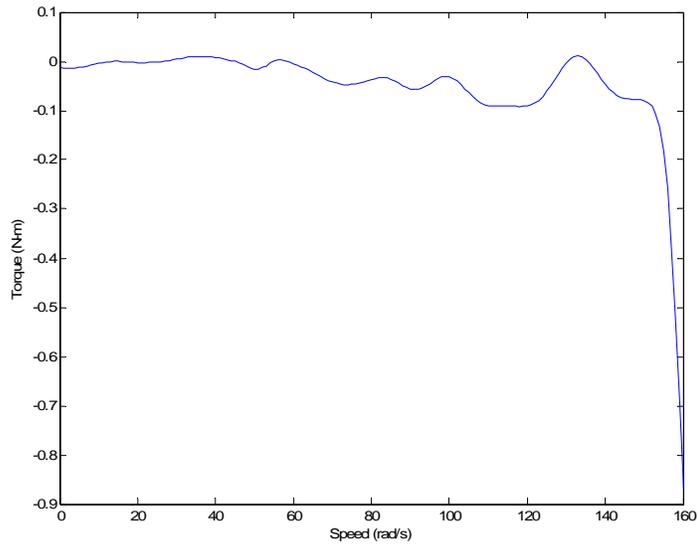


Figure 4.26 Real and Estimate Fan Load Torque Error from Recursive Simulation

Figure 4.27 shows the estimated and real current from the recursive simulations. It could be seen that the curves are almost the same. Furthermore, from Figure 4.28, the mean error is -0.1 Amperes. The highest errors occur during the step changes in the curve, which is something normal. The overall performance of the estimator is good because the value of 0.1 Amperes is low compared with the maximum current of 3.2 Amperes. These results demonstrate the good performance of the recursive parameter estimation algorithm.

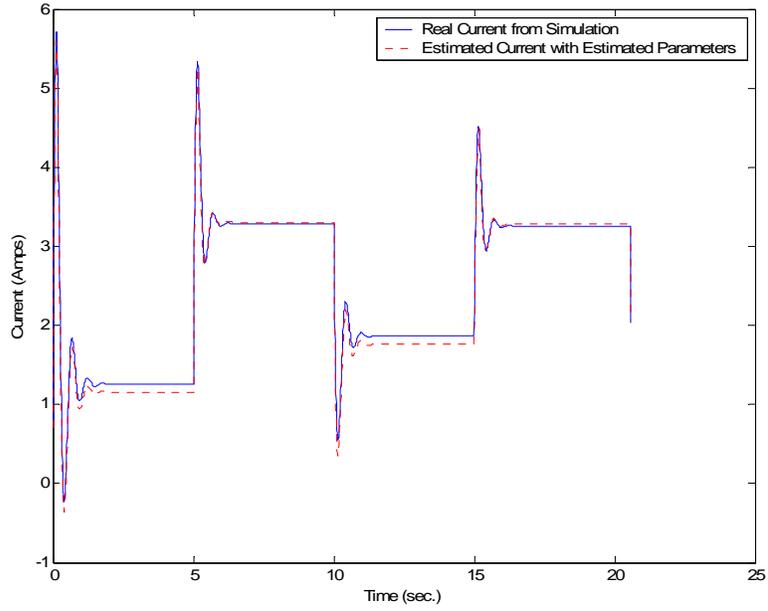


Figure 4.27 Real and Estimated Current from Recursive Simulation

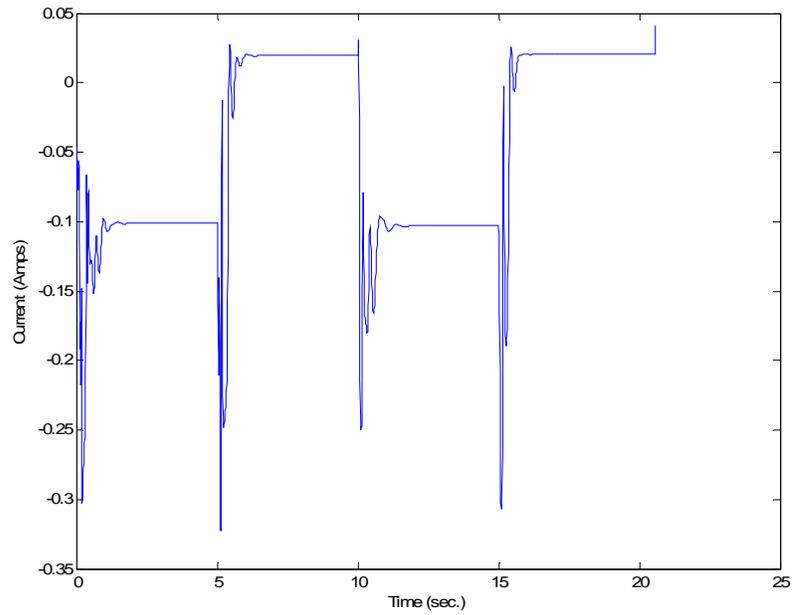


Figure 4.28 Real and Estimated Current Error from Recursive Simulation

In Figure 4.29, the real and estimated speed from the recursive method is presented. On this figure the estimated speed is behaving almost the same as the real speed from the simulation. In Figure 4.30, the estimation error is presented. The maximum error magnitude is about -0.2 rad/s. This is good because the highest speed is about 140 rad/s. Highest errors occur during fast changes in speed. In the steady state regions the average error is approximately -0.2 rad/s.

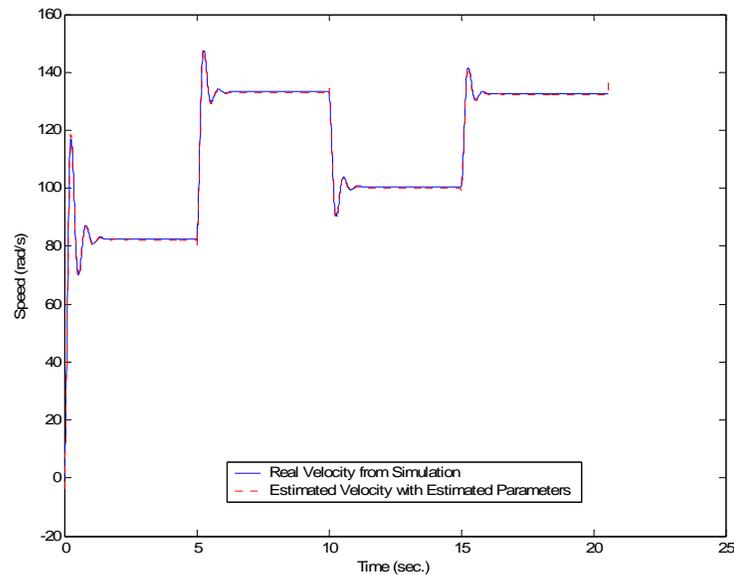


Figure 4.29 Real and Estimated Speed from Recursive Estimator

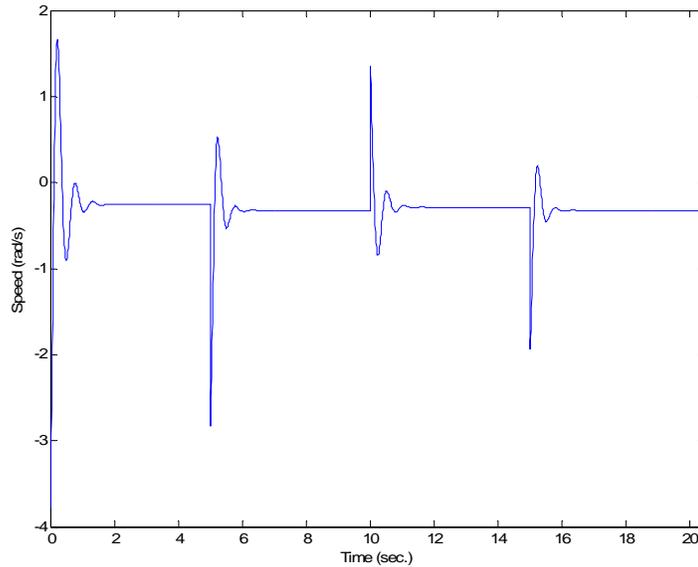


Figure 4.30 Speed Error from Recursive Estimator

4.7 Validation Results for the Recursive Two Stage Method

For the validation test, the same voltage input used for the validation of the batch method was selected for the recursive case. This voltage input is presented on Figure 4.8.

The real and estimated current from the recursive case is presented on Figure 4.31. Here the estimated current behaves almost as the real current. The error between the validation and real current is presented on Figure 4.32. The error has an average of almost zero Amperes with the exception of the period between each step, but it was less than 0.25 Amperes. The error varies between +0.05 and -0.05 Amperes. This error is minimum when compared with the current highest value of 2.5 Ampere. The identified model shows good performance with the validation data.

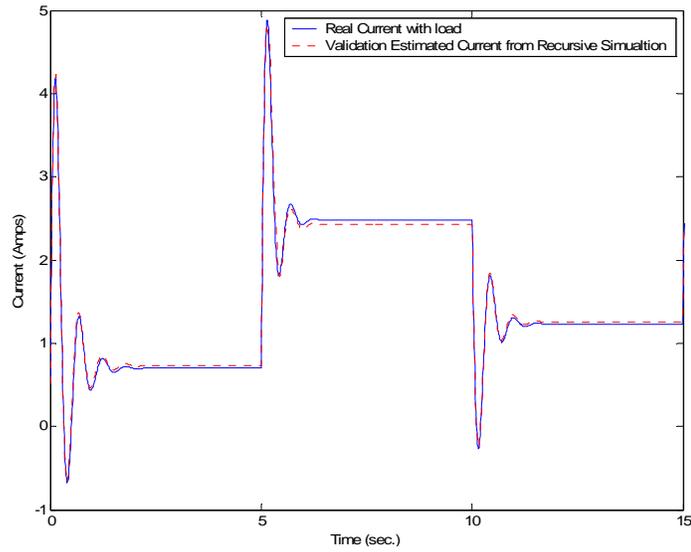


Figure 4.31 Real and Estimated Current Validation from Recursive Estimator

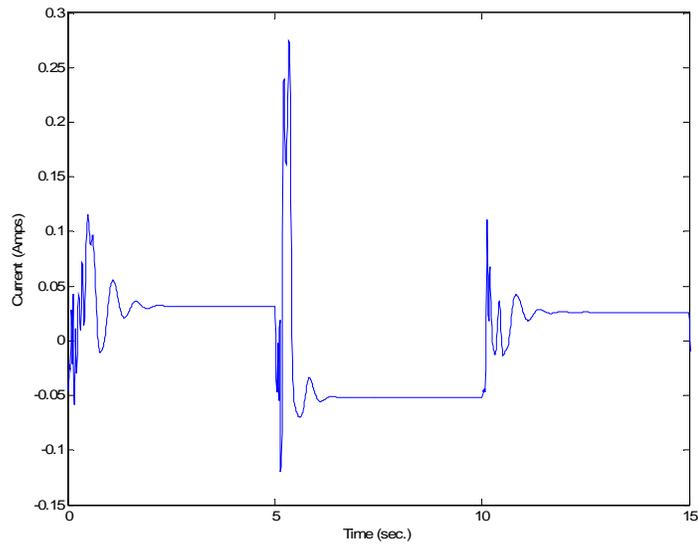


Figure 4.32 Validation error from Real and Estimated Current from Recursive Estimator

The Figure 4.33 presents the validation and real speed for the recursive case. Figure 4.34, presents the error between the validation and the real simulated current. The average

error is -0.2 rad/s with the highest error during speed changes. This again shows good performance for the identified model.

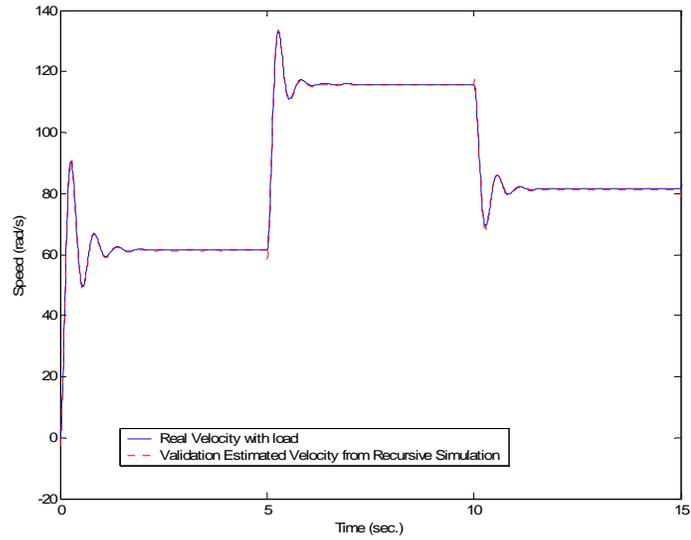


Figure 4.33 Real and Estimated Speed Validation for Recursive Estimation

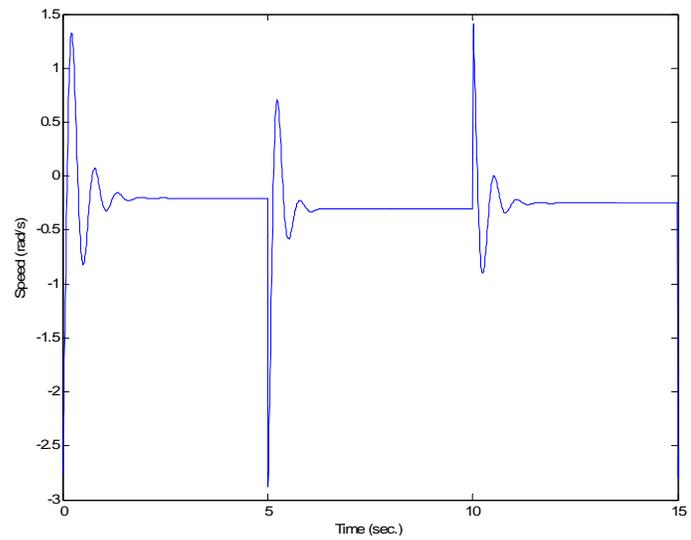


Figure 4.34 Error between Real and Estimated Speed for Recursive Estimation: Validation

From this validation results and the validations results of the batch two-stage method, we could say that the identified model shows good performance to reproduce the response of the system to an input different from training data. Also the proposed recursive algorithm is good for identifying the system although there are slightly large errors in the estimated physical parameters.

4.8 Simulation Results for Recursive Two Stage Method: Noise Case

Table 4.4 Parameter Estimates for the Recursive Two Stage Method using Simulated Data

| Parameters | Real Value | Estimated value | %Error |
|------------|------------|-----------------|--------|
| L_a | 0.4094 | 0.4279 | 4.52 % |
| R_a | 1.587 | 1.563 | 1.51 % |
| K_a | 0.3409 | 0.3422 | 0.38 % |
| J_m | 0.002387 | 0.002371 | 0.67 % |

The parameter estimates using the recursive method are presented in Table 4.4. These parameters are estimated using the input voltage in Figure 4.1. From the results it could be seen that all the estimates have errors less than 5 %, being the highest error of 4.52% on the armature resistance (R_a). Figure 4.35 show the time response of the electrical parameter estimates. On this figure the response of the linear regression model parameters

are presented. The blue line is R_a/L_a , the Red line is $1/L_a$, and the green line is K_a/L_a . It could be seen that the estimated parameters converge to a value near the nominal parameters values (dashed lines). These nominal values are 3.876 for the R_a/L_a line, 2.4425 for the $1/L_a$ line and 0.8326 for the K_a/L_a line. In steady state, the estimation errors were low as the Table 4.4 shows. The Figure 4.36 shows the estimates of K_a/J_m . The parameter almost converges to the nominal value of 142.815 (Blue dashed line). The estimated parameter is the dotted blue line. The error is almost zero as could be seen on Table 4.4.

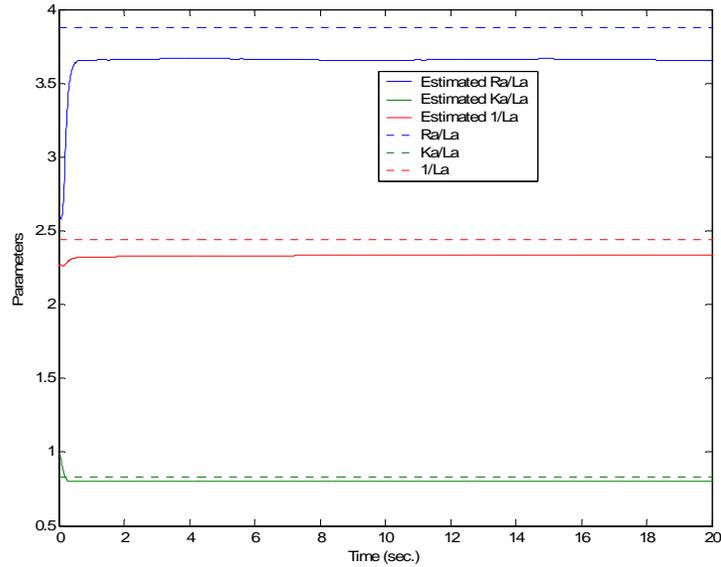


Figure 4.35 Real and Estimated Electrical Parameters: Recursive Estimator with Noise

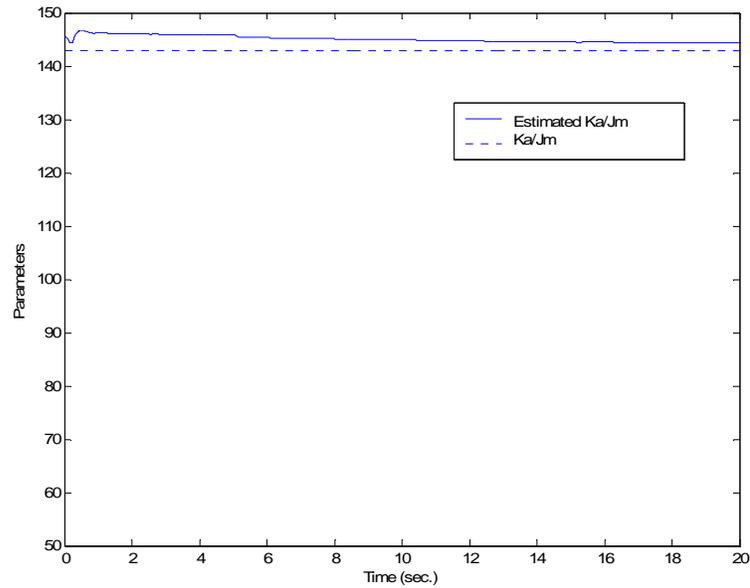


Figure 4.36 Real and Estimated Mechanical Parameters: Recursive Estimator with Noise

Figure 4.37 presents a comparison between the load torque characteristic estimated with the recursive algorithm and the real one. From Figure 4.38, it could be seen that the error is small, but is higher than in the noise free case presented in Section 4.6. The largest difference occurs when the velocity reaches about 155 rad/s. This is because the neural network was trained for a velocity range between 0 to 150 rad/s. These results demonstrate that the approximation of the load characteristic is very good inside the training range.

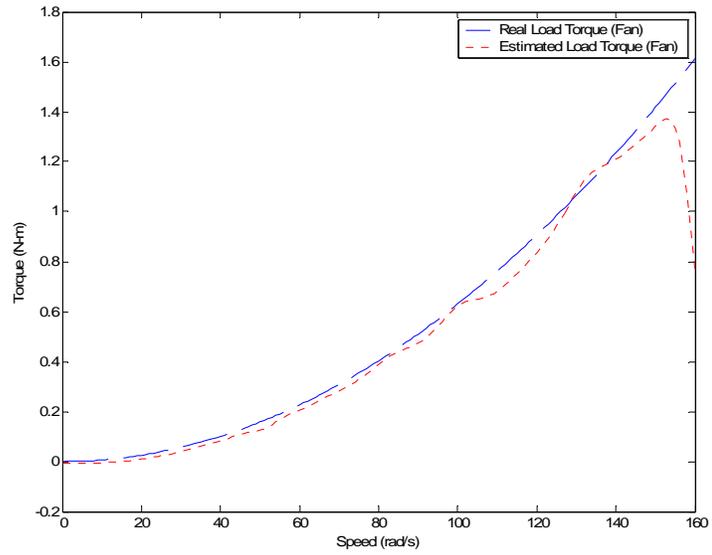


Figure 4.37 Real and Estimate Fan Load Torque: Recursive Estimator with Noise

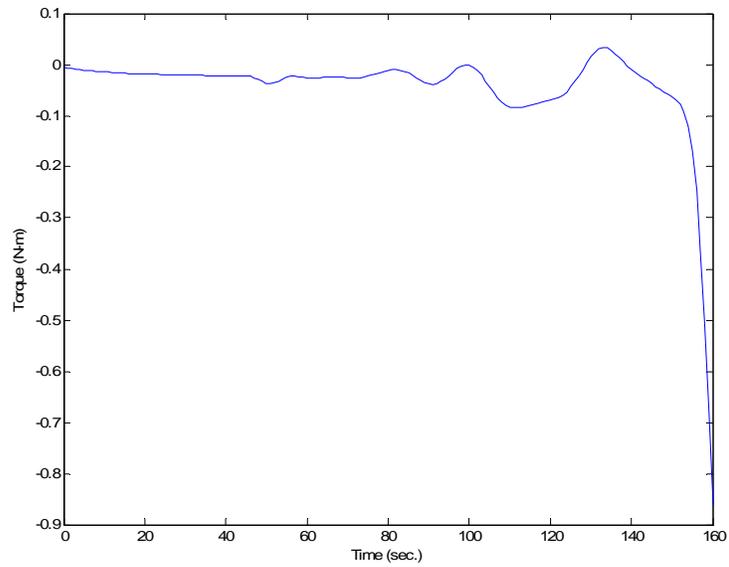


Figure 4.38 Real and Estimate Fan Load Torque Error from Recursive Simulation

Figure 4.39 shows the estimated and real current from the recursive simulations. It could be seen that the curves are almost the same. Furthermore, from Figure 4.40, the average error is 0.01 Amperes. The highest errors occur during the step changes in the curve, which is something normal. The overall performance of the estimator is good because the value of 0.01 Amperes is low compared with the maximum current of almost 4.0 Amperes. These results demonstrate the good performance of the recursive parameter estimation algorithm.

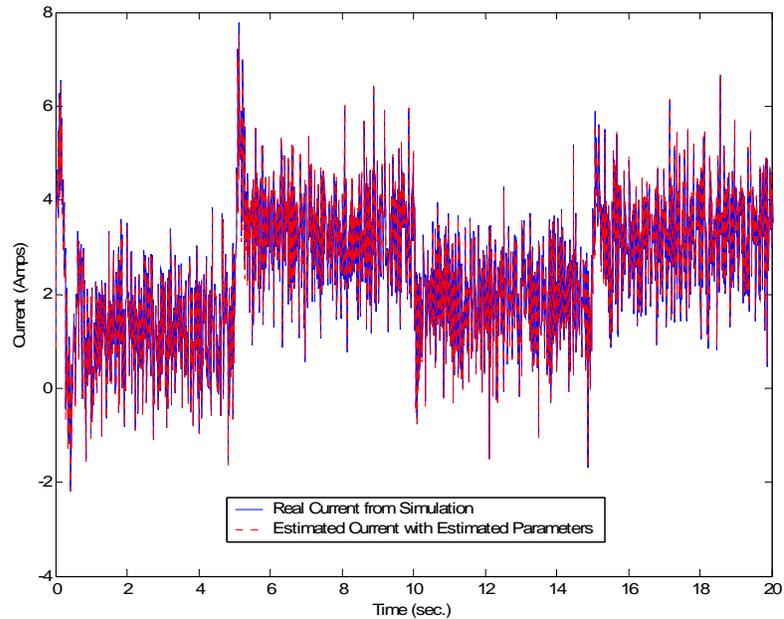


Figure 4.39 Real and Estimated Current from Recursive Simulation: Noise Case

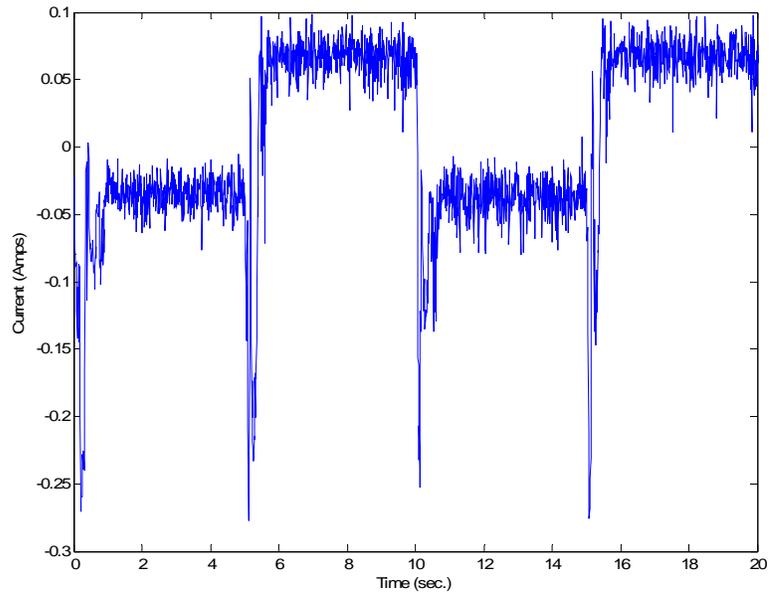


Figure 4.40 Real and Estimated Current Error from Recursive Simulation

In Figure 4.41, the real and estimated speed from the recursive method is presented. On this figure the estimated speed is behaving almost the same as the real speed from the simulation. In Figure 4.42, the estimation error is presented. The maximum error magnitude is about 2 rad/s. This is good because the highest speed is about 140 rad/s. Highest errors occur during fast changes in speed. In the steady state regions the average error is approximately -0.2 rad/s.

This recursive results compare with the recursive results without noise demonstrate the good performance of the two-stage method.

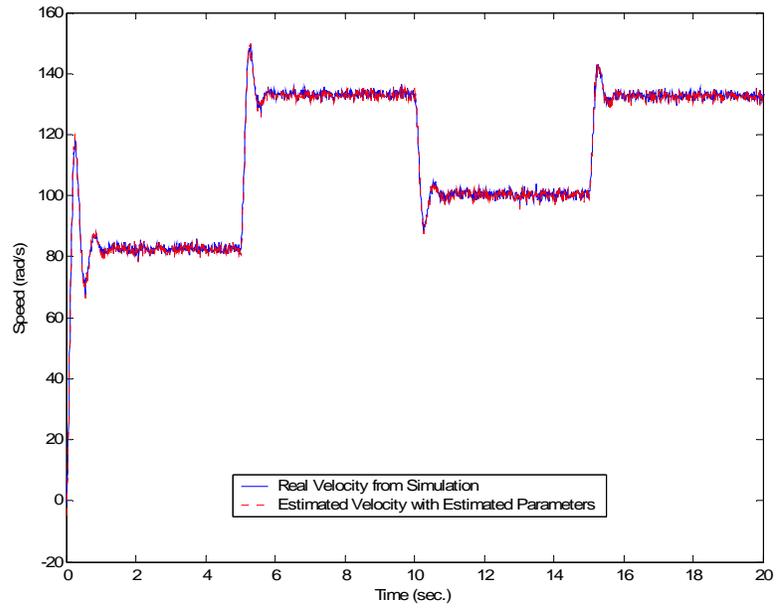


Figure 4.41 Real and Estimated Speed from Recursive Estimator with Noise

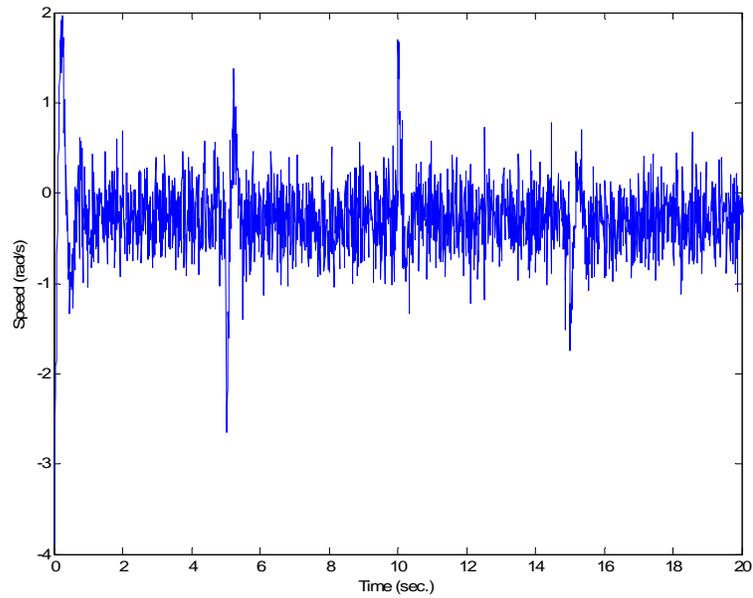


Figure 4.42 Speed Error from Recursive Estimator

4.9 Validation Results for the Recursive Two Stage Method: Noise Case

For the validation test, the same voltage input used for the validation of the batch method was selected for the recursive case. This voltage input is presented on Figure 4.8.

The real and estimated current from the recursive case is presented on Figure 4.43. Here the estimated current behaves almost as the real current. The error between the validation and real current is presented on Figure 4.44. The error has an average of almost zero Amperes with the exception of the period between each step, but it was less than 0.3 Amperes. The error varies between +0.05 and -0.05 Amperes. This error is minimum when compared with the current highest value of almost 3 Amperes. The identified model shows good performance with the validation data.

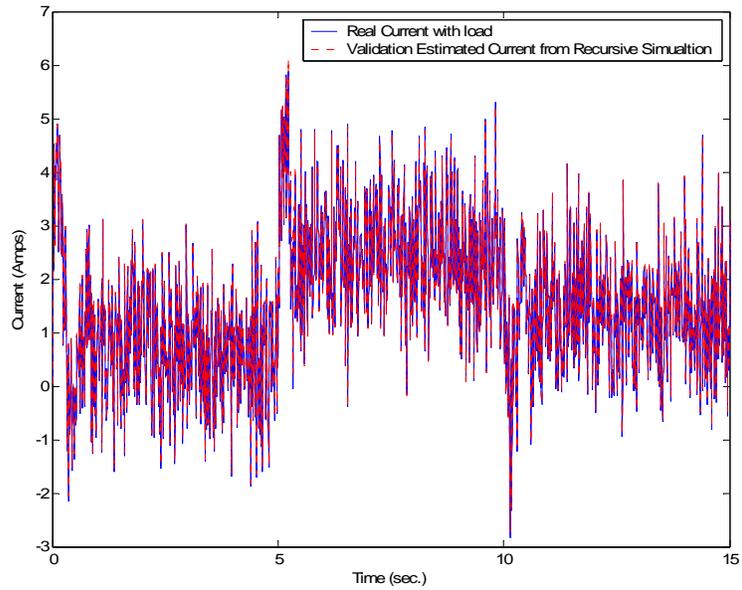


Figure 4.43 Real and Estimated Current Validation from Recursive Estimator with Noise

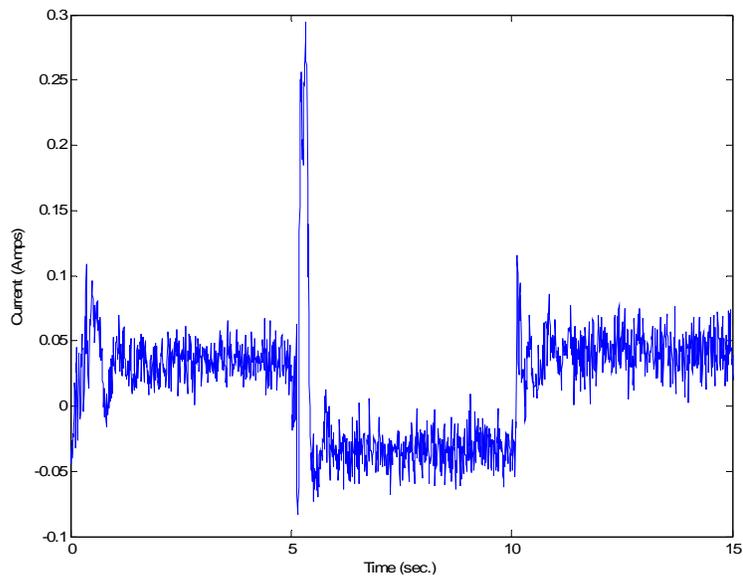


Figure 4.44 Validation error from Real and Estimated Current from Recursive Estimator

The Figure 4.45 presents the validation and real speed for the recursive case. Figure 4.46, presents the error between the validation and the real simulated current. The average error is -0.1 rad/s with the highest error during speed changes. This again shows good performance for the identified model.

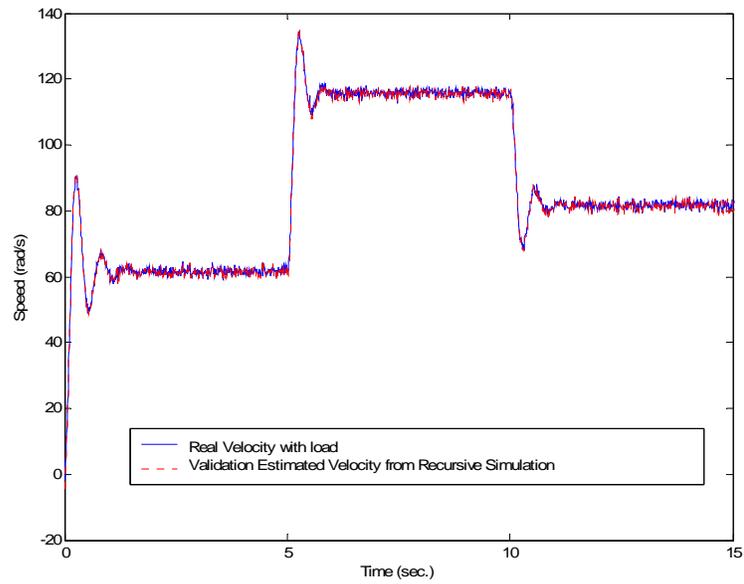


Figure 4.45 Real and Estimated Speed Validation for Recursive Estimator with Noise

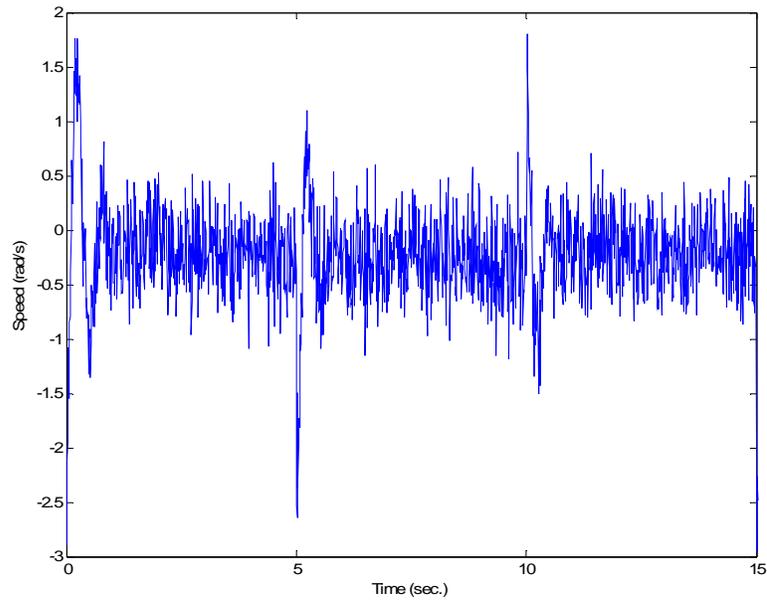


Figure 4.46 Error between Real and Estimated Speed for Recursive Estimation: Validation

From this validation results and the validations results of the batch two-stage method with noise, we could say that the identified model shows good performance to reproduce the response of the system to an input different from training data. Also the proposed recursive algorithm is good for identifying the system although there are slightly large errors in the estimated physical parameters.

CHAPTER 5

EXPERIMENTAL RESULTS

This chapter presents the experimental work to test and validate the identification algorithms. A block diagram of the experimental setup is presented on Figure 5.1. Each of these components is going to be explained in detail next.

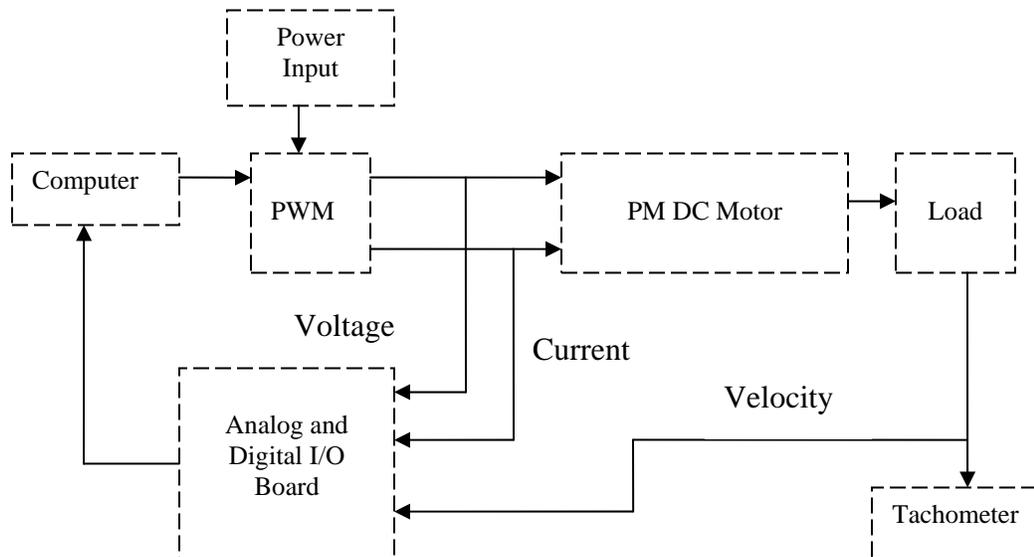


Figure 5.1 Experimental Setup

5.1 Experimental Setup Overview

The experimental setup consists of a Personal Computer with a dSPACE[®] data acquisition board, signal conditioning modules, a PWM power driver amplifier with power supply for the motor drive, and a permanent magnet DC motor with a tachometer. Figure 5.1

shows the block diagram of this system. The characteristics of the individual components are as follows:

- **PM DC Motor:** The motor is a ½ HP, 90 Volts, PM DC motor of General Electric Company with an speed of 2500 RPM and an armature current of 5.2 A.
- **Computer:** The Computer is a personal computer that is expected to read the voltage, current, and speed feedback, then calculate the parameters of the motor.
- **Analog and Digital I/O Board:** The analog/input channels are used to sample discrete values of armature current and output voltage from the tachometer and store them into the computer memory.
- **PWM Driver:** Is a Copley 421 PWM servo amplifier, which provides a 5 Amperes continuous, 10 Amperes peak at switching frequency of 25 kHz. The servo amplifier is configured to work as a DC voltage amplifier with a gain of 10.
- **Power Input:** The power input to the PWM driver is an uncontrolled rectifier that produces 135 Volts at 6 Amperes.
- **Load:** A fan of 16” with 26 degrees of pitch was used.
- **Tachometer:** Instrument to measure the speed of the motor.

5.1.1 DSPACE Data Acquisition Board

As mentioned at the beginning of this chapter a dSPACE data acquisition board is used for obtain the measurements needed from the motor. The dSPACE board used on this work was the DS1103 board.

Special attention must be taken when loading a heavy computational process to the DS1103. If the test is too complex, the simulation step time must be increase in order to avoid overloading of the control board, but increasing the step time too much will introduce convergence related problems. The main reason is because of the computer used for this work, which is a Gateway Computer with a 200 MHz Pentium 2 microprocessor.

The dSPACE system consists of the DS1103 control board and Matlab[®] software. DS1103 main processing unit is Texas Instrument's TMS320C31 DSP with 60Mhz system clock. It also includes a second Texas Instrument's TMS320P14 with 25Mhz clock slave DSP used as a co-processing unit.

DS1103 provide twenty ADC (for data acquisition), eight DAC, fifteen programmable digital I/O and ISA bus interface (to connect the board to a personal computer). For a more detailed description of the DS1103 please refer to dSPACE Manual [17]. The DS1103 board is installed in a personal computer used to control the drive system, through an ISA bus.

Matlab[®] Real-Time workshop and Simulink[™] is the main software tools used to program the DS1103.

Once the simulation is started, data can be stored with a software called Trace. Any signal from the simulation model can be stored or displayed with this software. On a test simulation, the DS1103 will basically receive voltage signals representing armature currents, armature voltage and rotor speed from the DC motor. Also it will produce the input voltage control signals to the PWM driver. The Matlab[®] Simulink block diagram used for control

and data collection from the DC motor is presented in Figure 5.2. Inside this figure is also presented the state variable filters used to calculate the derivatives of the armature current and the rotor speed. These derivatives are needed when implementing the two-stage algorithm to estimate the system parameters. The dSPACE blocks that interface the computer and the drive to read and input data are also shown in this figure.

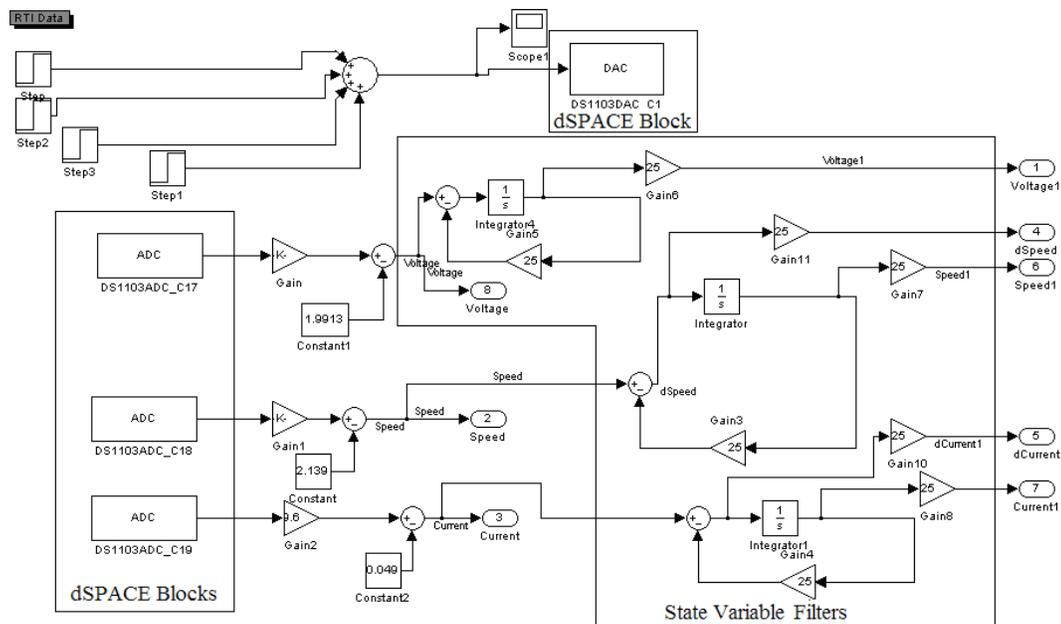


Figure 5.2 dSPACE Matlab Simulink Block Diagram

5.1.2 Signal Conditioning for Data Acquisition

Several signal conditioning circuits are built in order to read the armature current, the rotor speed, and the armature voltage of the DC motor drive.

Armature Current Measurement:

The circuit presented on Figure 5.3 was used to measure the armature current. We used a resistor $R_i=0.01\Omega$ in series with the motor to sense the voltage drop that is proportional to the armature current. The output of the sensor is given by the following equation:

$$V_i = R_i * i_a \quad (5.1)$$

The range of this voltage drop is in the order of millivolts. An amplifier circuit was used to amplify this sensor voltage. One thing to take into consideration when designing this amplifier is the isolation of this circuit. To solve this, a ground loop isolation stage in the drive system is used to reduce noise on the signals. Electromagnetic interference (EMI) is very common on drive systems because of the switching of the power inverters. This EMI can affect all parts of the drive system if it is not considered during design. The ground loop isolator stage was build to break a ground loop between the dSPACE stage and the co-processing stage. The ground loop isolator circuit use a low distortion Operational Amplifier and isolation amplifiers. This approach helps isolate the voltage level difference between grounds of both stages.

The low distortion operational amplifier used on this work was the Burr-Brown OPA2604. This amplifier was used at the output of the Isolation amplifier and on the Post Amplification Stage.

The isolation is done with Agilent Technologies isolation amplifier HCPL-7840. The HCPL-7840 consists of a sigma-delta analog-to-digital converter optically coupled to a digital-to-analog converter. Analog signals like armature Current (i_a) and armature Voltage

(V_a) are decoupled using these amplifiers to break all electrical connections between stages. The isolation circuit presented on Figure 5.3 was taken from the HCPL-7840 manual [18] for motor signal sensing. Some modifications to fit our work were made, like adding a post amplification stage.

Also its to reduce the effect of the EMI, all components are enclosed in metal cases and shielded cable was plenty used. Electronic components were selected with a high immunity to EMI, and circuit design was also done with noise reduction in mind.

On each isolation circuit a post-amplification stage with a gain of 2 was used. This stage used the low distortion amplifier OPA2604. Now including all of the gains the circuit stage, the total gain of this circuit is described with:

$$V_t = 0.915 * i_a \quad (5.2)$$

This voltage (V_t) was measured by the dSPACE board and converted by the computer to the corresponding armature current.

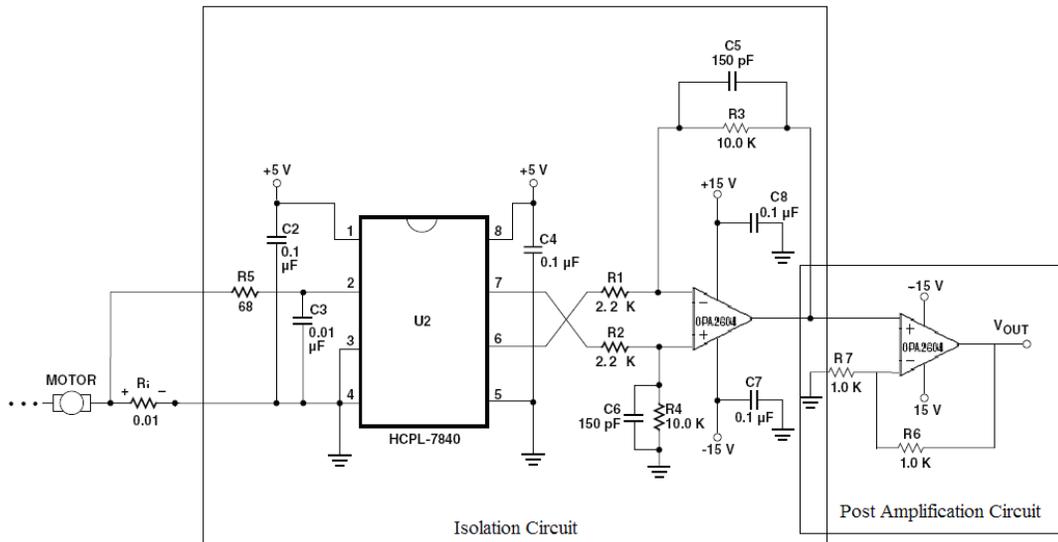


Figure 5.3 Current Measurement Circuit

The implemented passive low pass filter in the OPA2604 amplifiers was a capacitor C in parallel with a resistor R. The cutoff frequency of the filters is calculated with equation $f_c = \frac{1}{2\pi RC}$. For this armature current circuit we used a value of $R = 10\text{k}\Omega$ and $C = 150\text{pf}$, for a cutoff frequency of 106 kHz.

Armature Voltage Measurement:

The Figure 5.4 presents the Armature Voltage measurement circuit implemented. For the measurement of the Armature Voltage, a similar circuit for the current was used. To step down the voltage, to input it to the dSPACE card a voltage divider was used. The input voltage to the conditioning circuit given by:

$$V_i = \left(\frac{R_i}{R_i + R_j} \right) * V_a \quad (5.3)$$

where $R_i=3.9 \text{ K}\Omega$ and $R_j=4.7 \text{ M}\Omega$. The rest of the circuit is the same as the used for the armature current. The post amplification circuit has the same structure presented for the current measurement but with a gain of 1.68. Now combining all of the gains, of this amplifier the total gain of this circuit is given by:

$$V_t = 0.05131 * V_a \quad (5.4)$$

This voltage (V_t) was measured by the dSPACE board and converted by in the computer as a the armature voltage.

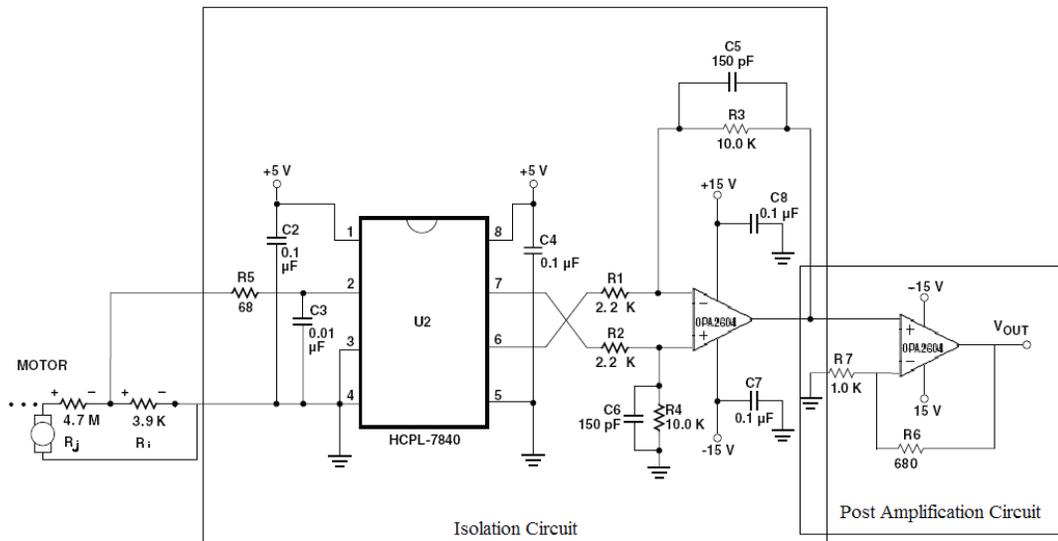


Figure 5.4 Voltage Measurement Circuit

The implemented active low pass filter was a capacitor C in parallel with a resistor R. The cutoff frequency of the filters is calculated with equation $f_c = \frac{1}{2\pi RC}$. For this armature voltage circuit we used a value of R = 10kΩ and C = 150pf, for a cutoff frequency of 106 kHz.

Speed Measurement:

The rotor angular speed is measured by a tachometer coupled to the shaft of the motor. The tachometer employed is a servo DC motor with a gain of 5V/900RPM. The Figure 5.5 shows the circuit used for the speed measurements. To remove the noise of this signal, a filter with a cutoff frequency at 723 Hz was used to scale the input voltage since the data acquisition system input voltage is limited to 10 Volts. A voltage divider using $R_i=330\text{ K}\Omega$ in series with another resistor $R_j=1.5\text{ M}\Omega$ was used.

The total DC gain for this circuit is equal to:

$$V_t = 0.3607 * V_{Tach} \quad (5.5)$$

This voltage (V_t) was measure by the dSPACE board and converted by the computer to speed.

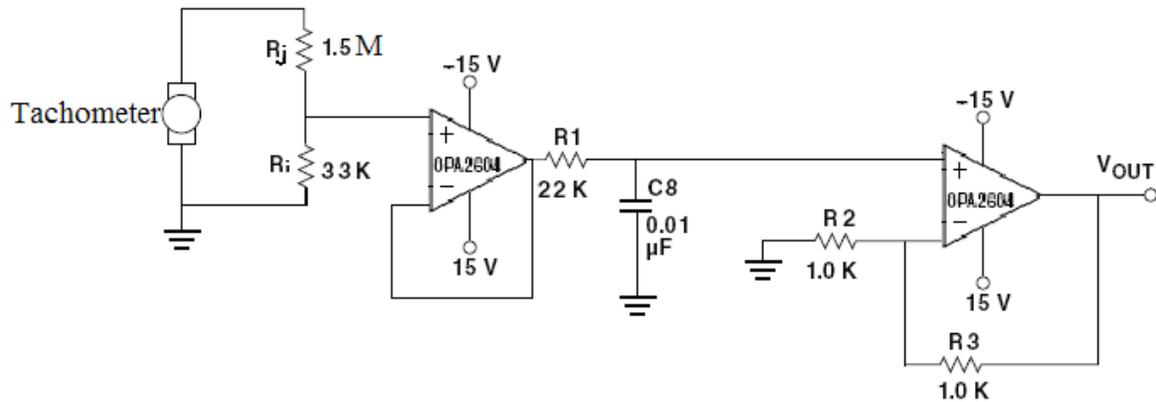


Figure 5.5 Speed Measurement Circuit

The implemented active low pass filter was a capacitor C8 in parallel with R1. The cutoff frequency of the filters is calculated with equation $f_c = \frac{1}{2\pi RC}$. For this speed circuit, a value of $R1 = 22k\Omega$ and $C8 = 0.01\mu F$, for a cutoff frequency of 723 Hz was used.

5.2 Permanent Magnet DC Motor

Before any simulation was performed we have to know the physical parameter of the DC motor model. The motor is a ½ HP, 90 Volts, PM DC motor from the General Electric Company with a maximum speed of 2500 RPM and an armature current of 5.2 A. The following experimental methods were for the determination of the physical parameters.

The resistance R_a was measured by the voltmeter-ammeter method [19]. The resistance R_a can be obtained by measuring the DC resistance across the motor terminals.

When the motor is running on steady state, turn off the power and repeat this procedure more than five times, then take the average of the measurements to obtain the resistance R_a value.

The induced emf constant K_a is determined by measuring the open circuit terminal voltage when the motor is operated as a generator. The back emf of the motor is given by:

$$E_{emf} = K_a \omega(t) \quad (5.6)$$

E_{emf} (back emf) vs. $\omega(t)$ (angular velocity) is collected and linear least square fit is used to determine the value of K_a .

The self inductance L_a is calculated from the electric equation 2.1 during transient step. Then this equation would become:

$$L_a \frac{di_a(t)}{dt} = V_a(t) - R_a i_a(t) - K_a \omega(t) \quad (5.7)$$

Since the value of the resistance R_a and the back emf K_a constant was previously calculated, then using linear least squares the inductance L_a could be obtain.

The viscous friction coefficient B_m is calculated by running the motor at constant speed unloaded. By doing this in equation 2.1 and 2.2, $\frac{d\omega}{dt} = 0$, $\frac{di}{dt} = 0$ and $T_L = 0$ so,

$$\omega(t) = \frac{K_a}{B_m} i_a \quad (5.8)$$

and linear least square can be used to determine K_a/B_m .

The motor inertia J_m is determined by the retardation test speed versus time characteristic [19] due to switching off the motor after steady state is reached. Using equation 2.2 and considering that now $T_L = 0$ and $\tau_{em} = 0$, results in

$$J_m \frac{d\omega}{dt} + B_m \omega(t) = 0 \quad (5.9)$$

and the solution of the above linear differential equation give the following:

$$\omega(t) = \omega(0)e^{-B_m t / J_m} = \omega(0)e^{-t / \tau_m} \quad (5.10)$$

where $\tau_m = \frac{J_m}{B_m}$ is the mechanical time constant of the motor. Using the curve of ω (angular speed) vs. time the value of J_m can be obtained. We can obtain the mechanical time constant solving equation 5.10, using logarithms, to obtain this time constant.

5.3 Mechanical Load

A fan load model presented in (5.13) has a parameter μ (Nm*s²). We used a 16 inch fan with 26 degrees of pitch as the load torque for this work. To estimate the fan torque, a simple test was made. The motor speed is increase with small steps from zero rad/s to a value of at least 160 rad/s. At every step the velocity is left constant for at least 5 seconds before applying a new step and the armature current was measured. By doing this to equation 2.2, $\frac{d\omega}{dt} = 0$, during that period with constant velocity (2.2) becomes (5.12):

$$\tau_F = \mu \text{sign}(\omega(t)) \omega^2(t) \quad (5.11)$$

$$\tau_{em}(t) = K_a i_a(t) - B_m \omega(t) = \tau_L(\omega(t)) \quad (5.12)$$

Now taking current and speed measurements points during steady state, we can calculate the load torque of the fan from 0 to 160 rad/s. The resulting curve is presented on Figure 5.6. This curve was used to evaluate the estimates from the linear and recursive parameter estimation methods presented in this work.

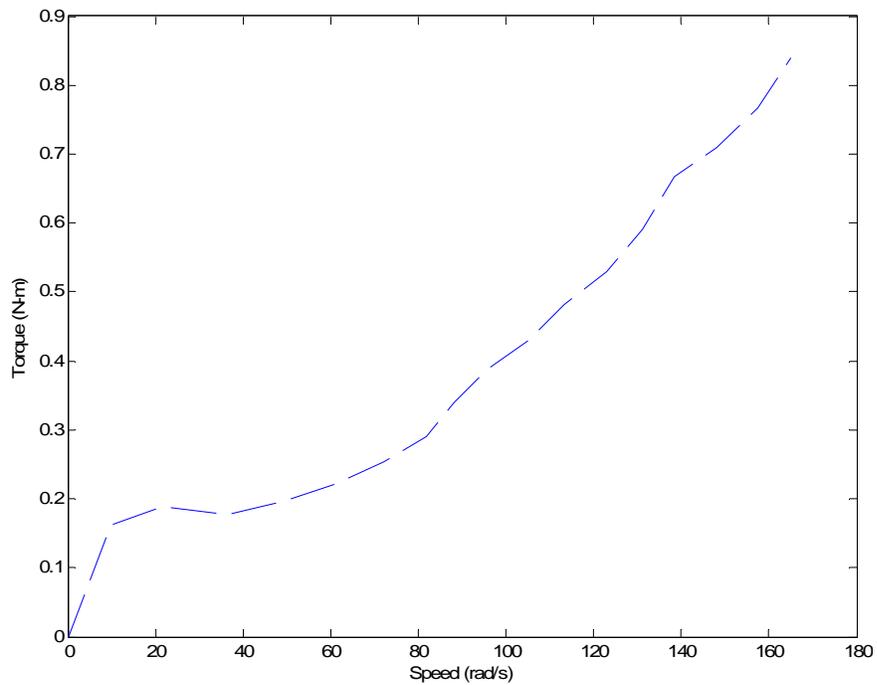


Figure 5.6 Load Torque for the Fan used in this Work

5.4 Experiment Initialization

The input data used for training the system was a voltage signal that would produce a speed velocity of a range between 0 rad/s to approximately 150 rad/s. This signal was produce with voltage steps using the Matlab[®] Simulink interface with dSPACE. This voltage

steps should produce the same voltage signal used also for the simulation tests. The voltage steps for this experimental test are different from the voltage curve used in the simulation. The difference resides in the initial shaving of the curve between each voltage step as is presented in Figure 5.7. This initial shaved rising occurs because of the Copley 421 PWM amplifier used to control the motor. This amplifier produced a signal that resembled a DC voltage signal produce by switching signals and during this process it introduces some delay of about 1.5s and that is why the signal is different from the simulation curve. Besides this difference the radial basis function model was selected with 160 points as in the simulations cases.

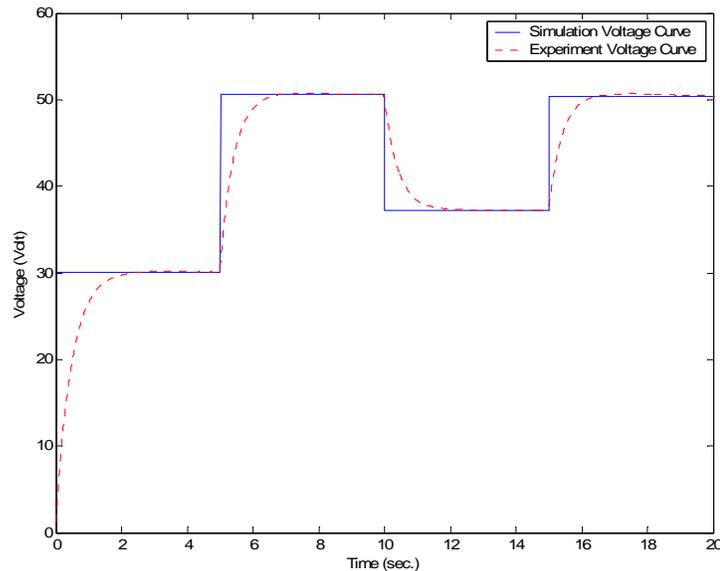


Figure 5.7 Input Voltage used for Training the Experimental Model

5.5 Experimental Results for the Batch Two Stage Method

Table 5.1 Experimental Results for the Batch Two Stage Method

| Parameters | Real Value | Estimated value | %Error |
|------------|------------|-----------------|----------------|
| L_a | 0.4094 | 0.4868 | 18.9 % |
| R_a | 1.587 | 1.414 | 10.9 % |
| K_a | 0.3409 | 0.3381 | 0.82 % |
| J_m | 0.002387 | 0.006062 | NO FAN INERTIA |

The experimental parameter estimates are presented in Table 5.1. From the results on the table, we can see that the torque coefficient (K_a) is estimated with good results. The inertia J_m has an error of 154% but this is because the value used in the simulations does not include the fan inertia. The only problem with the estimates comes from the resistance R_a and the inductance L_a . The real values were calculated without load and in ideal conditions. The change in the resistance and the inductance could be a result of wire heating due to the addition of the fan and also to the effect of the noise from the PWM amplifier. Besides these results, the algorithm is identifying the load characteristics with the radial basis functions with good precision as is shown in Figure 5.8.

Figure 5.8 compares the load torque estimate with the real load torque. From Figure 5.9 it could be seen that the error is less than 0.1 N-m and -0.1N-m in the training range. These demonstrate the results obtained from the simulation, where the approximation of the

load characteristic is very good inside the nominal operation range, but begin to loss the characteristic outside it.

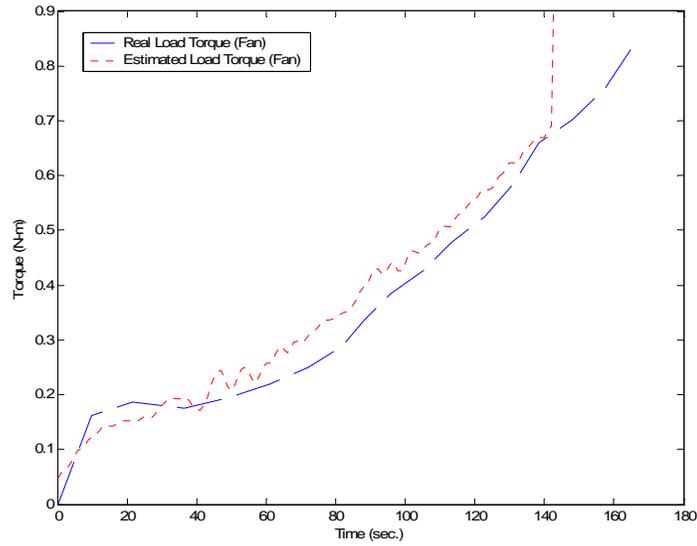


Figure 5.8 Real and Estimated Fan Load Torque

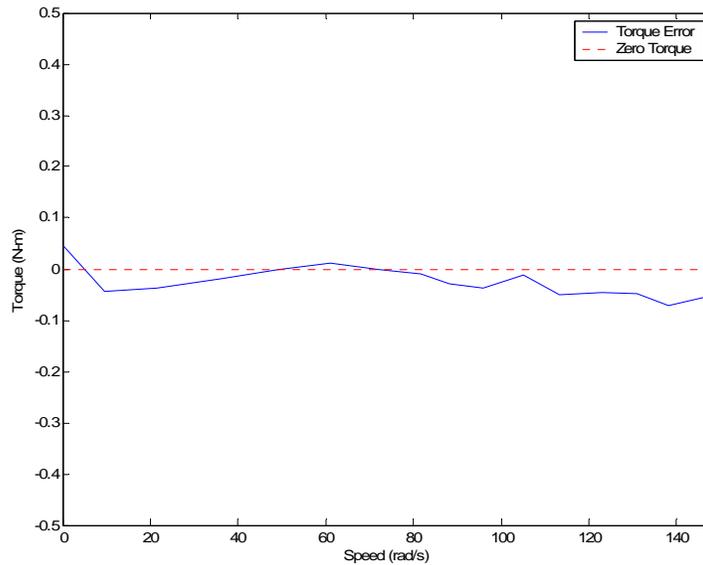


Figure 5.9 Estimated and Real Fan Load Torque Error

Figure 5.10 compares the estimated and real current from the simulations. It could be seen from the figure that the curves are almost the same. Furthermore, from Figure 5.11, it could be seen that the error in magnitude is less than 0.1 Ampere. The highest error is on the transition from one step to the other. The rest of the curve shows good results because the error was almost zero. The current has some spikes because of the noise produce by the PWM switching. This noise was greatly reduce because of the analog filters and the state variable filters used for the current data acquisition.

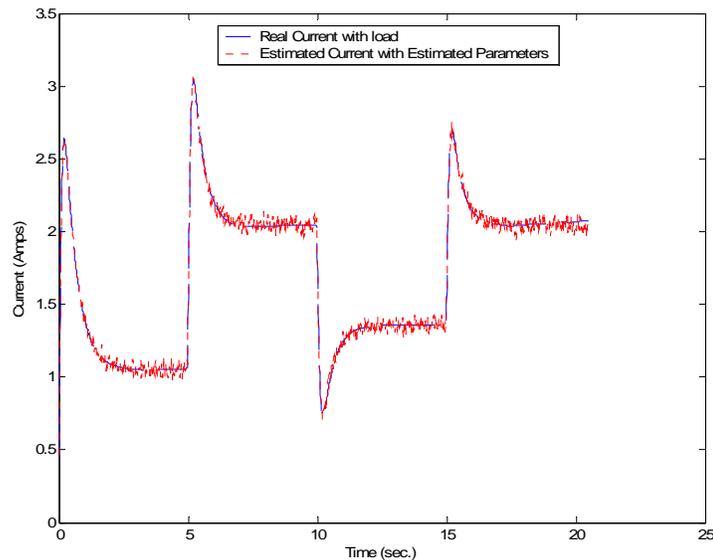


Figure 5.10 Real and Estimated Current from Experiment

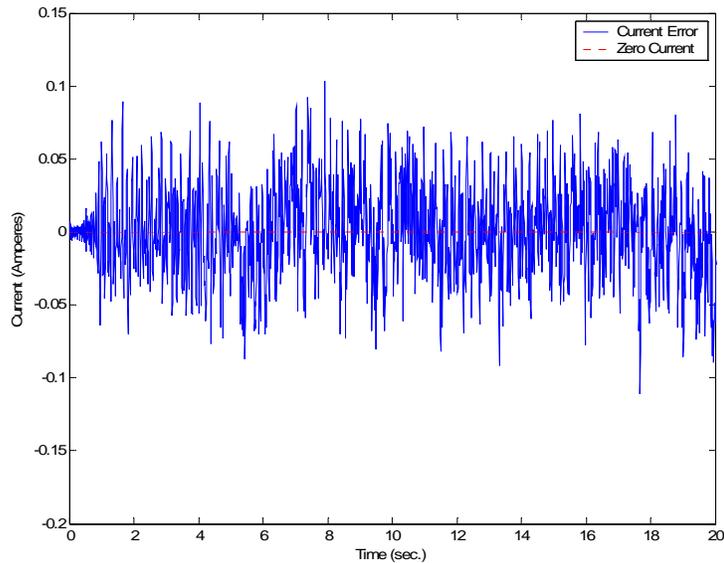


Figure 5.11 Real and Estimated Current Error from Experiment

In Figure 5.12, the real and estimated speed from the experiment are presented. On this figure, the measured speed is almost identical to the simulation speed. Figure 5.13 show the speed estimation error. The highest error magnitude is between 5 to 10 rad/s and occurs in the transition between steps on the speed curve. The rest of the curve has an average error of almost zero rad/s.

If we compare the experimental batch with the simulations, we could see that error has increase. It is still small but the simulations errors were in the order of 1×10^{-7} or less for the current and the speed curves. This is because of the errors in the armature resistance R_a and the armature inductance L_a . We would see the same behavior in the recursive two stage method which is presented later in this Chapter.

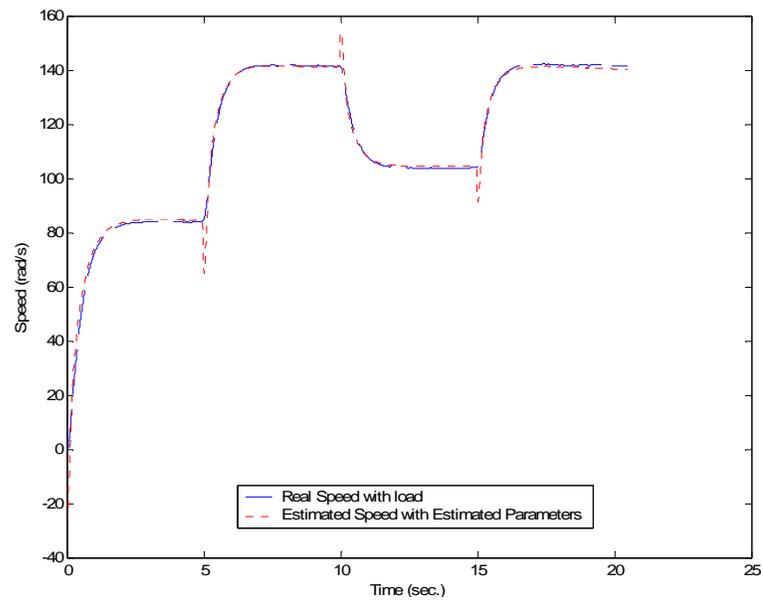


Figure 5.12 Real and Estimated Speed from Experiment

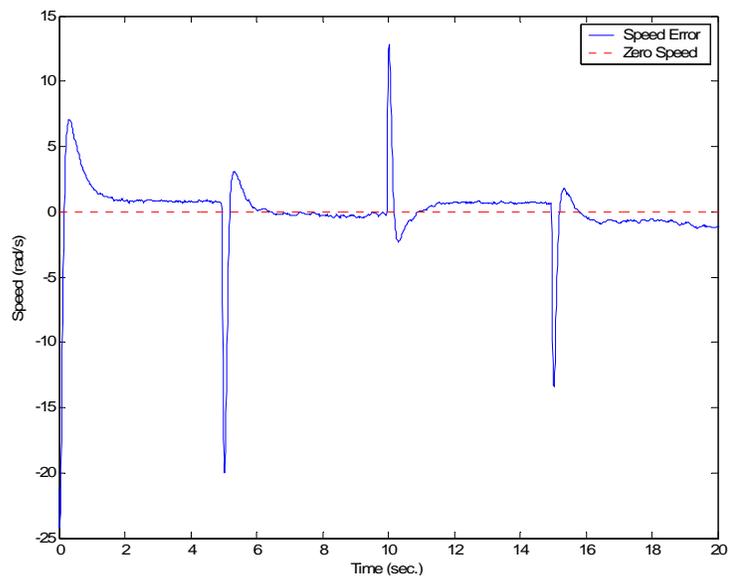


Figure 5.13 Real and Estimated Speed Error from Experiment

5.6 Validation Results for the Batch Two Stage Method

For the validation test, a new voltage input was selected for this purpose. This new voltage input has the same shape as the voltage input used for the simulations. This curve is different from that used in the simulations for the reason explained in Section 5.4.

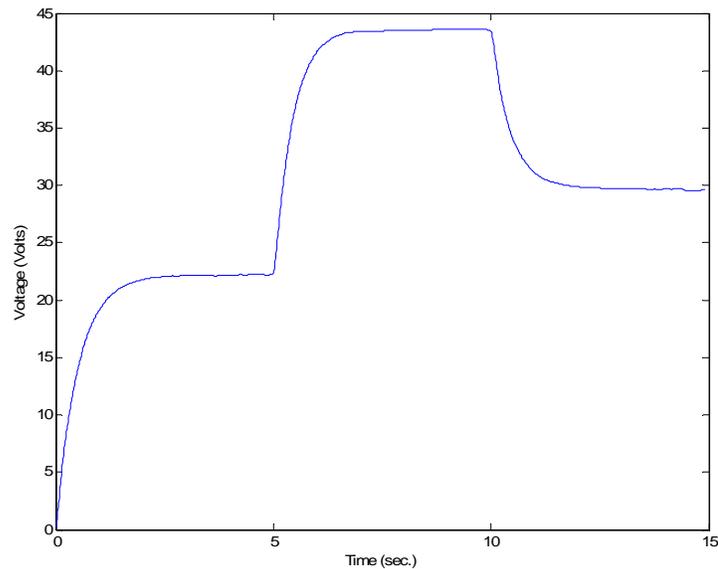


Figure 5.14 Input Armature Voltage used for Validation

The real and validation current are presented on Figure 5.15. Here the validation current curve behaves almost as the real simulated current but with some errors. The error between the validation and real current is presented on Figure 5.16. The error has an average of almost -0.086 Amperes with the exception of the initial transient of the step. In the initial transient the error was less than 0.2 Amperes.

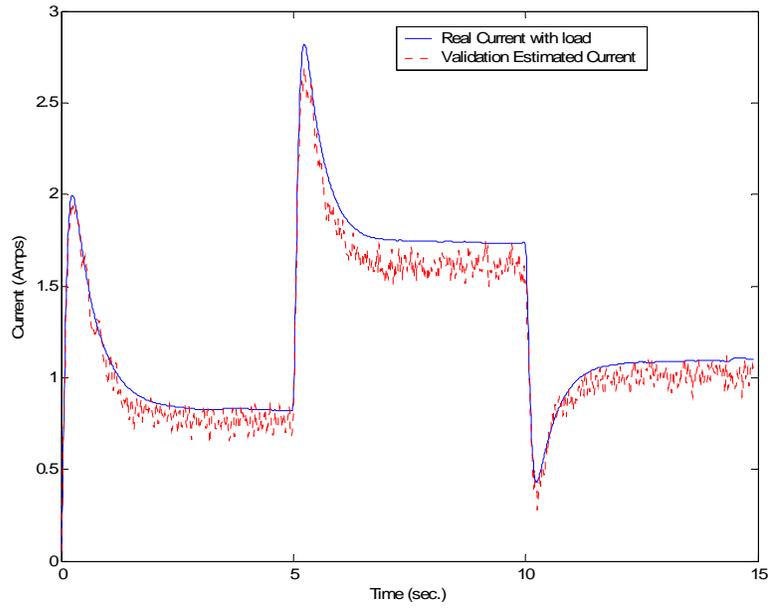


Figure 5.15 Real and Estimated Current Validation from Experiment

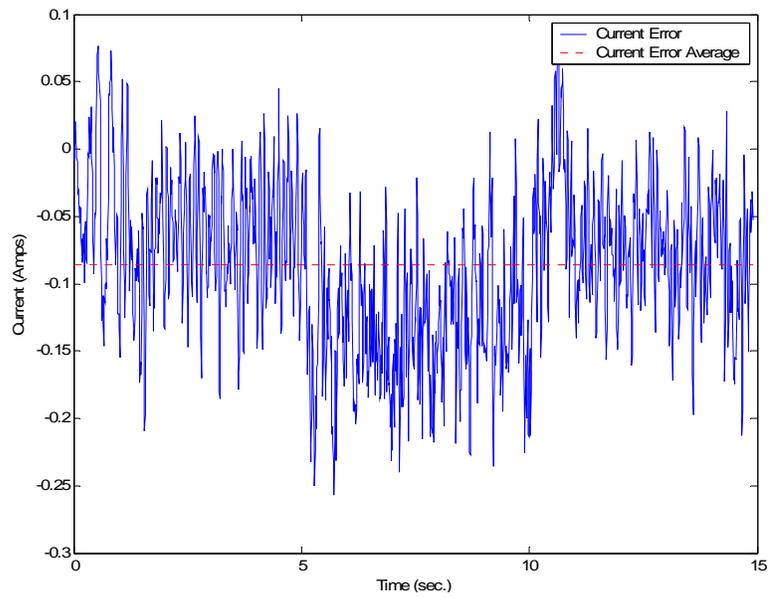


Figure 5.16 Validation error from Real and Estimated Current

Figure 5.17 presents the validation results for speed. Here both curves had similar shapes but with an average error of 2.265 rad/s. Moreover, Figure 5.18 presents the error between the validation and simulated speed, shows that the error is having an average of less than 4 rad/s with the highest error occurring on the steps of the curve.

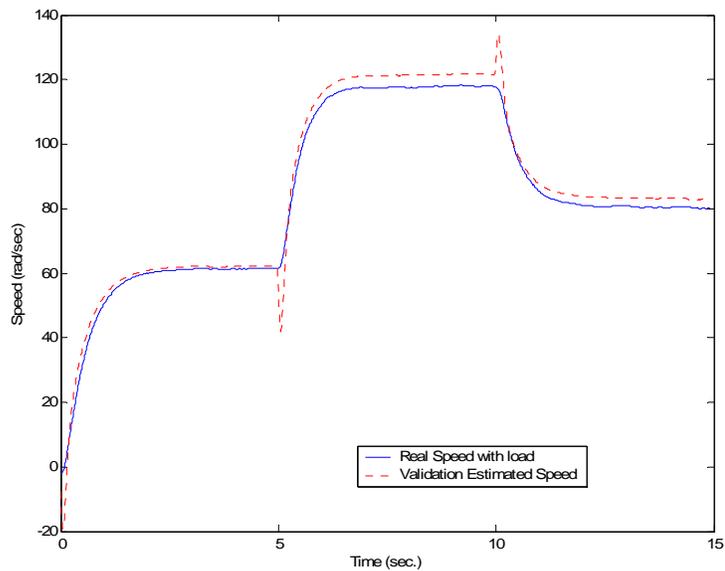


Figure 5.17 Real and Estimated Speed Validation from Experiment

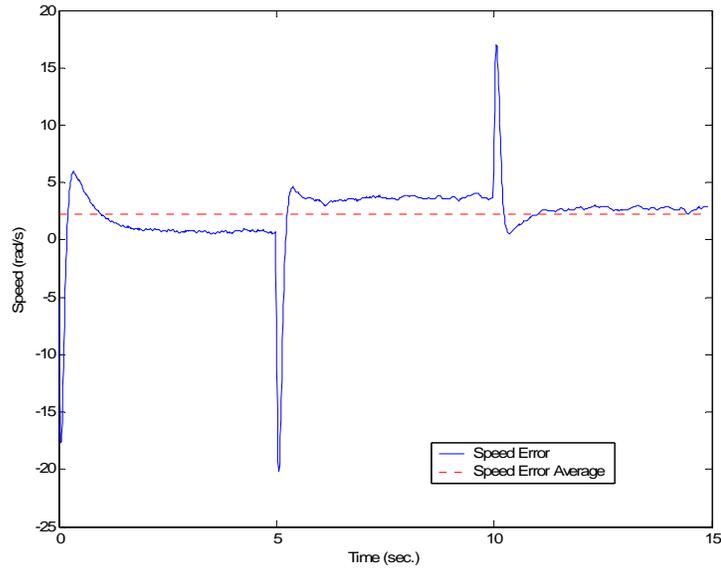


Figure 5.18 Validation Error from Real and Estimated Speed

We could see that besides the two stage method identified the R_a and the L_a with some error the model is showing an acceptable validation. Later on this chapter a solution for this error is presented.

5.7 Experimental Results for the Recursive Two Stage Method

Table 5.2 Parameter Estimates from Experiment for the Recursive Two Stage Method

| Parameters | Real Value | Estimated value | %Error |
|------------|------------|-----------------|----------------|
| L_a | 0.4094 | 0.5120 | 25.06 % |
| R_a | 1.587 | 1.408 | 11.28 % |
| K_a | 0.3409 | 0.3401 | 0.23 % |
| J_m | 0.002387 | 0.005702 | NO FAN INERTIA |

The steady state experimental parameters estimates using the recursive method are presented in Table 5.2. These parameters are estimated using the input armature voltage in Figure 5.7. From the results it could be seen that the estimates have high errors on the armature resistance (R_a) and the armature inductance (L_a). These errors are higher than the 10.9 % and the 18.9% error from the R_a and the L_a , respectively, from the batch two-stage case. On Figure 5.19 the time response of electrical parameter estimates are presented. We could see the high error on the resistance and the inductance. The blue dashed line is the R_a/L_a parameter, the Red dashed line is the I/L_a parameter and the green dashed line is the K_a/L_a parameter. It could be seen that the estimated parameters don't converge to the parameters values obtained from off-line tests and to the batch values (dotted dashed lines). These values are 3.876 for the dotted R_a/L_a line, 2.4425 for the dotted I/L_a line and 0.8326 for the dotted K_a/L_a line. We could see that the highest error occur in the R_a/L_a parameter. The figure shows that it never converge to a constant parameter. This points out telling us that there are problem to identify the resistance (R_a). This problem was also seen in [18] for induction motor. Figure 5.20 shows the time response for the K_a/J_m . The neural network parameters are not presented because they are 161.

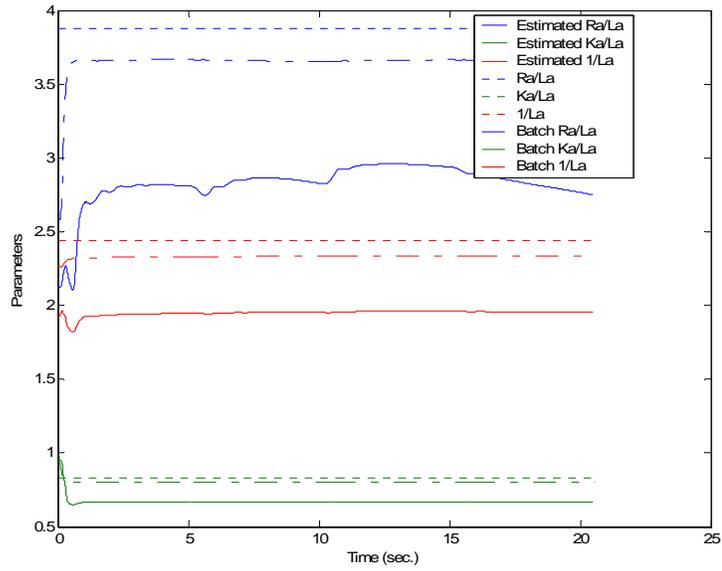


Figure 5.19 Real and Estimated Electrical Parameters from Recursive Experiment

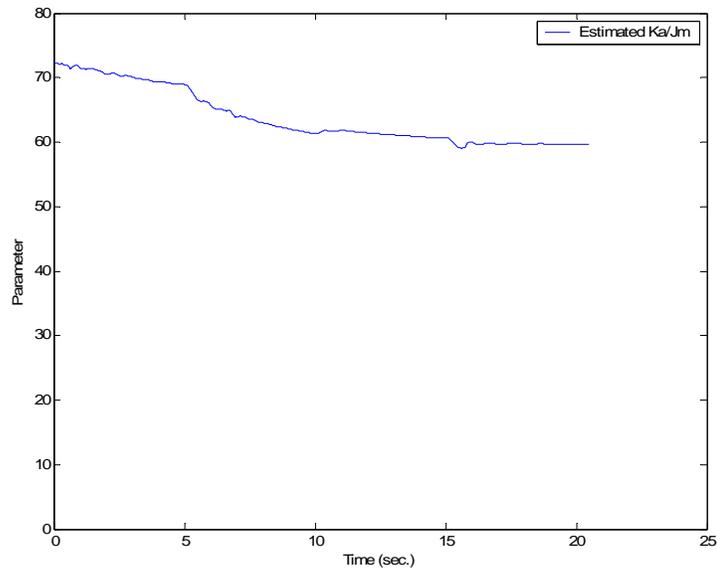


Figure 5.20 Real and Estimated Mechanical Parameters from Recursive Experiment

Figure 5.21 presents the load torque characteristic estimated using the recursive algorithm in comparison with the real load torque. From Figure 5.22, it could be seen that the error is higher than the batch experiment case presented in Section 5.6.

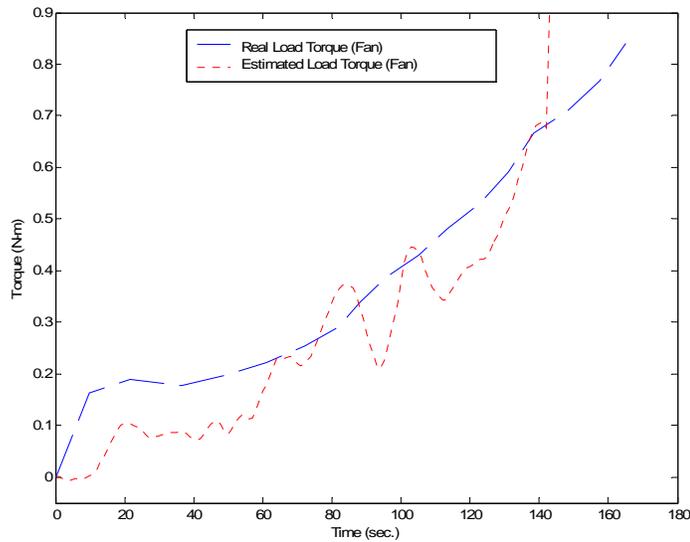


Figure 5.21 Real and Estimated Fan Load Torque from Recursive Experiment

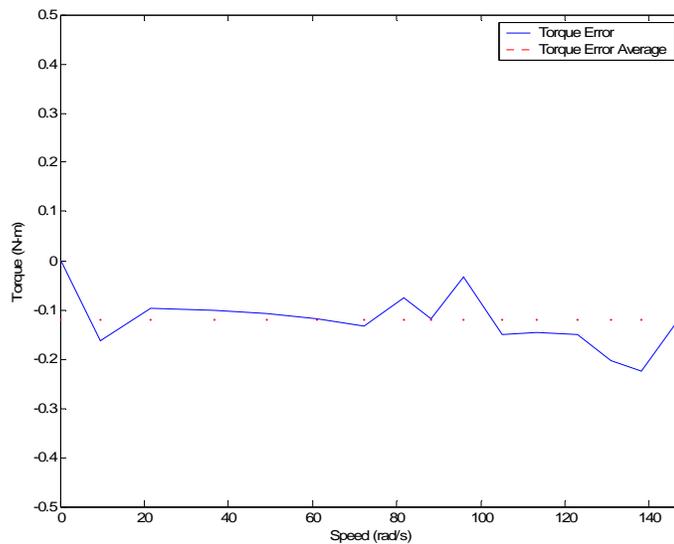


Figure 5.22 Real and Estimated Fan Load Torque Error from Recursive Experiment

Figure 5.23 compares the measured and simulated current using the parameters estimated with the recursive algorithm. It could be seen from the figure that the curves have some errors especially on the transition between steps. In Figure 5.24, it could be seen that the errors on this steps transitions had a value of almost -1 Ampere. This current error happens because of the estimation error on the load torque.

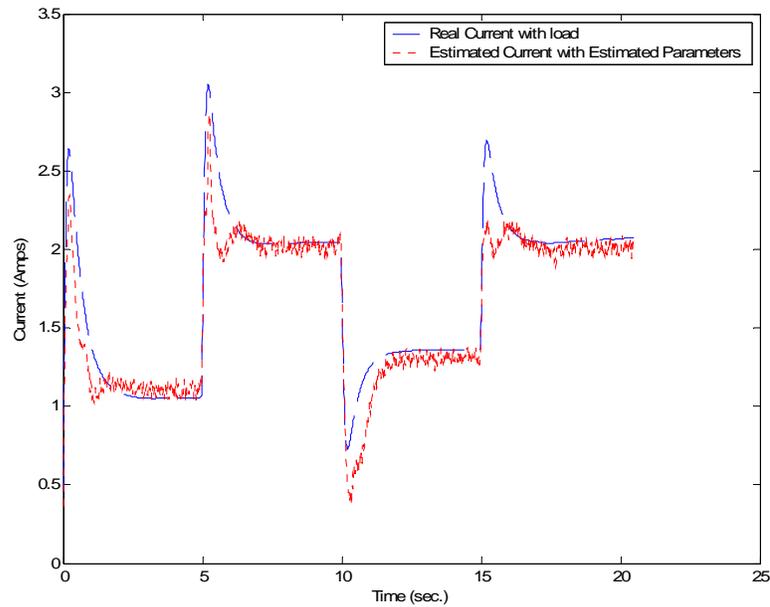


Figure 5.23 Real and Estimated Current from Recursive Experiment

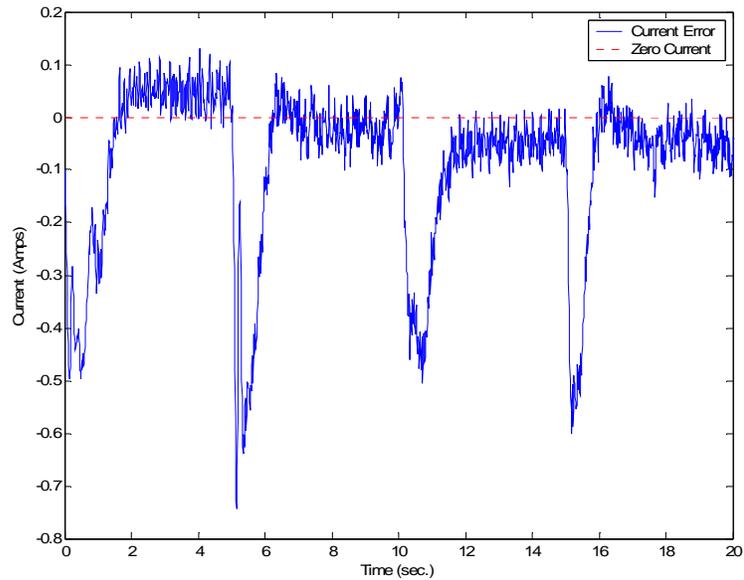


Figure 5.24 Real and Estimated Current Error from Recursive Experiment

In Figure 5.25, the real and simulated speed from the recursive method is presented. In Figure 5.26, the difference between the simulated and real speed is presented. The mean error magnitude is about 1 rad/s, which is less than 1 % since the highest speed is about 140 rad/s. Also during step changes is when the highest errors occur.

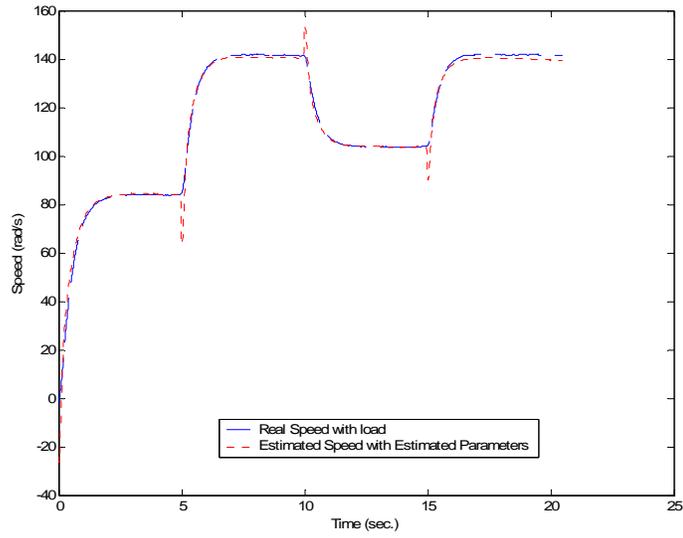


Figure 5.25 Real and Estimated Speed from Recursive Experiment

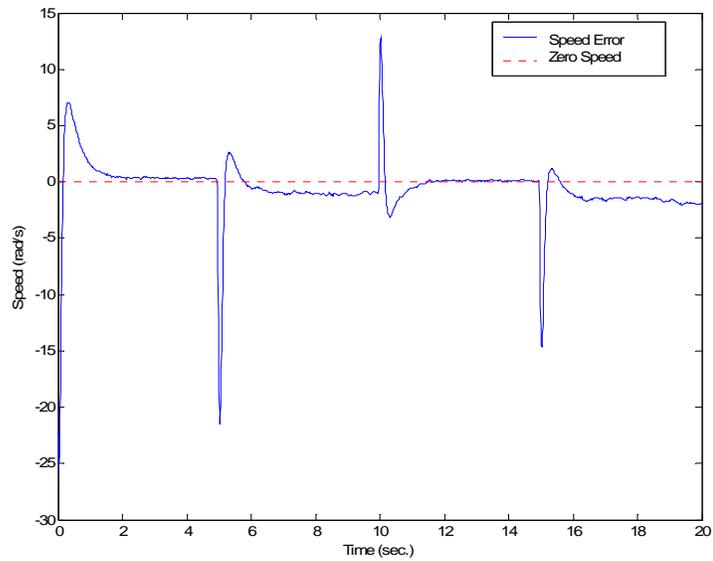


Figure 5.26 Real and Estimated Speed Error from Recursive Experiment

5.8 Validation Results for the Recursive Two Stage Method

For the validation test, the same armature voltage input used for the validation of the experiment batch method was selected for the recursive case. This voltage input was presented in Figure 5.14.

The measured and validation current from the recursive case are presented in Figure 5.27. Here the validation current curve has similar errors as the estimated current presented in the previous section. The error between the validation and real current is presented on Figure 5.28. The error has an average of almost 0.1 Amperes with the exception of the period during the transients between steps that was less than 0.25 Amperes. This error is 10% of the current peak values of 2.5 Ampere. The identified model shows acceptable performance with the validation data set.

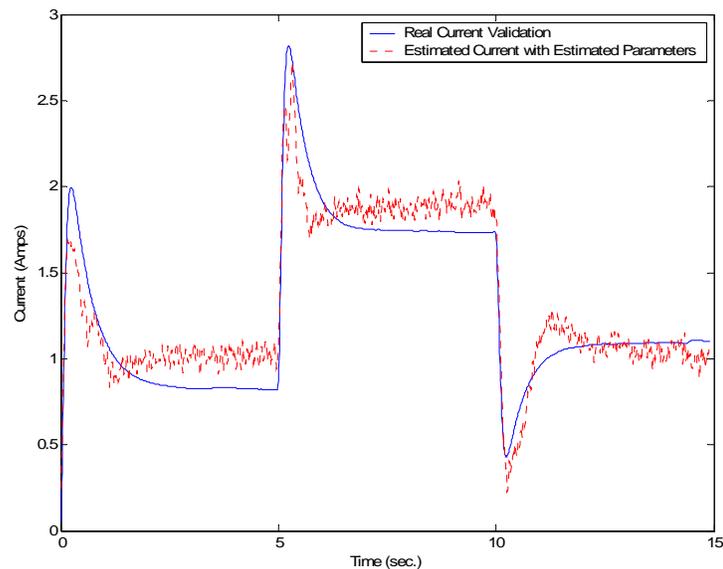


Figure 5.27 Real and Estimated Current Validation from Recursive Experiment

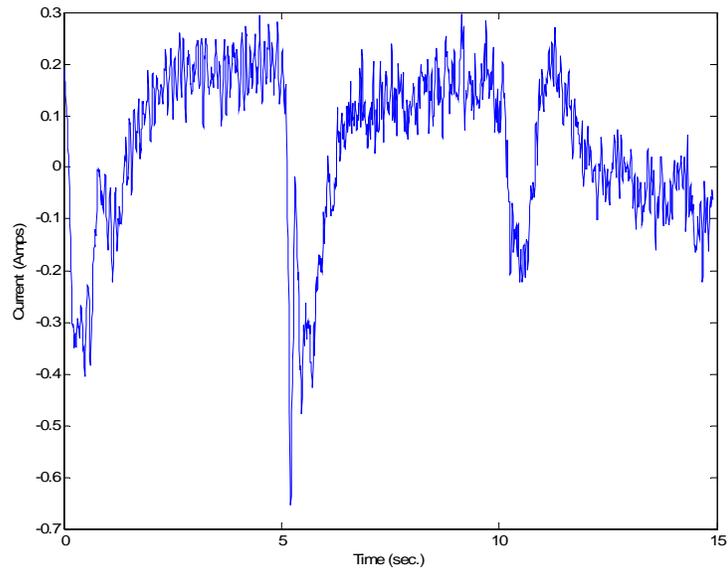


Figure 5.28 Validation error from Real and Estimated Current from Recursive Experiment

The Figure 5.29 presents the validation and measured speed from the recursive case. Here both curves are almost identical. Moreover, Figure 5.30 presents the error between the validation and the real simulated current; it shows that the error has an average of 2 rad/s. The highest error occurs on the steps of the curve. These results are similar to those in batch two stage method.

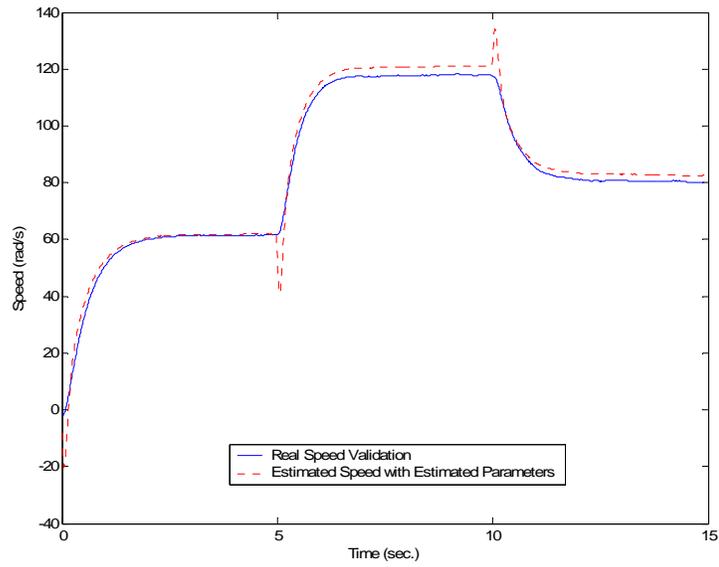


Figure 5.29 Real and Estimated Speed Validation from Recursive Experiment

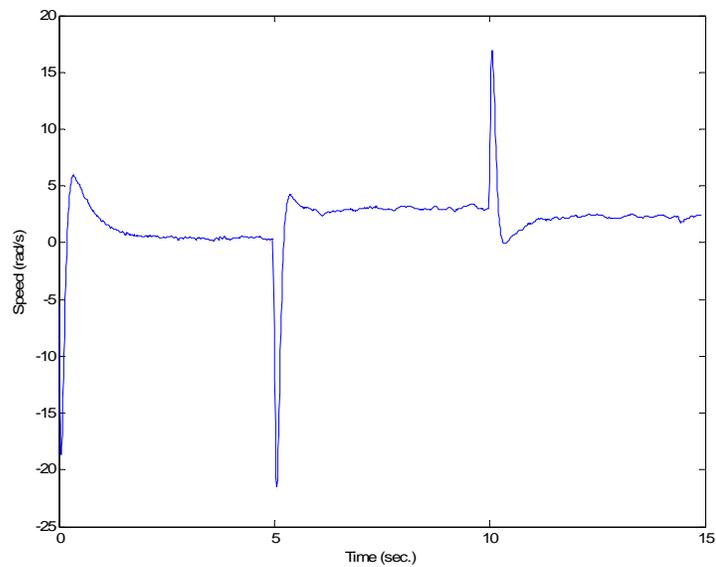


Figure 5.30 Validation Error from Real and Estimated Speed from Recursive Experiment

From the results on Section 5.5-5.8, we can conclude that the armature resistance errors is causing problems with the overall system identification. A solution for this problem is presented in the next section. The solution is going to be implemented on both algorithms, the batch and the recursive two stage methods.

5.9 Experimental Results for Batch Two Stage Method with Fixed Armature Resistance

The performance of the parameter estimation algorithms is affected by different aspects of system modeling and quality of measurements. Electric drives models are created from the analysis of the physical phenomena that describe the system behavior. But many times the measurements are not often good enough to correctly reflect all of the parameters effects. This difference between high detail models and low richness in the measurements leads to very sensitive or ill-conditioned parameter estimation problems [20]

To solve this problem, in [19] they reduce the order of the system by fixing the ill-conditioned or sensitive parameters. Also in [21] estimation of resistance from operational data is an ill-conditioned problem for the induction motor. In this work, the ill-conditioned parameter was also believed to be the armature resistance (R_a). We arrive to this conclusion from the results on the recursive and also batch experiment presented on Section 5.7. From a brief analysis of the DC permanent magnet equivalent circuit presented in Figure 2.2 we

could see what is causing this problem with estimating R_a . In this work, the nominal value of the resistance is 1.587Ω and if the nominal current of this motor is 5.2 Amperes, the voltage drop on the resistance would be 8.25 V. Since the voltage drop in the inductor L_a with a DC current is zero, the rest of the 90 Volts nominal motor voltage drop would be in the back emf voltage. The emf voltage would be 81.7476 V. This is a 9.16 % from the nominal motor voltage of 90 V. If for example, the resistance change because of heating to a value of 2.0Ω , the voltage drop on the resistance would be 10.4 V. The back emf voltage would now be 79.6 V, resulting in a resistance voltage drop of 11.55 % from the nominal value. As you could see the resistance is not so sensitive to the effect of the voltage and this made the resistance difficult to identify from the measurement. For this reason, the armature resistance became an ill-conditioned parameter. If the prior estimation of the ill-conditioned parameter is incorporated into the estimation process would result in the sensitive reduction and the numerical performance of the estimation process would improve [21]. Figure 5.13 showed that the parameters R_a/L_a and $1/L_a$ do not converge to a constant value. We decide to fix the parameter R_a to the off-line measured value of 1.5875 ohms based on the results from [19] and [21]. Here we present the estimation results when fixing R_a for the batch two-stage algorithm.

We need to rearrange the linear regression equation for the electrical parameters (5.13) as follows:

$$\frac{di_a(t)}{dt} = -\frac{R_a}{L_a} i_a(t) - \frac{K_a}{L_a} \omega(t) + \frac{V_a}{L_a}(t) \quad (5.13)$$

$$\mathbf{y}_e(t) = \mathbf{a}_e(t)^T \boldsymbol{\theta}_e \quad (5.14)$$

where:

$$\mathbf{a}_e^T = \left[(v_a - R_a i_a) - \omega(t) \right]^T \quad (5.15)$$

$$\hat{\boldsymbol{\theta}}_e = \left[\hat{\theta}_{e1} \quad \hat{\theta}_{e2} \right]^T \quad (5.16)$$

and

$$\mathbf{y}_e = \frac{di_a(t)}{dt} \quad (5.17)$$

The remaining electrical parameters are obtained as follows:

$$\hat{L}_a = \frac{1}{\hat{\theta}_{e1}} \quad (5.18)$$

$$\hat{K}_a = \frac{\hat{\theta}_{e2}}{\hat{\theta}_{e1}} \quad (5.19)$$

Table 5.3 Results from Experiment for the Batch Two Stage Method with Fixed R_a

| Parameters | Real Value | Estimated value | %Error |
|-------------------------------|------------|-----------------|----------------|
| <i>Fixed R_a</i> | 1.587 | N/A | N/A % |
| L_a | 0.4094 | 0.4506 | 10.06 % |
| K_a | 0.3409 | 0.3357 | 1.52 % |
| J_m | 0.002387 | 0.006017 | NO FAN INERTIA |

In Table 5.3, the results from the batch two stage method with fixed R_a are shown. We could notice that the armature inductance L_a has an error of 10.06%. When we compare this result with the inductance estimate shown in Table 5.1 for the full-parameter estimator

we see a reduction of 8.9%. The induced emf constant (K_a) increases his error from 0.82 % to 1.52 %. This increase is small and has no significant effect. The inertia J_m change less than 1% from that in Table 5.1. In Figure 5.31, we could see that the estimated load torque characteristic is very close to the real torque. The error which is presented in Figure 5.32 shows that the error is less than 0.1 N-m inside the training speed range.

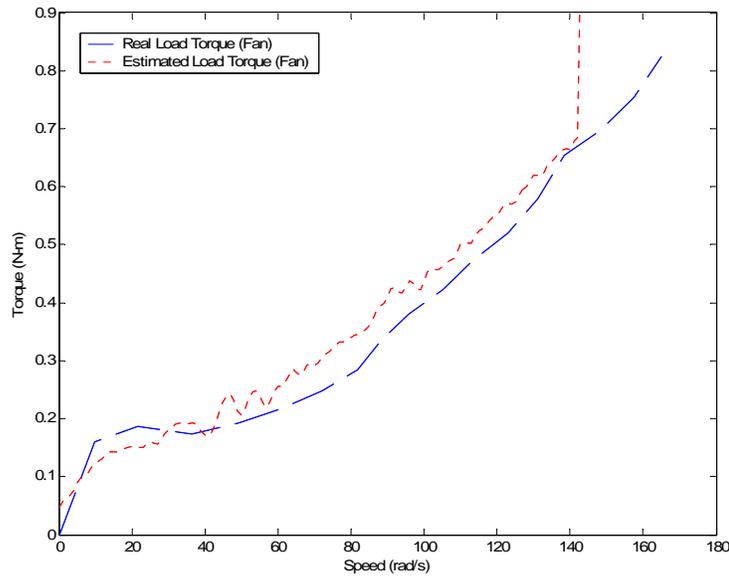


Figure 5.31 Real and Estimated Fan Load Torque with Fixed R_a

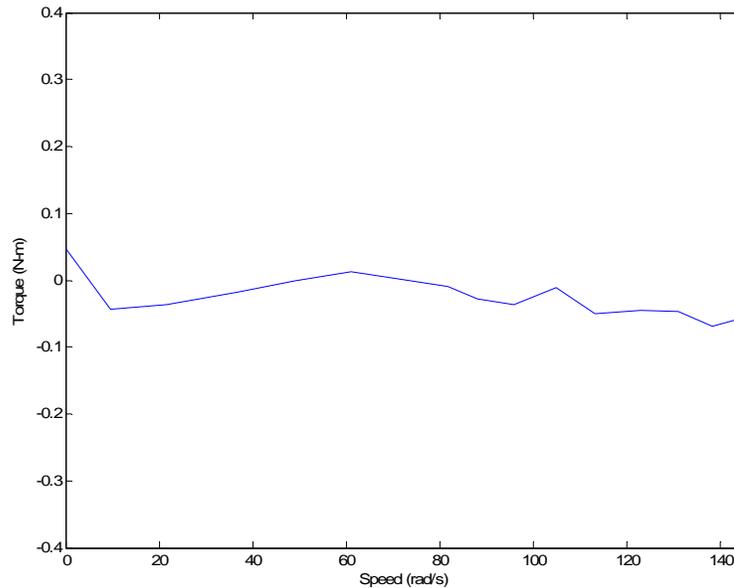


Figure 5.32 Estimated and Real Fan Load Torque Error with Fixed R_a

Figure 5.33 shows the measured and estimated current. We could see that the estimated current is behaving almost as the real current as the error presented in Figure 5.34 shows. Here the error has an average of almost zero Amperes and maximum error is 0.05 Ampere. Figure 5.35 presents the real and estimated speed with fixed R_a and Figure 5.36 presents the estimation error. It could be seen that the estimated speed error is less than 2 rad/s except during transients.

From this figures we could conclude that the identified model is good and that the estimated parameters are close to the real values.

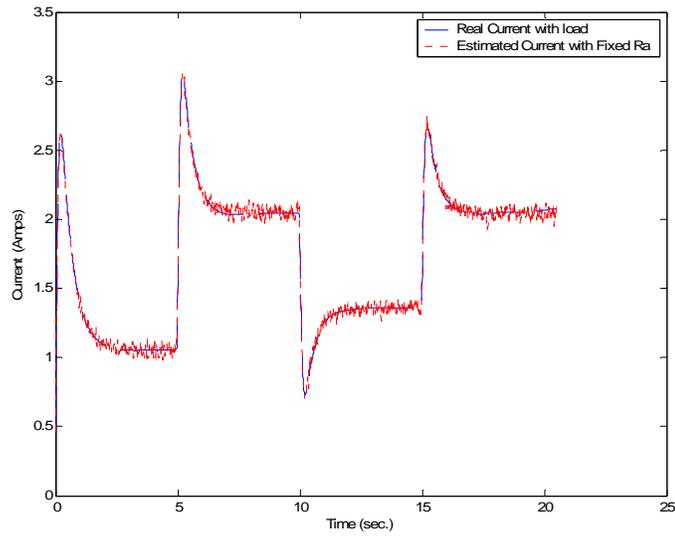


Figure 5.33 Real and Estimated Current from experiment with Fixed R_a

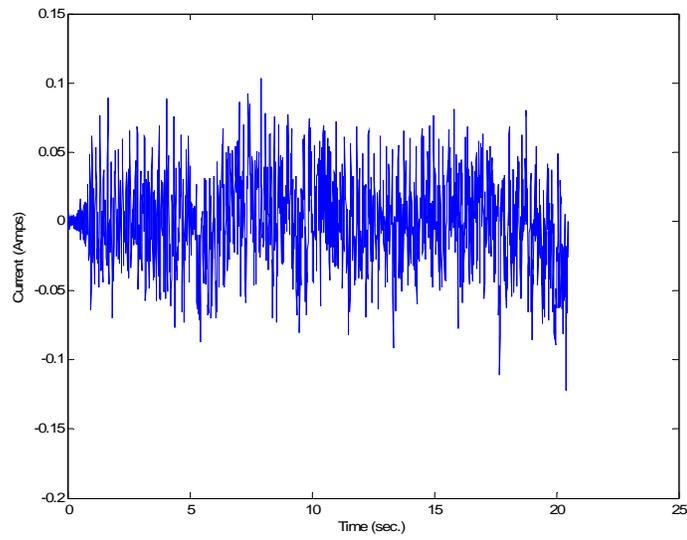


Figure 5.34 Real and Estimated Current Error from Experiment with Fixed R_a

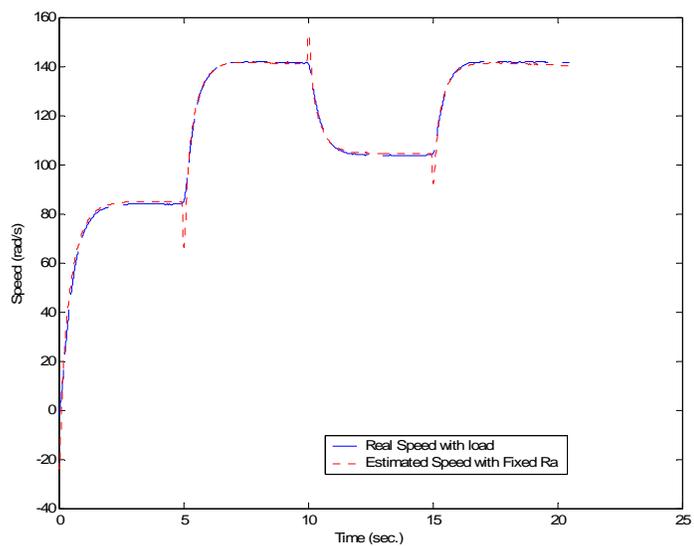


Figure 5.35 Real and Estimated Speed from Experiment with Fixed R_a

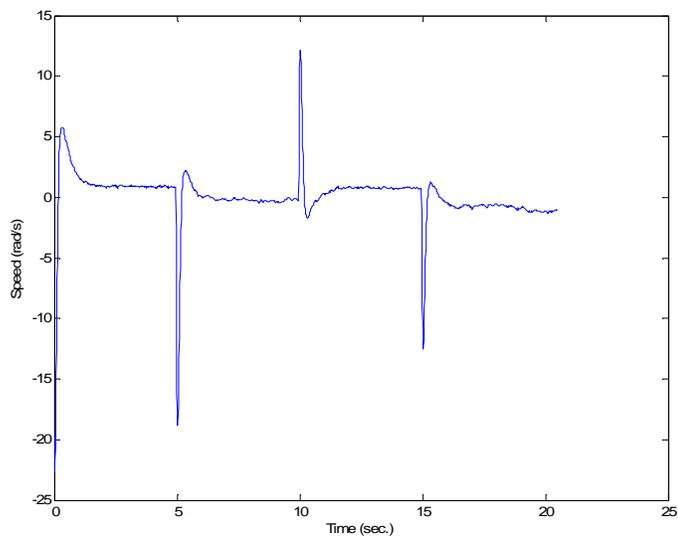


Figure 5.36 Real and Estimated Speed Error from Experiment with Fixed R_a

5.10 Validation Results for the Batch Two Stage Method with Fixed R_a

In this section, the validation results from the batch two stage method with fixed R_a are presented. The same input armature voltage as described in Section 5.6 was used. Here it could be seen that the estimated armature current (Figure 5.37) and the speed (figure 5.39) are behaving very close to the measured signal. If you compare these figures with the validation figure from Section 5.6, you could not see much difference. In the full parameter validation case, besides the error on the parameters, the system was identified fairly well. But the only big difference is that with the fixed R_a , the parameter estimates had improved, especially the inductance L_a .

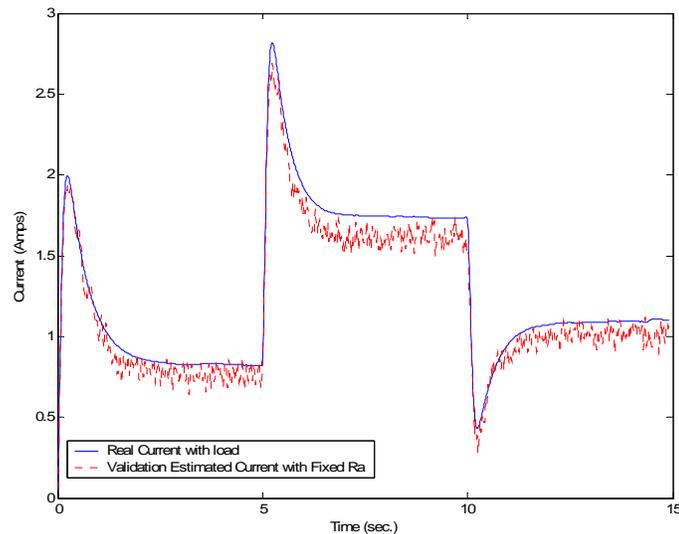


Figure 5.37 Real and Estimated Current Validation from Experiment with Fixed R_a

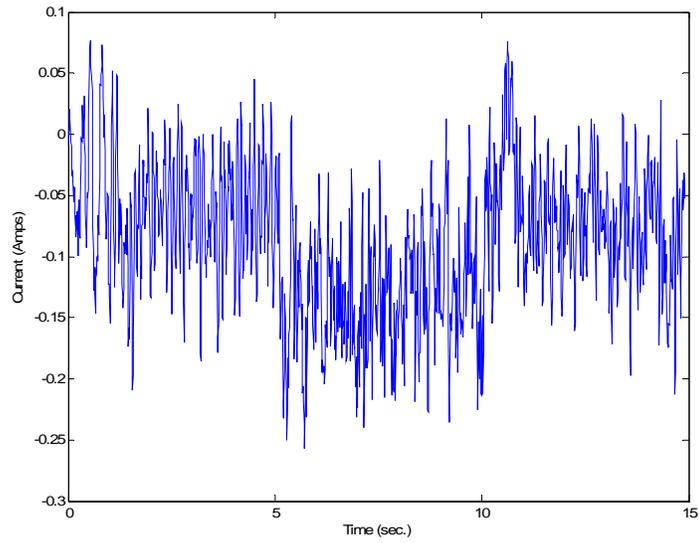


Figure 5.38 Validation error from Real and Estimated Current with Fixed R_a

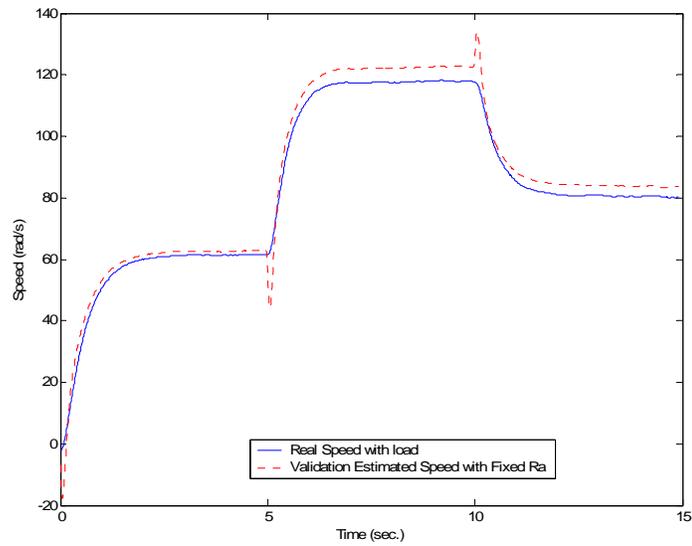


Figure 5.39 Real and Estimated Speed Validation from Experiment with Fixed R_a

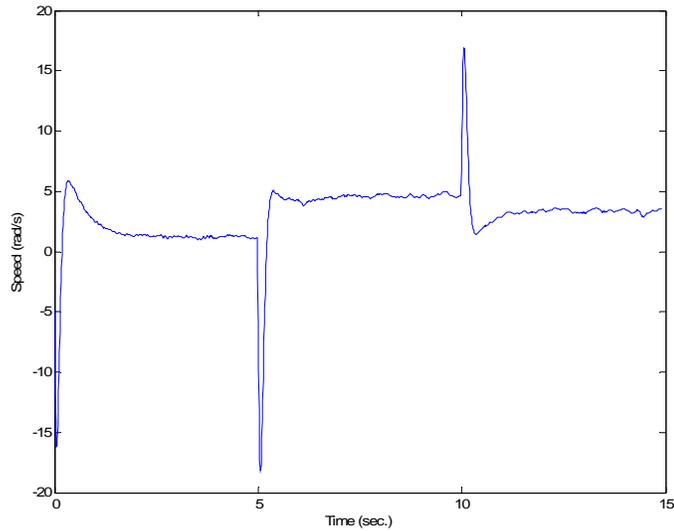


Figure 5.40 Validation Error from Real and Estimated Speed with Fixed R_a

5.11 Results for the Recursive Two Stage Method with Fixed R_a

On this section we present the results from the recursive two stage method with fixed R_a . For this case we did the same modification for the Newton-Raphson recursive least square method, as in the batch two stage method of Section 5.9. The results were even better than in the batch results presented in Section 5.9.

Table 5.4 Parameter Estimates from Experiment for the Recursive Two Stage Method with Fixed R_a

| Parameters | Real Value | Estimated value | %Error |
|-------------------------------|-------------------|------------------------|----------------|
| <i>Fixed R_a</i> | 1.587 | N/A | N/A % |
| L_a | 0.4094 | 0.4158 | 1.56 % |
| K_a | 0.3409 | 0.3397 | 0.35 % |
| J_m | 0.002387 | 0.005783 | NO FAN INERTIA |

In Table 5.4, we could see the steady state results for this case. The inductance L_a was estimated with an error of 1.56%. The induced electric emf was estimated with an error of 0.35%. When you compare the L_a parameter obtained with fixed R_a , with the L_a obtained from full parameter estimation, the difference is clearly noticeable. In the full parameter recursive version, L_a estimate was 0.5120 H with an error of 25.06 %. We could see that the error was almost eliminated by fixing the resistance.

Also it could be noticed in Figure 5.40 how fast the electrical parameters converge to a value near the nominal value. This is a big difference when comparing the convergence results for the full parameter version described in Section 5.7. In that section, most parameters convergence slowly and the R_a/L_a parameter does not never converge. In the fixed R_a estimator, the R_a/L_a converges very fast to a value near the nominal value. What this is telling us is that the resistance R_a parameter was ill-conditioned or non observables from from the measurements. Fixing this parameter greatly improves the estimation results in terms of faster convergence and more accurate results.

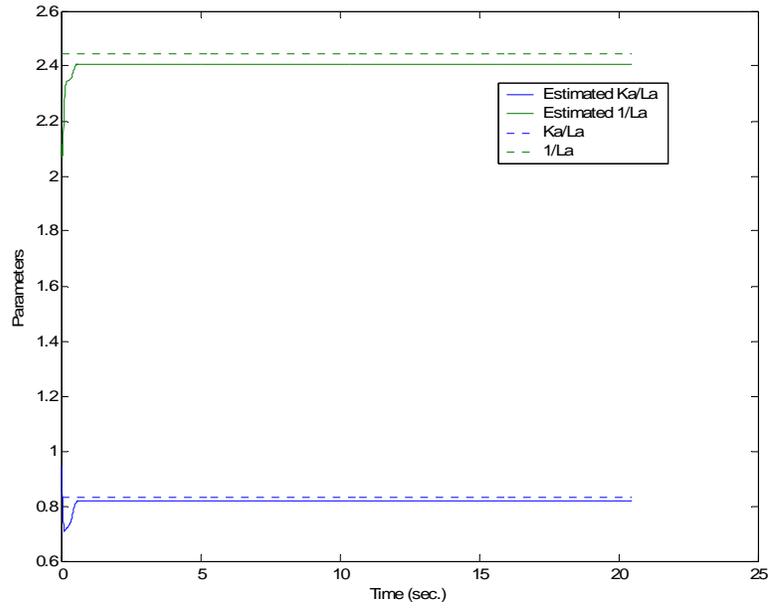


Figure 5.41 Real and Estimated Electrical Parameters from Recursive Experiment with Fixed R_a

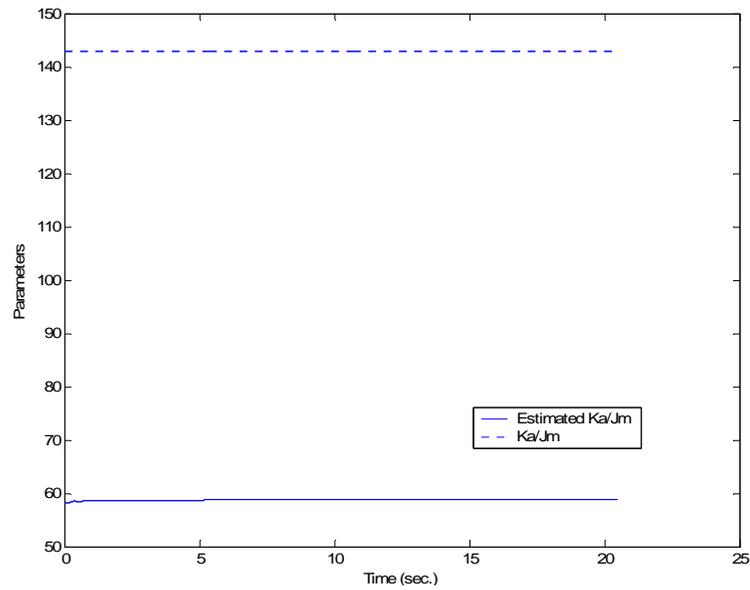


Figure 5.42 Real and Estimated Mechanical Parameters from Recursive Experiment with Fixed R_a

Figure 5.43 presents the estimated and real load torque characteristic. When this figure is compared with the corresponding figure on Section 5.7, it could be seen that the estimated torque is better identified with the fixed R_a . The torque error (figure 5.44) shows that in this case the error has an average of almost zero N-m. This is a big improvement over the full parameter case. This could be the result of fast convergence, for K_a where in the full parameter case convergence is slow and had some oscillation through the time.

Also the current presented on Figure 5.45 shows a better estimate than the full parameter case. With the fixed R_a , the current error (Figure 5.46) has an average of almost 0 amperes. In this simulation, the estimated current behaves almost identical as the real current including the transition between the steps. This was a problem in the full parameter case.

This tendency continues with the estimated speed (Figure 5.47). The estimated speed was almost similar to the real speed and the error (Figure 5.48) was almost the same as in the full parameter version. In the full parameter version, the speed was estimated fairly good and the differences between the cases are minimal. But the great difference has come on the estimated current which results in improved torque estimation when compared with the simulation results from the Chapter 4.

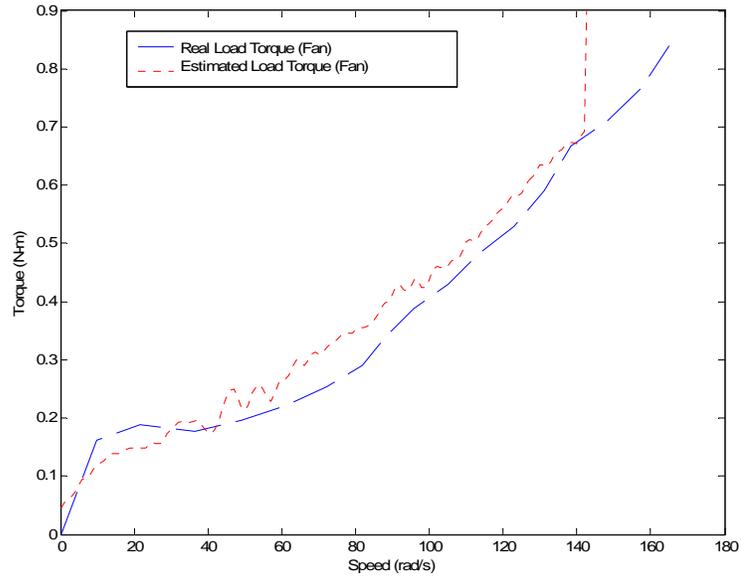


Figure 5.43 Real and Estimated Fan Load Torque from Recursive Experiment with Fixed R_a

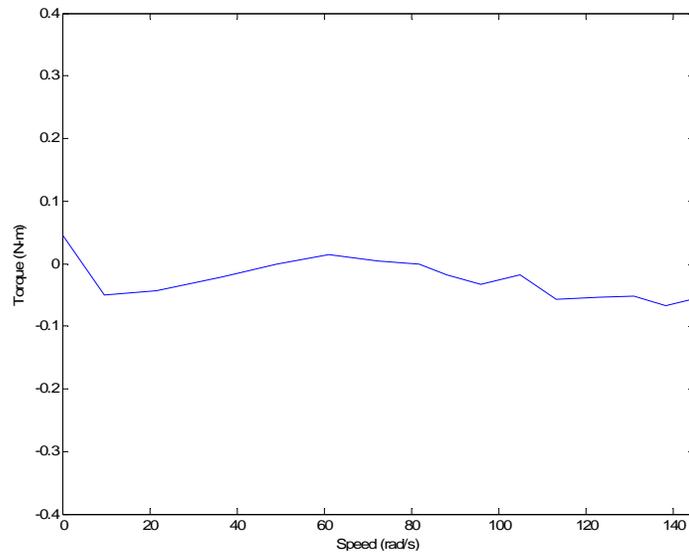


Figure 5.44 Real and Estimated Fan Load Torque Error from Recursive Experiment with Fixed R_a

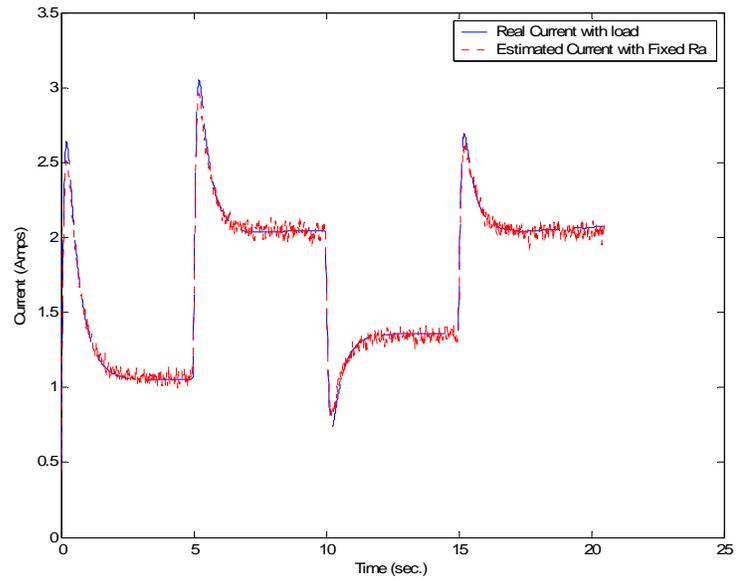


Figure 5.45 Real and Estimated Current from Recursive Experiment with Fixed R_a

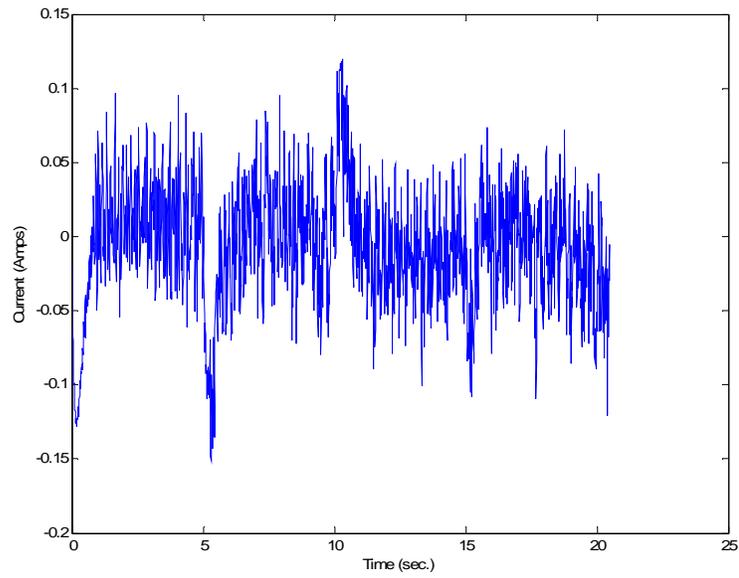


Figure 5.46 Real and Estimated Current Error from Recursive Experiment with Fixed R_a

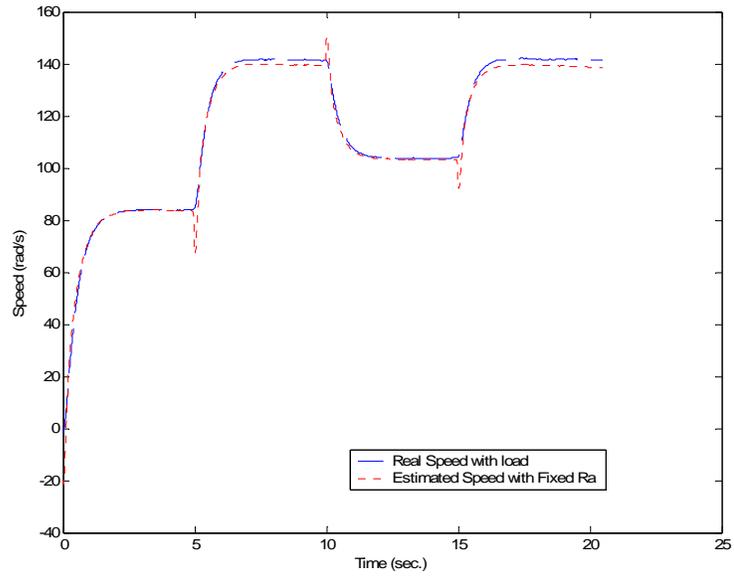


Figure 5.47 Real and Estimated Speed from Recursive Experiment with Fixed R_a

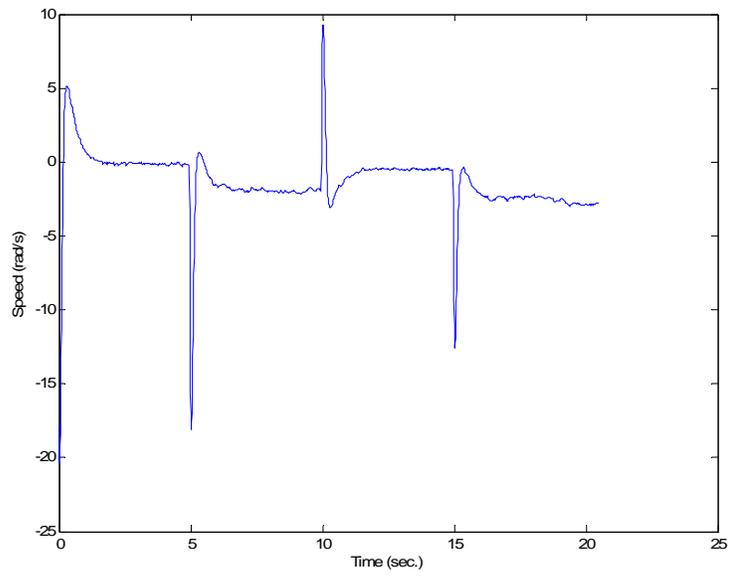


Figure 5.48 Real and Estimated Speed Error from Recursive Experiment with Fixed R_a

5.12 Validation Results for the Recursive Two Stage Method

Experiment with Fixed R_a

The validation of the recursive two stage method with the fixed R_a confirms the excellent ability of this algorithms to estimate the parameters and identify the system. Here the estimated and real currents are presented in Figure 5.49 and the error in Figure 5.50. We could say that the system was correctly identified when the estimated parameters are tested with a new set of data.

The estimated and real speed is presented in Figure 5.51 and the error in Figure 5.52 corroborates the results obtained with the batch fixed R_a case. What we could conclude now is that the overall identification and estimation of the system improves with fixing the ill-conditioned parameter, which in our case was the armature resistance.

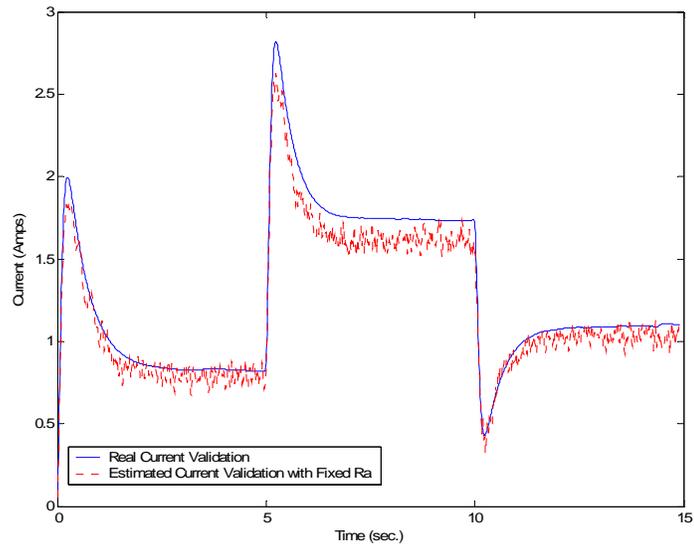


Figure 5.49 Real and Estimated Current Validation from Recursive Experiment with Fixed R_a

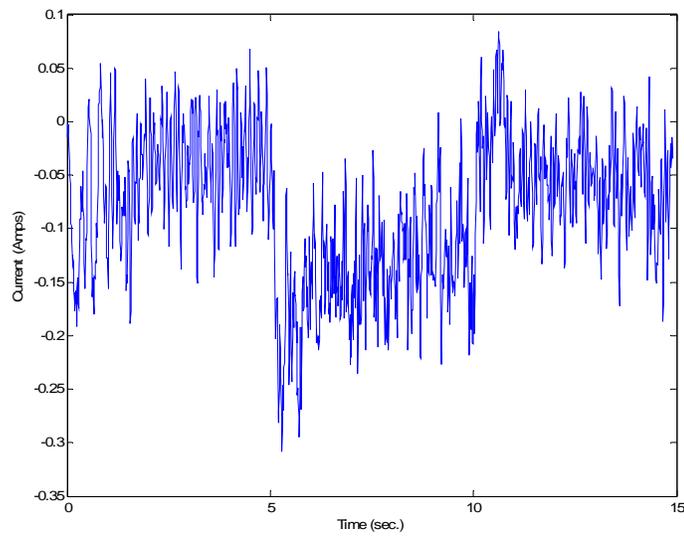


Figure 5.50 Validation error from Real and Estimated Current from Recursive Experiment with Fixed R_a

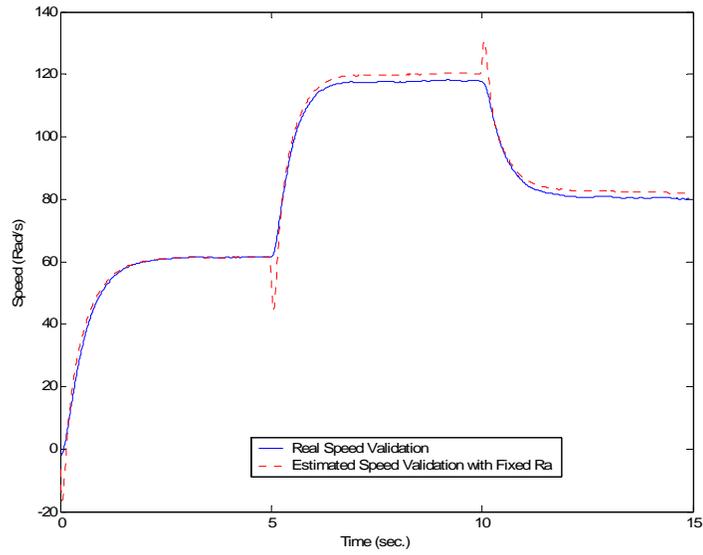


Figure 5.51 Real and Estimated Speed Validation from Recursive Experiment with Fixed R_a

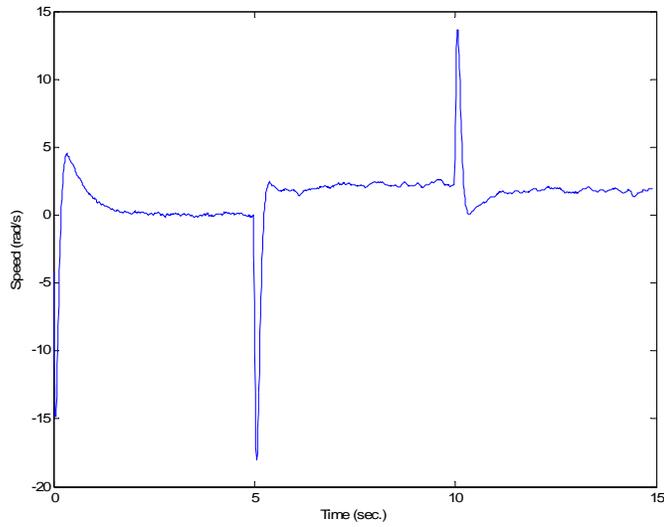


Figure 5.52 Validation Error from Real and Estimated Speed from Recursive Experiment with Fixed R_a

CHAPTER 6

FEEDBACK LINEARIZATION

This chapter presents an application of the proposed scheme to an adaptive feedback linearization controller. Here we present the development of the feedback linearization motor control scheme and the simulations results.

6.1 Feedback Linearization Theory

In this work we want to control the motor speed to a reference speed. If we want to develop a self-commissioning scheme for the motor drive we need a control scheme that would be adaptive. Also to control the motor speed, we need to design a control scheme that would be adequate for the gray-box model structure presented in this work. The feedback linearization control scheme was selected for this purposes. In this method, the state equations could be completely linearized (full-state linearization) or the input-output is linearized, while the state equation may be only partially linearized (input-output linearization) [21]. In this work, we would demonstrate that our system state equations are completely linearized. Now we will introduce some concepts that would help understand the development of the feedback linearization control law.

6.1.1 Lie Derivatives Notation

The first concept we need to define is called the Lie Derivatives. To introduce the concept of a Lie Derivative first we have to introduce the following simple-input-single-output (SISO) system:

$$\dot{x} = f(x) + g(x)u \quad (6.1)$$

$$y = h(x) \quad (6.2)$$

where $f(x)$ is the system plant equation and the $g(x)$ is the equation related to the system input and the $h(x)$ is the output equations. If we want the first derivative of the output (\dot{y}), we define the following notion:

$$\dot{y} = \frac{\partial h}{\partial x} [f(x) + g(x)u] \stackrel{def}{=} L_f h(x) + L_g h(x) \quad (6.3)$$

where:

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) \quad (6.4)$$

which is called a Lie Derivative. The Lie derivative is the directional derivative of h with the direction of f . This notation is convenient for repeated calculation of the derivative with respect to the same vector field. For convenience the following notation is mostly used:

$$L_g L_f h(x) = \frac{\partial(L_f h)}{\partial x} g(x) \quad (6.5)$$

$$L_f^2 h(x) = L_f L_f h(x) = \frac{\partial(L_f h)}{\partial x} f(x) \quad (6.6)$$

$$L_f^k h(x) = L_f L_f^{k-1} h(x) = \frac{\partial(L_f^{k-1} h)}{\partial x} f(x) \quad (6.7)$$

and
$$L_f^0 h(x) = h(x) \quad (6.8)$$

6.1.2 Feedback Linearization for Nonlinear Systems

If we consider the system presented in (6.1) and (6.2), we first have to define the relative degree of this (SISO) system. The relative degree ρ is equal to the number of times the output y has to be differentiated in order to have the value of the input explicitly appearing. In other words, to find the relative degree we had to perform the following procedure. First we calculate the first derivative of y , denoted in this case by $y^{(1)}$, instead of \dot{y} :

$$y^{(1)} = \frac{\partial h}{\partial x} [f(x) + g(x)u] = L_f h(x) + L_g h(x)u \quad (6.9)$$

If the relative degree is larger than 1 we have that $L_g h(x) = 0$.

Now we calculate the second derivative of y to obtain the following:

$$y^{(2)} = \frac{\partial(L_f h)}{\partial x} [f(x) + g(x)u] = L_f^2 h(x) + L_g L_f h(x)u \quad (6.10)$$

If the term $L_g L_f h(x) = 0$, it mean that $y^{(2)}$ is independent of the input u . We have to repeat this process until we find that $h(x)$ satisfies the following:

$$L_g L_f^{i-1} h(x) = 0, \text{ for } i = 1, 2, \dots, \rho - 1 \quad (6.11)$$

$$L_g L_f^{\rho-1} h(x) \neq 0 \quad (6.12)$$

This means that the input u does not appear in the equations of $y, \dot{y}, \dots, y^{(\rho-1)}$ and appears in the equation $y^{(\rho)}$:

$$y^{(\rho)} = L_f^\rho h(x) + L_g L_f^{\rho-1} h(x)u \quad (6.13)$$

Equation 6.13 shows that the system is input-output linearizable using the following state feedback control law to cancel the nonlinearities of the system

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} [-L_f^\rho h(x) + v] \quad (6.14)$$

Substituting (6.14) into (6.13) reduces the input-output map to;

$$y^{(\rho)} = v \quad (6.15)$$

This is similar to connect a chain of ρ integrators to the system. This integer ρ is what is called the relative degree of the system [21].

Now the next step is to develop a nonlinear version for the nonlinear system presented in (6.1) – (6.2) that would had a relative degree of ρ . We have to find a transformation $z = T(x)$ that would be a diffeomorphism on the domain of interest. It would change the system into the normal form. This means that the change of variables $z = T(x)$ had to be invertible and differentiable. In other words, it must have an inverse map $T^{-1}(\cdot)$ such that $x = T^{-1}(z)$ for all z in the domain of T , and since x and z are continuous, both $T(\cdot)$ and $T^{-1}(\cdot)$ have to be continuously differentiable. The following theorem summarize this concepts [21].

Theorem 6.1: Consider the system 6.1 – 6.2, and suppose it has relative degree $\rho \leq n$ in the domain of interest D . If $\rho = n$, then for every $x_0 \in D$ a neighborhood N of x_0 exists such that the map

$$z = T(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} = [\xi] \quad (6.16)$$

restricted to N , is a diffeomorphism on N .

For the case when $\rho < n$ please refer to [21] for more information.

Now we can linearized the system via the state feedback equation:

$$u = [\alpha(x) + \beta(x)v] \quad (6.17)$$

where $\beta(x) = \gamma^{-1}(x)$ and from (6.14) we have that:

$$\gamma(x) = L_g L_f^{\rho-1} h(x) \text{ and } \alpha(x) = -\frac{L_f^\rho h(x)}{L_g L_f^{\rho-1} h(x)} \quad (6.18)$$

They can be expressed in the new coordinates by setting:

$$\alpha_0(\eta, \xi) = \alpha(T^{-1}(z)) \text{ and } \gamma_0(\eta, \xi) = \gamma(T^{-1}(z)) \quad (6.19)$$

When $\rho = n$, and we used the transformation in (6.16) the system reduces to to:

$$\dot{z} = A_c z + B_c \gamma(x)[u - \alpha(x)] \quad (6.20)$$

$$y = C_c z \quad (6.21)$$

where $z = [h(x), \dots, L_f^{n-1} h(x)]^T$.

In summary, the system (6.1) to (6.2) is feedback linearizable if and only if a function $h(x)$ exists such that the system (6.1) to (6.2) has relative degree n (equal to the degree of the system) or equivalent, h satisfies the partial differential equations (6.11) when $\rho = n$, subject to the condition of (6.12). For this case when $\rho = n$, the system is said to be full-state linearizable. This means that the system state equations are completely linearizable.

These concepts about feedback linearization were implemented in the gray-box model of the electric drive system. The procedure is going to be presented in the next section.

6.1.3 Feedback Linearization for Gray-Box Modeling of Electric Drive

Now we are going to apply the feedback linearization scheme to the permanent magnet DC drive. First, we define the nonlinear electric and mechanical equations as were presented in Chapter 3.

$$\frac{di_a(t)}{dt} = \frac{V_a(t)}{L_a} - \frac{R_a}{L_a} i_a(t) - \frac{K_a}{L_a} \omega(t) \quad (6.22)$$

$$\frac{d\omega(t)}{dt} = \frac{K_a}{J_m} i_a(t) - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(\frac{(\omega(t)-c_i)^2}{2\sigma_i^2} \right)} \right) \quad (6.23)$$

On this work we want to control the speed of the motor so we define the output as follows:

$$y = \omega(t) \quad (6.24)$$

Now we have to arrange the equations as are presented in equations (6.1) and (6.2) in order to proceed with the feedback linearization analysis. For this purpose we define that $x_1=ia(t)$, $x_2=\omega(t)$ and $V_a(t) = u$, then we could define $f(x)$, $g(x)$ and $h(x)$:

$$f(x) = \begin{bmatrix} -\frac{R_a}{L_a} x_1 - \frac{K_a}{L_a} x_2 \\ \frac{K_a}{J_m} x_1 - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(-\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \end{bmatrix} \quad (6.25)$$

$$g(x) = \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} \quad (6.26)$$

$$h(x) = [x_2] \quad (6.27)$$

Now the first procedure we performed was to find the relative degree of the system.

Performing the first derivative, we arrive to the following expression:

$$\dot{y} = \frac{K_a}{J_m} x_1 - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(-\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \quad (6.28)$$

Since on this equation is independent of the input we had to derivate for a second time to see if was dependent of the output. We arrive with the following:

$$\begin{aligned}
y^{(2)} &= \frac{K_a}{J_m} \left[-\frac{R_a}{L_a} x_1 - \frac{K_a}{L_a} x_2 + \frac{1}{L_a} u \right] - \\
&\frac{1}{J_m} \left[\sum_{i=1}^N \left(\alpha_i \left[\frac{c_i - x_2}{\sigma_i} \right] \left(e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right) \right] \left[\frac{K_a}{J_m} x_1 - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right] \quad (6.29)
\end{aligned}$$

We find that the second derivative of the output is dependent on the input. This means that $\rho = 2$ since the order of the derivative is 2. For this reason, the system has a relative degree of 2. Because the order of the system is 2, the system is also input-state linearizable. Now using equation (6.13) and (6.14), we could proceed to calculate the corresponding linearizing control law as follows.

$$L_g L_f h(x) = \frac{K_a}{J_m L_a} \quad (6.30)$$

$$\begin{aligned}
L_f^2 h(x) &= -\frac{R_a K_a}{J_m L_a} x_1 - \frac{K_a^2}{J_m L_a} x_2 - \frac{K_a}{J_m^2} \left[\sum_{i=1}^N \left(\alpha_i \left[\frac{c_i - x_2}{\sigma_i} \right] \left(e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right) \right] x_1 \\
&+ \frac{\sum_{i=1}^N \left(\alpha_i \left[\frac{c_i - x_2}{\sigma_i} \right] \left(e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right) \left(\sum_{i=1}^N \alpha_i e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right)}{J_m^2} \quad (6.31)
\end{aligned}$$

Now substituting equation (6.30) and (6.31) into equation (6.14), we arrive to the linearizing control law:

$$\begin{aligned}
u = & R_a x_1 + K_a x_2 + \frac{L_a}{J_m} \left[\sum_{i=1}^N \left(\alpha_i \left[\frac{c_i - x_2}{\sigma_i} \right] \left(e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right) \right] x_1 \\
& - \frac{L_a}{J_m K_a} \left[\sum_{i=1}^N \left(\alpha_i \left[\frac{c_i - x_2}{\sigma_i} \right] \left(e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right) \left(\sum_{i=1}^N \alpha_i e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \right] + \frac{J_m L_a}{K_a} v
\end{aligned} \tag{6.32}$$

where v are the two gain for the control law that would stabilized the linear state feedback. The input (6.32) and the following change of variable would transform the system to the normal form.

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = T(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{K_a}{J_m} x_1 - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(\frac{(x_2 - c_i)^2}{2\sigma_i^2} \right)} \right) \end{bmatrix} \tag{6.33}$$

The normal form is presented in (6.34). This form is obtained when the variables in the system are changed to the z -variables presented in (6.33). This is the reason we insert the input with the control law calculated in equation (6.32). With this procedure the nonlinearities are eliminated, but with the original x variables this could not be performed.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{K_a}{J_m} \left[-\frac{R_a}{L_a} x_1 - \frac{K_a}{L_a} z_1 + \frac{1}{L_a} u \right] - \frac{1}{J_m} \left[\sum_{i=1}^N \left(\alpha_i \left[\frac{c_i - z_1}{\sigma_i} \right] \left(e^{\left(\frac{(z_1 - c_i)^2}{2\sigma_i^2} \right)} \right) \right) \right] \left[\frac{K_a}{J_m} x_1 - \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(\frac{(z_1 - c_i)^2}{2\sigma_i^2} \right)} \right) \right] \right] \end{bmatrix} \tag{6.34}$$

where,

$$x_1 = \frac{J_m}{K_a} \left[z_2 + \frac{1}{J_m} \left(\sum_{i=1}^N \alpha_i e^{\left(-\frac{(z_1 - c_i)^2}{2\sigma_i^2} \right)} \right) \right] \quad (6.35)$$

Figure 6.1 presents the feedback linearization scheme. It could be seen that the control law receive inputs from the output signals of the motor. With this control input the system would linearize completely and would behave like a cascade of integrators, a double integrator in our case, as is presented in Figure 6.2. If you applied this analysis with the recursive parameter estimation algorithm (Figure 6.3) presented in Section 3.3 the system would be adaptive. The electrical and mechanical parameter in the equation (6.32) are going to be estimated with the recursive linear least square. This converts the feedback linearization control law into an adaptive one because the parameters used for controlling the system are being estimated in every step time. The step time used for the simulations was 0.01 seconds.

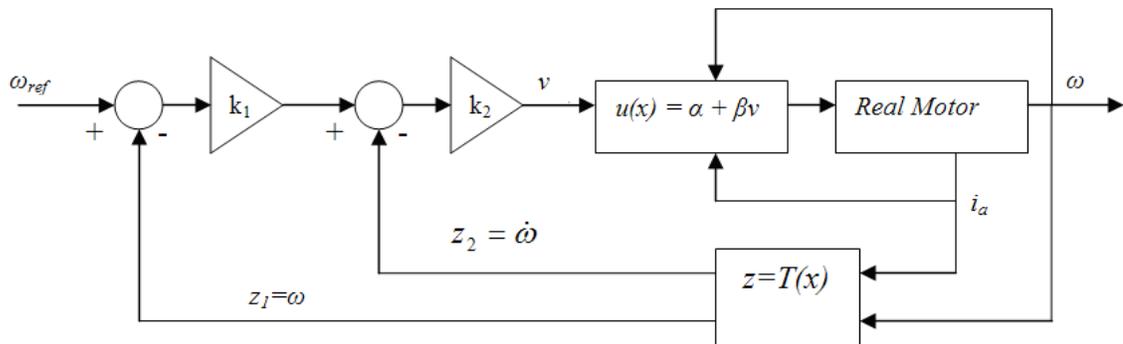


Figure 6.1 Feedback Linearization with the Motor Drive

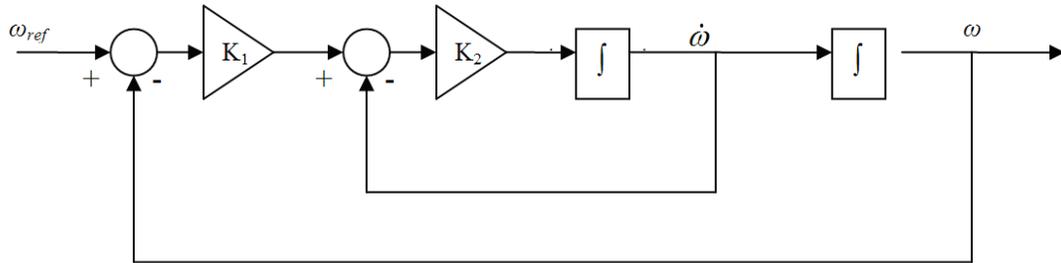


Figure 6.2 Linearize System with Double Integrator

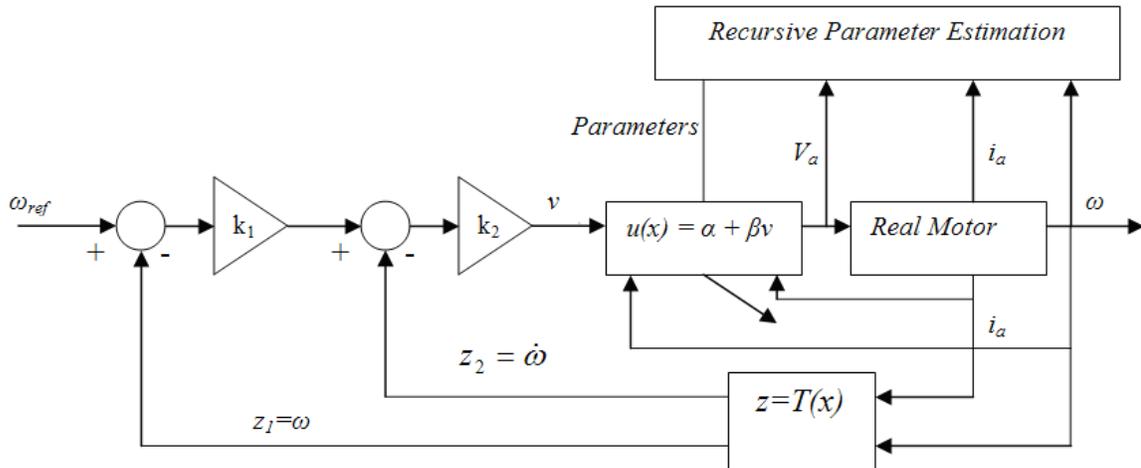


Figure 6.3 Feedback Linearization Diagram with Recursive Linear Least Squares

These methods were implemented on a Matlab[®] Simulink[™] mode. In the Simulink model, the z transformation used the same speed from the motor and for the speed acceleration a state variable filter was used to derive the speed. The Simulink Model implemented for this work is presented here:

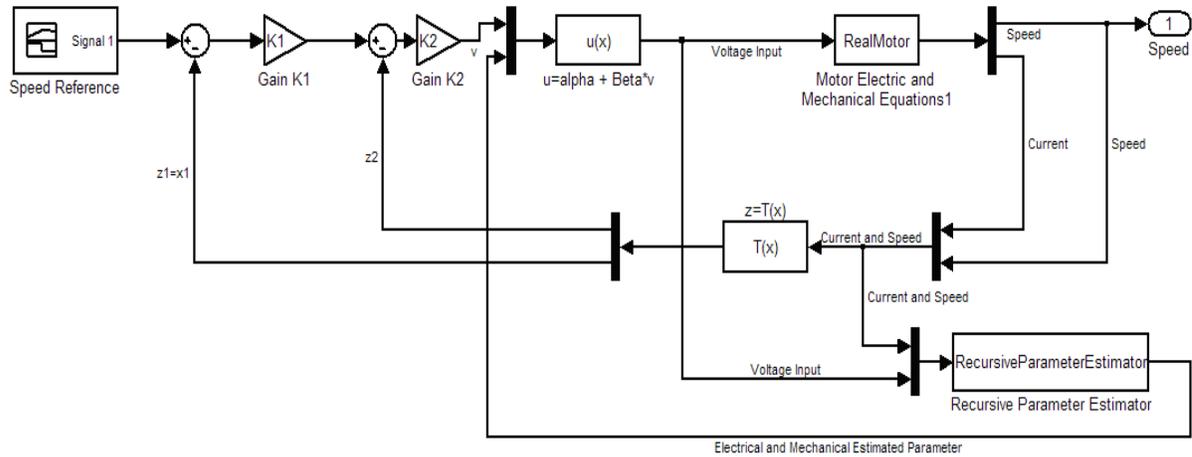


Figure 6.4 Matlab Simulink Model for the Feedback Linearization of the System

6.2 Feedback Linearization Simulation Results

Figure 6.2 shows that the linearized system behaves like a double integrator. Since in our case the system has an order of two, it could be represented as a second order system. From that figure we find that the transfer function of the system would be:

$$G(s) = \frac{k_1 k_2}{s^2 + k_2 s + k_1 k_2} \quad (6.36)$$

where k_1 and k_2 are the control gain of the system. A general second order system has the following transfer function:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (6.37)$$

where ω_n is the natural frequency of the system and ζ is the damping ratio of the system. With this values, we could design the control gain of the system. In our case we decide that the natural frequency of the system would be 17.32 rad/s and the damping ratio would be 0.87 s/rad. With this the gains for this system were found to be approximately: $k_1 = 10$ and $k_2 = 30$.

These gains were implemented in the Matlab Simulink feedback linearization scheme without noise. Figure 6.5 presents the speed output compared with the speed output of an ideal double integrator scheme as in Figure 6.2. It could be seen that the system is behaving almost identical as the ideal case. Figure 6.6 presents the results from the speed output versus the reference signal used to control the speed. It could be seen that the reference speed is been track by the speed output with small errors. These results demonstrate that the adaptive feedback linearization scheme is good for controlling the motor drive.

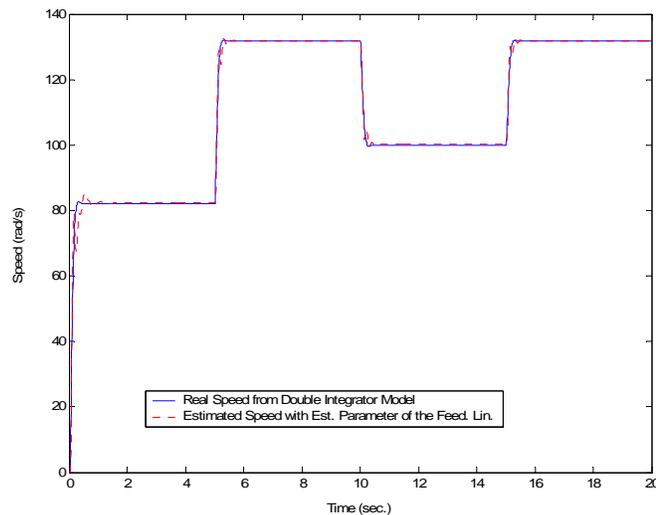


Figure 6.5 Simulated Motor Speed and Double Integrator Output

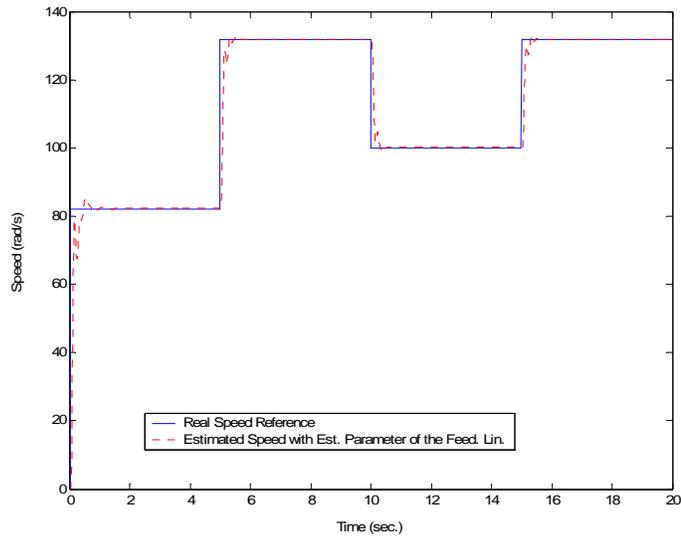


Figure 6.6 Motor Speed Output and Reference Speed

Figure 6.7 presents the speed output of the system compared with the ideal double integrator. In this case, an approximately 0.1 dB noise is added to the system. It could be seen that the system is behaving almost as the ideal case. Figure 6.8 presents the results from the speed output versus the reference signal used to control the speed. It could be seen that the reference speed is been tracked by the speed output with small errors. These results corroborate the results found without noise demonstrating that the feedback linearization scheme is good for controlling the motor drive.

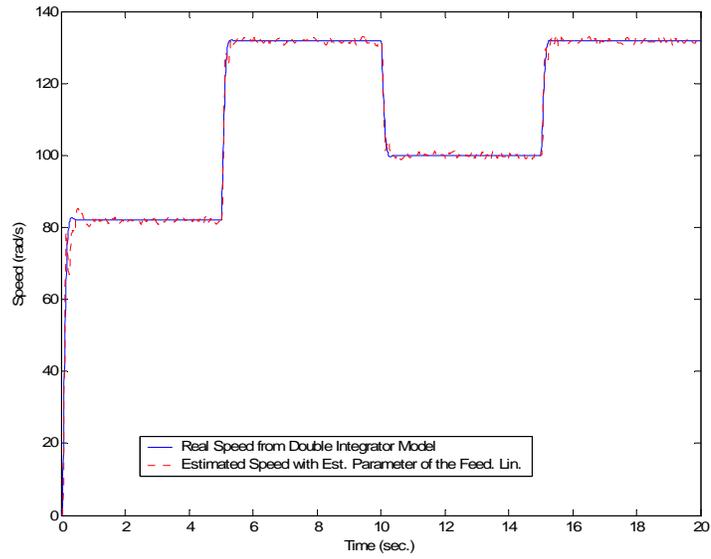


Figure 6.7 Simulated Motor Speed and Double Integrator Output with Noise

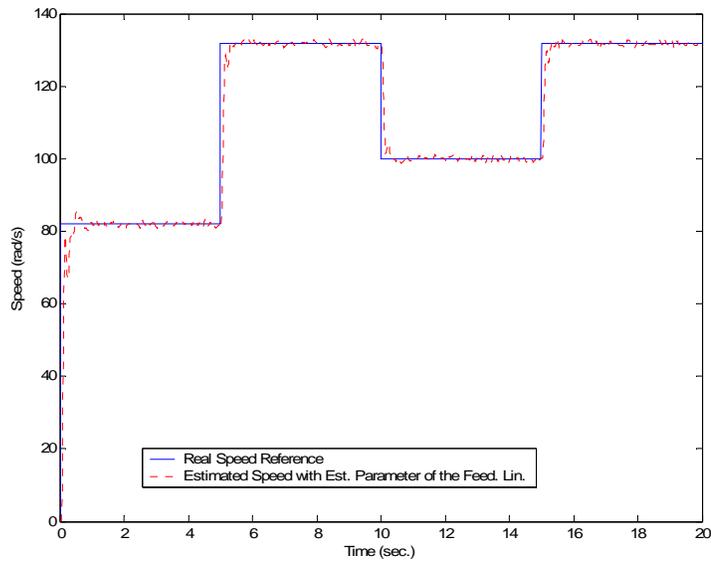


Figure 6.8 Motor Speed Output and Reference Speed with Noise

These results demonstrate that the feedback linearization scheme can effectively control an electric motor drive. This adaptive linearizing controller can be easily applied to an on-line self-commissioning control scheme.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The main objective of this work was to develop an automated methodology for the identification of electric motor drives that can be used in a self commissioning scheme. This methodology had to be flexible and simple enough to accommodate a wide variety of loads for its eventual implementation in the industry. Using the two-stage modeling approach the identification of mechanical loads meets these desired objectives.

The two-stage modeling approach presented in this work illustrates the potential of gray-box modeling for the identification of electric drives with unknown mechanical loads. The convenience of this model is that the identified physical parameters give information about the physical system and can be used to evaluate the goodness of the model. The neural network based black-box model would allow to model different types of loads independently of their actual form. The recursive algorithm identifies the system correctly and therefore, it can be implemented on-line and later used for self-tuning or self-calibration of electric drives.

The radial basis functions neural networks used for modeling the load demonstrated excellent results and validations.

The results from the simulations demonstrate the effectiveness of the methodologies presented in this work. Some problems were encountered during the implementation of the

methodologies. This is because conditions that were not affecting the simulations decreased the estimation performance during the test-bed implementation. The solution proposed in this work was to fix the parameter that was less observable from the measurements. In our case, was the armature resistance (R_a). Fixing this parameter, improves the performance of the estimation methodologies, especially for the recursive two-stage method. For example the estimated inductance (L_a) error was reduced from 25.06 %, in the full parameter case, to 1.56 %, in the fixing R_a . This is important because eventually this method is the one that is going to be implemented on a self-commissioning scheme.

It was also observed that the noise created by the switching frequency of the PWM had to be taken into consideration when designing the experimental setup. Since the methodologies presented on this work identify the electric drives the quality of them is very important. Adding analog and digital filters (State Variable Filters) to reduce the noise of the measurement is very important and almost mandatory to obtained precise and accurate results.

The simulations results, also demonstrate that the feedback linearization scheme can be effectively applied to control electric motor drives. This proposed adaptive linearizing controller can be later applied to a self-commissioning control scheme.

The work shows that gray-box the methodologies had the potential for drive identification with unknown load and its applicability for later implementation in a self-commissioning drive.

7.2 Future Work

Our interest is to continue this work by looking at other types of loads, including dynamic loads. Also is very important to implement the models on-line. We are also interested in using these schemes in other types of motor, for example the induction motor that is also commonly used in the industry.

We also want to pursue on-line implementation of the feedback linearization control scheme.

We are also interested in studying schemes to grow and prune the neural network to improve modeling flexibility while keeping a small dimension.

REFERENCES

- [1] R. Rivera-Sampayo and M. Velez-Reyes, "Gray-box modeling of electric drive systems using neural networks", *2001 IEEE International Conference on Control Applications*, pp. 146 – 151, Sept. 2001.
- [2] S. Weerasooriya and M.A. El-Sharkawi, "Identification and Control of a DC motor using Back-Propagation Neural Networks", *IEEE Transaction on Energy Conversion*, Vol. 6, No. 4, pp. 663 – 669, Dec. 1991
- [3] S. Beineke, F. Schutte, H. Wertz and H. Grotstollen, "Comparison of parameter identification schemes for self-commissioning drive control of nonlinear two-mass system", *Proceedings of IEEE IAS Annual Meeting*, Vol. 1, pp.-493-500, Oct. 1997
- [4] S. Beineke, F. Schutte and J. Grotstollen, "Online identification of nonlinear mechanics using extended Kalman filter with basis function networks", *Proceedings of the Industrial Electronics Conference*, pp. 316-321, 1997
- [5] L. R. Ray and J. S. Remine, "Machine friction for modeling, diagnostics and control", *Proceedings of the American Control Conference*, pp. 2737 – 2741, June 1998
- [6] R. H. A. Hensen, G. Z. Angelis, M. J. G. van de Molengraft, A. G. de Jager and J. J. Kok, "Gray-box modeling of friction: An Experimental Case-study", *European Journal of Control*, pp. 258 – 267, 2000
- [7] H. Wertz, S. Beineke, N. Frohleke, S. Bolognani, K. Unterkofler, M. Zigliotto and M. Zordan, "Computer aided commissioning of speed and position control for electrical drives with identification of mechanical load", *IEEE Industry Application Conference 1999*, pp. 2372-2379, Vol. 4, Oct. 1999
- [8] Y. Baéz and M. Velez-Reyes, "Two-Stage Identification of Motor Drives using Gray Box Models", *2003 North American Power Symposium*, 2003
- [9] H. Shierling, "Self-commissioning – A novel feature of modern-fed induction motor drives", *IEEE Power Electronics and Variable-Speed Drives*, pp. 287 – 290, July 1988
- [10] H. Ren-Mu and A.J. Germond, "Comparison of dynamic load modeling using neural network and traditional method", *1993 ANNPS Neural Networks to Power Systems*, pp. 253-258, Apr. 1993

- [11] M. Kirby, *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns*, Wiley, 2001
- [12] S. J. Leon, *Linear Algebra with Applications*, Prentice Hall, New Jersey, 2002
- [13] R. Johansson, *System Modeling Identification*; Prentice Hall, New Jersey, 1993
- [14] M.H.A. Davis and R.B. Vinter, *Stochastic Modelling and Control*, Chapman and Hall, New York, 1985.
- [15] T. Bakken, J. Choma, “Stability of a continuous-time state variable filter with ORA-L and current amplifier integrators”, *IEEE 1998 International Symposium on Circuit and Systems*, Vol. 3, pp. 643 – 646, June 1998
- [16] L.S.H. Ngia, “Separable nonlinear least-squares methods for on-line estimation of neural nets Hammerstein models”, Proc. of the Signal Processing Society 2000 IEEE Neural Networks for Signal Processing X, Vol. 1, pp. 65 – 74, Dec. 2000.
- [17] dSPACE Manual, Floating-point Controller Board DS1102 User’s Guide, Digital signal processing and control engineering GmbH, Paderborn, Germany, Manual, 1996
- [18] *Isolation Amplifier HCPL-7840 Technical Data*, Agilent Technology, Manual, March 2005
- [19] J. C. Tafur, and M. Velez-Reyes, “An adaptive feedback linearizing controller for a shunt DC motor”, *2002 IEEE Workshop on Computers in Power Electronics*, pp. 144 – 151, June 2002.
- [20] M. Velez-Reyes, W. L. Fung and J. E. Ramos-Torres, “Developing robust algorithms for speed and parameter estimation in induction machines”, *2001 IEEE Conference on Decision and Control*, Vol. 3, pp. 2223 – 2228, Dec. 2001
- [21] M. Vélez-Reyes and G.C. Verghese, “Subset Selection in Identification, and Application to Speed and Parameter Estimation for Induction Machines”, In *Proceedings of the 4th IEEE Conference on Control Applications*, Albany NY, September 28-29, 1995.
- [22] H. K. Khalil, *Nonlinear Systems 3rd Edition*, Prentice Hall, 2002, pp. 505 – 550.

APPENDIX

Appendix A –Matlab File for the Experiment Batch Linear Least Square Case

```
%%% This code calculate the model parameters%%%  
%%% as well as the system responses %%%  
%%% without the damping factor %%%  
%%% The Data is obtained by the Experimental Set-up %%%
```

```
load motorvaldcsa111
```

```
t2=zeros(1,2051);  
t2(1,:)=motorvaldcsa111.X.Data(116:2166);  
t2=t2-motorvaldcsa111.X.Data(116);
```

```
speed=zeros(1,2051);  
speed(1,:)=motorvaldcsa111.Y(4).Data(116:2166);
```

```
speed2=zeros(1,2051);  
speed2(1,:)=motorvaldcsa111.Y(3).Data(116:2166);
```

```
voltage=zeros(1,2051);  
voltage(1,:)=motorvaldcsa111.Y(6).Data(116:2166);
```

```
current=zeros(1,2051);  
current(1,:)=motorvaldcsa111.Y(1).Data(116:2166);
```

```
current2=zeros(1,2051);  
current2(1,:)=motorvaldcsa111.Y(2).Data(116:2166);
```

```
dspeed=zeros(1,2051);  
dspeed(1,:)=motorvaldcsa111.Y(8).Data(116:2166);
```

```
dcurrent=zeros(1,2051);  
dcurrent(1,:)=motorvaldcsa111.Y(7).Data(116:2166);
```

```
Vol=[voltage]';  
Vel=[speed]';  
Cor=[current]';  
didt=[dcurrent]';  
dwdt=[dspeed]';  
Cor2=[current2]';  
Vel2=[speed2]';
```

```
%%% Parameter initialization %%%
```

```

centro=0:160;
L = 720.19997;
sig=4;
Rae=1.587
%%% Electrical Parameter Estimation %%%

y1 = didt;
Xe1=Vol-Rae*Cor;
Xe = [Xe1 -Vel];
theta1 = Xe \ y1;
Kae = theta1(2) / theta1(1)
Lae = 1 / theta1(1)

Tem = Kae* Cor;

for b=1:length(Vel)
    for k=1:161
        phi (b,k)= exp(-((Vel(b) - centro(k)).^2) / (2*(sig^2)));
    end
end

%%% Mechanical Parameter Estimation %%%
Xm=[dwdt phi];
theta2 = Xm \ Tem;
Jme = theta2(1)

alpha_estim = theta2(2:162);

vel = centro;
for j=1:161
    for h=1:161
        phi_fan (j,h)= exp(-((vel(j) - centro(h)).^2) / (2*(sig^2)));
    end
end
%
TL_fan_estim = phi_fan*alpha_estim;

iae=(Jme/Kae)*dwdt + (1/Kae)*phi*alpha_estim ;

%Estimated velocity
wme = (1/Kae)*Vol-(Rae/Kae)*Cor-(Lae/Kae)*diddt;
%%% Real Torque %%%
load torque

t9=zeros(1,5287);
t9(1,:)=torque.X.Data(446:5732);
t9=t9-torque.X.Data(446);

```

```

speedt=zeros(1,5287);
speedt(1,:)=torque.Y(4).Data(446:5732);

speed2t=zeros(1,5287);
speed2t(1,:)=torque.Y(3).Data(446:5732);

voltageg=zeros(1,5287);
voltageg(1,:)=torque.Y(6).Data(446:5732);

currentt=zeros(1,5287);
currentt(1,:)=torque.Y(1).Data(446:5732);

current2t=zeros(1,5287);
current2t(1,:)=torque.Y(2).Data(446:5732);

dspeedt=zeros(1,5287);
dspeedt(1,:)=torque.Y(8).Data(446:5732);

dcurrentt=zeros(1,5287);
dcurrentt(1,:)=torque.Y(7).Data(446:5732);

Volt=[voltageg]';
Velt=[speedt]';
Cort=[currentt]';
didtt=[dcurrentt]';
dwdtt=[dspeedt]';
Cor2t=[current2t]';
Vel2t=[speed2t]';

sig= 4;
centro=0:160;

Velft=zeros(1,18);
pos=290;
ii=2;
while ii < 19
    Velft(ii)=Velt(pos);
    pos=pos+300;
    ii=ii+1;
end
Corft=zeros(1,18);
pos=290;
jj=2;
while jj < 19
    Corft(jj)=Cort(pos);
    pos=pos+300;

```

```

    jj=jj+1;
end
miui=0.000033;

Bme=0.0008599;

TL=Kae*Corft-Bme*Velft;
jj=1;
pos=1;
TLfan=zeros(1,18);
while jj < 19
    TLfan(jj)=TL_fan_estim(pos);
    pos=pos+9;
    jj=jj+1;
end

%Errors
e_fan=(TLfan-TL);
e_ifan=(iae-Cor);
e_wfan=(wme-Vel);

%%% Results Plots %%%

figure(1);
plot(Velft,TL,'--');
hold on;
plot(vel,TL_fan_estim,':r')
legend('Real Load Torque (Fan)', 'Estimated Load Torque (Fan)');

figure(2);
plot(Velft,e_fan);

figure(3);
plot(t2,Cor,'--');
hold on;
plot(t2,iae,':r')
legend('Real Current with load', 'Estimated Current with Fixed Ra');

figure(4);
plot(t2,e_ifan);

figure(5);
plot(t2,Vel,'--');
hold on;
plot(t2,wme,':r');
legend('Real Speed with load', 'Estimated Speed with Fixed Ra');

```

```
figure(6);  
plot(t2,e_wfan);
```

```
figure(7);  
plot(t2,Vol);
```

Appendix B - Matlab File for the Experiment Recursive Batch Linear Least Square Case

```
% This file implements the Recursive Newton-Raphson algorithm for parameter
% estimation with separated Parameters, a fan as
% a load and RBF as a Nonlinear Load.
% The Data is obtained by the Experimental Set-up
%%%%%%%%%%
%%%%%%%%%%
```

```
load motorvaldcsa111
```

```
t2=zeros(1,2051);
t2(1,:)=motorvaldcsa111.X.Data(116:2166);
t2=t2-motorvaldcsa111.X.Data(116);
```

```
speed=zeros(1,2051);
speed(1,:)=motorvaldcsa111.Y(4).Data(116:2166);
```

```
speed2=zeros(1,2051);
speed2(1,:)=motorvaldcsa111.Y(3).Data(116:2166);
```

```
voltage=zeros(1,2051);
voltage(1,:)=motorvaldcsa111.Y(6).Data(116:2166);
```

```
current=zeros(1,2051);
current(1,:)=motorvaldcsa111.Y(1).Data(116:2166);
```

```
current2=zeros(1,2051);
current2(1,:)=motorvaldcsa111.Y(2).Data(116:2166);
```

```
dspeed=zeros(1,2051);
dspeed(1,:)=motorvaldcsa111.Y(8).Data(116:2166);
```

```
dcurrent=zeros(1,2051);
dcurrent(1,:)=motorvaldcsa111.Y(7).Data(116:2166);
```

```
Vol=[voltage]';
Vel=[speed]';
Cor=[current]';
didt=[dcurrent]';
dwdt=[dspeed]';
Cor2=[current2]';
Vel2=[speed2]';
```

Rai=1.41388;
Lai=0.48680;
Kai=0.33816;
Jmi=0.00606;
miui=0.000019;
alpha=1.0e+007 *[-0.00010822140508
0.00038620111603
0
-0.00209538190804
0.00359578304453
0
-0.00512996005802
0
0.01137316641420
0
-0.04136899213133
0.05794999345755
0
-0.06620031636389
0
0.21507951822246
-0.39536901135816
0.36147825257340
-0.16235576638670
0
0.02825563608850
0
0
-0.01171613692283
0
0.01202218434448
0
-0.01069762083173
0
0.00804710137676
0
0
-0.01045433937233
0
0.02041034356443
0
-0.03814818651615
0
0.10425995156697
-0.12280819875728
0
0.09023170114954
0
-0.13744549978376
0.11518509196788

0
0
-0.06864876014350
0
0.10946631226116
0
0
-0.99542143046008
3.08405800035408
-4.73277604033409
4.21244861535349
-1.87174445759372
0
0
0.62498435928445
0
-1.17611326515839
0
4.59167025463039
-9.40207447140170
9.91812076927363
-5.42652682043403
0
1.92072308162285
0
-2.49963946294102
2.85274480194947
-1.37173020990126
0
0.28154597812890
0
-0.11013558537336
0
0.05171735354867
0
-0.02403591478512
0
0
0.02516924924257
-0.02412648847713
0
0.00799274451480
0
0
-0.00469104404858
0
0.00523733279599
0
-0.00483173895297
0
0.00375121518192

0
0
-0.00731818005445
0.00845972408847
0
-0.00403060044804
0
0
0.00411409956702
0
-0.00686978566212
0
0.01009844251366
0
-0.01476869427395
0
0.02318702069817
0
-0.05171681319189
0.05146432844019
0
0
-0.07711474668470
0.10454149086181
0
-0.11904812006401
0.10374756064118
0
-0.04081285392470
0
0.02162192362869
0
0
-0.02140540884878
0
0.04976864681874
-0.04849572408032
0
0.01860626583943
0
0
-0.01425023696082
0
0.01857509263244
0
-0.01919885758832
0
0.01619873628611
0
-0.00934979190759
0

```

0
0
0.00681873535022
0
0
-0.01138675931051
0
0
0.01481652113959
0
0
0
0
0];

po=[Kai/Lai 1/Lai Kai/Jmi miui/Jmi]';
%%% Initialization Parameters
pobe=zeros(2,2050);
pobm=zeros(162,2050);

% Initial Parameters

Raii=2.0045;
Laii=0.58126;
Kaii=0.4736;
Jmii=0.00583;
alphai=0.9*alpha';
sig=4;

%Parameter Separation
pE=[Raii/Laii Kaii/Laii 1/Laii]';
pM=[Jmii
alphai(1)
alphai(2)
alphai(3)
alphai(4)
alphai(5)
alphai(6)
alphai(7)
alphai(8)
alphai(9)
alphai(10)
alphai(11)
alphai(12)
alphai(13)
alphai(14)
alphai(15)
alphai(16)
alphai(17)
alphai(18)
alphai(19)

```

alphi(20)
alphi(21)
alphi(22)
alphi(23)
alphi(24)
alphi(25)
alphi(26)
alphi(27)
alphi(28)
alphi(29)
alphi(30)
alphi(31)
alphi(32)
alphi(33)
alphi(34)
alphi(35)
alphi(36)
alphi(37)
alphi(38)
alphi(39)
alphi(40)
alphi(41)
alphi(42)
alphi(43)
alphi(44)
alphi(45)
alphi(46)
alphi(47)
alphi(48)
alphi(49)
alphi(50)
alphi(51)
alphi(52)
alphi(53)
alphi(54)
alphi(55)
alphi(56)
alphi(57)
alphi(58)
alphi(59)
alphi(60)
alphi(61)
alphi(62)
alphi(63)
alphi(64)
alphi(65)
alphi(66)
alphi(67)
alphi(68)
alphi(69)
alphi(70)

alphi(71)
alphi(72)
alphi(73)
alphi(74)
alphi(75)
alphi(76)
alphi(77)
alphi(78)
alphi(79)
alphi(80)
alphi(81)
alphi(82)
alphi(83)
alphi(84)
alphi(85)
alphi(86)
alphi(87)
alphi(88)
alphi(89)
alphi(90)
alphi(91)
alphi(92)
alphi(93)
alphi(94)
alphi(95)
alphi(96)
alphi(97)
alphi(98)
alphi(99)
alphi(100)
alphi(101)
alphi(102)
alphi(103)
alphi(104)
alphi(105)
alphi(106)
alphi(107)
alphi(108)
alphi(109)
alphi(110)
alphi(111)
alphi(112)
alphi(113)
alphi(114)
alphi(115)
alphi(116)
alphi(117)
alphi(118)
alphi(119)
alphi(120)
alphi(121)

```
alpha(122)
alpha(123)
alpha(124)
alpha(125)
alpha(126)
alpha(127)
alpha(128)
alpha(129)
alpha(130)
alpha(131)
alpha(132)
alpha(133)
alpha(134)
alpha(135)
alpha(136)
alpha(137)
alpha(138)
alpha(139)
alpha(140)
alpha(141)
alpha(142)
alpha(143)
alpha(144)
alpha(145)
alpha(146)
alpha(147)
alpha(148)
alpha(149)
alpha(150)
alpha(151)
alpha(152)
alpha(153)
alpha(154)
alpha(155)
alpha(156)
alpha(157)
alpha(158)
alpha(159)
alpha(160)
alpha(161)];
```

```
alph = 1;
```

```
n=1;
xn=ones(2,1);
xn2=ones(332,1);
xn=0.01*xn;
cw=zeros(2,2051);
cw(1,:)=didt';
cw(2,:)=0.3409*Cor';
xn2=0.01*xn2;
```

```

uu=zeros(1,2051);
tt=zeros(1,2051);
JJ=zeros(3,2051);
t=t2;

% Voltage Input
Va=zeros(1,2001);

t3=0;
h=1;

u=Vol';

the1=po(1);    % 1. parameter to be estimated
the2=po(2);    % 2. parameter to be estimated
the3=po(3);    % 3. parameter to be estimated
the4=po(4);    % 4. parameter to be estimated
the5=po(5);    % 5. parameter to be estimated

didt;%=1/Lai*Vol-Rai/Lai*Cor-Kai/Lai*Vel;
dwdt;%=Kai/Jmi*Cor-miui/Jmi*(sign(Vel).*(Vel));
xnn=cw;
cwv=zeros(3,2051);
cwv(1,:)=Cor';
cwv(2,:)=Vel';
cwv(3,:)=Vol';
mear=cwv;
e1=0;
sal=zeros(2,2051);

%%% Beginning of recursive algorithm
while(n<2051)

%%% Recursive Newton-Raphson

y2=xnn';
mea=mear';

xn=xn;
xn2=xn2;

```

```

i=1;
change=100000;

if n < 2052
[ee,em,xn,yxx] = errerdnew(pE,pM,u,t,y2,xn,n);
yxx=yxx;
[Je,Jm,xn2]= regradnew(pE,pM,u,t,y2,xn2,mea,yxx,n);
% uu(1,n)=u(1,n);
tt(1,n)=t(1,n);
xn=xn;
sal(:,n)=xn;
xn2=xn2;
Je=Je';
Jm=Jm';
ee=ee;
em=em;

end

if n==1
PNe=4*[1 0 0 ;0 1 0 ;0 0 1];
PNm=0.2*eye(162);
end
PNoe=PNe;
PNom=PNm;
if n>1

PNe=(PNoe-(PNoe*Je'*Je*PNoe)/(1+Je*PNoe*Je'));
PNm=(PNom-(PNom*Jm'*Jm*PNom)/(1+Jm*PNom*Jm'));
end

i=i+1;

h=1;
if n>0

alphe=0.0068;
alpm=0.000508;

pE=pE - (alphe/1)*PNe*Je'*ee;
pM=pM - (alpm/1)*PNm*Jm'*em;

h=h+1;
end

```

```

p_prevelec=pE
p_prevmech=pM;
Jme = pM(1)
pobe(:,n)=p_prevelec;
pobm(:,n)=p_prevmech;
n=n+1;

end
% figure(1)
% plot(tt(1:2050),pobe);
% figure(2)
% plot(tt(1:2050),pobm);

the1=pE(1); % 1. parameter to be estimated
the2=pE(2); % 2. parameter to be estimated
the3=pE(3); % 3. parameter to be estimated
the4=pM(1); % 4. parameter to be estimated

Rai
Lai
Kai
Jmi

Rae = the1/the3
Lae = 1/the3
Kae = the2/the3
Jme = (the4)
alphae=zeros(161,1);
ss=1
while ss < 162
    alphae(ss)=pM(ss+1);
    ss=ss+1;
end

alphae=[alphae];
alphaee=zeros(161,2);
alphaee(:,1)=alphae;
alphaee(:,2)=alpha;
%alphaee=[alphaee]

centro=0:160;
for b=1:length(Vel)
    for k=1:161
        phi (b,k)= exp(-((Vel(b) - centro(k)).^2) / (2*(sig^2)));
    end
end

didt=didt;
dwdt=dwdt;

```

```

%Estimated current
iaee=sal(1,:);%(Jme/Kae)*dwdt + (1/Kae)*phi*alphae;
iae=(Jme/Kae)*dwdt + (1/Kae)*phi*alphae;
%Estimated velocity
wmee = sal(2,:);%(1/Kae)*Vol-(Rae/Kae)*Cor-(Lae/Kae)*didt;
wme = (1/Kae)*Vol-(Rae/Kae)*Cor-(Lae/Kae)*didt;

vel = centro;
for j=1:161
    for h=1:161
        phi_fan (j,h)= exp(-((vel(j) - centro(h)).^2) / (2*(sig^2)));
    end
end
%
TL_fan_estim = phi_fan*alphae;

%%% Real Torque %%%
load torque

t9=zeros(1,5287);
t9(1,:)=torque.X.Data(446:5732);
t9=t9-torque.X.Data(446);

speedt=zeros(1,5287);
speedt(1,:)=torque.Y(4).Data(446:5732);

speed2t=zeros(1,5287);
speed2t(1,:)=torque.Y(3).Data(446:5732);

voltageget=zeros(1,5287);
voltageget(1,:)=torque.Y(6).Data(446:5732);

currentt=zeros(1,5287);
currentt(1,:)=torque.Y(1).Data(446:5732);

current2t=zeros(1,5287);
current2t(1,:)=torque.Y(2).Data(446:5732);

dspeedt=zeros(1,5287);
dspeedt(1,:)=torque.Y(8).Data(446:5732);

dcurrentt=zeros(1,5287);
dcurrentt(1,:)=torque.Y(7).Data(446:5732);

```

```

Volt=[voltage];
Velt=[speed];
Cort=[current];
didtt=[dcurrent];
dwdtt=[dspeed];
Cor2t=[current2t];
Vel2t=[speed2t];

sig= 4;
centro=0:160;

Velft=zeros(1,18);
pos=290;
ii=2;
while ii < 19
    Velft(ii)=Velt(pos);
    pos=pos+300;
    ii=ii+1;
end
Corft=zeros(1,18);
pos=290;
jj=2;
while jj < 19
    Corft(jj)=Cort(pos);
    pos=pos+300;
    jj=jj+1;
end
miui=0.000063;
Bme=0.0008599;
% TL=(miui*sign(vel).*((vel).^2));
TL=0.3409*Corft-Bme*Velft;
jj=1;
pos=1;
TLfan=zeros(1,18);
while jj < 19
    TLfan(jj)=TL_fan_estim(pos);
    pos=pos+9;
    jj=jj+1;
end
e_fan=(TLfan-TL);
e_ifan=(iae-Cor);
e_wfan=(wme-Vel);
% e_fullfan=(fulltorque-fullfan);

%%% Results Plots %%%

figure(1);
plot(Velft,TL,'-');

```

```

hold on;
plot(vel,TL_fan_estim,':r')
legend('Real Load Torque (Fan)','Estimated Load Torque (Fan)');

figure(2);
plot(Velft,e_fan);

figure(3);
plot(t,Cor,'--');
hold on;
plot(t,iae,':r')
legend('Real Current with load','Estimated Current with Estimated Parameters');

figure(4);
plot(t,e_ifan);
%
figure(5);
plot(t,Vel,'--');
hold on;
plot(t,wme,':r');
legend('Real Speed with load','Estimated Speed with Estimated Parameters');

figure(6);
plot(t,e_wfan);

pob2e=zeros(3,2050);
pob2e(1,1:2050)=3.876;
pob2e(2,1:2050)=0.8326;
pob2e(3,1:2050)=2.4426;

figure(7)
plot(tt(1:2050),pobe);
hold on;
plot(tt(1:2050),pob2e,':');
legend('Estimated Ra/La','Estimated Ka/La','Estimated 1/La','Ra/La','Ka/La','1/La');

pobm1=zeros(1,2050);
pobm1(1:2050)=pobm(1,1:2050);
pob2m=zeros(1,2050);
pob2m(1:2050)=142.81525;

figure(8)
plot(tt(1:2050),pobm1);
hold on;
plot(tt(1:2050),pob2m,':');
legend('Estimated Ka/Jm','Ka/Jm');

```

*****errednew Function*****

%This function computes the error function of the recursive algorithm

%between the measurements and the prediction model for the example system.

%%%

```
function [ee,em,xc,yxx] = errerdnew(pe,pm,u,t,y,x,n);
```

```
the1=pe(1);    % 1. parameter to be estimated
the2=pe(2);    % 2. parameter to be estimated
the3=pe(3);    % 3. parameter to be estimated
the4=pm(1);    % 4. parameter to be estimated
Kaet=the2/the3;
```

```
B = [the3;0];
C = [1 1];
D = [0];
```

```
cen=0:160;
xn=x;
th=0.01;
t1=t(1,n);
```

```
u1=u(1,n);
```

```
sig=4;
c=1;
A22=0;
while c < 162
A2o=(pm(c+1)/the4)*exp(-((xn(2,1)-cen(c))^2)/(2*sig^2));
A22=A22+A2o;
c=c+1;
end
A22=A22;
```

```
A = [ -the1  -the2
      Kaet/the4  -A22/((xn(2,1)))];
xn=(eye(2,2)+th*A)*xn+th*B*(u1);
```

```
yx=C*xn;
```

```
diidt=-the1*xn(1,1)-the2*xn(2,1)+u1*the3;
dwwdt=(Kaet/the4)*xn(1,1)-A22;
```

```
t2=Kaet*xn(2,1);
ee=y(n,1)-diidt;
```

```

em=y(n,2)-t2;

yxx=zeros(3,1);
yxx(1,1)=diidt;
yxx(2,1)=t2;
yxx(3,1)=dwwdt;
yxx=yxx;
xc=xn;

*****regradnew*****

%This function calculates the Jacobian of the recursive algorithm
%for the system with fixed Ra.

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

function [Je,Jm,xn3]=regradnew(pe,pm,u,t,y,x,me,yx,n);

the1=pe(1);    % 1. parameter to be estimated
the2=pe(2);    % 2. parameter to be estimated
the3=pe(3);    % 3. parameter to be estimated
the4=pm(1);    % 4. parameter to be estimated

xb=me;
yxn=yx;

cen=0:160;
xn=x;
sig=4;
h=2;

Jfm=zeros(162,1);
g=1;
cv=1;
while g < 162
    Jfm(h,1)=exp(-((xn(2,1)-cen(cv))^2)/(2*sig^2));
    cv=cv+1;
    g=g+1;
    h=h+1;
end
Jfm(1,1)=yxn(3,1);
Jm=-Jfm;

uu=u(1,n);
xn3=zeros(162,1);
Je=zeros(3,1);

```

$Je(1,1)=-xn(1,1);$
 $Je(2,1)=-xn(2,1);$
 $Je(3,1)=uu;$
 $Je=-Je;$