University of South Florida

Digital Commons @ University of South Florida

USF Tampa Graduate Theses and Dissertations

USF Graduate Theses and Dissertations

April 2022

A Model-Based Fault Diagnosis in Dynamic Systems via Asynchronous Motors System Identification or Testing, and Control Engineering Observers

Kenelt Pierre
University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Electrical and Computer Engineering Commons

Scholar Commons Citation

Pierre, Kenelt, "A Model-Based Fault Diagnosis in Dynamic Systems via Asynchronous Motors System Identification or Testing, and Control Engineering Observers" (2022). *USF Tampa Graduate Theses and Dissertations.*
https://digitalcommons.usf.edu/etd/9436

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact scholarcommons@usf.edu.

A Model-Based Fault Diagnosis in Dynamic Systems via Asynchronous Motors System Identification or Testing, and Control Engineering Observers

by

Kenelt Pierre

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Co-Major Professor: Wilfrido Moreno, Ph.D.
Co-Major Professor: Chung Seop Jeong, Ph.D.
Andrew Hoff, Ph.D.
Alex Volinsky, Ph.D.
Brig Bjorn, Ph.D.

Date of Approval:
April 15, 2022

Keywords: Detection, Isolation, Threshold, Ambiguity, Digital Twin

Copyright © 2022, Kenelt Pierre

Dedication

I dedicate this dissertation to my thirteen-year-old son Kenneth who always asks me: “Dad, when are you going to be done with this equipment? When you are done, may I have them?” I would also like to dedicate this dissertation to my sister Maggie and my nephew Joshua who always believe in me.

Acknowledgments

As in one of my published papers in IEEE, I thank my advisors Wilfrido Moreno, PhD., PE, and Chung Seop Jeong, Ph.D., both Professors at the University of South Florida for encouraging me to write a paper as a prelude to my doctoral dissertation in Control Engineering Model-Based Fault Diagnostics and for understanding my schedule constraints due to my full-time employment. I thank Dr. Moreno for all his technical advice and input. I also thank Dr. Thomas Weller for referring me to Dr. Moreno when I switched focus from Microwave Engineering to Control Engineering Fault Diagnostics.

Finally, I thank my Mentor John Perazza a Lockheed Martin Sr. Fellow for teaching me how to implement Reliability Engineering and Fault Management System methods in writing proposals, in executing government contracts, and in presenting technical data to Lockheed Martin customers. I thank John for patiently answering all my technical and career related questions. All credits and thanks are due to the One Infinite Creator for giving me opportunities, strength and persistence to carry on and for connecting me with such great, intelligent, understanding, and supportive people in my life.

Table of Contents

List of Tables

List of Figures

Abstract

Induction machines are the workhorse of many military and commercial types of equipment. They are asynchronous machines referred to in this dissertation as induction motors. They are widely used due to their reliability in Naval Unmanned Systems like Remotely Operated Vehicles (ROV), Unmanned Untethered Underwater Vehicles (UUV), Autonomous Undersea Vehicles (AUV), and in platforms like Missiles and Satellites. These induction motors are mainly controlled using variable speed drive functions, i.e., thrusters, propellers, actuators. Though they are very robust, to harness the full technical benefits of induction motors, state-of-the-art drivers or controllers must be used to control such motors.

In model-based fault diagnosis, i.e., fault detection, fault isolation, parameter characteristics monitoring, also called Prognostic Health Monitoring (PHM) system, drivers or controllers must be used with induction motors of AUV systems to assess when to abort the mission due to malfunctioning hardware issues, as well as sonar detected obstacles to bring the AUV to the surface for recovery.

Induction motor drivers or controllers and model-based motor fault diagnostic systems require exact knowledge of information about all the major parameter characteristics of the motor, which are usually not available. These characteristics may be functions of joule heating, skin depth, motor linear or non-linear region of operation, and environmental operating conditions, to name a few.

Novel Model-based Fault Diagnostic Systems are the primary goal of this research. In order to validate the proposed novel approach, an induction motor-based system was selected to implement and test the unified approach developed and used in this doctoral dissertation. Such a unified approach comprises of the following items: a) Proposed Model-based Fault Diagnostic System theoretical foundation b) applicable Control Engineering Techniques required in Fault Diagnostic c) System Identification & Verification d) the Induction Motor-based Validation system. Each of these items is a vast field and each constitutes a field of advanced research. This dissertation addresses only their relevant aspects as applicable to advance the research carried out and presented in this dissertation. The driven motivation of this work, the Model-Based Fault Diagnosis through Induction Motors, is mainly to uphold the Department of Defense (DoD) strenuous effectiveness, safety, and performance system requirements [129].

Chapter 1: Introduction

Electrical Engineering Systems are increasingly becoming more complex; thus, achieving reliability, maintainability, availability, and affordability is increasingly challenging. Model-based Fault Diagnosis, Detection, and Isolation (FDI) with all its required accompanied fields is an essential tool in managing these complex challenges. Model-based FDI is widely accepted as a powerful technique in engineering system fault diagnosis and more recently, in Digital Twin (DT), Digital Transformation using Model-Based Systems Engineering (MBSE). Model-based FDI requires robust mathematical modeling of the system when using state space or input-output based models. It must account for noises, uncertainties, and disturbances ever present in the monitored system due to its design and operating environment.

The Defense Industry is on the verge of fully integrating Model-based FDI systems into vehicle control systems, robots, transport systems, power systems, manufacturing processes, and process control systems. Model-based FDI efficacy in detecting faults in these systems has been fully proven [140]. For nearly forty years, fault diagnosis in dynamic systems and processes has been a research topic in Control Engineering. These research studies have resulted in numerous proven varieties of model-based fault diagnosis approaches and techniques [78], [82-83], [85], [87], [90], [98], [104], [108]. In fact, the observer-based fault diagnosis has always been an active field of research in control theory and engineering. The essential difference between the state observer and diagnostic observer is that the diagnostic observer is an output observer while a state observer is often used for control purposes. The main role of the observer in a fault diagnosis system is to make the generated residual signals independent of the system input and initial conditions.

This doctoral dissertation focuses on unraveling the underlying design aspects and computational analysis of fault diagnostic observers for electrical systems in a coherent way to advance its implementation in all the related industries and applications. It is important to highlight that the dissertation research outcomes align with the DoD efforts under the on-going Digital Transformation initiatives. Model-based Systems Engineering is the core to the Digital Transformation era and is presently being used to develop the Digital Twin and Hardware-in-the-Loop efforts. The methodology proposed herein is a practical solution for complex multidisciplinary engineering applications. Model-based Systems Engineering, Digital Transformation, Digital Twin, and Hardware-in-the-Loop techniques lend themselves easily to Electrical engineering.

Since electrical engineering covers a vast area ranging from the fundamental circuit card assembly to low frequency electric/electronic systems to microwave, infrared, electro-optical systems, power station system control, motor control, among others, the layout of diagnosis systems in electrical engineering systems is a complex challenge that entails many individual steps. Complete understanding of each element of the design requires knowledge of theory from various subjects. Engineers working in these multidisciplinary areas need to have a broad understanding of these multiple elements and see how many of them are analogous due to their physics properties, or which display mathematical dualities. For instance, analogies between electrical, mechanical, thermal, and fluid systems are extremely helpful for engineers who can readily use these analogies in their design, analysis, and or implementation.

In system engineering, the coupling of different disciplines of science, engineering, technology, and applied mathematics make many Model-based fault diagnosis methods challenging to implement into real systems. Such implementation requires an engineer with broadband theory and practical knowledge in the relevant disciplines that may be beyond the scope of the engineer's ability if one has not studied or not yet worked in these disciplines. The Model-based fault diagnosis will monitor the system about an expected point of operation. This will allow the option of linear models as a simple alternative to the more complex nonlinear model systems.

The proposed method is a novel approach to model-based fault diagnosis. This research is composed of steps of exploiting linear machine identification processes connected with the model-based residual technology challenge. The novel techniques set forth herein are validated using an induction motor that is widely used as electrical drives to convert electricity to mechanical energy for commercial and Aerospace/Defense Industries.

This dissertation is most beneficial for students and engineers. It may serve as a basic understanding of Model-Based Fault Diagnosis Methods. Furthermore, it can lead to Digital Transformation via Model-Based Systems Engineering (MBSE), Digital Twin (DT), and Hardware in the Loop (HIL). For this reason, it is imperative to introduce the key parts of all areas involved to facilitate the understanding. The interested readers should refer to the reference section, which is quite exhaustive.

1.1 Objective, Motivation, State of the Art, and Background

There are many issues in complex engineering systems that keep eluding the model-free fault diagnosis method. Modular testing of assemblies is common for systems that have several levels of testing and assembly. These systems are susceptible to "ReTest OK" (RTOK) diagnostic conditions or "CanNot Duplicate" (CND) that considerably increases life cycle costs. It is vital to reduce the number of "RTOKs" and "CNDs." Focusing on fault accountability and test tolerances are the classical approaches used for that goal. To have an effect in reducing "RTOK" or "CND," these approaches require an exchange of data between design groups during the project development phase [142].

Fault accountability analysis ensures faults that are detectable by the higher levels of testing are also detectable by the lower levels of testing. For example, if a circuit card fault-open causes a system to fail, the fault must also be detectable by a Built-in-Test (BIT) system. A fault accountability analysis must verify that the same fault is detectable by offline testing of the circuit card. Failure to do so creates the diagnostic condition of RTOK or CND. Similarly, a CND or RTOK may occur if tolerances are not “tighter” at each lower level. The fault detection test report will record the test parameter regime of operation and all information about detected faults or failures for all modes of the BIT testing. This report can be designed to record the health information of the system, i.e., the fault parameter values that affect regular system operation. A Failure Mode and Effects Analysis (FMEA), verifying test tolerances, and creating a fault accountability matrix is essential in reducing CNDs.

An FMEA task is an excellent example of a hierarchical fault accountability analysis. Failure Mode and Effects Analysis (FMEA) is the mapping of faults to the functional operations of the system.

An FMEA will typically identify the effect of component level failures or functional system failures on system operation. Besides a hierarchical analysis, it is essential to identify the boundaries between the levels of indenture. The diagnostic point is one solution to the boundary problem. It can be any point in the system where an assembly joins another assembly. Assemblies may join higher or lower indentures within a vertical accountability axis. In a horizontal axis, assemblies may join other assemblies of the same indenture. That is, assemblies or items inside the same subsystem may be linked together and connected to the subsystem. The diagnostic point captures the physical connectivity as a single data element. The diagnostic point is a physical representation of the hardware. The measurements made at a diagnostic point are the result of test implementation at each level of assembly.

A solid-state power amplifier (SSPA) bias network in a communication device can illustrate a fault causing the DC offset to exceed acceptable tolerances if the amplifier serves as an input to a low pass and high pass filter. The fault will quickly propagate through the low pass filter. However, the high pass filter will block the effects of the DC offset fault. A fault detected by a function may not affect the operation or propagate through adjacent functions. The model-free fault diagnosis method will have difficulty identifying and isolating this noncontiguous fault (Filtered or Blocked) fault condition.

In RF-Microwave circuit design and analysis, there is a simple case of a noncontiguous fault, also called filtered or blocked fault that can elude the model-free fault diagnosis method even when the engineer is an RF-Microwave engineer and is aware of this condition as described.

These classical approaches, fault accountability, fault accountability matrix, and FMEA, can become tedious, cumbersome, and ineffective for certain types of faults. Therefore, advanced methods of fault supervision and fault diagnosis that can perform early detection of minor amplitude faults with abrupt or incipient time behavior are needed. These methods must be able to perform diagnosis of faults in all the monitored constituents (i.e., LRUs and CCAs, the actuator, and process components), which can perform fault detection in closed loops and monitor systems in transient states.

The model-based fault diagnosis method is a scheme that uses a mathematical model of the monitored systems. In contrast to the model-free fault diagnosis, the model-based fault diagnosis uses analytical redundancy, which compares system measurements with analytically computed values of their respective variable. Although developed for different purposes by control engineering researchers, all model-based fault diagnosis systems use the monitored system model. Algorithms are developed and implemented based on the system model for interpreting data that are collected and recorded during the system operation.

Even when using the most advanced model-based fault diagnosis method, fault detection becomes challenging to implement because of unknown inputs like disturbances, the effect of noise, and modeling errors. When unknown inputs like disturbances influence the system, it is impossible to detect faults if their influence is smaller than the influence of the disturbance in the residual signals. Recognizing this fact, model-based fault diagnosis with various strategies that decouple disturbances from residuals to handle noise is used. Among these strategies are parity space design observer based, Kalman filter, and parameter estimation techniques [20], [30], [32], [49], [54], [74]. It is worthy to note there exists a one-to-one mapping correspondence between parity space and observed-based residual generators.

It is certain that the fault detection scheme has succeeded if it is robust to unavoidable modeling uncertainty challenges to prevent any false alarm

[126]. Those uncertainties are external disturbances and system model mismatches. In this work, a novel robust linear fault diagnosis methodology using a linear analytic redundancy technique is developed that can easily be adjusted to handle nonlinear conditions. The methods in this research may also be used for Fault Prognosis i.e., System of Prognosis Health Management; and Fault Accommodation, which is the ability for the system to continue the mission under a fault condition. The detailed theoretical development, along with the simulation results, is presented in this dissertation.

Although many linear and nonlinear approaches are in use in many applications, robust FDI is still a challenge open to research. Guaranteeing that faults can be detected and isolated, mathematical models of the process under investigation are required, either in state-space or input-output forms. Residuals should then be processed to detect faults and rejecting any false alarms. However, in an actual fault detection situation, the straightforward application of model based FDI techniques can be difficult due to the dynamic model complexity.

The system analytical model is configured to carefully encompass all relevant details to the analysis and the deployment of the real system in its operating environments. This renders almost unfeasible the use of many cited FDI methods because of their intrinsic complexity. Thus, an alternative procedure for the practical application of FDI techniques is necessary in practical cases.

Two practical aspects of this dissertation are stressed. A thorough mathematical, and physical models may not be necessary if a dynamic model identification method for FDI that is successfully used. Therefore, if a thorough mathematical and physical models of the system is not readily available from the design team, a linear mathematical model of the state variables or the input-and-output links may be obtained through system identification techniques. The latter approach provides a reliable model of the system under investigation as well. Linear prototypes for designs of linear output estimators have been developed instead of using nonlinear models. This is considered essential to avoid the complexity that would otherwise be inevitable when nonlinear models are used. Since monitoring the operation and performance of the system to an expected point of operation is the issue, linear system methods are very valid. Because this operation will be seen and pointed out during development and the implementation of the fault diagnosis that is done using an induction motor. Induction motors are the workhorse commonly used in systems that require control speed. They have nonlinear characteristics that can present a challenge to system identification and fault diagnosis. However, as stated above, the operation and the performance of the Induction Motor for an expected point of operation can be monitored. The linear system methods for nonlinear systems have proven to be very valid [105]. The main challenges are to provide the technology for the system fault diagnosis. The approach being evaluated has an important implication on the use of on-line fault diagnostic tools.

Additionally, Model-Based FDI method has overlapping tasks with Digital Transformation using Model-Based Systems Engineering (MBSE) principles, Digital Twin (DT), Hardware in the Loop (HIL) that facilitate system effectiveness, system requirements and system affordability.

This dissertation sets forth an innovative hybrid method for fault diagnosis, with a wide field application that includes fault accommodation, anti-tampering, system security, health monitoring, and prognosis in dynamic engineering and structural systems. This dissertation has been developed and written by concentrating on both the theoretical and application aspects of the hybrid method, with a lesser focus on experiment and validation. The Induction Motor Driver and Controller section provides explanation of underlying motor physics modeling and where motor and control are discussed. The Induction Motor Driver and Controller used in this research is a Commercial Off the Shelf Item (COT) purchased from Texas Instrument (TI). The testing section where parameter estimation, experiments, and all related topics are discussed, and a Fault Diagnosis section where Fault Diagnosis physics and techniques are discussed and implemented in MATLAB. The way to connect the different parts stated above is laid out in detail so that only a minimum background is required to follow the denouement from introduction to conclusion.

The State of the Art of fault diagnosis means gauging fault diagnosis as it is mainly implemented in the industry today. Detecting fault and isolating failures or fault states of in an aerospace or military equipment are always part of a request for proposal (RFP) from the Department of Defense because fault detection and isolation is necessary for operation and maintenance of the equipment and the digital twin facilitator. Conventionally, classical fault diagnosis is performed in-situ or ex-situ. The in-situ fault diagnosis is plagued with challenges as stated above. The ex-situ fault diagnosis is plagued with challenges and is not dynamic detection.

Failure Modes, Effects, and Criticality Analysis (FMECA) is the classical approach in the design of fault diagnosis systems. It involves the following steps:

- System Built-in-Test (BIT) requirement
- Allocating the BIT requirement to the Subsystems

- Develop the Concept of the BIT Testability in Hardware and Software Design Requirement Documents
- Analyze the Test Requirement Documents to ensure where to place the BIT electrical circuits and instruct the Software Engineer how to write the test algorithm for the device to be monitored.
- Review the draft Design Requirement Documents with Designer, Electrical, Software Engineers.
- Develop the initial FMECA.
- Document and evaluating failure modes and safety critical issues.
- Perform the BIT analysis and prediction based on the FMECA.
- Develop the BIT based on new and existing tests.
- Proving the hardware and software Design Requirement Documents to the software engineer to implement the BIT algorithm and software.

The FMECA approach has many weaknesses. For instance, FMECA may not be able to identify some failure modes during the design phase, as well as other undetectable root causes. Thus, the engineer must combine the FMECA with failure mode distribution (FMD) historical data to attempt to capture most of them. Not only do the FMECA or the classical based fault diagnosis not up to the tasks, but in hardware and software Design Requirement Documents the BIT algorithms does not account for Gaussian and Laplacian Noises Effects on Parameters. Refer to dissertation IEEE published paper for more drawbacks on classical fault diagnosis [142].

In addition, the fault isolation of this approach requires subsequent Bayesian test networks, Bayesian and other statistical analyses of the BIT test results especially in the cases where many BIT tests are used when there is ambiguity or non-linearity condition, thus, the fault cannot be easily isolated to a particular component or module. The isolation result is forced to give three or more possible candidates that may potentially have failed.

The fault isolation problem is tedious and has high data dependency. It estimates dependencies and relations between the BIT tests and the item, with a set of training data. Then, the BIT software algorithm calculates the diagnoses when BIT test results make an isolation. The tedious computations may be performed on-line or offline depending on the system storage capacity.

FMEA is an inductive approach, whether it is a Systems FMEA (SFMEA), Design FMEA (DFMEA), Process FMEA (PFMEA) or a Failure Modes, Effects, and Criticality Analysis (FMECA) are used to identify defects at the device, component, module, line replaceable unit and subsystem level. Their effects are assessed on the next two or more assembly levels.

The industry has been searching a fault detection and isolation system that considers and solves some of the issues with the classical fault detection and isolation as well as serving as a springboard for the Digital Transformation era initiated by the DoD. The model-based fault diagnosis proposed in this dissertation serves that purpose. It provides a control engineering approach to detect fault aerospace of military equipment without the tedious ineffective steps in the classical approach described above. Model-based fault diagnosis methods uses the LRU, module or the equipment physical model of the system and the expected values in normal operating conditions. In addition, it makes use of observers which are a mathematical structure that combines sensor and or system output and system voltage or current input signals with its physical models. The observer provides optimum feedback signals. Its results have no dependence on the data source for fault isolation and provide good dynamic performance.

In the chapter that follows, a brief summary of the induction motor principle of operation is in order so that the link can be established how Electrical engineering lends itself easily to Model-based Systems Engineering, Digital Transformation, Digital Twin, and Hardware-in-the-Loop techniques.

Chapter 2: Induction Motor Brief Characteristics

2.1 Characteristic Understanding for FDI Implementation

The use of AC induction motors in dynamic system engineering is challenging because of their complex mathematical model, nonlinear characteristic during saturation, and the temperature-dependent nature of their electrical parameter oscillation.

These complexities render induction motors a challenge for motor controller, driver, and Model-Based Fault Diagnostic designers. Vector or scalar control, drive algorithms, and a microcontroller are needed to operate the type of motors. In this dissertation, a TI scalar controller controls the motor during the system identification experiments. The environmental operation of the motor is steady-state using simple voltage excitation, current-controlled, or speed-controlled techniques. The motor scalar control transient and the nonlinear characteristics are not ideal; however, its steady-state results are helpful for this purpose.

In this chapter, the induction motor model for transient and steady-state operating conditions is derived. The model parameter values are calculated using System Identification and Parameter Estimation methods.

The magnetic field's configuration in the motor rotor is created by induction. Thus, it is credited the name induction or asynchronous motor. The induction motor external excitation or power source is connected to the stator windings; however, the rotor itself in an induction motor is not externally excited, so there is no need for mechanical connectors like slip rings and brushes utilized in other motor types.

The key factor here is in the aluminum or copper squirrel-cage type induction motor. The three-phase squirrel cage induction motor is widely used, effortless, inexpensive, and has a reliable machine design. Its rotor has electrically shorted and non-accessible conductors with conducting bars installed in the rotor. The slots are short-circuited at both ends of the rotor structure and the bars are arranged parallel to the rotor axis and attached by a thick conducting ring at the ends, forming a short circuit.

The alternating currents (AC) in the stator windings generate a rotating magnetic field in the stator and rotor air gaps. The rotating rotor and the stator magnetic field generate or induce a voltage in the rotor windings. This induced voltage is done at different speeds. The rotating stator magnetic field voltage induces currents in the short-circuited rotor conducting elements, which produce the rotor magnetic field interacting with the stator magnetic field and generates the mechanical torque output work. These generated torque characteristics depend on the angle between the rotor and the stator magnetic fluxes. Therefore, the conditions to be met to keep the motor running at a constant speed in a steady state is that the rotor slip should be a function of the rotation speed of the stator field. Induction Motors are asynchronous machines. The physical definition is "In steady-state, the rotor rotation is not synchronized with the stator magnetic field". The induction motor operates at a frequency less than that of the synchronous speed, i.e., the stator magnetic field. This rotating speed difference of the rotor and the stator magnetic fields is termed the slip of the motor. As the rotor speed picks up toward the synchronous one, the magnetic field of the stator, the relative speed of the rotor, and the stator flux (or slip) decreases. The slip decrease causes the induced voltage in the stator to drop, resulting in a reduced torque.

2.2 Operating Principles to Facilitate FDI Implementation

The AC induction motor has a solid rotor and stator. The currents in the stator windings are generated by the exciting phase voltages, which drive the induction motor. These currents generate a rotating magnetic field or a stator field governed by the winding currents and the number of turns in the phase windings. The rotating magnetic field is analogous to the electric voltage and the magnetic flux to the electrical current. The stator magnetic field has a higher rotation speed than the rotor speed; thus, it generates an induction current in the rotor, which creates a rotor magnetic field. The fluxes produced by the stator and rotor magnetic fields give rise to the rotor torque. The stator, rotor technical explanation, and detail operating principles of induction motors are found in the 6th Edition publication of Electrical Machines, Drives and Power Systems, by Theodore Wildi [1].

A 3-pole pair stator with 5-turn winding each connect in series to 3-phase terminal configuration [a-a], [b-b], and [c-c] giving rise to three identical sets of windings. They are as follows: phase/line A to neutral N, phase/line B to neutral N, phase/line C to neutral N, that are mechanically spaced at 120° to each other [1].

When the stator is excited by a 3-phase supply and the rotor is short-circuited, the induced voltages in the rotor conductor bars generate a rotor current that interacts with the air gap field to create the torque. If the rotor is unlocked, it will rotate in the direction of the stator rotating field. According to Lens law, the relative speed between the stator rotating field and the rotor speed will decrease. At a steady state, the rotor speed is less than the synchronous speed of the stator rotating field in the air gap. During operation, the only time the induced voltage and current in the rotor conductors are zero; is when the rotor speed equals the synchronous speed of the stator rotating field in the air gap.

The rotational speed of the field depends upon the duration of one cycle. The duration of one cycle depends on the frequency of the source. For example, suppose the frequency is 60Hz (60 cycles/second) or 3,600 revolutions per minute in an induction motor with P poles, one cycle of variation of the current will cause the magneto-motive wave to rotate 2/P revolutions. The revolutions per minute "n" of the traveling wave in a P-pole motor is given below.

$$n = \frac{2*f*60}{P} = \frac{120f}{P} \qquad (1)$$

The physical meaning of slip is described as the difference between the rotor speed "n" and the stator synchronous speed "ns", of the rotating field.

$$S = \frac{n_s - n}{n_s} \text{ or } S = \frac{\omega_s - \omega_r}{\omega_s} \qquad (2)$$

That is, the rotor is slipping behind the rotating field by the slip rpm given below as:

$$S_{rpm} = n_s - n = S * n_s \qquad (3)$$

Let the frequency "f_2" be the rotor circuit slip frequency because this is the relative velocity between the non-rotor rotating field and the solid rotor field. Thus, the slips frequency is the frequency of the induced voltage and current in the rotor.

$$f_2 = \frac{P}{120}(n_s - n) = \frac{P}{120} S * n_s = Sf_1 \quad (4)$$

$$f_1 = \frac{P}{120} n_s \quad (5)$$

At the slip velocity, one may the induced voltage formula is given below.

$$E_{2s} = 4.44\, f_2 N_2 \phi_p K_{w2} = Sf_1 N_2 \phi_p K_{w2} = SE_2 \quad (6)$$

where:

E_2 is the induced voltage in the rotor circuit at a standstill that is a stator frequency f_1. "Kw" range [0.85 to 0.95] is the winding reduction factor 3-phase motors. "N" is considered to be the overall number of series turns for each phase with turns forming a concentrated full pitch winding.

ϕ_p is the air gap flux per pole

$\phi_p = 2B_{max} l_r$

"I" is the axial length of the stator.

"r" is the radius of the stator at the air gap.

"B" is the magnetic field.

The induced currents in the rotor generate a rotating field. Its speed rpm_n2 with respect to the rotor is:

$$n_2 = \frac{120 f_2}{P} = \frac{120 S f_1}{P} = Sn_s \quad (7)$$

The rotor is rotating at a speed n_rpm. The rotor field rotation is confined in the air gap. Its speed in "rpm" is formulated as follows:

$$n - n_2 = (1 - S)n_s + Sn_s = n_s \quad (8)$$

Both the stator and the rotor field rotations are synchronized and confined in the air gap. Therefore, the stator magnetic-field vector and the rotor magnetic-field vector are stationary. The stator magnetic-field and the rotor magnetic-field coupling produces the torque phenomenon.

In the above chapter, the induction motor, principle of operation, characteristics, and parameters of primary importance to the Model Based Fault Diagnosis were recapped as basis parameters that will either be measured during the experiments; or calculated by the MATLAB code or the observers.

In the following chapter, the various control schemes of the Induction Motor, the Texas Instrument (TI) High Voltage Digital Motor Control Kit used in the dissertation experiments are explained. Additionally, the induction motor and its datasheet showing the parameters whose specific shift will cause a fault, are introduced. Finally, the TI macro definitions are given so that the experiment can be reproduced easily.

Chapter 3: Control Scheme of Induction Machines

3.1 Hardware Overview for Scalar Control

The motor drive system used in this dissertation is connected to a single-phase AC power supply. Such system contains additional hardware to generate the required three-phase power to the AC motor including a Power Factor correction block. The Power Factor is formulated as the useful power divided by the total power. It has no effect in the operation of the motor and it serves to neutralize the magnetizing current. It is needed because the motor magnet or its magnetizing current as well as the voltage are out of phase. The additional Power control sub-systems are configured and identified on the Texas Instrument High Voltage Digital Motor Control and the Power Factor Correction tool kit TMDSHVMTRPFCKIT motor driver board. This driver board is composed of numerous macroblocks [110]. Figure 3.1 depicts the power factor correction connection in the motor drive configuration.

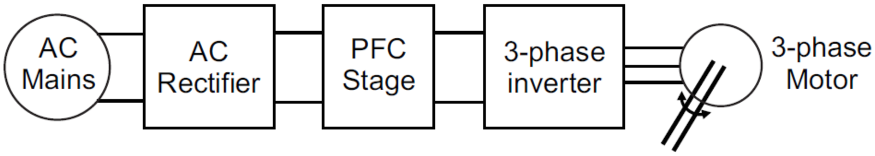


Figure 3.1 Motor Drive System Using Power Factor Correction [111]

Note. From “High Voltage Digital Motor Control Kit Quick Start Guide” by TI Quick Start Guide, page 4.
Copyright © 2010, Texas Instruments Incorporated. Reprinted with permission.

The tool kit TMDSHVMTRPFCKIT offers a convenient way to implement digital control methods in high voltage induction motors.

3.1.1 Features of the High Voltage Motor Control

The Texas Instrument (TI) High Voltage Digital Motor Control Kit, shown in Figure 3.2, contains [110-118] one High Voltage DMC board, called F28035 controlCARD, a 15 Volts DC Power Supply slot for the control CARD, an AC power Cord, a Banana Plug Cord, an USB Cable CCS4 CD and an USB Stick with QuickStart GUI and Guide.

Figure 3.2 TI TMDSHVMTRPFCKI, [117-118]

Note. From “High Voltage Digital Motor Control Kit Quick Start Guide” by TI Quick Start Guide, page 2.
Copyright © 2010, Texas Instruments Incorporated. Reprinted with permission.

3.1.2 System Overview of Scalar Control

The High Voltage Motor control contains a 3-Phase Inverter Stage capable of “sensor-less” and “sensor-red” Field-oriented Control (FOC) of high voltage ACI and Permanent Magnet Synchronous Motor (PMSM) motor and trapezoidal; and sinusoidal control of the high voltage brushless DC Motor

(BLDC). In addition, it contains a 350 Volts DC maximum input voltage source and one KW maximum load. In the bullets below, other board specifications are presented.

- Power Factor Correction stage rated for 750 Watts which takes rectified AC input with two different ranges, i.e., one from 85 volts to 132 Volts AC and a second one from 170 Volts to 250 Volts AC with a 400 Volts DC Maximum output voltage
- AC Rectifier stage rated for 750 Watts maximum power which accepts. 85 Volts to 132 Volts AC with 170 Volts to 250 Volts AC input
- Aux Power Supply Module (400Vto15V&5V module) generates 15V and 5V DC from rectified AC voltage or the PFC output (input Max voltage 400V, min voltage 90V).
- Isolated CAN, SCI & JTAG
- Four PWM DACs to observe the system variables on an oscilloscope.

The Motor picture is shown in Figure 3.3 below.

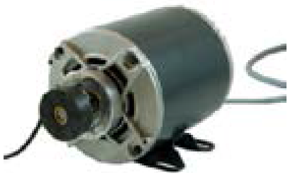


Figure 3.3 AC Induction Motor Model #HVACIMTR [118]

Note. From "High Voltage Digital Motor Control Kit Quick Start Guide" by TI Quick Start Guide, page 3.
Copyright © 2010, Texas Instruments Incorporated. Reprinted with permission.

Table 3.1 Marathon Induction Motor Parameter

Marathon 5K33GN2A ¼ Induction Motor Parameter	**Values**
Stator resistance (Rs)	11.05 ohms
Rotor resistance (Rr)	6.11 ohms
Mutual inductance (Lm)	0.293939 H
Stator inductance (Ls)	0.316423 H
Rotor inductance (Lr)	0.316423 H
Number of pole (p)	4
Power	¼ hp
Load inertia (J)	0.0006 2 kg-m^2
Viscous friction coefficient (F)	0.0008 (N.m)/(rad/s)

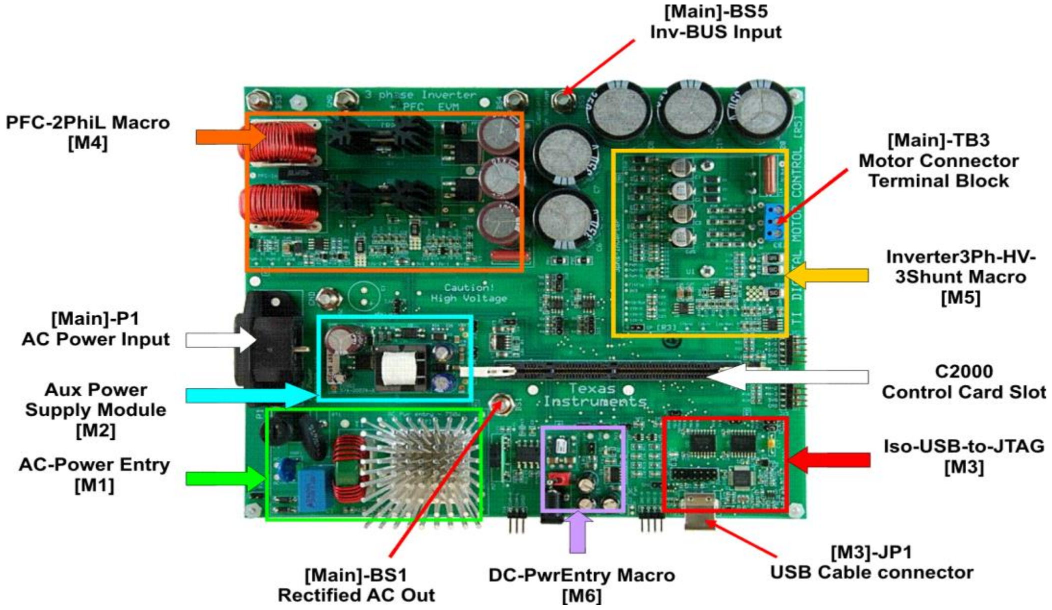


Figure 3.4 TI HVDMCMTRPFC Kit Board Macros [110 –115]

Note. From "High Voltage Digital Motor Control Kit Quick Start Guide" by TI Quick Start Guide, page 5.
Copyright © 2010, Texas Instruments Incorporated. Reprinted with permission.

The Texas Instrument (TI) list of all the macroblock names present on the board and a short description of their functions are described below and detailed in references [110-118].

- HVDMC: Main Board [Main] – Consists of control CARD socket, communications (isoCAN) block, Instrumentation Digital-to-analog converters (DAC's), Quadrature Encoder Pulse (QEP) and CAP connection, and routing of signals between the macros and to the control.
- M1: AC-Power Entry that rectifies the AC power that it receives house wall power supply source. This rectified voltage of M1 may be either served as the input of the PFC stage or to excite directly the DC bus for the inverter.
- M2: Aux Power Supply Module is able to receive up to 400V input and output powers of 5 Volts and 15 Volts DC. The rectified AC input may be directly connected to the Aux Power module. PFC stage Output can be used with appropriate jumper settings Aux Power module as well.
- M3: Iso-USB-to-JTAG Macro supplies the board with an isolated JTAG connecting the USB and the host. It can be used for communication for connection with the Graphic Unit Interface (GUI).
- M4: PFC-2PhiL Macro is a 2-phase interleaved Power Factor Correction stage that serves to improve drive efficiency.
- M5: 2Ph-HV-3 shunt Macro, an analytical 3-phase inverter for control of AC motors.

- M6: DC-Pwr Entry Macro is the DC power entry, used for the 15 Volts, 5 Volts, and 3.3 Volts for the board from 15 Volts DC power supply supplied with the TI kit.

In order to quickly find a component, i.e., a jumper, a wire, components are referred to their macro number in the brackets. For example, [M3]-J1 would refer to the jumper J1 located in the macro M3, and [Main]-J1 would refer to the J1 located Figures 3.5 and 3.6 below illustrate the jumper and connectors that need to be connected.

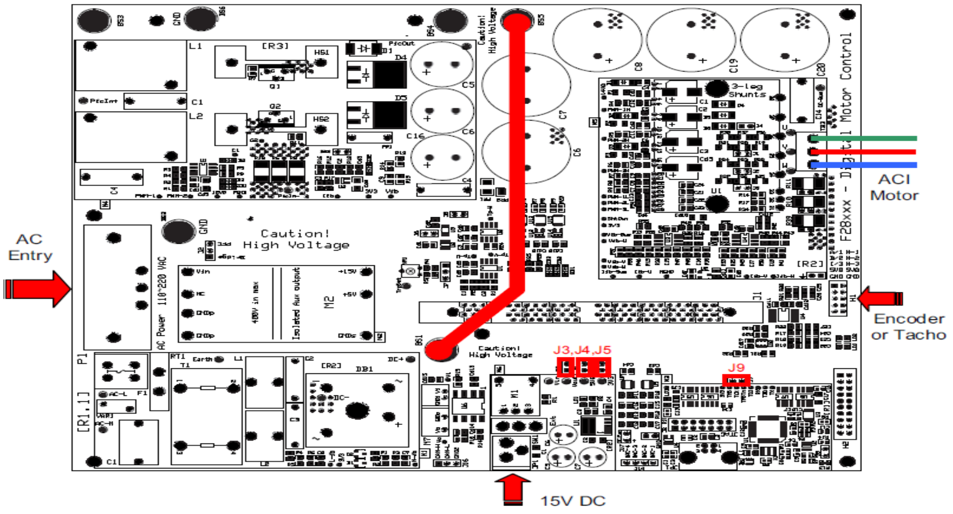


Figure 3.5 Generating DC Bus Power with AC Power [111]

Note. From "Sensored Field Oriented Control of 3-Phase Induction Motors" by TI, page 16.
Copyright © 2013, Texas Instruments Incorporated. Reprinted with permission.

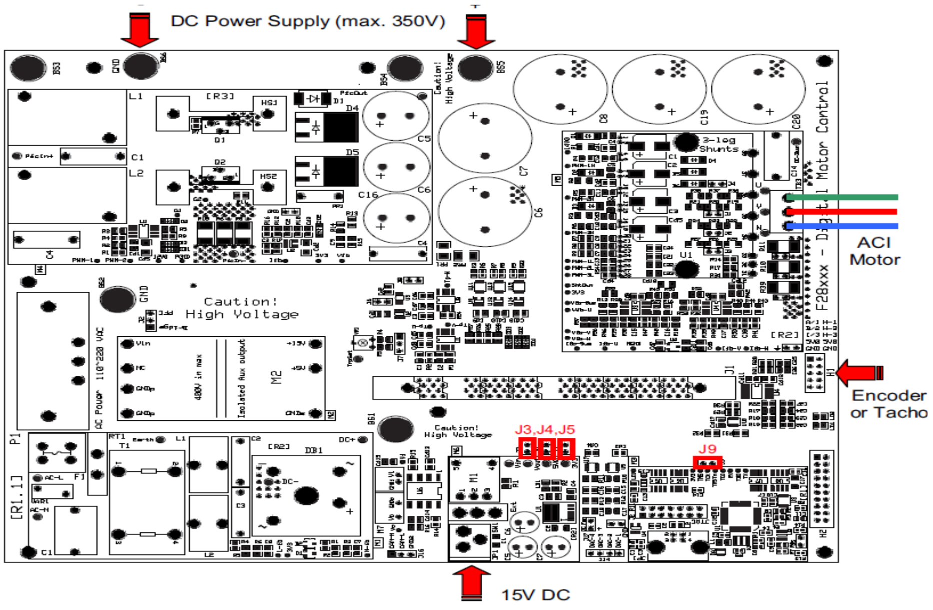


Figure 3.6 DC-Bus for the Inverter with DC Power Supply [111]

Note. From “Sensored Field Oriented Control of 3-Phase Induction Motors” by TI, page 17.
Copyright © 2013, Texas Instruments Incorporated. Reprinted with permission.

The real-time control framework "C" used to demonstrate the scalar control of induction motors for the TI TMS320F2803x-based controllers Code Composer Studio is shown in Table 3.2, which contains the list of the TI Macro names [110-118]. The overall system implementing the induction motor V/Hz drive implementation used in this dissertation is illustrated in Figure 3.7. The illustration of a) the inverter of the voltage source type to control the motor b) the TI TMS320F2803x to provide the pulse-width-modulated six-pulse excitation are shown.

Table 3.2 TI Macro Names

TI Macro Names	**Definition/Explanation**
Proportional Integral (PI) Micro means:	Proportional Integral (PI) Regulator
Ramp Controller (RC) Micro means:	Ramp Controller (slew rate limiter)
Voltage and Hertz (VHZ) PROFILE Micro means:	Voltage and Hertz (VHZ) Profile
Quadrature Encoder Pulse (QEP) / CAP Micro means:	Quadrature Encoder Pulse (QEP) and CAP Drives
SPEED_PR Micro means:	Speed Measurement, derived from sensor signal period
SPEED_FR Micro means:	Speed Measurement, derived from sensor signal frequency
SVGEN_MF Micro means:	Space Vector Pulse Width Modulator (PWM) (based on magnitude and frequency)
PWM / PWMDAC Micro means:	Vector Pulse Width Modulator (PWM) and PWMDAC Drives

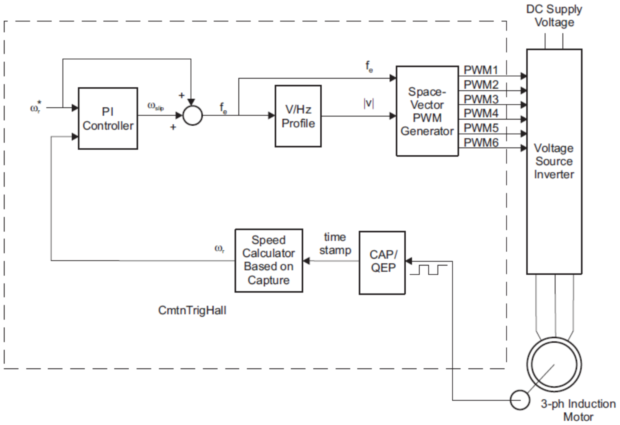


Figure 3.7 The Induction Motor V/Hz Drive Implementation [111]

Note. From “Scalar (V/f) Control of 3-Phase Induction Motors” by TI, page 9. Copyright © 2013, Texas Instruments Incorporated. Reprinted with permission.

The hardware of the control system contains various solid-state devices. Solid-state control provides smoother control and higher efficiency. Multiple types of converters used to control electric machines are listed below.

- AC Voltage Controller (AC to AC): An AC voltage controller converts a fixed voltage AC to a variable voltage alternating current used to control the speed of an induction machine (voltage-control-method) and smooth induction motor starting.
- Controlled Rectifier (AC to DC): A controlled rectifier converts a fixed voltage AC to a variable voltage Direct Current (DC) used primarily as parts of the DC motor speed control device. The output voltage and output power can be controlled by controlling the instants at which the semiconductor devices switch. Thus, controlled rectifiers are parts of the DC motor speed control device. Some controlled rectifiers can convert DC power to AC power, which is known as inversion. This inversion mode of operation is used for regenerative braking of motors
- Inverter: Inverters are static circuits that convert power from a DC source to AC power at a specified output voltage and frequency.
- Voltage Source Inverters (VSI): In the voltage source inverter, the input is a DC voltage supply, and the inverter converts the input de voltage into a square wave AC output voltage source.

- Current Source Inverters (CSI): In the current source inverter, the input is a DC current source, and the inverter converts the input DC current into a square-wave AC output current.
- Chopper (DC to DC): A chopper converts a fixed voltage DC to a variable voltage DC. It is used primarily to control the speed of de motors.
- Inverter (DC to AC): One role of an inverter is to convert a fixed Direct Current (DC) voltage to an either fixed or variable Alternating Current (AC) voltage with variable frequency. It has the ability that can be used to control AC motors.
- Cyclo Converter (AC to AC): One of the roles of a cyclo converter is to transform a fixed voltage and fixed frequency Alternating Current (AC) to a variable voltage and variable frequency Alternating Current (AC).
- Pulse Width-Modulation (PWM) Inverters: While induction motors speed is controlled, its voltage varies as a function of the frequency to keep the flux level constant. The output voltage of an inverter can be varied by changing the pulse width of each half-cycle of the inverter output voltage inverters with multiple pulses in each half-cycle of the inverter output voltages can reduce the harmonic content.

The illustration in Figure 3.8 is a Texas Instrument (TI) TMS320F2803x. It has a modulating six-pulse width. Each inverter leg is a 2-way switch. The voltage source inverter makes it possible to connect the 3-phase induction motor to the positive or negative side of the voltage supply inverter. The six power switching devices in the inverter or the six inverter states yield active vectors ($\vec{U}_1$, $\vec{U}_2$, $\vec{U}_3$, $\vec{U}_4$, $\vec{U}_5$, $\vec{U}_6$) for the terminal voltage.

The configuration for Voltage/Hertz (V/Hz) Drive in Induction Motor is given in Figure 3.8 below [112.

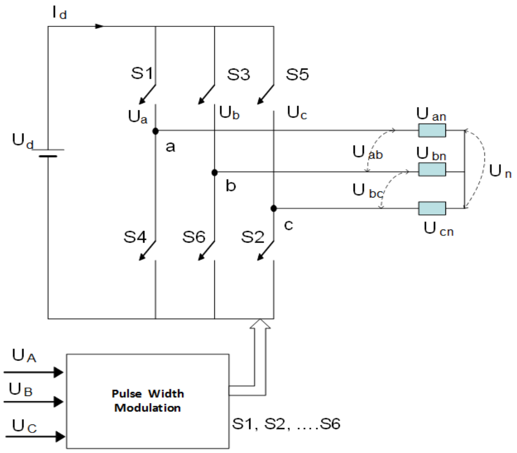


Figure 3.8 Voltage Source Inverter Feeding Induction Machine [110]

Note. From “Scalar (V/f) Control of 3-Phase Induction Motors” by TI, page 3. Copyright © 2013, Texas Instruments Incorporated. Reprinted with permission.

The Voltage/Hertz (V/Hz) control system is gradually built up, so the final system can be confidently operated. [110–118]

The Incremental System Build Block contains the following development levels:

- Step1, Level 1 checks the independent target modules V/Hz and SVGENMF.
- Step2, Level 2 checks the PWM excitation, A/D conversion units.
- Step3, Level 3 checks PI Regulator and Closed Loop functionality.

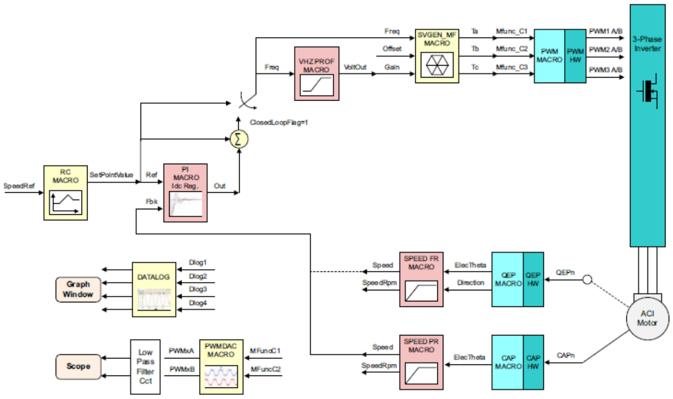


Figure 3.9 V/Hz Control System Final Build Block Diagram [118]

Note. From “Scalar (V/f) Control of 3-Phase Induction Motors” by TI, page 23.
Copyright © 2013, Texas Instruments Incorporated. Reprinted with permission.

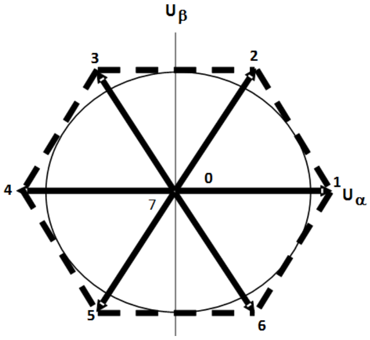


Figure 3.10 Space Vector PWM Used in TI V/Hz Drive Control [118]

Note. From “Scalar (V/f) Control of 3-Phase Induction Motors” by TI, page 6.
Copyright © 2013, Texas Instruments Incorporated. Reprinted with permission

The frequency of the input voltage can be varied by the duration of each active vector. The amplitude can be varied by varying either the DC voltage Ud or by using PWM, the two switching states for which the machine is short-circuited. Based on the inverter 8 switching states, the output voltages (Ua, Ub, Uc), the phase voltages (Uan, Ubn, Ucn), and the line voltages (Uab, Ubc, Uca) can easily be calculated.

3.1.3 Constant Current Controlled

The induction machine is controlled and operated at constant current by providing a current loop and frequency around the AC-DC converter of the control apparatus of the machine. Since the motor rms current I_1 is the rms stator or terminal current out of the inverter is proportional to the PFC stage DC link current I or Id, the motor current can be kept at the same value as the control current Ic.

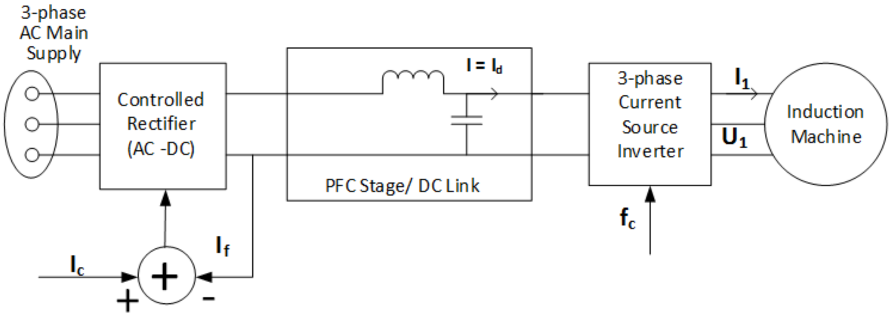


Figure 3.11 Detailed Block Diagram for Motor Drive Control System [118]

In constant-current controlled operation of induction machines, the rotor frequency at which maximum torque is developed is much smaller.

$$f_{2\_break\_down} = \frac{R_2'}{2\pi(L_m+L_2')} \qquad (9)$$

Table 3.3 Torque and Current Expressions Summary

Condition	Per Phase Value	All Phase Value
Constant Slip Frequency	$T_e = \frac{4\pi p L_m^2}{R_2'} I_1^2 \frac{L_2}{1+\left\lvert\frac{2\pi(L_m+L_2')f_2}{R_2'}\right\rvert^2}$	$T_e = \frac{\psi_s^2}{R_r}\frac{\omega_r(1-\sigma)^2}{1+(\sigma\tau_r\omega_r)^2}$
Constant Rotor Frequency	$T_e = \frac{p}{4\pi R_2'}\left(\frac{V_1}{f_1}\right)^2 \frac{f_2}{1+\left(\frac{2\pi f_2 L_2'}{R_2'}\right)^2}$	
Constant Flux	$T_{max} = \frac{p}{4\pi R_2'}\left(\frac{V_1}{f_1}\right)^2 \frac{1}{4\pi L_2'}$	$T_{pull} = \frac{\psi_s^2(1-\sigma)^2}{2L_m\sigma}$
Constant Current	$T_{max} = \frac{\pi p L_m^2}{R_2'} I_1^2 \frac{R_2'}{2\pi(L_m+L_2')}$	$T_e = \tilde{I}_s^2 L_m \frac{\tau_r\omega_r}{(\tau_r\omega_r)^2+1}$
	$T_{max} = \frac{pL_m^2}{2(L_m+L_2')} I_1^2$	$T_{max} = \frac{\omega_r}{R_r}\psi_r^2$
Current	I_1	$I_s^S = I_t^S = \tilde{I}_s e^{j\omega_s t}$
		$\tilde{I}_s = \frac{1}{L_m}\sqrt{(\tau_r\omega_r)^2+1}\tilde{\psi}_r$
Flux	$\psi_{p\_phase} = \frac{V_1}{f_1}$	$\psi_r^R = \frac{L_m}{\tau_r j\omega_r + 1}\tilde{I}_s e^{j\omega_r t}$
		$\psi_s^R = \ \psi_r^R L_\sigma \tilde{I}_s e^{j\omega_r t}$

where: ω_s is stator field revolution frequency or the supply frequency in radiant per second (rad/s)

ω_m is the mechanical speed of the motor

ω_r is the difference between stator field revolution frequency and the mechanical speed of the motor ω_m. Thus, ω_r is the slip frequency. The frequency of the magnetic field arises in the rotor windings. The 1-pole pair relative slip is formulated in equation 10 as follows:

$$s = \frac{\omega_s - \omega_m}{\omega_s} = \frac{\omega_r}{\omega_s} \tag{10}$$

The motor is rotating a synchronous speed implies that the slip is zero.

$$\omega_s - \omega_m = 0 \tag{11}$$

The slip rpm (revolution per minute) is formulated in equation 12 as follows:

$$s\omega_s = (\omega_s - \omega_m) \tag{12}$$

The revolution per minute of a travelling of a field wave in a p-pole machine for a frequency f cycles per second the current in the rotor winding is given by:

$$\omega_m = \frac{2}{p} f * 60 = \frac{120f}{p} \tag{12a}$$

The slip frequency is the rotor circuit frequency in cycles per second.

$$f_2 = \frac{p}{120}(\omega_s - \omega_m) = \frac{p}{120}\omega_r = \frac{p}{120}s\omega_s \tag{13}$$

$$f_2 = sf_1 \tag{14}$$

$$f_1 = \frac{p}{120}\omega_s \tag{15}$$

If the supply frequency is in radiant per second (rad/s), the stator rotating field speed and the rotor rotating speed are:

$$\omega_s = \frac{f_1}{p}\text{(rad/s)} \tag{16}$$

$$\omega_m = (1 - s)\omega_s = (1 - s)\frac{f_1}{p}\text{(rad/s)} \tag{17}$$

The frequency of the induced voltage and current in the rotor corresponds to the slip rpm. This is the rotating speed between the rotating field and the rotor winding.

The induced current in the rotor winding produces a rotating field as well, known as the rotor speed (rpm) with respect to the rotor as:

$$\omega_2 = \frac{120}{p} f_2 = \frac{120}{p} s f_1 = s\omega_s \qquad (18)$$

The stator magnetic field and the rotor magnetic field interaction align both fields; this alignment force generates the machine torque. The rotor is rotating at a speed ω_m (rpm). The induced rotor field rotates in the air gap at the speed:

$$\omega_m + \omega_2 = (1 - \mathrm{s})\omega_s + s\omega_s \text{ , thus, } \omega_m + \omega_2 = \omega_s \,\mathrm{rpm} \qquad (19)$$

Thus, the stator field and the induced rotor field rotate in the air gap at the same synchronous speed ω_s.

Chapter 4: Induction Motor System Testing

4.1 Induction Motor Offline Test

Induction Motor tests can be performed during normal operation or by designed experimental test identification. The operation conditions and input signals can be tailored in the experimental tests or the off-line identification tests to estimate one or more motor parameters. The no-load and locked rotor tests are part of off-line identification experiments and are the common tests used to identify the electrical machine parameters. Other experiment tests can be performed when the machine is at a standstill and the inverter of the drive is used to generate the signals required for parameter estimations. Some of the equivalent circuit parameters, i.e., R_c, X_m, R_1, X_1, X_2, and R_2, may be determined by just measuring the resistance of the stator winding.

Three tests are required to evaluate the impedances of an induction motor:

- DC measurements of stator DC resistance
- AC measurements with No-load test
- AC measurements with Blocked rotor

These tests were implemented using the TI (TMDSHVMTRPFCKI) board controller where the three-line terminals of the stator winding are accessible.

Consider the winding connected in a wye-configuration, or in a Delta-configuration for calculation purposes. It is less difficult to address a wye-configuration circuit. Therefore, the DC resistance among any two terminals of the winding is twice the resistance of one stator phase.

4.1.1 DC Measurements of Stator DC Resistance

The stator winding resistance is preferably measured before any other test is performed since the resistance is dependent on the temperature. The experiment is conducted at room temperature, which means that the resistance must be adjusted during steady state. The following equation [1] can be used:

$$R_T = R_{T_0}\left[\frac{T+k}{T_0+k}\right] \tag{20}$$

where:

k: is the characteristic constant for the winding material

T: is the limiting temperature according to the insulation classification

T_0: is the temperature at which the experiment was conducted

R_{T_0}: is the measured armature resistance at room temperature

$\hat{R}_s$: is the measured stator resistance

$$R_{T_0} = \hat{R}_s \tag{21}$$

4.1.2 Test Procedure

The step-by-step test procedure is described below:

- Excite the two series connected phases with the variable DC supply. A relatively low current should be used so that the resulting I2R loss will not cause a significant change in temperature during the time of application. The resulting current in the stator must be at its rated value.
- Read the value of the resulting current using an ampere-meter.
- Derive the mean value of the stator winding resistance.

A current limiting load box is used to limit the current through the motor. Its resistance is set to bring the current to its rated value. Then, the potential difference across the induction motor poles is measured and given by:

$$R_s = \frac{1}{2}\frac{V_{DC}}{I_{DC}} \tag{22}$$

The IEEE recommended Induction Motor equivalent circuit for a wye-configured load shown in Figure 4.1 where $U_T = U_1$ is the DC variable voltage excitation, $X_1 = X_m = 0$. The rotor impedance is not in connection with the stator.

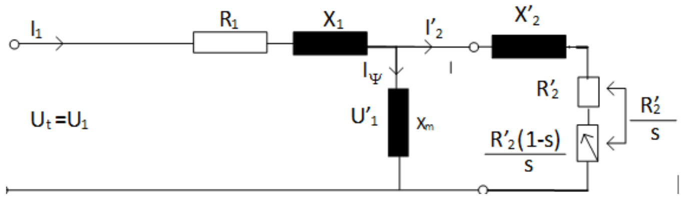


Figure 4.1 Induction Motor IEEE Recommended Equivalent Circuit

U_1: is the line to neutral AC difference of potential

R_1: is the per phase stator resistance

R_2: is the per phase rotor resistance in reference to the stator

X_2: is the per phase rotor leaky or spurious reactance in reference to or referred to the stator

X_m: is the shunt reactance

s: is the slip

The shunt reactance forms a path for the magnetizing element of the current in the stator. This current gives rise to the revolving field in the motor. In this case, note that core losses and rotational losses are not accounted for in the equivalent circuit. Small negligible errors are usually due to core losses. The generated values from the equivalent circuit are used to calculate mechanical power and torque. Rotational losses are subtracted from the generated values to obtain actual power and torque output values. Another assumption is that R_2 is constant. The resistance identified as R_2 is a function of the rotor frequency current and temperature. It is necessary to use the correct value of R_2. Note - The frequency of the rotor is equal to the slip times the number of pole divided by 120 as follows: f_rotor = Ns*P/120. If the slip is equal to zero, the frequency of the rotor is equal to zero. When the operating speed is zero the frequency of the rotor current is equal to frequency of the stator current. The slip is the field speed minus the rotor speed.

4.1.3 Blocked Rotor Test and No-Load Test

Blocked Rotor or Locked Rotor Test [46] is like the short-circuit test on a transformer. Leakage impedance information can be extracted from the test. During the test the rotor shaft is locked while a 3-phase voltage is applied to the stator terminals. During the blocked rotor test, rotor current and frequency values should not exceed those under normal operating conditions.

If the performance characteristics of the induction motor in the low slip region are required, the motor blocked-rotor test should be performed at a reduced voltage, reduced frequency, and rated current. The low slip is the stable region, with high slip, the torque increases to its maximum: this is called the unstable or breakdown region. Especially for double-cage, deep-bar rotors, the reduced frequency corresponds to the lower values of slip. It is worth noting that the effective rotor resistance and leakage inductance at the reduced frequency test differ from their values at the rated frequency. Thus, the blocked-rotor test is to be done at 25 percent of the rated frequency according to the IEEE recommendation. Since the motor used in this dissertation is rated less than 20-hp, the effects of frequency are negligible, therefore, the locked rotor test may be performed at the rated frequency.

The blocked rotor test can be viewed as a short-circuited test on a transformer. As mentioned earlier, in this test, the motor rotor shaft is clamped so that it cannot turn, that is, ω_m = 0 rad/sec. The motor terminals are excited with a 3-phase supply. The rotor plays the role of the shorted secondary winding of a transformer operating at the supply voltage frequency. Because an induction motor is a basic transformer with its primary winding as the motor stator and its shorted secondary winding as the motor rotor, the flux due to the rotor current creates a counter electromagnetic force that opposes the stator flux. This flux phenomenon increases the stator current like the increase in the transformer secondary current that causes an increase in its primary winding.

Using the Induction Motor IEEE Recommended Equivalent Circuit, at locked rotor condition slip in unity, the magnetizing reactance is shunted by the low impedance circuit branch. Since the magnetizing reactance is much greater than the rotor circuit branch impedance, it can be omitted in the locked rotor equivalent circuit. Using the equivalent circuit below, the parameters resulting from the blocked rotor tests are determined.

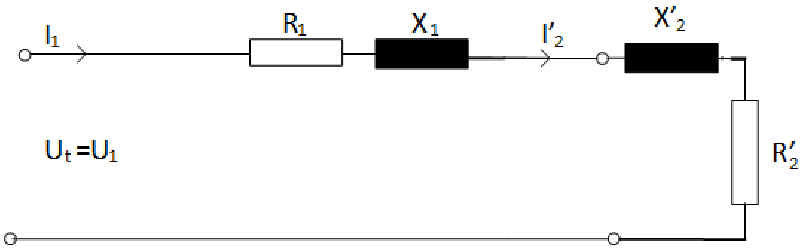


Figure 4.2 Equivalent Circuit for Locked Rotor Test Based on IEEE Equivalent Circuit (per Phase)

The locked rotor resistance is:

$$R_{LK} = \frac{P_{LK}}{3I_1^2} \tag{23}$$

$$R_2' = R_{LK} - R_1 \tag{24}$$

The locked rotor Impedance is:

$$Z_{NL} = \frac{V_1}{I_1} \tag{26}$$

The locked rotor reactance is:

$$X_{LK} = \sqrt{Z_{LK}^2 - R_{LK}^2} \tag{27}$$

$$X_{LK} = X_1 + X_2' \tag{28}$$

$$X_1 = X_2' \tag{29}$$

Since at no-load:

$$X_{NL} = X_1 + X_m \tag{30}$$

the magnetizing reactance, X_m, is formulated in equation 31.

$$X_m = X_{NL} - X_1 \tag{31}$$

Using the equivalent circuit below, a more accurate calculation of the rotor equivalent resistance "R'2" is found.

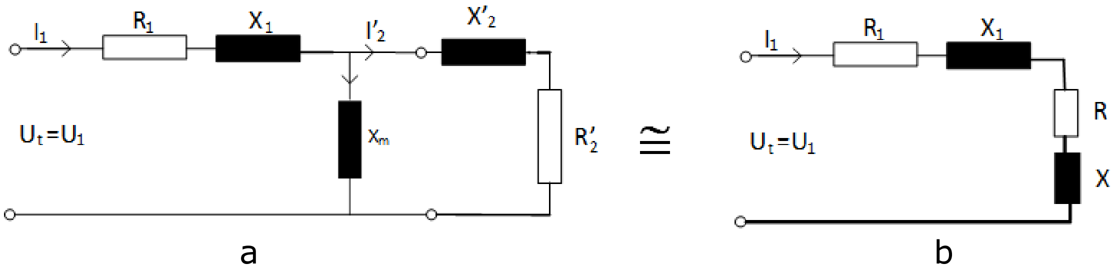


Figure 4.3 Locked Rotor Test Circuit per Phase [46]

The locked resistance is the sum of R₁ and an equivalent resistance R in the equivalent circuit shown in Figure 4.3, b side.

$$R_{LK} = R_1 + R \qquad (32)$$

where from the circuit shown in Figure 4.3:

$$R = \frac{X_m^2}{\left(R_2'\right)^2+\left(X_2'+X_m\right)^2} R_2' \qquad (33)$$

$$R_2' = \left(\frac{X_2'+X_m}{X_m}\right)^2 R \qquad (34)$$

$$R = R_{LK} - R_1 \qquad (35)$$

$$U_{th} = \frac{X_m}{\left(R_1^2+(X_1+X_m)^2\right)^{0.5}} U_1 \text{ ; } R_{th} = \left(\frac{X_m}{X_1+X_m}\right)^2 R_1 = K_{th}^2 U_1 \quad \text{and } X_{th} = X_1 \qquad (36)$$

Using a Thevenin equivalent voltage, resistance, and reactance from a Short Circuit Transformer Equivalent Circuit, the Locked Rotor Test values are obtained.

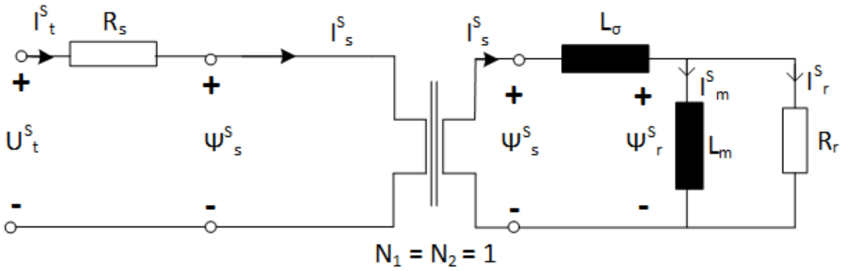


Figure 4.4 Locked Rotor Circuit as a Short-Circuited Transformer

The input impedance at stator frequency "ω₀" is as formulated in equation 37.

$$Z_{input} = \frac{U_t^S(\omega_0)}{I_t^S(\omega_0)} \approx R_s + j\omega_0 L_\sigma + \frac{R_r L_m j\omega_0}{R_r + L_m j\omega_0} \qquad (37)$$

As stated for the per phase approach, since the input impedance of the motor with a locked rotor is very low, during a locked rotor test, the voltage amplitude must be reduced compared to the rated amplitude to prevent large stator currents. The magnetizing current is given by equation 38 as:

$$I_m^S \approx 0 \quad (38)$$

The input impedance at stator frequency becomes

$$Z_{input} = \frac{U_t^S(\omega_0)}{I_t^S(\omega_0)} \approx R_s + j\omega_0 L_\sigma + R_r = R_t + j\omega_0 L_\sigma \quad (39)$$

From the stator measured voltage and current, the leakage inductance and the total stator and rotor resistance can be found. The stator DC test resistance can be used to find the rotor resistance [46]. If iron losses, skin-effect, and saturation of the leakage inductance are insignificant, the rotor resistance and the leakage inductance can be calculated from the locked rotor test measurement.

In the No-Load Test Plan [46], the induction motor no-load experimental test is analog to transformer open circuit tests. This test is performed by applying a balanced 3-phase voltages to the stator windings at the rated frequency of the motor. The rotor has no mechanical load attached. The motor loss in the core, windage, and the friction account for very little power loss at no-load. The total rotational loss at no-load is the total rotational loss at the rated voltage and frequency under load.

Using the Induction Motor IEEE Recommended Equivalent Circuit, Figure 4.1 at no-load condition, (R'$_2$/s) is high, causing the magnetizing reactance "X$_m$" to be shunted by the rotor highly resistive circuit branch. Thus, the reactance of this parallel combination can be safely estimated to the magnetizing reactance "X$_m$". Hence, at the stator terminals, the total reactance measured at no-load is: X$_{NL}$= X$_1$ + X$_m$. From the no-load Induction Motor IEEE Recommended Equivalent Circuit given in Figure 4.5 below, the parameters resulting from the no-load rotor tests can be determined [46].

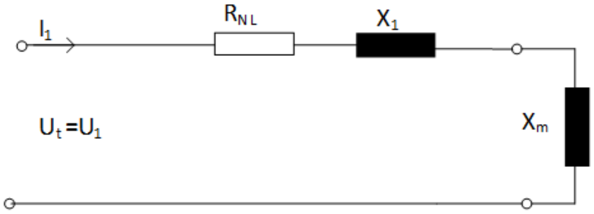


Figure 4.5 Circuit for No Load Test per Phase

The no-load model parameters are given in the equations below

$R_{NL} = \frac{P_{NL}}{3I_1^2}$:   is the no-load resistance (40)

$X_{NL} = \sqrt{Z_{NL}^2 - R_{NL}^2}$: is the no-load reactance (41)

$X_{NL} = X_1 + X_m$ (42)

The rotational loss power, the input per phase voltage and the No-Load Impedance are given respectively by equations 43, 44 and 45 below.

$P_{Rot\_Loss} = P_{NL} - I_1^2 R_1$ (43)

$V_1 = \frac{V_{Line}}{\sqrt{3}}$ (44)

$Z_{NL} = \frac{V_1}{I_1}$ (45)

The measured DC resistance per stator phase is obtained in equation 46 below, modeled from an Open Circuit Transformer Equivalent Circuit for No-Load Rotor Test.

$R_1 = R_s$ (46)

As stated for the per phase approach, at no-load, the induction motor produces no torque. Thus, the rotor current is zero. Assuming iron-losses are negligible, the terminal impedance at no-load is given below:

$Z_{input} = \frac{U_t^S(\omega_0)}{I_t^S(\omega_0)} = R_s + j\omega_0(L_\sigma + L_m)$ (47)

Once the stator resistance and the leakage inductance are obtained during the locked rotor test, the rotor flux can be computed using the formula below.

$\psi_t^S(\omega_0) = \frac{1}{j\omega_0}\left(U_t^S(\omega_0) - \hat{R}_s I_t^S(\omega_0)\right) - \hat{L}_\sigma I_t^S(\omega_0)$ (48)

$I_t^S = I_m^S$ (49)

Using the reconstructed rotor flux and the measured stator current, which is at no-load equal to the magnetizing current, the main inductance can be calculated. The magnetizing curve of the induction motor is obtained by taking many measurements at no-load at different stator voltages. That is, the magnetizing curve is the rotor flux amplitude as a function of stator voltage versus the magnetizing current amplitude. Thus, the main inductance can be calculated as a function of the magnetizing current.

4.1.4 Draw Backs in DC, Locked Rotor and No-load Tests

In identifying no-load test drawbacks, the operation is assumed to be at synchronous speed. Thus, the mechanical load is to be adjusted to compensate for the mechanical losses. In applications, the load can only be mechanically uncoupled from the motor. Losses are still present in the motor at no-load. Thus, they must be accounted for through additional measurements at various input voltages and rated supply frequency. The results are then extrapolated to low and zero voltages to account for all loss compensation [71], [101].

In identifying the locked rotor test drawbacks, locking the rotor is not always mechanically feasible due to large torques production. Ignoring the magnetizing path in this test, locked rotor test flux low level, higher slip frequency, and significant rotor winding skin depth effect; are sources of inaccuracies. However, all these inaccuracies in no-load test and Locked rotor test experiments can be compensated.

As stated, the traditional tests present some drawbacks. For instance, the locked-rotor test requires the rotor to be externally locked, and the no-load test needs the rotor to rotate freely at almost the synchronous speed. Performing these tests can be impractical or even impossible in case the machine has already been coupled to its mechanical load. Thus, it is best to replace these traditional tests with standstill tests.

4.1.5 Standstill Tests

Induction motor electrical parameters can be obtained from the stator voltages and currents at a standstill, i.e., the drive inverter produces the signals required for the parameter estimation. Time varying models are no longer applied; thus, simpler identification algorithms can be used easily. The physical parameters can be calculated easily using two output models between the measured voltages and currents. At different levels of DC input signal, the motor is excited various flux levels this help identify the motor inductances with saturation.

4.1.6 Model for Standstill Tests

Using the voltage as the input signal and the current as of the output signal, the model of an induction motor at a standstill is described by the following transfer given as the terminal admittance function expressed as [31], [36]:

$$Y_t\left(q,\theta_p\right) = \frac{I_t^S(s)}{U_t^S(s)} = \frac{1}{Z_p\left(s,\theta_p\right)} \tag{50}$$

$$Y_t\left(q,\theta_p\right) = \frac{R_r+L_mS}{L_\sigma L_mS^2+(R_sL_m+R_rL_\sigma+R_rL_m)S+R_s+,R_r} \tag{51}$$

The transfer function parameter vector representing the motor admittance:

$$\theta_p = [R_r, R_s, L_\sigma\ , L_m]^T \tag{52}$$

With n measurements sample, the physical motor parameters can be calculated using the indirect approach discussed in this dissertation. Note that: $U_t^S(k)$ is Stator voltage, and $U_t^S(k)$ is stator current.

The indirect approach will be:

- Using the physical model, the selected continuous to discrete-time transformation a discrete time model is arrived at
- Using the measurements an optimal model derived from the discrete time model
- Using the identified discrete time parameters, the physical parameter is derived

The physical parameters must be uniquely defined. At a standstill, the model is linear time invariant. Thus, the indirect approach requires less computation than the direct method. Thus, the differential equation describing the relation between the directly sampled stator voltage and stator current with Tustin approximation is:

$$U(k) = U_t^S(kT_s) + U_t^S(kT_s - T_s) \tag{53}$$

$$U(k) = (1 + q^{-1}) + U_t^S(kT_s) \tag{54}$$

$$y(k) = I_t^S(kT_s) \tag{55}$$

The discrete time model structure after the application of the Tustin approximation is:

$$G_d\left(q, \theta_q\right) = \frac{y(k)}{u(k)} = \frac{b_0 - b_1 q^{-1}}{1 + f_1 q^{-1} + f_2 q^{-2}} \tag{56}$$

The discrete time parameter vector is modeled in equation 57 below.

$$\theta = [b_0, b_1, f_1\ , f_2]^T \tag{57}$$

The discrete time transfer function admittance between the stator current and voltage is obtained from the transfer function given in this section.

$$.Y_t(q, \theta_d) = \frac{I_t^S(k)}{U_t^S(k)} = (1 + q^{-1}) G_d(q, \theta_d) \tag{58}$$

The discrete time parameters, the corresponding physical parameters are derived uniquely. The relation between the discrete time parameters and the physical parameters is given in the equation below, and the inverse Jacobian can be found using the equation below as well.

$$\begin{bmatrix} \hat{R}_r \\ \hat{R}_r \\ \hat{L}_\sigma \\ \hat{L}_m \end{bmatrix} = \begin{bmatrix} \left(\frac{1 - f_2}{b_0 - b_1} - \frac{1}{2} \frac{(b_0 + b_1)(f_2 - f_1 + 1)}{(b_0 - b_1)^2} - \frac{1}{2} \frac{f_2 + f_2 + 1}{b_0 - b_1} \right) \\ \left(\frac{1}{2} \frac{f_2 + f_2 + 1}{b_0 + b_1} \right) \\ \left(\frac{T_S}{4} \frac{f_2 - f_2 + 1}{b_0 - b_1} \right) \\ \left(-\frac{T_S}{4} \frac{(b_0 - b_1)(f_2 + f_2 + 1)}{(b_0 + b_1)^2} + \frac{T_S}{2} \frac{-f_2 + 1}{b_0 + b_1} - \frac{T_S}{4} \frac{(f_2 - f_1 + 1)}{b_0 - b_1} \right) \end{bmatrix} \tag{59}$$

The Jacobian matrix will help transform discrete-time parameters in the physical parameters coordinate system. The inverse Jacobian is used to relate the covariance of the discrete-time parameter with the physical parameter covariance. Thus, discrete-time and physical parameters variability may be assessed.

4.1.7 The Induction Motor Parameters

Testing in stator coordinates can identify all machine parameters from measurements of the terminal voltage, terminal current, mechanical position, or mechanical speed. Identification in stator coordinates uses the direct method because the indirect method is not applicable when all machine parameters must be calculated simultaneously.

Given: U_t^S: The terminal voltage; I_t^S: The terminal current; ω_m: The mechanical speed; θ_m: The mechanical Position

The Continuous time prediction model for the electromagnetic equations is given below.

$$M(\theta,\ \omega_m): \begin{cases} x' = A(\theta,\ \omega_m)x + B(\theta)u \\ \hat{y} = C(\theta)x + D(\theta)u \end{cases} \tag{60}$$

where: $u \in C = U_t^S$: The measured terminal voltage; $\hat{y} \in C = I_t^S$:The measured stator current

$$x = \begin{bmatrix} \hat{\psi}_s^S \\ \hat{\psi}_r^S \end{bmatrix} \tag{61}$$

$$A(\theta, \omega_m) = \begin{bmatrix} -\frac{R_s}{L_\sigma} & +\frac{R_s}{L_\sigma} \\ \frac{R_r}{L_\sigma} & -\frac{R_r}{L_\sigma} - \frac{R_r}{L_m} + j\omega_m \end{bmatrix} \tag{62}$$

$$\theta = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } C(\theta) = \begin{bmatrix} \frac{1}{L_\sigma} & -\frac{1}{L_\sigma} \end{bmatrix} \tag{63}$$

The parameter vector is

$$\theta = \begin{bmatrix} R_r \\ R_s \\ L_m \\ L_\sigma \end{bmatrix} \tag{64}$$

The indirect identification method is not suited to uniquely estimate the physical parameters as the transfer function below revealed. These parameters cannot be identified with the indirect identification process as the speed cannot be implemented in the discrete-time model. As the transfer function below describes the relation between the terminal voltage and current at constant mechanical speed is constant. The transfer function contains four parameters to be calculated and one known parameter, which is the mechanical speed.

$$Y_t^S(S) = \frac{I_t^S(S)}{U_t^S(S)} = \frac{\frac{1}{L_\sigma}S + \frac{R_r}{L_\sigma L_m} - j\omega_m \frac{1}{L_\sigma}}{S^2 + \left(\frac{R_r}{L_m} + \frac{R_r}{L_\sigma} + \frac{R_S}{L_\sigma} - j\omega_m\right)S + \frac{R_S R_r}{L_m L_\sigma} - j\omega_m \frac{R_S}{L_\sigma}} \tag{65}$$

The related black box transfer function contains eight parameters, of which four are complex parameters with four real parts and four imaginary parts. For example, for the direct identification method, let set the following parameters as:

$$A_\sigma = \frac{1}{L_\sigma} \text{ and } A_m = \frac{1}{L_m} \tag{66}$$

The parameter vector is

$$\theta = \begin{bmatrix} R_r \\ R_s \\ A_m \\ A_\sigma \end{bmatrix} \tag{67}$$

The system matrices become

$$A(\theta, \omega_m) = \begin{bmatrix} -R_s A_\sigma & R_s A_\sigma \\ R_r A_\sigma & -R_r A_\sigma - R_r A_m + j\omega_m \end{bmatrix} \tag{68}$$

$$C(\theta) = [A_\sigma \quad -A_\sigma] \tag{68a}$$

The gradient model calculation is illustrated in equations 69 to 72 below.

$$z_i'(\theta) = A(\theta, \omega_m) z_l(\theta) + A_i(\theta, \omega_m) x(\theta) \tag{69}$$

$$\psi_l(\theta) = C(\theta) z_l(\theta) + C_l(\theta) x(\theta) \tag{70}$$

$$A_l(\theta, \omega_m) = \frac{\delta A(\theta, \omega_m)}{\delta \theta_l} \tag{71}$$

$$C_l(\theta) = \frac{\delta C(\theta)}{\delta \theta_l} \tag{72}$$

The gradient matrices for the continuous prediction models are as follows:

1. $\frac{\delta A(\theta,\omega_m)}{\delta R_r} = \begin{bmatrix} 0 & 0 \\ A_\sigma & -A_\sigma - A_m \end{bmatrix}$ and $\frac{\delta C(\theta)}{\delta R_r} = [0 \quad 0]$ (73)

2. $\frac{\delta A(\theta,\omega_m)}{\delta R_s} = \begin{bmatrix} -A_\sigma & A_\sigma \\ 0 & 0 \end{bmatrix}$ and $\frac{\delta C(\theta)}{\delta R_s} = [0 \quad 0]$ (74)

3. $\frac{\delta A(\theta,\omega_m)}{\delta A_m} = \begin{bmatrix} 0 & 0 \\ 0 & -R_r \end{bmatrix}$ and $\frac{\delta C(\theta)}{\delta A_m} = [0 \quad 0]$ (75)

4. $\frac{\delta A(\theta,\omega_m)}{\delta A_\sigma} = \begin{bmatrix} -R_s & R_s \\ R_r & -R_r \end{bmatrix}$ and $\frac{\delta C(\theta)}{\delta A_\sigma} = [1 \quad -1]$ (76)

5. $\frac{\delta A(\theta,\omega_m)}{\delta L_m} = \begin{bmatrix} 0 & 0 \\ 0 & R_r A_m^2 \end{bmatrix}$ and $\frac{\delta C(\theta)}{\delta L_m} = [0 \quad 0]$ (77)

6. $\frac{\delta A(\theta,\omega_m)}{\delta L_\sigma} = \begin{bmatrix} R_s A_\sigma^2 & -R_s A_\sigma^2 \\ -R_r A_\sigma^2 & R_r A_\sigma^2 \end{bmatrix}$ and $\frac{\delta C(\theta)}{\delta L_\sigma} = [-A_\sigma^2 \quad A_\sigma^2]$ (78)

Equation set (1 to 6 above).

The related fluxes are

$$\psi_{L_m}(\theta) = \frac{\delta \hat{y}(\theta)}{\delta L_m} = \frac{\delta \hat{y}(\theta)}{\delta A_m}\frac{\delta A_m}{\delta L_m} = -A_m^2 \psi_{A_m}(\theta) \quad (79)$$

$$\psi_{L_\sigma}(\theta) = \frac{\delta \hat{y}(\theta)}{\delta L_\sigma} = \frac{\delta \hat{y}(\theta)}{\delta A_\sigma}\frac{\delta A_\sigma}{\delta L_\sigma} = -A_\sigma^2 \psi_{A_\sigma}(\theta) \quad (80)$$

Only one set of state evolution equations need calculating for all the sensitivity functions, thus rendering the computation easy. The sensitivity functions are computed and explained in [91].

where: $E = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{R_s R_r} \end{bmatrix}$ (81)

The parameter vector can be estimated using an iterative estimate or the recursive Gauss Newton algorithm.

$U_{t\sigma}$, $U_{t\beta}$, $\hat{I}_{t\alpha}$ and $\hat{I}_{t\beta}$ must be known in both axes.

The prediction model based on the mechanical position is as follows:

$$M(\theta,\, \theta_m)\colon \begin{cases} \hat{x} = A(\theta,\, \theta_m)x + Bu \\ \quad \hat{y} = \ C(\theta,\, \theta_m)x \end{cases} \quad (82)$$

where: $u \in C = U_t^S$: The measured terminal voltage; $\hat{y} \in C = I_t^S$: The measured stator current, $\hat{x}$ and $\hat{y}$ are the predicted or estimated value of x and y.

$$x = \begin{bmatrix} \hat{\psi}_s^S \\ \hat{\psi}_r^R \end{bmatrix} \tag{83}$$

$$A(\theta,\ \theta_m) = \begin{bmatrix} -R_s A_\sigma & R_s A_\sigma e^{j\theta_m} \\ A_\sigma e^{-j\theta_m} & -R_r A_\sigma - R_r A_m \end{bmatrix} \tag{84}$$

$$\theta = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and} C(\theta,\ \theta_m) = [A_\sigma \quad -A_\sigma e^{j\theta_m}] \tag{85}$$

The gradient matrices for the continuous prediction model are as follows:

$$\frac{\delta A(\theta, \theta_m)}{\delta R_r} = \begin{bmatrix} 0 & 0 \\ A_\sigma e^{-j\theta_m} & -A_\sigma - A_m \end{bmatrix} \quad \text{and} \quad \frac{\delta C(\theta, \theta_m)}{\delta R_r} = [0 \quad 0]$$

$$\frac{\delta A(\theta, \theta_m)}{\delta R_s} = \begin{bmatrix} -A_\sigma & A_\sigma e^{j\theta_m} \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \frac{\delta C(\theta, \theta_m)}{\delta R_s} = [0 \quad 0]$$

$$\frac{\delta A(\theta, \theta_m)}{\delta A_m} = \begin{bmatrix} 0 & 0 \\ 0 & -R_r \end{bmatrix} \quad \text{and} \quad \frac{\delta C(\theta, \theta_m)}{\delta A_m} = [0 \quad 0]$$

$$\frac{\delta A(\theta, \theta_m)}{\delta A_\sigma} = \begin{bmatrix} -R_s & R_s e^{j\theta_m} \\ R_r e^{-j\theta_m} & -R_r \end{bmatrix} \text{and} \frac{\delta C(\theta, \theta_m)}{\delta A_\sigma} = [1 \quad -e^{-j\theta_m}] \tag{86}$$

They assess the sensitivity of certain motor parameters as a function of frequency. Thus, the parameter that is most sensitive can be predicted more accurately. For example, the gradients such as the stator resistance, the rotor resistance, the main inductance, and the leakage inductance can be graphed for a certain mechanical speed. The shape of the rotor sensitivity may be an electrical signature indicating whether the rotor resistance is observable for a specific range of frequencies at no-load.

In this chapter, the induction motor test plan and procedure are laid out and explained as used in the defense industry. Various tests, including the no-load, locked rotor, DC measurement tests, and other tests performed at standstill tests are also explained. Additionally, the ways standard tests identify the induction motor parameters used in the Fault Diagnosis are explained so that any engineer familiar with the topic can reproduce the work. This chapter also explains how the equivalent circuit parameters, i.e., Rc, Xm, R1, X1, X2, and R2, may be determined by just measuring the resistance of the stator winding. These tests and experiments are needed in conjunction with the IEEE Equivalent Circuit Analysis to develop the parameters to use in the MATLAB codes used in this dissertation. In the following chapter, the design of the Model-Based Fault Diagnosis is laid out and explained.

Chapter 5: Robust Fault Detection and Isolation

5.1 The Robust Observer Approach to Fault Detection and Isolation

For the fault diagnosis design and implementation, the induction motors can be assumed to be an uncertain Lipschitz nonlinear system, defined in section 5.4 and references [130] and [139] or a linear system. Thus, detecting and isolating faults in a nonlinear system like an induction motor may be analyzed as working with an uncertain Lipschitz nonlinear system. In this fault detection and isolation method, the system to be diagnosed must be transformed into two subsystems. The first subsystem contains both motor faults and motor uncertainties; the second subsystem only contains motor faults. A Luenberger observer [106] is used to detect faults in the motor subsystem_2. Due to their inherent robustness, a bank of sliding mode observers [109] is used to isolate faults in subsystem_1 and subsystem_2.

Recall the Lipschitz condition about the state x means that if the function $\phi(x,t)\colon [a,b]$ is Lipschitz then, there exists a constant $\mathcal{L}$ called Lipschitz constant such that $|\phi(x,t) - \phi(x',t)| \leq \mathcal{L}\,|x - x'| \ \ \forall\, x, x' \ \in [a,b]$; $\mathcal{L}$ must be the smallest constant meeting the Lipschitz condition [130], [139].

The effects of system uncertainties to the residual of the subsystem are decoupled by performing a coordinate transformation of the original system model, enabling us to perform early detection while suppressing false alarms.

For Hard to Detect Faults (HDF), i.e., incipient and drift faults, the approach to use is fault estimation which assesses the magnitude, the shape, and the duration of the fault. Fault estimation plays an important role in a system that requires fault accommodation or fault tolerant systems. As in the case of fault detection, a linear coordinate transformation of the system state vector into two parts will be used such that the induction motor faults are represented in the second state vector. The induction motor system with uncertainties can be assumed to be unstructured, and thus, cannot be decoupled entirely from faults but minimized.

It is easier to implement model-based fault diagnosis using linear system models. The scheme to handle non-linear system behaviors has been to approximate the non-linear systems by linearizing the system models at an operating point and applying robust fault diagnostic techniques to the manipulated model. However, this approach does not always work since linear and non-linear models usually have considerable mismatches, the operating range becomes wider. Therefore, the need arises to design the fault diagnosis which can handle system non-linear behavior.

5.2 On-line Fault Detection and Isolation

In on-line fault detection and isolation, the operations of detecting and isolating faults are performed while the system is running. Thus, all the information needed by the model-based system is available during system operation. The schematic below illustrates the dynamic between the fault diagnosis and the control loop.

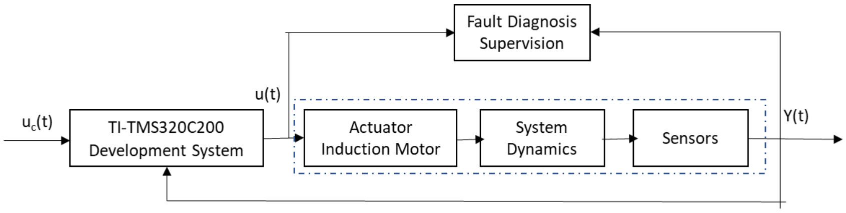


Figure 5.1 On-Line Fault Detection and Isolation

The system can be breaking up into actuators, system dynamics, and sensors as illustrated in the figure below.

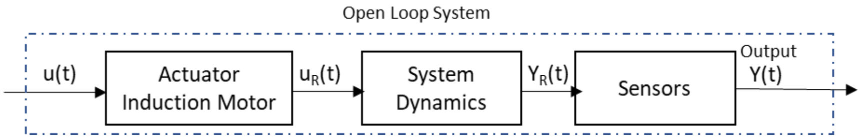


Figure 5.2 Open-Loop System without On-Line Fault Supervision

$U_R(t)$: the output of the Induction Motor becomes the input of the System Dynamic block

$Y_R(t)$: the output of the system Dynamic block becomes the input of the sensors, assuming that sensors are uses.

The System Dynamics state space model below is illustrated in the figure below.

$$\dot{x}(t) = Ax(t) + Bu_i(t) + f_c(t) \qquad (87)$$

$$y(t) = Cx(t) + Du_i(t) \qquad (88)$$

$f_c(t)$: a component fault, i.e., the motor fault vector

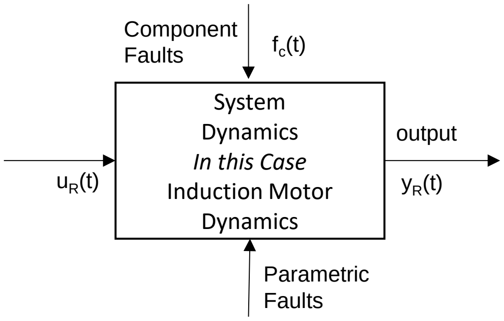


Figure 5.3 The System Dynamics with Faults

where:

A, B, C, and D are known system matrices

x $\in$ R^n: The state vector

$u_R \in$ R^r: The input vector to the actuator

$y_R \in$ R^m: The real system output vector

When sensors are used their output state space is illustrated in Figure 5.4. Equation 89 are used if the sensors are incorporated into the system. In that case the output of the of sensor block Y(t) in Figure 5.4 contains the induction motor fault and the sensor fault, and Y_R(t) the output of the motor becomes the sensor input.

$$y(t) = y_R(t) + f_s(t) \qquad (89)$$

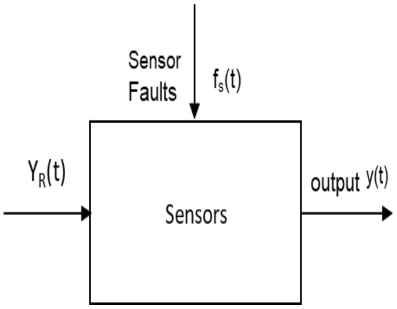


Figure 5.4 The Sensors with Faults

u_R is the actuator response to an actuator command u(t) as illustrated in Figure 5.5.

$$u_R(t) = u(t) + f_a(t) \tag{90}$$

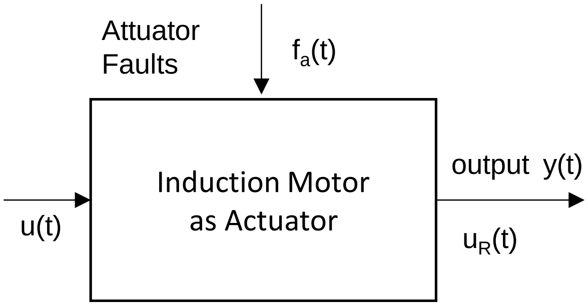


Figure 5.5 The Induction Motor Actuator with Inputs

The entire system is the system that comprises sensor faults, actuator, and induction motor faults. The entire system state vector model is written in equations 91 and 92.

$$\dot{x}(t) = Ax(t) + Bu(t) + Bf_a(t) + f_s(t) \tag{91}$$

$$y(t) = Cx(t) + Du(t)+Df_a(t) + f_s(t) \tag{92}$$

However, in this dissertation, the interest is only in the actuator, i.e., the induction motor. Thus, it is illustrated in Figure 5.6.

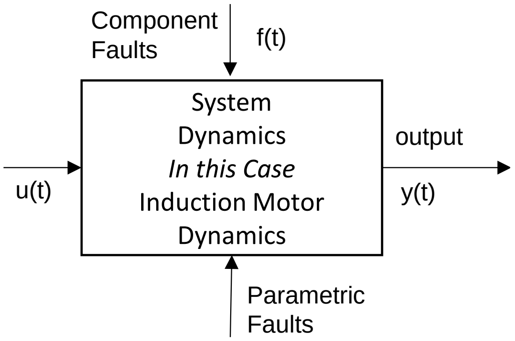


Figure 5.6 The Induction Motor as the Sole Part of the System

5.3 Model Based Fault Detection Implementation in System

To be able to perform the tasks of fault diagnosis at its optimum capability, a fault diagnosis scheme must detect and isolate the fault, including incipient faults, abrupt faults in sensors, actuators (induction motor), and components. However, in this dissertation, the fault diagnosis must only detect and isolate faults in the induction motor, including incipient and abrupt faults. Therefore, the guideline to develop the fault diagnosis system in this dissertation is that the fault diagnosis system must be able to handle:

- Noise in the system
- Multiple faults
- Disturbances i.e., additive uncertainty
- Modeling errors i.e., multiplicative uncertainty
- Nonlinearities
- Detection delays

- FDI algorithm design complexity and implementation

More details on these types of faults are given in references [132-136].

The fault diagnostic will be performed using an observer approach where the fault isolation is performed using a set structured residual obtained using the observer scheme. This observer fault isolation method is very suitable to handle:

- Incipient faults, which are very fast. [135]
- Detecting and isolating faults in induction motors in sensors
- Able of detecting and isolating faults in parameters
- Able of detecting and isolating multiple faults
- Able of handling nonlinearity with Non-linear observers
- Able of using robustness

If the noise statistics in the system is unknown, a filter may be added to the residual scheme, based on assumed characteristics of the noise. A Kalman filter may be added to reduce false and missed detection [137].

5.4 Linear and Non-linear Observer in Fault Diagnosis

In fault diagnostic systems, when a failure occurs during operation, the system is no longer operating at a fixed point. Therefore, it is better to use non-linear observers to represent non-linear systems than multiple linear observers. Here Sliding Mode Observers (SMO) are used as much as possible so they can deal with uncertainties, i.e., system disturbances. The main advantage of SMO fault diagnosis is that, regardless of model uncertainties and system disturbances, the errors for the output estimation between the system and the observer can be minimized.

The original system model is transformed into two subsystems, as explained in this document in sections 5.1, 5.7 and 5.9. Once a fault is detected, it must be isolated to a particular system component or function subsystem. One single observer may be enough to isolate faults if they have a different effect on residual space. Multiple or a bank of observers handle fault isolation better, especially during non-linear system behavior [133].

There are two observer approaches for fault isolation. In the dedicated observer fault isolation, a certain number of observers generate the same number of residuals, and each residual is sensitive to the corresponding number of faults only. In the generalized observer approach, certain observers are employed to generate an equal number of residuals. In this case, the corresponding residual to the observer is not sensitive to the related observer fault but is sensitive to all faults that may occur in the system. Thus, our approach to fault detection is to use a dedicated observer for the second subsystem and a bank of robust observers, i.e., sliding mode observers for each fault in the system; and the induction motor, to estimate and generate the output vector.

One of the typical state vector nonlinear models is adapted below in equations 93 and 94 [105]. Faults and model discrepancy and nonlinearity are represented in the motor model as follows:

$\dot{x}(t) = Ax(t) + Bu(t) + Df_{IM}(t) + \phi(x,t) + E\Delta\psi(x,t)$ (93)

$y(t) = Cx(t)$ (94)

Given that:

X $\in$ R^{m} and A $\in$ R^{nxn} : System Matrix (SM)

U$\in$ R^{m} and B $\in$ R^{nxm} : System Matrix (SM)

Y$\in$ R^{p} and C $\in$ R^{pxn} : Measurement Input Matrix (MIM)

D $\in$ R^{nxm} = B : Fault Distribution Matrix (FDM)

f_{IM} $\in R^{m}$: Vector of actuator/motor fault

E $\in$ R^{nxr} : Known Constant Rectangular Matrices (KCRM)

$\phi(x,t)$: $\in$ R^{nx1}: Known as the nonlinear Lipschitz term

$\Delta\psi(x,t)\in$ R^{r}: Unknown nonlinear term for model uncertainties, system disturbances, and model discrepancies.

The assumption on matrices C and E are as follows: C and E are considered full rank, that is, if C is a $p \times n$ matrix the maximum possible rank of matrix C is the smaller of p and n. The matrix C rank is the number of independent rows or columns in the matrix. Matrix property of full rank is explained in [137], [138]. The Term $\phi(x,t)$ is a known nonlinear term Lipschitz about the state x uniformly.

5.5 Fault Detectability Assumption

Assuming that the matrix pair (A, C) is detectable, then there is a matrix L that can be defined as follows:

$L \in R^{nxp}$ / A – LC is stable and for any Q > 0, the Lyapunov equation as given in equations 95 and 96 below [130].

$$(A - LC)^T P + P*(A - LC) = - Q \quad (95)$$

whose solution is P > 0

where P, Q $\in R^{nxn}$ and given as:

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2^T & P_3 \end{bmatrix}, Q = \begin{bmatrix} Q_1 & Q_2 \\ Q_2^T & Q_3 \end{bmatrix} \quad (96)$$

where if matrices P > 0, and Q > 0, then $P_1 > 0$, $Q_1 > 0$, $P_3 > 0$, and $Q_3 > 0$

It is worth noting that: $P_1 \,\&\, Q_1 \in \mathrm{R}^{(n-p)X(n-p)}$, $P_3 \,\&\, Q_3 \in \mathrm{R}^{pXp}$

5.6 Rank Assumption

This assumption requires a greater degree of freedom; that is, the number of required measurements must be larger than all the number of unknown inputs, i.e., the model uncertainties and the input disturbances, and the number of induction motor faults. Given the following stipulation:

- rank(CE) = rank(E)
- rank(*C*[*D E*]) = rank(*D*)+rank(*E*), Thus, the rank (*D E*) ≤ p is given.
- Given a complex number "s" with positive real part, then the rank becomes:

$$rank \begin{bmatrix} sI - A & E \\ C & 0 \end{bmatrix} = n + rank\ (E) \tag{97}$$

$$rank\ (E) = rank \begin{bmatrix} sI - A & E \\ C & 0 \end{bmatrix} - n \tag{98}$$

The rationale for the rank and detectability assumptions is in reference publication: Robust Model Based Fault Diagnosis for Dynamic Systems [140].

The function $\Delta\psi(x,t)$ is the unknown bounded structured uncertainty.

$\|\Delta\psi(x,t)\| \leq \mathcal{L}_{\psi}(x,t)$ or $\|\Delta\psi(x,t)\| \leq \mathcal{L}_{\psi}$ (99)

where $\mathcal{L}_{\psi}(x,t)$ is known and Lipschitz about *x* uniformly, and $\mathcal{L}_{\psi}$ is a constant.

For the fault vector it can be said:

$\|f_{IM}\| \leq \rho_a(t)$ or $\|f_{IM}\| \leq \rho_a$ (100)

where $\rho_a(t)$ is a known and Lipschitz about *x* uniformly and ρ_a is a constant. Depending on whether the induction motor fault is time-varying or a constant, ρ_a is the predefined threshold to be compared with the motor residual.

5.7 Output Matrix Assumption

Let assume that the output matrix C has full row rank and is as follows:

$$C = [0 \quad I_p] \tag{101}$$

C is a structured matrix because its matrix vector multiplication is simpler than the general matrix [145].

If C is not structured, the state and output transformation matrices must be such that given T_c:

$$CT_c^{-1} = [0 \quad I_p] \tag{102}$$

If the matrices (A, E and D) have the following structures in equation 102a,

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}; \;\; E \; = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \;\; \text{and D} = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \tag{102a}$$

where $A_1 \in \mathrm{R}^{(n-p)\times(n-p)}$; $E_1 \in \mathrm{R}^{(n-p)\times r}$; $D_1 \in \mathrm{R}^{(n-p)\times q}$; $A_4 \in \mathrm{R}^{p\times p}$; $E_2 \in \mathrm{R}^{p\times r}$; $D_2 \in \mathrm{R}^{p\times q}$

the original system typical state vector nonlinear models can be rewritten as:

$$\dot{x}_1(t) = A_1x_1 + A_2x_2 + B_1u(t) + D_1f_{IM} \; + \phi_1(x,t) + E_1\Delta\psi(x,t) \tag{103}$$

$$\dot{x}_2(t) = A_3x_1 + A_4x_2 + B_2u(t) + D_2f_{IM} \; + \phi_2(x,t) + E_2\Delta\psi(x,t) \tag{104}$$

$$y(t) = x_2 \tag{105}$$

$x = \text{column}\,[x_1,\; x_2\,]$; $\;x_1 \;\&\; \phi_1(x,t) \in \mathrm{R}^{(n-p)}$; $\quad \phi(x,t) \;\&\; \phi_2(x,t) \in \mathrm{R}^p$

5.8 The Sliding Mode Observer Fault Diagnosis

The sliding mode observer is designed by first performing a linear change of coordinates. This coordinate change is needed to structure the uncertainty and fault distribution matrices. Since E and C are nonsingular matrices; rank(CE) =rank(E), the state and output nonsingular transformation matrices are as follows:

$$\mathfrak{z} = Tx = \begin{bmatrix} \mathfrak{z}_1 \\ \mathfrak{z}_2 \end{bmatrix} \; ; \;\; \mathscr{y} = Sy = \begin{bmatrix} \mathscr{y}_1 \\ \mathscr{y}_2 \end{bmatrix} \tag{106}$$

where: $\mathfrak{z}_1 \in \mathrm{R}^r$; $\mathfrak{z}_2 \in \mathrm{R}^{n-\,r}$; $\;\mathscr{y}_1 \in \mathrm{R}^r$; $\mathscr{y}_2 \in \mathrm{R}^{p-r}$; $S \in \mathrm{R}^{p\times p}$; $T \in \mathrm{R}^{n\times n}$

and $\begin{bmatrix} I_{n-p} & P_1^{-1}P_2 \\ 0 & I_p \end{bmatrix}$, the transformation of T and S are satisfied as:

$$\begin{cases} T = \begin{bmatrix} I_1 & -D_1D_2^{-1} \\ 0 & I_2 \end{bmatrix} \\ T^{-1} = \begin{bmatrix} I_1 & -D_1D_2^{-1} \\ 0 & I_2 \end{bmatrix} \end{cases} \text{and} \begin{cases} x = T^{-1}\begin{bmatrix} \mathfrak{z}_1 \\ \mathfrak{z}_2 \end{bmatrix} \\ y = S^{-1}\begin{bmatrix} \mathscr{y}_1 \\ \mathscr{y}_2 \end{bmatrix} \end{cases}$$

Given a matrix A, a nonsingular Transformation Matrix T may exist such that one can transform the original matrix A as follows: TAT^{-1}. Furthermore, for the given matrix A, a column vector "v" can be found such that the transformation matrix can be calculated using T = [A^{n-n} v, $A^{n-(n-1)}$ v,…. $A^{n-(n-3)}$ v] or T = [A^0 v, A v,…. A^{n-1} v], where T is non-singular.

The new coordinate system matrices can be structured from the state input and output transformation matrices as given in equation 107.

$$TAT^{-1} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix};\ TB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix};\ TE = \begin{bmatrix} E_1 \\ 0 \end{bmatrix};\ SCT^{-1} = \begin{bmatrix} C_1 & 0 \\ 0 & C_4 \end{bmatrix};\ T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}; S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \quad (107)$$

The coordinate transformation must be performed to decouple the faults, incipient, and the unknown disturbance or signals. Then, the original system using the linear change state and output transformation can be rewritten as:

$$\mathfrak{z} = Tx = \begin{bmatrix} \mathfrak{z}_1 \\ \mathfrak{z}_2 \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \mathfrak{z} + \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} (T^{-1}\mathfrak{z}, t) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} (u + f_{IM}) + \begin{bmatrix} E_1 \\ 0 \end{bmatrix} \Delta\psi \quad (107b)$$

$$\mathscr{y} = Sy = \begin{bmatrix} \mathscr{y}_1 \\ \mathscr{y}_2 \end{bmatrix} = CT^{-1}\mathfrak{z} \quad (107c)$$

After some mathematical maneuvers, a compact version is obtained:

$$\dot{\mathfrak{z}}(t) = TAT^{-1}\mathfrak{z} + T\phi(T^{-1}\mathfrak{z}, t) + TB(u + f_{IM}) + TE\Delta\psi \quad (108)$$

$$\mathscr{y}(t) = CT^{-1}\mathfrak{z} \quad (109)$$

The original system is converted into two systems whose descriptions are given below.

System 1 is as follows:

$$\dot{\mathfrak{z}}_1(t) = A_1\mathfrak{z}_1 + A_2\mathfrak{z}_2 + B_1(u + f_{IM}) + \phi_1(T^{-1}\,\mathfrak{z}, t) + E_1\Delta\psi$$

$$\mathscr{y}_1 = C_1\mathfrak{z}_1$$

System 1 is in a slightly different form by replacing $\phi_1(T^{-1}\,\mathfrak{z}, t)$ by its value

$$\dot{\mathfrak{z}}_1(t) = A_1\mathfrak{z}_1 + A_2\mathfrak{z}_2 + B_1(u + f_{IM}) + T_1\phi(T^{-1}\,\mathfrak{z}, t) + E_1\Delta\psi \tag{110}$$

$$\mathscr{y}_1 = C_1\mathfrak{z}_1 \tag{110a}$$

System 2 as follows:

$$\dot{\mathfrak{z}}_2(t) = A_3\mathfrak{z}_1 + A_4\mathfrak{z}_2 + B_2(u + f_{IM}) + D_2f_{IM} + \phi_2(T^{-1}\,\mathfrak{z}, t) \tag{111a}$$

$$\mathscr{y}_2 = C_4\mathfrak{z}_2 \tag{111b}$$

System 2 in a slightly different form by replacing $\phi_2(T^{-1}\,\mathfrak{z}, t)$ by its value

$$\dot{\mathfrak{z}}_2(t) = A_3\mathfrak{z}_1 + A_4\mathfrak{z}_2 + B_2(u + f_{IM}) + D_2f_{IM} + T_2\phi(T^{-1}\,\mathfrak{z}, t) \tag{111c}$$

$$\mathscr{y}_2 = C_4\mathfrak{z}_2 \tag{111d}$$

If the rank assumption "$rank \begin{bmatrix} sI - A & E \\ C & 0 \end{bmatrix} = n + rank\ (E)$" mentioned above is true (A; C) is detectable [140].

Suppose that the matrix pair (A; C) is detectable, it follows also that the pair (A_4; C_4) is detectable. There is a matrix L defined as follows:

L $\in$ R^{nxp} / A_4 – LC_4 is stable and for any $Q_2 > 0$, thus, the Lyapunov equation is as follows:

$(A_4 - LC_4)^T P_2 + P_2(A_4 - LC_4) = -Q_2$ whose solution is $P_2 > 0$

A general Lyapunov equation in the context of nonlinear system stability or asymptotically stability definition maybe formulated as follows:

$LA + A^T L = -Q$; where Q > 0, L > 0 and symmetrical square matrices

Knowing A and Q, Q = I, the Lyapunov equation can be solved. In this dissertation YALIMP and appropriate functions in MATLAB are used to solve the equation.

The detectability of fault is the ability to detect the fault. Some faults become impossible to detect if the sliding mode observers are designed from the system before converting the original system into two subsystems because small magnitude faults can become zero, thus, unable to detect, thus, incipient.

In subsystem 2, whose equation is reproduced below:

$$\dot{\mathfrak{z}}_2(t) = A_3\mathfrak{z}_1 + A_4\mathfrak{z}_2 + B_2(u + f_{IM}) + D_2 f_{IM} + \phi_2(T^{-1}\,\mathfrak{z}, t) \qquad (112)$$

z_1 and z_2 are unaffected model uncertainties by faults prior to any system fault manifestation.

Suppose $\hat{\mathscr{y}}_2$ is the estimation of $\mathscr{y}_2$, then the output error/residual can be written as: $\mathscr{y}_2 - \hat{\mathscr{y}}_2$

Thus, a sliding mode observer for equation 110 can be formulated as follows:

$$\dot{\hat{\mathfrak{z}}}_1(t) = A_1\hat{\mathfrak{z}}_1 + A_2\hat{\mathfrak{z}}_2 + B_1 u + (A_1 - \ A_1^s)\, C_1^{-1}(\mathscr{y}_1 - \hat{\mathscr{y}}_1) + \phi_1(T^{-1}\hat{\mathfrak{z}}, t) + e_1^i \qquad (113)$$

$$\hat{\mathscr{y}}_1 = C_1\hat{\mathfrak{z}}_1$$

where:

$A_1^s \in \mathrm{R}^{r\times r}$: is a stable design matrix, which plays the role of a Luenberger observer gain

where $\hat{\mathfrak{z}} = \ col(\hat{\mathfrak{z}}_1, \hat{\mathfrak{z}}_2)$

e_1^i: is the added output error

$$e_1^i = \begin{cases} k_1 \frac{P_1(C_1^{-1}\,\mathscr{y}_1 - \hat{\mathfrak{z}}_1)}{\|P_1(C_1^{-1}\,\mathscr{y}_1 - \hat{\mathfrak{z}}_1)\|} & for\ \ C_1^{-1}\,\mathscr{y}_1 - \hat{\mathfrak{z}}_1 \neq 0 \\ 0 & else \end{cases} \qquad [140] \qquad (114)$$

It’s worth noting that in [140] the derivation and explanation of (114) can be found. where: k_1 is a component of $k(t, y, u)$ the calculated gain; $k_1 = \|E_1\|\mathcal{L}_{\psi+\eta_1}$; η_1: is a positive scalar to be determined; $P_1 \in \mathrm{R}^{r\times r} > 0$: is the symmetric definite Lyapunov matrix of A_1^s

$$k_1 = \ C_1^{-1}\,\mathscr{y}_1 = \ \ C_1^{-1}\, S_1 y$$

The Luenberger observer is designed for subsystem equation (111) as:

$$\dot{\hat{\mathfrak{z}}}_2(t) = A_4\hat{\mathfrak{z}}_2 + A_3 C_1^{-1}\mathscr{y}_1 + B_2 u + L(\mathscr{y}_2 - \hat{\mathscr{y}}_2) + \phi_2(T^{-1}\hat{\mathfrak{z}}, t)$$

$$\hat{\mathscr{y}}_2 = C_4\hat{\mathfrak{z}}_2 \qquad (115)$$

where

$L \in \mathrm{R}^{(n\,-\,r)\,\times(p\,-\,r)}$: is the gain of the Luenberger observer [133]

$e_1 = \mathfrak{z}_1 - \hat{\mathfrak{z}}_1$ and $e_2 = \mathfrak{z}_2 - \hat{\mathfrak{z}}_2$: are state estimation errors

$\dot{e}_1 = \dot{\mathfrak{z}}_1 - \dot{\hat{\mathfrak{z}}}_1$ and $\dot{e}_2 = \dot{\mathfrak{z}}_2 - \dot{\hat{\mathfrak{z}}}_2$: are state estimation error dynamics prior to system faults

The calculation of the state estimation error dynamics are as follows:

$$\dot{e}_1 = \dot{\mathfrak{z}}_1 - \dot{\hat{\mathfrak{z}}}_1 \;=\; A_1\mathfrak{z}_1 + A_2\mathfrak{z}_2 + \phi_1(T^{-1}\,\mathfrak{z}, t) + B_1 u + E_1\Delta\psi - A_1\hat{\mathfrak{z}}_1 - A_2\hat{\mathfrak{z}}_2$$

$$-\,\phi_1(T^{-1}\,\mathfrak{z}, t) - \, B_1 u - \,(A_1 - \, A_1^s)C_1^{-1}(\mathscr{y}_1 - \hat{\mathscr{y}}_1) - e_1^i$$

$$\dot{e}_1 = A_1^s e_1 + A_2 e_2 + [\phi_1(T^{-1}\,\mathfrak{z}, t) - \phi_1(T^{-1}\hat{\mathfrak{z}}, t)] + E_1\Delta\psi - e_1^i$$

$$\dot{e}_1 = A_1^s e_1 + A_2 e_2 + \Delta\phi_1 + E_1\Delta\psi - e_1^i \tag{116}$$

$$\dot{e}_2 = \dot{\mathfrak{z}}_2 - \,\dot{\hat{\mathfrak{z}}}_2 = A_3\mathfrak{z}_1 + A_4\mathfrak{z}_2 + \phi_2(T^{-1}\,\mathfrak{z}, t) - A_4\hat{\mathfrak{z}}_2 - A_3C_1^{-1}\mathscr{y}_1 - \phi_2(T^{-1}\hat{\mathfrak{z}}, t) - L(\mathscr{y}_2 - \hat{\mathscr{y}}_2)$$

$$\dot{e}_2 = \dot{\mathfrak{z}}_2 - \,\dot{\hat{\mathfrak{z}}}_2 = (A - LC_4)e_2 + [\phi_2(T^{-1}\,\mathfrak{z}, t) - \phi_2(T^{-1}\hat{\mathfrak{z}}, t)]$$

$$\dot{e}_2 = \dot{\mathfrak{z}}_2 - \,\dot{\hat{\mathfrak{z}}}_2 = (A - LC_4)e_2 + \Delta\phi_2 \tag{117}$$

where

$$\Delta\phi_1 = \phi_1(T^{-1}\,\mathfrak{z}, t) - \phi_1(T^{-1}\hat{\mathfrak{z}}, t) \tag{118}$$

$$\Delta\phi_2 = \phi_2(T^{-1}\,\mathfrak{z}, t) - \phi_2(T^{-1}\hat{\mathfrak{z}}, t) \tag{119}$$

When the induction motor is operating with no failure, the error dynamics formulation in equations 116 and 117 above are stable given that matrices “$A_1^s < 0$ and $P_1 = P_1^T > 0$", and the positive scalars “$\alpha_1 > 0$ and $\alpha_2 > 0$” exist.

Thus, the following inequality is given in equation 120 below.

$$\mathcal{M} = \begin{bmatrix} \mu_1 + \frac{1}{\alpha_1}(P_1)^2 & P_1 A_2 \\ A_1^T P_1 & \mu_2 + \frac{1}{\alpha_2}(P_2)^2 \alpha I_{n-r} \end{bmatrix} < 0 \qquad (120)$$

where

$\mu_1 = A_1^{sT} P_1 + P_1 A_1^s$

$\mu_2 = (A_4 - LC_4)^T P_2 + (A_4 - LC_4) P_2$

$\alpha = \alpha_1 \mathcal{L}_{\phi_1}^2 \|T^{-1}\|^2 + \alpha_2 \mathcal{L}_{\phi_2}^2 \|T^{-1}\|^2$

The proof that the error dynamics are asymptotically stable is given in detail in references [123-130] and [135].

The Linear Matrix Inequality (LMI) can be derived from the equation 120 above. For the LMI, the matrices: X, Y, P1 > 0, P, > 0 and the scalars: $\alpha_1 > 0$, $\alpha_2 > 0$ can be found.

Recall the Schur properties [141]. It states: Given a partitioned matrix

$S(x) = \begin{bmatrix} S_{11}(x) & S_{12}(x) \\ S_{21}(x) & S_{22}(x) \end{bmatrix}$;

where $S_{11}(x)$:a square matrix

Then the followings are true and known as the Schur complement [141].

$$S(x) < 0; \; S_{11}(x) < 0, \; S_{22}(x) - S_{21}(x) S_{11}^{-1}(x) S_{12}(x) < 0 \qquad (121)$$

$$S_{22}(x) < 0, \; S_{11}(x) - S_{12}(x) S_{22}^{-1}(x) S_{21}(x) < 0 \qquad (123)$$

Thus, the linear matrix inequality LMI is given below according to Schur decomposition and as in [130]:

$$\begin{bmatrix} X + X^T & P_1 & P_1 A_2 & 0 \\ P_1 & \alpha_1 I & 0 & 0 \\ A_1^T P_1 & 0 & A_1^T P_1 + P_2 A_4 - C_4^T Y^T - Y C_4 + \alpha I & P_2 \\ 0 & 0 & P_2 & -\alpha_2 I \end{bmatrix} < 0 \qquad (123)$$

where: $X = P_1 A_1^s$ and $Y = P_2 L$

The proof and decomposition of a LMI of the form formulated in equation 123 is given in [130].

5.8.1 Sliding Observer Design

The first task in designing a sliding mode observer is determining the sliding surface that ensures the desired stable dynamics. The second step is to identify the control gain. When the induction motor is fault free, the error dynamics in equations 116 and 117 are asymptotically stable according to equation 120.

The observer sliding mode control surface is formulated in equation 124.

$$S = (e_1 - e_2)|\, e_1 = 0 \qquad (124)$$

The sliding mode observer and the Luenberger observer [126], [133], [135] are reproduced respectively below as:

$$\dot{\hat{\mathfrak{z}}}_1(t) = A_1 \hat{\mathfrak{z}}_1 + A_2 \hat{\mathfrak{z}}_2 + B_1 u + (A_1 - A_1^s)\, C_1^{-1}(\mathcal{y}_1 - \hat{\mathcal{y}}_1) + \phi_1(T^{-1}\hat{\mathfrak{z}}, t) + e_1^i$$

$\widehat{w}_1 = C_1\hat{\mathfrak{z}}_1$ (125)

$\dot{\hat{\mathfrak{z}}}_2(t) = A_4\hat{\mathfrak{z}}_2 + A_3C_1^{-1}\mathcal{y}_1 + B_2u + L(\mathcal{y}_2 - \hat{\mathcal{y}}_2) + \phi_2(T^{-1}\hat{\mathfrak{z}}, t)$

$\hat{\mathcal{y}}_2 = C_4\hat{\mathfrak{z}}_2$ (126)

Equations 116 and 117 errors are set as in equation 124 assuming equation 121 can be solved, and the gain $k_1 = \|E_1\|\mathcal{L}_\psi + \eta_1$ is such that:

$\eta_1 \geq \left(\|A_2\| + \mathcal{L}_{\phi_1}\|T^{-1}\|\right)\varepsilon + \eta_2$

Recall the explanation of equation 114 is given in [140] where k_1 is a component of $k(t, y, u)$ the calculated gain; $k_1 = \|E_1\|\mathcal{L}_{\psi+\eta_1}$; η_1: is a positive scalar to be determined; $P_1 \in \mathrm{R}^{r\times r} > 0$: is the symmetric definite Lyapunov matrix of A_1^s

$k_1 = C_1^{-1}\mathcal{y}_1 = C_1^{-1}S_1y$

If the fault occurs at time t_f, then equations 116 and 117 are rewritten:

$\dot{e}_1 = A_1^s e_1 + A_2e_2 + [\phi_1(T^{-1}\mathfrak{z}, t) - \phi_1(T^{-1}\hat{\mathfrak{z}}, t)] + E_1\Delta\psi + B_1f_{IM} - e_1^i$ (127)

$\dot{e}_2 = (A_4 - LC_4)e_2 + [\phi_2(T^{-1}\mathfrak{z}, t) - \phi_2(T^{-1}\hat{\mathfrak{z}}, t)] + B_2f_{IM}$ (128)

The error e_2 is only affected by the induction motor faults f_{IM} unaffected by the motor model uncertainties $\Delta\psi$ and the error injection e_1^i. The bounded norm $\|e_2(t)\|$ depends on the initial condition on $e_2(0)$. Thus, the best candidate residual model for the induction motor fault is given in equation 129 below.

$\|e_{\mathcal{y}2}\| = \|C_4e_{\mathcal{y}2}\|$ (129)

C_4 is the system matrix new coordinate; it can be either positive or negative.

Suppose the residual given in equation 130 is greater than the set fault parameter threshold. In that case, a fault is detected, else the system is fault free within the set detection time given in equation 131 below, which is the first time it is detected that the residual value exceeds that of the threshold.

$$\|e_{\mathscr{y}2}\| \geq \mathcal{T} \tag{130}$$

$$t_d \geq t_f \tag{131}$$

5.9 Modified Observer in the Fault Isolation Approach

In the fault isolation part of the task, our objective is to identify where the faults or failures are; and to assess whether it is the occurrence of single, multiple faults, or failure in the system. It is assumed that the fault in the system, i.e., the induction motor, can be written as in equation 132 below.

$$f_{IM} = [f_{IM}^{1T}, f_{IM}^{2T} \dots, f_{IM}^{mT}]^T \tag{132}$$

Equation 132 represents the motor fault matrix.

Assessing the value of "f_{IM}^{i} for i =1 through m" equates to isolating the occurrence of such a fault. Thus, each nonzero fault value requires a sliding mode observer for subsystem 1 and system 2 sensitive to the particular fault.

The sliding mode observer for the subsystem is as follows:

$$\dot{\hat{\mathfrak{z}}}_1^i = A_1\hat{\mathfrak{z}}_1^i + A_2\hat{\mathfrak{z}}_1^i + B_1u + \phi_1(T^{-1}\,\mathfrak{z},t) + (A_1 - A_1^s)C_1^{-1}(\mathscr{y}_1^i - \hat{\mathscr{y}}_1^i) + e_1^{ii}$$

$$\hat{\mathcal{y}}_1^i = C_1 \hat{\mathfrak{z}}_1^i \tag{133}$$

Equation 133 contains the superscript “i” for i =1 to m faults, whereas equation 125 does not have this option.

where the state estimate is defined as:

$$\hat{\mathfrak{z}}^i = col\left(C_1^{-1}\mathcal{y}_1 , \hat{\mathfrak{z}}_2^i\right) \tag{134}$$

The output error injection is given and explained in detail in [143]:

$$e_1^{ii} = \begin{cases} (\|E_1\|\mathcal{L}_\psi + \|B_1\|\rho_a + \eta_1) \frac{P_1(C_1^{-1}\, \mathcal{y}_1 - \hat{\mathfrak{z}}_1^i)}{\|P_1(C_1^{-1}\, \mathcal{y}_1 - \hat{\mathfrak{z}}_1^i)\|} & for\ \ C_1^{-1}\, \mathcal{y}_1 - \hat{\mathfrak{z}}_1^i \neq 0 \\ 0 & else \end{cases} \tag{135}$$

where: η_1 is a positive scalar to be determined; $P_1 \in \mathrm{R}^{r\times r} > 0$: is the positive definite Lyapunov matrix of A_1^s

For the fault isolation, this time, a sliding mode observer is designed for the subsystem given by equation 115 is as follows:

$$\dot{\hat{\mathfrak{z}}}_2^i = A_4 \hat{\mathfrak{z}}_2^i + A_3 C_1^{-1} w_1^i + B_2 u + \phi_2\left(T^{-1}\hat{\mathfrak{z}}^i, t\right) + L\left(\mathcal{y}_2^i - \hat{\mathcal{y}}_2^i\right) + \bar{B}_2^i e_2^{ii}$$

$$\hat{\mathcal{y}}_2^i = C_1 \hat{\mathfrak{z}}_2^i \tag{136}$$

where: L: is the observer gain

$$B_2 = [B_2^i, \ldots\ B_2^m] \tag{136a}$$

The output error injection is given by [143]:

$$e_2^i = \begin{cases} (\rho_a + \eta_3)\ \frac{\bar{F}^i(\mathcal{y}_2^{-i} - \hat{\mathcal{y}}_2^i)}{\|\bar{F}^i(\mathcal{y}_2^{-i} - \hat{\mathcal{y}}_2^i)\|} & for\ \ \mathcal{y}_2^{-i} - \hat{\mathcal{y}}_2^i \neq 0 \\ 0 & else \end{cases} \tag{137}$$

where

η_3: is a positive scalar to be determined

$\bar{F}^i \in \mathrm{R}^{m\times(p-r)}$: is a matrix to be determined

$e_1^i = \mathfrak{z}_1^i - \hat{\mathfrak{z}}_1^i$: is the state estimation errors for the ith fault f_{IM}^i

$e_2^i = \mathfrak{z}_2^i - \hat{\mathfrak{z}}_2^i$: is the state estimation errors for the ith fault f_{IM}^i

There will always be an error after a fault has occurred in the system, and the error is calculated as follows:

$$\dot{e}_1^i = A_1^s e_1^i + A_2 e_1^i + \left[\phi_1(T^{-1}\,\mathfrak{z}, t) - \phi_1\left(T^{-1}\hat{\mathfrak{z}}^i, t\right)\right] + E_1 \Delta\psi + B_1 f_{IM} - e_1^{ii}$$

$$\dot{e}_2^i = (A_4 - LC_4) e_2^i + \left[\phi_2(T^{-1}\,\mathfrak{z}, t) - \phi_2\left(T^{-1}\hat{\mathfrak{z}}^i, t\right)\right] + B_2 f_{IM}^i - \bar{B}_2^i e_2^{ii}$$

$$\dot{e}_2^i = (A_4 - LC_4) e_2^i + \left[\phi_2(T^{-1}\,\mathfrak{z}, t) - \phi_2\left(T^{-1}\hat{\mathfrak{z}}^i, t\right)\right] + B_2^i f_{IM}^i + \bar{B}_2^i(\bar{f}_{IM}^i - e_1^{ii}) \qquad (138)$$

$\bar{f}_{IM}^i$: is only the vector fault of the system as stated earlier

Note that f_{IM}^i is the "ith" column in f_{IM}. $\bar{f}_{IM}^i$ is the motor fault vector; it does not contain f_{IM}^i

Given the assumption: $A_1^s < 0$, L, F, $P_1 = P_1^T > 0$, $P_2 = P_2^T > 0$, and scalars: $\alpha_1 > 0, \alpha_2 > 0$. That is, it is assumed that P_1, P_2, α_1 α_1 , and α_2 can be found.

Then afterwards equation 139, is obtained.

$$B_2^T P_2 = F C_4 \qquad (139)$$

and: $$\begin{bmatrix} \mu_1 + \frac{1}{\alpha_1}(P_1)^2 & P_1 A_2 \\ A_2^T P_1 & \mu_2 + \frac{1}{\alpha_2}(P_2)^2 \alpha I_{n-r} \end{bmatrix} < 0 \qquad (140)$$

$$\mu_1 = A_1^{sT} P_1 + P_1 A_1^s \qquad (141)$$

$$\mu_2 = (A_4 - LC_4)^T P_2 + (A_4 - LC_4) P_2 \qquad (142)$$

$$\alpha = \alpha_1 \mathcal{L}_{\phi_1}^2 \|T^{-1}\|^2 + \alpha_2 \mathcal{L}_{\phi_2}^2 \|T^{-1}\|^2 \qquad (143)$$

$$e_2^i \to 0 \;\; if \;\; f_{IM}^i = 0 \qquad (144)$$

$e_2^i \to 0 \;\; if \;\; f_{IM}^i = 0$, else $e_2^i \neq 0$ and then for $e_2^i \to 0$ e_2^i can be written as:

$$e_2^i = (A_4 - LC_4)e_2^i + \left[\phi_2(T^{-1}\,\mathfrak{z}, t) - \phi_2\left(T^{-1}\hat{\mathfrak{z}}^i, t\right)\right] + B_2^i f_{IM}^i + \bar{B}_2^i(\bar{f}_{IM}^i - e_2^{ii}) \qquad (145)$$

The equation $B_2^T P_2 = FC_4$ is rewritten as follows:

$$Trace\,[(B_2^T P_2 - FC_4)^T (B_2^T P_2 - FC_4)] = 0 \qquad (146)$$

There exists F as: $F \exists$ if $rank\,(C_4\,B_2) = rank\,(\,B_2)$

$$(B_2^T P_2 - FC_4)^T (B_2^T P_2 - FC_4) < \gamma^2 I_{n-r} \approx LMI \qquad (147)$$

$$(B_2^T P_2 - FC_4)^T (B_2^T P_2 - FC_4) < \gamma^2 I_{n-r} \approx \begin{bmatrix} -\gamma\, I_{n-r} & (B_2^T P_2 - FC_4)^T \\ B_2^T P_2 - FC_4 & -\gamma\, I_m \end{bmatrix} < 0 \qquad (148)$$

Equations 147, 148 are explained by the Schur properties discussed in reference [141] and in equation 130.

When the rank is formulated as "rank(C4, B2) = rank(B2)", then $B_2^T P_2 = FC_4$ can be solved because "F" exists, and the inequality in equation 148 can be solved by finding the minimizing γ in the equation. Thus, $Min\,\gamma$ implies:

- $P_1 > 0$ in inequality equation 125
- $P_2 > 0$ in inequality equation 148

Thus, in the fault diagnosis of the induction motor, a fault is detected if the fault value is different from zero, that is, if "$f_{IM}^i \neq 0$" within the detection time "t_d". Once a fault is detected in the system having "m" possible faults, the fault isolation is to be performed using "2*m" observers with one observer equation 133 for the state estimation and one observer equation 133 for the output estimation. When a fault is present, or when $f_{IM}^i \neq 0$, the dynamic error or the residual $\|e_{y2}^{i}\|$ exceeds the set threshold for some set time, i.e., $t_i > t_d$, else $f_{IM}^{i} = 0$. Thus, the residual is measured against the isolation threshold as in the equation below to isolate the fault.

$$\|e_{y2}^{i}\| = \|C_4 e_{y2}^{i}\| \leftrightarrow \mathcal{T}_i \tag{149}$$

The fault isolation is accomplished by comparing the residual from the fault isolation with the fault isolation threshold. The dissertation proposed approach for detecting and isolating fault in the induction motor is illustrated in the figure below. In this research, the induction motor parameters that is subject to contain the electrical fault signatures are:

- the rotor resistance
- the rotor inductance
- the stator resistance
- the stator inductance

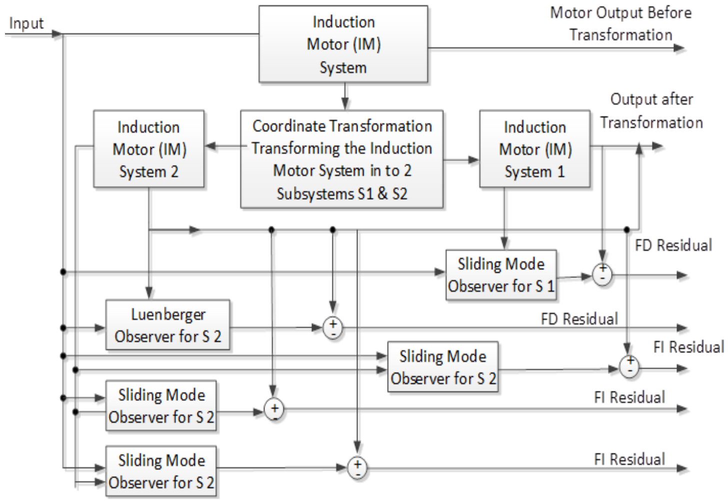

Input
Induction Motor (IM) System
Motor Output Before Transformation
Output after Transformation
Induction Motor (IM) System 2
Coordinate Transformation Transforming the Induction Motor System in to 2 Subsystems S1 & S2
Induction Motor (IM) System 1
FD Residual
Sliding Mode Observer for S 1
Luenberger Observer for S 2
FD Residual
FI Residual
Sliding Mode Observer for S 2
Sliding Mode Observer for S 2
FI Residual
Sliding Mode Observer for S 2
FI Residual
+
-

Figure 5.7 The Induction Motor Fault Detection and Isolation

Other observer methods can be designed. For instance, with a 3-observer design observer structure, the first observer generates a robust residual to the unknown disturbance input vector “d”. The second generates a residual insensitive to faults in the rotor resistance and the rotor inductance. Finally, the third observer produces or generates a residual, which is insensitive to faults in the stator resistance and in the stator inductance.

Table 5.1 The Nominal Induction Motor Parameters

Marathon 5K33GN2A ¼ Induction Motor Parameters	**Values**	**Bias Faults Magnitudes**
Stator resistance (Rs); (R_{s0})	11.05 ohms	$\delta R_s = 50\%$
Rotor resistance (Rr); (R_{r0})	6.11 ohms	$\delta R_r = 15\%$
Mutual inductance (Lm)	0.293939 H	
Stator inductance (Ls); (L_{s0})	0.316423 H	$\delta RL_s = 10\%$
Rotor inductance (Lr); (L_{r0})	0.316423 H	$\delta L_r = 5\%$
Number of pole(p)	4	
Power	¼ hp	
Torque Load (T_{L0})	1N.m	$\delta T_L = 20\%$
Load inertia (J); (J_0)	0.00062 kg*m^2	
Viscous friction coefficient (F) ; (f_0)	0.0008 (N.m)/(rad/s)	

For the sake of simplicity, the mutual inductance (Lm) may be assumed to be zero. Besides, the change in the mutual inductance is not considered as a fault.

The type of faults that can occur in the motor are represented in Figures 5.8, 5.9, 5.10, and 5.11 below.

The abrupt or jump fault is modeled by a step function usually caused by faults and represented in Figure 5.8.

The Intermittent or drift fault is modeled by many step functions usually caused by noises, represented in Figure 5.9.

The drift or incipient fault is modeled by a ramp function usually caused by faults or disturbances and represented in Figure 5.10.

In the fault frequency behavior, the fact that the incipient fault detection and isolation approach can be used in the presence of all faults is illustrated in Figure 5.11.

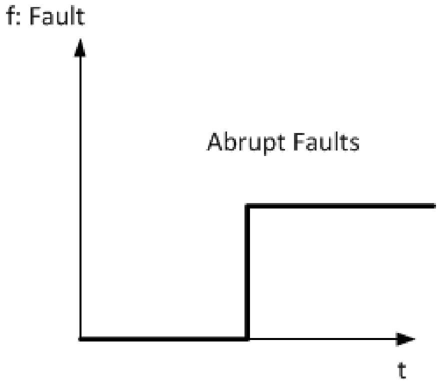


Figure 5.8 A Typical Abrupt Fault Representation

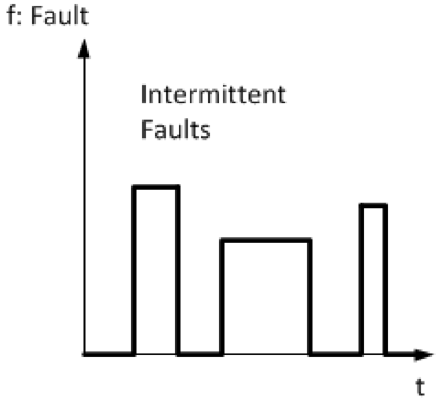


Figure 5.9 A Typical Intermittent Fault Representation

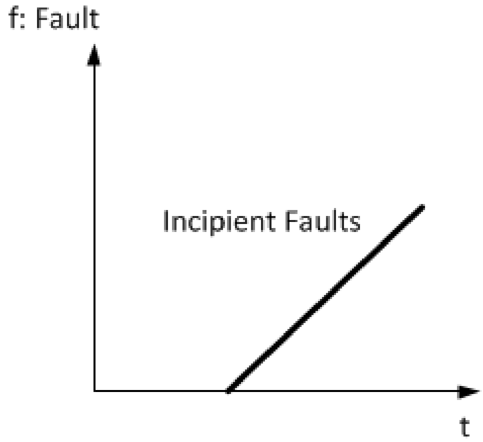


Figure 5.10 A Typical Incipient Fault Representation

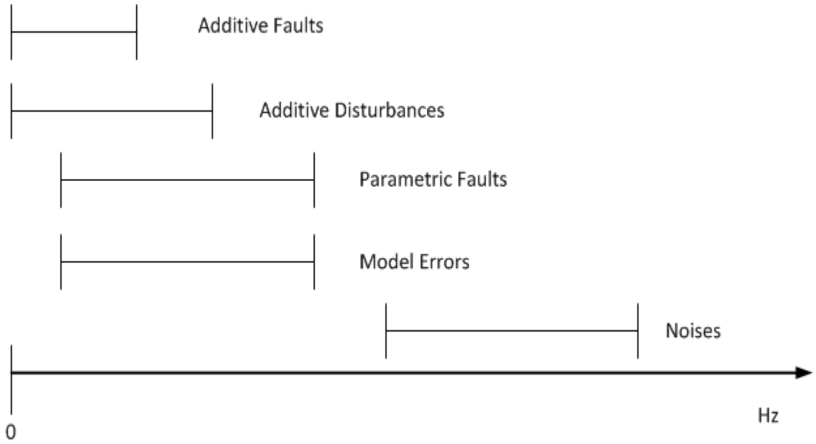


Figure 5.11 Faults Frequency Behaviors

The residual is evaluated at the nominal parameters of the motor. The nominal parameter values are estimated by the system identification method. In Figure 5.9 above, with an m-observer design observer structure, the fault detection observer generates a delta, which is the difference between the threshold and the estimated value. The generated delta is the residual. This differential value is attributed to all unknown disturbances and collected disturbances, I, and the unknown disturbance input vector “d”. The first observer generates a residual value insensitive to resistance faults in the rotor. The second observer generates a residual value insensitive to inductance faults in the rotor. The third observer generates a residual insensitive to faults in stator resistance, and the fourth observer generates a residual insensitive to faults in stator inductance. Let $\boldsymbol{r_d}$ be the fault detection residuals, r_i be the fault isolation residuals, and $r_1 \nRightarrow R_r$ which reads: residual 1 insensitive to the rotor resistor fault. Let 0 and 1 mean: = 0 or =1.

In the table below, only the four possible faults in the induction motor that manifest in the rotor and stator resistances and inductances are considered. However, in the simulation code, only two faults are used since two faults are enough to implement the detection and isolation schemes while keeping the code manageable.

Table 5.2 A Simple Fault Isolation Logic

r_d	$r_1 \nRightarrow R_r$	$r_2 \nRightarrow L_r$	$r_3 \nRightarrow R_s$	$r_4 \nRightarrow L_s$	**Interpretation is based on** $\forall(\boldsymbol{J}, \boldsymbol{f}, \boldsymbol{T_L})$
0	0	0	0	0	No Fault Detected
0	0	0	0	1	No Fault Detected
0	0	0	1	0	No Fault Detected
0	0	0	1	1	No Fault Detected
0	0	1	0	0	No Fault Detected
0	0	1	0	1	No Fault Detected
0	0	1	1	0	No Fault Detected
0	0	1	1	1	No Fault Detected
0	1	0	0	0	No Fault Detected
0	1	0	0	1	No Fault Detected
0	1	0	1	0	No Fault Detected
0	1	0	1	1	No Fault Detected
0	1	1	0	0	No Fault Detected
0	1	1	0	1	No Fault Detected
0	1	1	1	0	No Fault Detected
0	1	1	1	1	No Fault Detected
1	0	0	0	0	No Fault Detected
1	0	0	0	1	Fault Detected, but not a Ls fault
1	0	0	1	0	Fault Detected, but not a Rs fault
1	0	0	1	1	Fault Detected, not a stator fault
1	0	1	0	0	Fault Detected, not a Lr fault
1	0	1	0	1	Fault Detected, not a Lr or Ls fault
1	0	1	1	0	Fault Detected, not a Lr or Rs fault
1	0	1	1	1	Fault Detected, not a Lr, Rs, Ls fault
1	1	0	0	0	Fault Detected, not a Rrfault
1	1	0	0	1	Fault Detected, not a Rr or Ls fault
1	1	0	1	0	Fault Detected, not a Rr or Rs fault
1	1	0	1	1	Fault Detected, not a Rr, Rs, Ls fault
1	1	1	0	0	Fault Detected, not a Rr or Lr fault
1	1	1	0	1	Fault Detected, not a Rr, Lr, Ls fault
1	1	1	1	0	Fault Detected, not a Rr, Lr, Rs fault
1	1	1	1	1	No Fault Detected

For the induction motor in this application, a linear magnetic circuit and a balanced three-phase motor in the reference frame (a, b, c) is obtained. The reference frame (d, q) is used for the induction motor expression in the dual frame of reference (d, q): [1-3] and [144].

$$\begin{bmatrix} \psi_{sd} \\ \psi_{sq} \\ \psi_{rd} \\ \psi_{rq} \end{bmatrix} = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} * \begin{bmatrix} I_{sd} \\ I_{sq} \\ I_{rd} \\ I_{rd} \end{bmatrix}$$; see: [1-3] and [144] (150)

$$V_{sd} = R_s I_{sd} + \dot{\psi}_{sd} - \psi_{sq}\dot{\theta}_s$$

$$V_{sq} = R_s I_{sq} + \dot{\psi}_{sq} + \psi_{sq}\dot{\theta}_s$$

$$V_{rd} = R_r I_{rd} + \dot{\psi}_{rd} - \psi_{rq}\dot{\theta}_r = 0$$

$$V_{rd} = R_r I_{rd} + \dot{\psi}_{rd} + \psi_{rq}\dot{\theta}_s = 0$$; see: [1-3] and [144] (151)

Equations 150 to 155 are Power Engineering topics related to induction motor physics and operation explained in [1-3] and [144].

ψ_{sd} and ψ_{sq}: stator fluxes in (q, d) reference frame

ψ_{rd} and ψ_{rq}: rotor fluxes in (q, d) reference frame

I_{sd} and I_{sq} : the stator currents in (q, d) reference frame

I_{rd} and I_{rq}: the rotor currents in (q, d) reference frame

It's worth noting that: (ψ = flx in the matlab code)

The stator currents are assumed to be measured quantities in the (*a*, *b*, *c*) reference and expressed in the (*d*, *q*) reference frame.

V_{sd} and V_{sq} : the stator voltages in (q, d) reference frame

V_{rd} and V_{rq}: the rotor voltages in (q, d) reference frame

$R_{stator} = R_s$ and $R_{rotor} = R_r$ = stator and rotor resistances

$L_{stator} = L_s$ and $L_{rotor} = L_r$ = Stator and rotor inductances

$L_{stator} = L_s$ = Magnetizing inductances

$\theta_{stator} = \theta_s$ = Theta-Angle between reference frames (a, b, c), and (d, q)

$\theta_{rotor} = \theta_r$ = Theta-Angle between reference frames (a, b, c), and (d, q)

With the rotor voltages equal to zero in a squirrel cage induction motor, the squirrel-cage induction motor mechanical equation is obtained by adding inertia and friction effects.

$$\omega_m = \frac{d\theta_m}{dt} = (T_e - T_L - f)\,\frac{1}{J} \qquad (152)$$

The rotor speed is assumed to be measured quantity.

The electromagnetic torque is rewritten as:

$$T_e = \left(I_{sq}I_{rd} - I_{sd}I_{rq}\right)pL_m \qquad (153)$$

where:

ω_m: mechanical /rotor speed; T_L : load torque; J: the rotor inertia; f: the friction

Fixing the reference frame (d, q) to the stator gives the Park transformation in the following setting:

$$\theta_s = 0 \;\; ; \; \dot{\theta}_s = \omega_s ; \;\; s = \frac{\omega_e - \omega_s}{\omega_e} = \frac{\omega_m}{\omega_e} \qquad (154)$$

$$\dot{\theta}_m = \dot{\theta}_r = -\mathrm{p}\frac{d\theta_m}{dt} = -p\,\omega_m \tag{155}$$

ω_e: Electrical excitation frequency of the stator; ω_s : Stator speed

Thus, the induction motor nonlinear state space equations may be written as follows:

$$\dot{\mathrm{x}}(\mathrm{t}) = A\mathrm{x}(\mathrm{t}) + Bu(t) \tag{156}$$

$$y(t) = Cx(t) \tag{157}$$

where

$$x(t) = \begin{bmatrix} I_{sd} \\ I_{sq} \\ I_{rd} \\ I_{rq} \\ \omega_r \\ \omega_s \\ \omega_e \end{bmatrix};\ u(t) = \begin{bmatrix} V_{sd} \\ V_{sq} \end{bmatrix};\ \ y(t) = \begin{bmatrix} I_{sd} \\ I_{sq} \\ \omega_r \\ \omega_s \end{bmatrix}\ ;\ B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\Upsilon L_m & 0 \\ 0 & -\Upsilon L_m \\ 0 & 0 \\ \Upsilon L_r & 0 \\ 0 & \Upsilon L_r \end{bmatrix};$$

$$C = \begin{bmatrix} 0 & -0.451 & -0.40 & 1.55 & 0 & -0.447 & -0.052 \\ 0 & 1 & -1 & -5.2 & 0 & 0.347 & -1.14 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{158}$$

$$Ax(t) = \begin{bmatrix} (-L_rR_sx_1 + \mathrm{L}_m^2px_5x_2 + L_mR_rx_3 + pL_rL_mx_5x_4)\,\Upsilon \\ (-\mathrm{L}_r^2px_5x_1 - L_mR_sx_2 - pL_mL_rx_5x_3 + L_rR_rx_4)\,\Upsilon \\ (-\mathrm{L}_m^2px_5x_1 - L_rR_sx_2 - pL_rL_mx_5x_3 + L_mR_rx_4)\,\Upsilon \\ (L_mR_sx_1 - pL_sL_mx_5x_2 - L_sR_rx_3 - L_sL_rpx_5x_4)\,\Upsilon \\ (pL_sL_mx_5x_1 + L_mR_sx_2 + L_sL_rpx_5x_3 - L_sR_rx_4)\,\Upsilon \\ (pL_sL_mx_5x_1 + L_rR_sx_2 + L_sL_rpx_5x_3 - L_rR_rx_4)\,\Upsilon \\ (pL_mx_2x_3 - pL_mx_1x_4 - \ fx_5 - \mathrm{T}_L)\,\mathrm{J}^{-1} \end{bmatrix}\ ;\ \Upsilon = \frac{1}{L_rL_s - \mathrm{L}_m^2} \tag{159}$$

$$Ax(t)=\begin{bmatrix}\frac{-L_rR_sx_1}{L_rL_s-\mathrm{L}_m^2}+\frac{\mathrm{L}_m^2x_5x_2}{L_rL_s-\mathrm{L}_m^2}+\frac{L_mR_rx_3}{L_rL_s-\mathrm{L}_m^2}+\frac{pL_rL_mx_5x_4}{L_rL_s-\mathrm{L}_m^2}\\ \frac{-\mathrm{L}_r^2px_5x_1}{L_rL_s-\mathrm{L}_m^2}-\frac{L_mR_s\,x_2}{L_rL_s-\mathrm{L}_m^2}-\frac{pL_rL_mx_5x_3}{L_rL_s-\mathrm{L}_m^2}+\frac{L_rR_rx_4}{L_rL_s-\mathrm{L}_m^2}\\ \frac{-\mathrm{L}_m^2px_5x_1}{L_rL_s-\mathrm{L}_m^2}-\frac{L_rR_s\,x_2}{L_rL_s-\mathrm{L}_m^2}-\frac{pL_rL_mx_5x_3}{L_rL_s-\mathrm{L}_m^2}+\frac{L_mR_rx_4}{L_rL_s-\mathrm{L}_m^2}\\ \frac{L_mR_sx_1}{L_rL_s-\mathrm{L}_m^2}-\frac{pL_sL_mx_5x_2}{L_rL_s-\mathrm{L}_m^2}-\frac{L_sR_r\,x_3}{L_rL_s-\mathrm{L}_m^2}-\frac{L_sL_rpx_5x_4}{L_rL_s-\mathrm{L}_m^2}\\ \frac{pL_sL_mx_5x_1}{L_rL_s-\mathrm{L}_m^2}+\frac{L_mR_sx_2}{L_rL_s-\mathrm{L}_m^2}+\frac{L_sL_rpx_5x_3}{L_rL_s-L_m^2}-\frac{L_sR_rx_4}{L_rL_s-L_m^2}\\ \frac{pL_sL_mx_5x_1}{L_rL_s-\mathrm{L}_m^2}+\frac{L_rR_sx_2}{L_rL_s-\mathrm{L}_m^2}+\frac{L_sL_rpx_5x_3}{L_rL_s-L_m^2}-\frac{L_rR_rx_4}{L_rL_s-L_m^2}\\ \frac{pL_mx_2x_3}{J}-\frac{pL_mx_1x_4}{J}-\frac{fx_5-\mathrm{T}_L}{J}\end{bmatrix} \tag{160}$$

Using Park transformation, voltage measurements made in the (a, b, c) reference frame can be expressed in the (d, q) inequation 161 as follows:

$$u(t)=\begin{bmatrix}V_{sd}\\V_{sq}\end{bmatrix}=\begin{bmatrix}1 & -\frac{1}{2} & -\frac{1}{2}\\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}V_a\\V_b\\V_c\end{bmatrix}\quad \text{and} \quad \begin{bmatrix}i_a\\i_b\\i_c\end{bmatrix}=\begin{bmatrix}1 & 0\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}i_{sq}\\i_{sd}\end{bmatrix} \tag{161}$$

5.10 The Induction Motor Diagnostic Model

The most common faults encounter in the induction motors are stator windings, short circuits, and broken rotor bars. These faults manifest as electrical faults in the induction motors. Thus, they can be modeled as variations in the electrical parameters in the motor: the parameter values the stator and the rotor resistor and inductance.

A typical state vector nonlinear models may be as follows:

$$\dot{x}(t)=Ax(t)+Bu(t)+D(x)d+\phi(x,t)f_{IM}(t)+E\Delta\psi(t) \tag{162}$$

$$y(t)=Cx(t) \tag{163}$$

The nature of all the terms in equations 162 and 163 are exactly the same as those described for equations 93 and 94.

Now considering a (d, q) equivalent circuit on a synchronous frame for the equation 162 above, the matrices x, u, y, B and C are as follows:

$$x(t)=\begin{bmatrix} I_{sd}\\ I_{sq}\\ I_{rd}\\ I_{rq}\\ \omega_r\\ \omega_s\\ \omega_e \end{bmatrix};\ u(t)=\begin{bmatrix} V_{sd}\\ V_{sq} \end{bmatrix};\ \ y(t)=\begin{bmatrix} I_{sd}\\ I_{sq}\\ \omega_r\\ \omega_s \end{bmatrix}\ ;\ B=\begin{bmatrix} 0 & 0\\ 0 & 0\\ -\Upsilon L_m & 0\\ 0 & -\Upsilon L_m\\ 0 & 0\\ \Upsilon L_r & 0\\ 0 & \Upsilon L_r \end{bmatrix};$$

$$C=\begin{bmatrix} 0 & -0.451 & -0.40 & 1.55 & 0 & -0.447 & -0.052\\ 0 & 1 & -1 & -5.2 & 0 & 0.347 & -1.14\\ 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A=\begin{bmatrix} \frac{R_sL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_eL_sL_r-\omega_rL_m^2}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_rL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_sL_mL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_sL_mL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_sL_mL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_sL_mL_r}{L_rL_s-\mathrm{L}_m^2}\\ \frac{\omega_rL_m^2-\omega_eL_sL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{R_sL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{-\omega_sL_mL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_rL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_rL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_rL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_rL_m}{L_rL_s-\mathrm{L}_m^2}\\ \frac{-R_sL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{-\omega_sL_mL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_rL_sL_r-\omega_eL_m^2}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_rL_sL_r-\omega_eL_m^2}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_rL_sL_r-\omega_eL_m^2}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_rL_sL_r-\omega_eL_m^2}{L_rL_s-\mathrm{L}_m^2}\\ \frac{\omega_sL_mL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_sL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_eL_m^2-\omega_rL_sL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2}\\ \frac{\omega_sL_mL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_sL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_eL_m^2-\omega_rL_sL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2}\\ \frac{\omega_sL_mL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_sL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{\omega_eL_m^2-\omega_rL_sL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2}\\ \frac{\omega_eL_mL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{-R_sL_m}{L_rL_s-\mathrm{L}_m^2} & \frac{-\omega_sL_mL_r}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} & \frac{R_rL_s}{L_rL_s-\mathrm{L}_m^2} \end{bmatrix} \tag{164}$$

Thus, a fault vector is derived as: $f_{IM}=\begin{bmatrix} \delta R_s\\ \delta L_s\\ \delta R_r\\ \delta L_r \end{bmatrix}$, (165)

The unknown input vector disturbance: $\mathrm{d}=\begin{bmatrix} \delta J\\ \delta\mathrm{f}\\ \delta\mathrm{T}_L \end{bmatrix}$, (166) Assuming the disturbance will impact the states thus:

$$D(x) = \begin{bmatrix} \frac{\delta \mathrm{Ax}}{\delta \mathrm{J}} & \frac{\delta \mathrm{Ax}}{\delta \mathrm{f}} & \frac{\delta \mathrm{Ax}}{\delta T_L} \end{bmatrix}_0 \tag{167}$$

where

$$\frac{\delta \mathrm{Ax}}{\delta \mathrm{J}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \tau_1 \end{bmatrix}; \ \frac{\delta \mathrm{Ax}}{\delta f} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \tau_2 \end{bmatrix}; \ \frac{\delta \mathrm{Ax}}{\delta T_L} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \tau_3 \end{bmatrix}$$

$$\text{Thus: } D(x) = \begin{bmatrix} 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \\ \tau_1 \tau_2 \tau_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \tau_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_3 \end{bmatrix}^T_{(R_{s0}\,, L_{s0}\,, R_{r0,}\ L_{r0}\,, J_0\,,\ f_0\,, T_{L0}\ )} \tag{168}$$

where

$$\tau_1 = \frac{(-pL_m x_2 x_3 + pL_m x_1 x_4 + f x_5 - T_L\,)}{J^2}, \ \ \tau_2 = -\frac{x_5}{J} \text{ and } \tau_3 = -\frac{1}{J} \tag{169}$$

and $\mathrm{D}(x)$ to be evaluated at the nominal values of the following parameter:

$(R_{s0}\,, L_{s0}\ \ , R_{r0}\ \ , L_{r0}\,, J_0\,,\ f_{0\,,} T_{L0}\ )$

$$Bu(t) = \begin{bmatrix} 0 \\ 0 \\ -\Upsilon L_m V_{sd} \\ -\Upsilon L_m V_{sq} \\ 0 \\ \Upsilon L_r V_{sd} \\ \Upsilon L_r V_{sq} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{-L_m V_{sd}}{L_r L_s - \mathrm{L}_m^2} \\ \frac{-L_m V_{sq}}{L_r L_s - \mathrm{L}_m^2} \\ 0 \\ \frac{L_r V_{sd}}{L_r L_s - \mathrm{L}_m^2} \\ \frac{L_r V_{sq}}{L_r L_s - \mathrm{L}_m^2} \end{bmatrix} \tag{170}$$

Assuming the known nonlinear term will impact the states thus:

$$\phi(x,u) = \begin{bmatrix} \frac{\delta \mathrm{Ax}}{\delta R_s} & \frac{\delta \mathrm{Ax}}{\delta L_s} & \frac{\delta \mathrm{Bu}}{\delta L_s} & \frac{\delta \mathrm{Ax}}{\delta R_r} & \frac{\delta \mathrm{Ax}}{\delta L_r} & \frac{\delta \mathrm{Bu}}{\delta L_r} & \frac{\delta \mathrm{Ax}}{\delta J} \end{bmatrix}_{(R_{s0}\,,L_{s0}\ \,,R_{r0}\ \ L_{r0}\,,J_0\,,\ f_0\,,T_{L0}\ )} \tag{171}$$

Equation 171 states that $\phi(x,u)$ must be evaluated for the values of "$R_{s0}\,,L_{s0}\ \,,R_{r0}\ \ L_{r0}\,,J_0\,,\ f_0\,,T_{L0}$ "

For simplicity let us denote the matrix elements as: $\phi_{column,\ row}$

$$\frac{\delta \mathrm{Ax}}{\delta R_s} = \begin{bmatrix} \phi_{11} = \frac{-L_r x_1}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{12} = \frac{-L_r\, x_2}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{13} = \frac{L_m x_1}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{14} = \frac{L_m x_2}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{15} = 0 \\ \phi_{16} = 0 \\ \phi_{17} = 0 \end{bmatrix} \tag{172}$$

$$\frac{\delta \mathrm{Ax}}{\delta L_s} =$$

$$\begin{bmatrix} \phi_{21} = \frac{L_r L_r R_s x_1}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{L_r \mathrm{L}_m^2 x_5 x_2}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{L_r L_m R_r x_3}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{L_r p L_r L_m x_5 x_4}{(L_r L_s - \mathrm{L}_m^2)^2} \\ \phi_{22} = \frac{L_r \mathrm{L}_m^2 p x_5 x_1}{(L_r L_s - \mathrm{L}_m^2)^2} + \frac{L_r L_r R_s\, x_2}{(L_r L_s - \mathrm{L}_m^2)^2} + \frac{L_r p L_r L_m x_5 x_3}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{L_r L_m R_r x_4}{(L_r L_s - \mathrm{L}_m^2)^2} \\ \phi_{23} = -\frac{L_r L_m R_s x_1}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{(p L_m x_5 x_2)(L_r L_s - \mathrm{L}_m^2) - L_r p L_s L_m x_5 x_2}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{(R_r\, x_3)(L_r L_s - \mathrm{L}_m^2) - L_r L_s R_r\, x_3}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{(L_r p x_5 x_4)(L_r L_s - \mathrm{L}_m^2) - L_r L_s L_r p x_5 x_4}{(L_r L_s - \mathrm{L}_m^2)^2} \\ \phi_{24} = \frac{(p L_m x_5 x_1)(L_r L_s - \mathrm{L}_m^2) - L_r p L_s L_m x_5 x_1}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{L_r L_m R_s x_2}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{(L_r p x_5 x_3)(L_r L_s - \mathrm{L}_m^2) - L_r L_s L_r p x_5 x_3}{(L_r L_s - \mathrm{L}_m^2)^2} - \frac{(R_r x_4)(L_r L_s - \mathrm{L}_m^2) - L_r L_s R_r x_4}{(L_r L_s - \mathrm{L}_m^2)^2} \\ \phi_{25} = 0 \\ \phi_{26} = 0 \\ \phi_{27} = 0 \end{bmatrix}$$

(173)

$$\frac{\delta \mathrm{Bu}}{\delta L_s} = \begin{bmatrix} \phi_{31} = 0 \\ \phi_{32} = 0 \\ \phi_{33} = \frac{L_r L_m V_{sd}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{34} = \frac{L_r L_m V_{sq}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{35} = 0 \\ \phi_{36} = -\frac{L_r L_r V_{sd}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{37} = \frac{L_r L_r V_{sq}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \end{bmatrix}; \qquad \frac{\delta \mathrm{Ax}}{\delta R_r} = \begin{bmatrix} \phi_{41} = \frac{L_m\, x_3}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{42} = \frac{L_m\, x_4}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{43} = \frac{-L_s x_3}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{44} = \frac{-L_s x_4}{L_r L_s - \mathrm{L}_m^2} \\ \phi_{45} = 0 \\ \phi_{46} = 0 \\ \phi_{47} = 0 \end{bmatrix} \tag{174}$$

$$\frac{\delta \mathrm{Ax}}{\delta L_r} =$$

$$\begin{bmatrix} \phi_{51} = \frac{-R_s x_1\left(L_r L_s - \mathrm{L}_m^2\right) + L_s L_r R_s x_1}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{L_s \mathrm{L}_m^2 x_5 x_2}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{L_s L_m R_r x_3}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} + \frac{p L_m x_5 x_4\left(L_r L_s - \mathrm{L}_m^2\right) - p L_r L_r L_m x_5 x_4}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{52} = \frac{L_r \mathrm{L}_m^2 p x_5 x_1}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{R_s x_2\left(L_r L_s - \mathrm{L}_m^2\right) - L_s L_r R_s x_2}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{p L_m x_5 x_3\left(L_r L_s - \mathrm{L}_m^2\right) - L_s p L_r L_m x_5 x_3}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} + \frac{-L_s L_m R_r x_4}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{53} = \frac{-L_s L_m R_s x_1}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{-L_s p L_s L_m x_5 x_2}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{-\, L_s L_s R_r x_3}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{\left(L_s p x_5 x_4\right)\left(L_r L_s - \mathrm{L}_m^2\right) - L_s L_s L_r p x_5 x_4}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{54} = \frac{-L_s p L_s L_m x_5 x_1}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} + \frac{-L_s L_m R_s x_2}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} + \frac{\left(L_s p x_5 x_3\right)\left(L_r L_s - \mathrm{L}_m^2\right) - L_s L_s L_r p x_5 x_3}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} - \frac{-L_s L_s R_r x_4}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{55} = 0 \\ \phi_{56} = 0 \\ \phi_{57} = 0 \end{bmatrix}$$

$$\tag{175}$$

$$\frac{\delta \mathrm{Bu}}{\delta L_r} = \begin{bmatrix} \phi_{61} = 0 \\ \phi_{62} = 0 \\ \phi_{63} = \frac{L_s L_m V_{sd}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{64} = \frac{L_s L_m V_{sq}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{65} = 0 \\ \phi_{66} = \frac{V_{sd}\left(L_r L_s - \mathrm{L}_m^2\right) - L_s L_r V_{sd}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \\ \phi_{67} = \frac{V_{sq}\left(L_r L_s - \mathrm{L}_m^2\right) - L_s L_r V_{sq}}{\left(L_r L_s - \mathrm{L}_m^2\right)^2} \end{bmatrix} \tag{176}$$

Following the same mathematical maneuvers above $\frac{\delta \mathrm{Ax}}{\delta J}$ may be calculated and, thus equation 177.

$$\phi(x,u) = \begin{bmatrix} \frac{\delta \mathrm{Ax}}{\delta R_s} & \frac{\delta \mathrm{Ax}}{\delta L_s} & \frac{\delta \mathrm{Bu}}{\delta L_s} & \frac{\delta \mathrm{Ax}}{\delta R_r} & \frac{\delta \mathrm{Ax}}{\delta L_r} \frac{\delta \mathrm{Bu}}{\delta L_r} & \frac{\delta \mathrm{Ax}}{\delta J} \end{bmatrix}_{(R_{s0}\,, L_{s0}\ \ , R_{r0}\ , L_{r0}\,, J_0\,,\ f_0\,,\ T_{L0}\ )} \tag{177}$$

Then equation 177 becomes:

$$\phi(x) = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} & \phi_{16} & \phi_{17} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} & \phi_{26} & \phi_{27} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} & \phi_{36} & \phi_{37} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} & \phi_{45} & \phi_{46} & \phi_{47} \\ \phi_{51} & \phi_{52} & \phi_{53} & \phi_{54} & \phi_{55} & \phi_{56} & \phi_{57} \\ \phi_{61} & \phi_{62} & \phi_{63} & \phi_{64} & \phi_{65} & \phi_{66} & \phi_{67} \\ \phi_{71} & \phi_{72} & \phi_{73} & \phi_{74} & \phi_{75} & \phi_{76} & \phi_{77} \end{bmatrix} \tag{178}$$

The known nonlinear independent of the input may be written as:

$$\phi(x) = \begin{bmatrix} \frac{\delta \mathrm{Ax}}{\delta R_s} & \frac{\delta \mathrm{Ax}}{\delta L_s} & \frac{\delta \mathrm{B}}{\delta L_s} & \frac{\delta \mathrm{Ax}}{\delta R_r} & \frac{\delta \mathrm{Ax}}{\delta L_r} & \frac{\delta \mathrm{B}}{\delta L_r} & \frac{\delta \mathrm{A}}{\delta \mathrm{J}} \end{bmatrix}_{(R_{s0}\,, L_{s0}\ \ , R_{r0}\ \ L_{r0}\,, J_0\,,\ f_0\,, T_{L0}\ )} \tag{179}$$

$\phi(x)$ to be evaluated at the nominal values of the following parameters: $(R_{s0}\,, L_{s0}, R_{r0}, L_{r0}, J_{0,}\, f_{0\,,} T_{L0}\ )$

Note that the nonlinear term of the input may be calculated using the evaluated equations above or selected using any nonlinear function.

The new fault vector is formulated in equation 180 given below.

$$f_{IM} = \begin{bmatrix} \delta R_s \\ \delta L_s \\ u^T \delta L_s \\ \delta R_r \\ \delta L_r \\ u^T \delta L_r \end{bmatrix} \tag{180}$$

Rewriting the matrix B from above as:

$$B=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{L_m}{L_rL_s-\mathrm{L}_m^2} & 0 \\ 0 & -\frac{L_m}{L_rL_s-\mathrm{L}_m^2} \\ 0 & 0 \\ \frac{L_r}{L_rL_s-\mathrm{L}_m^2} & 0 \\ 0 & \frac{L_r}{L_rL_s-\mathrm{L}_m^2} \end{bmatrix};\ \frac{\delta\mathrm{B}}{\delta L_s}=\begin{bmatrix} \phi_{31}=0 \\ \phi_{32}=0 \\ \phi_{33}=-\frac{-L_rL_m}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \\ \phi_{34}=-\frac{-L_rL_m}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \\ \phi_{35}=0 \\ \phi_{36}=\frac{-L_rL_r}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \\ \phi_{37}=\frac{-L_rL_r}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \end{bmatrix};\ \frac{\delta\mathrm{B}}{\delta L_r}=\begin{bmatrix} \phi_{61}=0 \\ \phi_{62}=0 \\ \phi_{63}=-\frac{-L_mL_s}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \\ \phi_{64}=-\frac{-L_mL_s}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \\ \phi_{35}=0 \\ \phi_{36}=\frac{L_rL_s-\mathrm{L}_m^2-L_rL_s}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \\ \phi_{37}=\frac{L_rL_s-\mathrm{L}_m^2-L_rL_s}{\left(L_rL_s-\mathrm{L}_m^2\right)^2} \end{bmatrix}$$

(181)

When the known nonlinear terms are independent of the input, $\frac{\delta\mathrm{B}}{\delta L_s}$ and $\frac{\delta\mathrm{B}}{\delta L_r}$ must be used in the $\phi(x,t)$ not $\frac{\delta\mathrm{Bu}}{\delta L_s}$ and $\frac{\delta\mathrm{Bu}}{\delta L_r}$.

where the additional terms associated with the non-linearity and system uncertainty can be set in equation 182 as follows:

$$E=\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$ and $\Delta\psi(t)=\ 2\cos t$ or $\Delta\psi(t)=\ 2\sin t$

$$\phi(x,t)=\begin{bmatrix} cos(x_3+1) \\ cos(x_3+1) \\ 0 \\ 0 \\ cos(x_3+1) \\ 0 \\ 0 \end{bmatrix} \quad \text{or } \phi(x,t)=\begin{bmatrix} sin(x_3+1) \\ sin(x_3+1) \\ 0 \\ 0 \\ sin(x_3+1) \\ 0 \\ 0 \end{bmatrix} \qquad (182)$$

In equation 192, nonlinear function, i.e., cosine or sine may be used to represent the nonlinear input term.

One may as well omit and neglect these additional terms, or set them to zero. Some of these terms are already identified in the motor model. They come to manifest as an excessive change in specific parameters, i.e., resistance and inductance. Other faults manifest in other electrical signatures since the focus is on incipient faults that are not abrupt but time dependent.

$$f_{IM} = \begin{bmatrix} \delta R_s \\ \delta L_s \\ \delta R_r \\ \delta L_r \end{bmatrix} \tag{183}$$

Modeling the fault using positive exponential distribution shows the change in the parameter value increasing with time as in incipient faults. Thus, the motor fault vector becomes $f_{IM} = [f_{IM1} \quad f_{IM2}]$, known as the column values of the fault.

where

$$f_{IM1} = e^{0.02*t} \tag{184}$$

However, for all time periods less than 25 seconds, f_{IM1} faults are ignored due to possible electrical parameter value excursion in the electronic system.

where

$$f_{IM2} = e^{0.045*t} \tag{185}$$

However, for times less than 35 seconds, all f_{IM2} fault values are also ignored due to possible electrical parameter value excursion in the electronic system.

5.11 Nonlinear Disturbance Decoupling from a Nonlinear Perspective

Using a more general nonlinear model to describe the induction motor is given as follows below:

$$\dot{x}(t) = A(x) + Bu(t) + D(x)d + \phi(x)f_{IM} \tag{186}$$

$$y(t) = C(x) \tag{187}$$

The above model can be rewritten by means of a change of variables given by the state transformation matrix below. It defines a new state "z" uncoupled or decoupled from the unknown input [122-127], [130], [135].

$$\mathfrak{z} = T(x) \tag{189}$$

$$\dot{\mathfrak{z}} = \dot{T}(x) = \frac{\partial T(x)}{\partial x}(A(x) + B(x)u + D(x)d + \phi(x)f_{IM}) \tag{190}$$

$\frac{\partial T(x)}{\partial x}$ in equation 190 means that the transformation matrix T will be acting the entire state equation matrices to transform the system and obtain the coordinate in equation 189.

For the transformed model to be unaffected by the unknown inputs but still able to detect faults, it is required that:

$$\frac{\partial T(x)}{\partial x} D(x)d = 0 \tag{191}$$

Thus, the disturbance decoupled is independent of the unknown inputs d.

$$\dot{\mathfrak{z}} = \dot{T}(x) = \frac{\partial T(x)}{\partial x}[A(x) + B(x)u + \phi(x)f_{IM}] \tag{192}$$

The ranks are given in equation 193 below.

$$\text{rank}[D(x)] = n_d \quad \text{and rank}\left(\frac{\partial T(x)}{\partial x}\right) = rank\begin{pmatrix} \frac{\partial T_1(x)}{\partial x} \\ \vdots \\ \frac{\partial T_{n-n_d}(x)}{\partial x} \end{pmatrix} = n - n_d \qquad (193)$$

$$\frac{\partial T(x)}{\partial x} = \begin{pmatrix} \frac{\partial T_1(x)}{\partial x} \\ \vdots \\ \frac{\partial T_{n-n_d}(x)}{\partial x} \end{pmatrix} \qquad (194)$$

The complete transformation is: $\mathfrak{z} = T(x) = \begin{pmatrix} T_1(x) \\ \vdots \\ T_{n-n_d}(x) \end{pmatrix}$ (195)

Thus, $\mathfrak{z}$ decoupled is stated as follows:

$$\dot{\mathfrak{z}} = \dot{T}(x) = \frac{\partial T(x)}{\partial x}\left(A(x) + B(x)u + \phi(x)f_{IM}\right)_{x=V^{-1}(z,y')} \qquad (196)$$

$$y(t) = C(x)_{x=V^{-1}(z,y')} \qquad (197)$$

$$V(x) = \begin{bmatrix} \mathfrak{z} \\ y' \end{bmatrix} \qquad (198)$$

$$V(x) = \begin{bmatrix} T(x) \\ V'(y)_{y=C(x)} \end{bmatrix} \qquad (199)$$

The observer gain must be calculated under no-fault condition so that the observer is locally asymptotically stable. For the residual vector to be a function of fault or to reflect faults, it is required that:

$$rank\left[\frac{\partial T(x)}{\partial x}\ \phi(x)\right] = rank[\phi(x)] \qquad (200)$$

Equation 202 is the condition for fault sensitivity; the rank of the derivative of transformation function T(x), and the function $\phi(x)$ equals the rank of the function $\phi(x)$.

5.12 Residuals and Other Observers for Fault Isolation

As illustrated the fault isolation is performed using the fault vector by the aid of observer methods.

f_{IM}^{i} is the set of faults to which a particular observer is sensitive.

$\bar{f}_{IM}^{i}$ is the rest set of faults to which a particular observer is insensitive.

$F^{i}(x)$ and $\bar{F}^{i}(x)$ are partitions of F^{i}

then

$$\frac{\partial T(x)}{\partial x} D(x)d = 0 \tag{201}$$

$$\frac{\partial T(x)}{\partial x} D(x)d = 0 \sim \frac{\partial T(x)}{\partial x}\left[D(x) \quad \bar{F}^{i}(x)\right] = 0 \tag{202}$$

$$rank\left[\frac{\partial T(x)}{\partial x} \ F^{i}(x)\right] = rank\left[F^{i}(x)\right] \tag{203}$$

In the design of the fault isolation observer, the observers are restricted as follows: The first observer or the fault detection observer residual is robust to the unknown input disturbance vector, the i^{th} observer residual is insensitive to faults in the rotor, and the $(i+1)^{th}$ observer residual is insensitive faults in the stator.

The observer residuals, as shown in Figure 5.7 can be further structured by assuming that the fault detection observer generates five residuals.

- an unknown disturbance related residual
- a rotor resistance-related residual
- a rotor inductance related residual

- a stator resistance-related residual
- a stator inductance related residual

These residuals are robust to the unknown disturbance input vector “d”, insensitive to faults in the rotor resistance, insensitive to faults in the rotor inductance, insensitive to faults in stator resistance, and insensitive to faults in stator inductance.

In the implementation of the dedicated observer, all faults and their respective residuals must be specified. Each fault has its unique residual that serves to isolate one fault from another. For the generalized observer, all faults and their respective residuals must be specified. However, residual 1, for instance, is affected by all faults except fault 1.

What follows is a digressionary explanation regarding the dedicated observer and the generalized observer.

Thus, for a non-linear model given in equations 207 and 208 its related non-linear observer is given in equation 209 to 211.

$$\dot{x}(t) = g\big(x(t), u(t)\big) + R_1 f(t) \tag{207}$$

$$y(t) = h\big(x(t), u(t)\big) + R_2 f(t) \tag{208}$$

$$\dot{\hat{x}}(\mathrm{t}) = g\big(\hat{x}(t), u(t)\big) + K\big(\hat{x}(t), u(t)\big)\big(y(t) - \hat{y}(t)\big) \tag{209}$$

$$y(t) = h\big(\hat{x}(t), u(t)\big) \tag{210}$$

$$r(t) = y(t) - \hat{y}(t) \tag{211}$$

The reduction error in the state estimation and the residual model equations are:

$$e(t) = x(t) - \hat{x}(t) \tag{212}$$

$$\dot{e}(t) = F(t)e(t) + O_1(e^2(t), t\ ) + R_1 f(t) - K\big(\hat{x}(t), u(t)\big)R_2\, f(t) \tag{213}$$

$$r(t) = h(t)e(t) + O_2(e^2(t), t\ ) + R_2 f(t) \tag{214}$$

where:

$$F(t) = \frac{\delta g\big(\hat{x}(t), u(t)\big)}{\delta \hat{x}(t)} - K\big(\hat{x}(t), u(t)\big)\, H(t) \tag{215}$$

$$H(t) = \frac{\delta h\big(\hat{x}(t), u(t)\big)}{\delta \hat{x}(t)} \tag{216}$$

As stated from the outset, this fault detection and isolation method is implemented using the described induction motor; however, instead of using all the motor's possible faults, two failures or faults are used for the validation of the proposed fault diagnosis method. Equations 207 to 211 are used as a digressionary explanation regarding dedicated the observer and the generalized observer. Chapter 5 lays out the steps and the mathematics of the fault detection and isolation. The results are illustrated in chapter 6 and the rest of the dissertation.

Chapter 6: Fault Diagnosis Research Validation and Result Analysis

6.1 MATLAB Code/Simulation and Outcomes

Many equations and matrices used in this research are presented in this chapter. In sections 6.1, 6.2, and 6.3, results and preliminary matrices are obtained and discussed; concluding with the explanation of "Fault Diagnosis Research Validation Result Analysis" and discussing the "Technical Points to Note" in section 6.4.

The Matrices A, B, C used in the model-based fault detection and isolation simulation code are calculated from the motor parameters given by the manufacturer, the measured parameters in the electrical laboratory experiments, from the non-measured parameters calculated by MATLAB code whose codes and results are in appendixes A and B, and equation in sections 5.10, 5.11.

The additional terms associated with the non-linearity and system uncertainty mentioned and explained earlier in the previous sections are restated here. They are the matrices E, $\Delta\psi(t)$ and $\phi(x,t)$

where the equations are reproduced as:

$E = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ and $\Delta\psi(t)$ to be $\Delta\psi(t) = 2\cos t$ or $\Delta\psi(t) = 2\sin t$ are chosen

$\phi(x,t)$ to be $\phi(x,t) = \begin{bmatrix} cos(x_3 + 1) \\ cos(x_3 + 1) \\ 0 \\ 0 \\ cos(x_3 + 1) \\ 0 \\ 0 \end{bmatrix}$ or $\phi(x,t) = \begin{bmatrix} sin(x_3 + 1) \\ sin(x_3 + 1) \\ 0 \\ 0 \\ sin(x_3 + 1) \\ 0 \\ 0 \end{bmatrix}$ are chosen as well.

The fault vector of incipient and time-dependent faults is immediately below.

$$f_{IM} = \begin{bmatrix} \delta R_s \\ \delta L_s \\ \delta R_r \\ \delta L_r \end{bmatrix}$$

The fault is modeled using positive exponential distribution as given in equations 183 to 185.

Depending on the number of faults anticipated in the monitored equipment, a column for each fault is used.

6.2 MATLAB Code for the System Matrix Transformation

For a calculated system matrix, A, if there exists a column vector a, such that T = [a, Aa, · · ·, A^{n-1}a] is non-singular, the system matrix A can be transformed into a companion-like matrix. The column vector can be generated randomly; then, the System matrix is transformed and calculated. The transformed matrix is not unique. If one does need the standard companion form, the following statements can further be given.

From motor nonlinear models explained above and restated here.

$$\dot{x}(t) = Ax(t) + Bu(t) + D(t)d + \phi(x,t)f_{IM} + E\Delta\psi(t)$$

$$y(t) = Cx(t)$$

$$where\ x(t) = \begin{bmatrix} I_{sd} \\ I_{sq} \\ I_{rd} \\ I_{rq} \\ \omega_r \\ \omega_s \\ \omega_e \end{bmatrix};, u(t) = \begin{bmatrix} V_{sd} \\ V_{sq} \end{bmatrix};\ \ and\ y(t) = \begin{bmatrix} I_{sd} \\ I_{sq} \\ \omega_r \\ \omega_s \end{bmatrix}\ ;B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\Upsilon L_m & 0 \\ 0 & -\Upsilon L_m \\ 0 & 0 \\ \Upsilon L_r & 0 \\ 0 & \Upsilon L_r \end{bmatrix};$$

The following system matrices and vectors are obtained. See section 5.10 for the derivation and appendixes for the MATLAB codes.

```
A=[0, 0, 1, 0, 0, 0, 0; 0, -0.154, -0.04, 1.54,  0, -0.744, -0.032; 0,  0.249,
-1, -5.2,  0,   0.337,  -1.12; 0.0386,  -0.996,  0, -2.117,  0, 0.02, 0; 0, 0.5,
0, 0, -4, 0, 0; 0, 0, 0, 0, 0,   -20, 0; 0, 0, 0,  0, 0, 0,  -25];
```

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\Upsilon L_m & 0 \\ 0 & -\Upsilon L_m \\ 0 & 0 \\ \Upsilon L_r & 0 \\ 0 & \Upsilon L_r \end{bmatrix} = B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 7 & 0 \\ 0 & 19 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & -0.451 & -0.40 & 1.55 & 0 & -0.447 & -0.052 \\ 0 & 1 & -1 & -4.5 & 0 & 0.47 & -1.14 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \phi(x,t) = \begin{bmatrix} cos(x_3+1) \\ cos(x_3+1) \\ 0 \\ 0 \\ cos(x_3+1) \\ 0 \\ 0 \end{bmatrix} \text{ or } \phi(x,t) = \begin{bmatrix} sin(x_3+1) \\ sin(x_3+1) \\ 0 \\ 0 \\ sin(x_3+1) \\ 0 \\ 0 \end{bmatrix}$$

MATLAB Codes for calculating the state space inputs and the output transformations are in appendix C. They yield the input transformation matrix T and the following matrices T1, T2, T3, T4, A1, A2, A3, and A4, the output transformation matrix S and the following matrices S1, S2, S3, S4, C1, C2, C3, and C4.

6.3 The Matrices Needed in MATLAB-Ready Format

The system matrices (A, B, C, E, $\phi(x,t)$, $\Delta\psi(t)$, T, and S) above in previous sections are explained and calculated and reproduced below in the MATLAB-ready form.

A= [0, 0, 1, 0, 0, 0, 0; 0, -0.154, -0.04, 1.54, 0, -0.744, -0.032;

0, 0.249, -1, -5.2, 0, 0.337, -1.12; 0.0386, -0.996, 0, -2.117, 0, 0.02, 0; 0, 0.5, 0, 0, -4, 0, 0; 0, 0, 0, 0, 0, -20, 0; 0, 0, 0, 0, 0, 0, -25];

B= [0, 0; 0, 0; 0, 0; 0, 0; 0, 0; 7, 0; 0, 19];

C= [0, -0.451, -0.41, 1.55, 0, -0.44, -0.052; 0, 1, -1, -4.55, 0, 0.474, -1.14; 1, 0, 0, 0, 0, 0, 0; 0, 0, 0, 0, 1, 0, 0; 0, 0, 0, 0, 0, 0, 0; 0, 0, 0, 0, 0, 10, 0; 0, 0, 0, 0, 0, 0, 1];

E = [1; 1; 0; 0; 1; 0; 0];

$\phi(x,t)$ =[cos(x_3+1); cos(x_3+1); 0; 0; cos(x_3+1); 0; 0]; or

$\phi(x,t)$ =[sin(x_3+1); sin(x_3+1); 0; 0; sin(x_3+1); 0; 0];

$\Delta\psi(t)$ is chosen to be $\Delta\psi(t)$= 2*cos(t)

T= [0.9, 0.16, 0.04, -1.6, 0, 0.844, 0.03; -1, 1, 0, 0, 0, 0, 0;
0, 0, 1, 0, 0, 0, 0; 0, 0, 0, 1, 0, 0, 0; -1, 0, 0, 0, 1, 0, 0; 0, 0, 0, 0, 0, 1, 0;
0, 0, 0, 0, 0, 0, 1];

S=[1, 0, -1, 0; - 1, 1, -0.5, 0; 1, 0, 0, 0; 1, 0, -0.9, 1];

$f_{IM1} = e^{0.02*t}$

However, for all time periods less than 25 seconds, f_{IM1} faults are ignored due to possible electrical parameter value excursion in the electronic system.

$f_{IM2} = e^{0.045*t}$

However, for times less than 35 seconds, all f_{IM2} fault values are also ignored due to possible electrical parameter value excursion in the electronic system.

Depending on the number of faults monitored, detectable or critical faults are only monitored. The generic column fault expression is used below.

$f_{IMn} = e^{x*t}\ for\ t = time =\ t\ seconds$

The MATLAB code that calculates matrix products “T*A/T”; “S*C/T”; “T*B” and “T*E” is in appendix C and the results are given below.

The MATLAB matrix calculation of T*A/T result is in MATLAB form below.

```
(T*A)* inv(T)



ans =

    1.4313    1.3499    0.7963    5.7157    0.0000  -18.2256   -0.8429
   -0.1453   -0.1308   -1.0342    1.3075   -0.0000   -0.6214   -0.0276
    0.2349    0.2114   -1.0094   -4.8242    0.0000    0.1387   -1.1270
   -0.9032   -0.8515    0.0361   -3.5621   -0.0000    0.7823    0.0271
   -3.3019    1.0283   -0.8679   -5.2830   -4.0000    2.7868    0.0991
         0         0         0         0         0  -20.1100         0
         0         0         0         0         0         0  -25.2100
```

The MATLAB matrix calculation of S*C/T result is in MATLAB form below.

```
(S*C) * inv(T)

ans =

   -1.3689   -0.2320   -0.3552   -0.6402   -0.0000    0.7153   -0.0109
    0.8972    1.3075   -0.6259   -4.6645    0.0000    0.1568   -1.1149
   -0.4255   -0.3829   -0.3930    0.8692   -0.0000   -0.0809   -0.0392
   -0.3311   -0.3980   -0.3968    1.0202    1.0000   -0.1605   -0.0421
```

The MATLAB matrix calculation of T*B result is in MATLAB form below.

```
(T*B)

ans =

    5.9080    0.5700
         0         0
         0         0
         0         0
         0         0
    6.9110         0
         0   18.9990
```

The MATLAB matrix calculation of T*E result is in MATLAB form below.

(T*E)

ans =

[1.0600, 0, 0, 0, 0, 0, 0]^T

6.4 The Fault Research Validation and Results Analysis Explanation

The Fault Diagnosis in appendix D performs the Linear Matrix Inequality (LMI) parameter calculation necessary for all the necessary MATLAB codes and function codes to work. For instance, the resulting parameters after running the code in appendix D, are: O1, O2, F A1s L, alpha1, and gam. These parameters mentioned above are not as unique as they are resulting from Linear Matrix Inequality calculations. These parameters are used in the Simulink, whose block diagram is given in appendix D as well. The code that graphs these parameters is in Appendix E. Figures 6.1 to 6.6 support These resulting graphs demonstrate the dissertation's goal: to detect and isolate incipient faults.

It is worth summarizing the steps taken to arrive at the results illustrated in Figures 6.1 to 6.6. The summary below may serve as an executive summary for readers interested only in the essence of the dissertation.

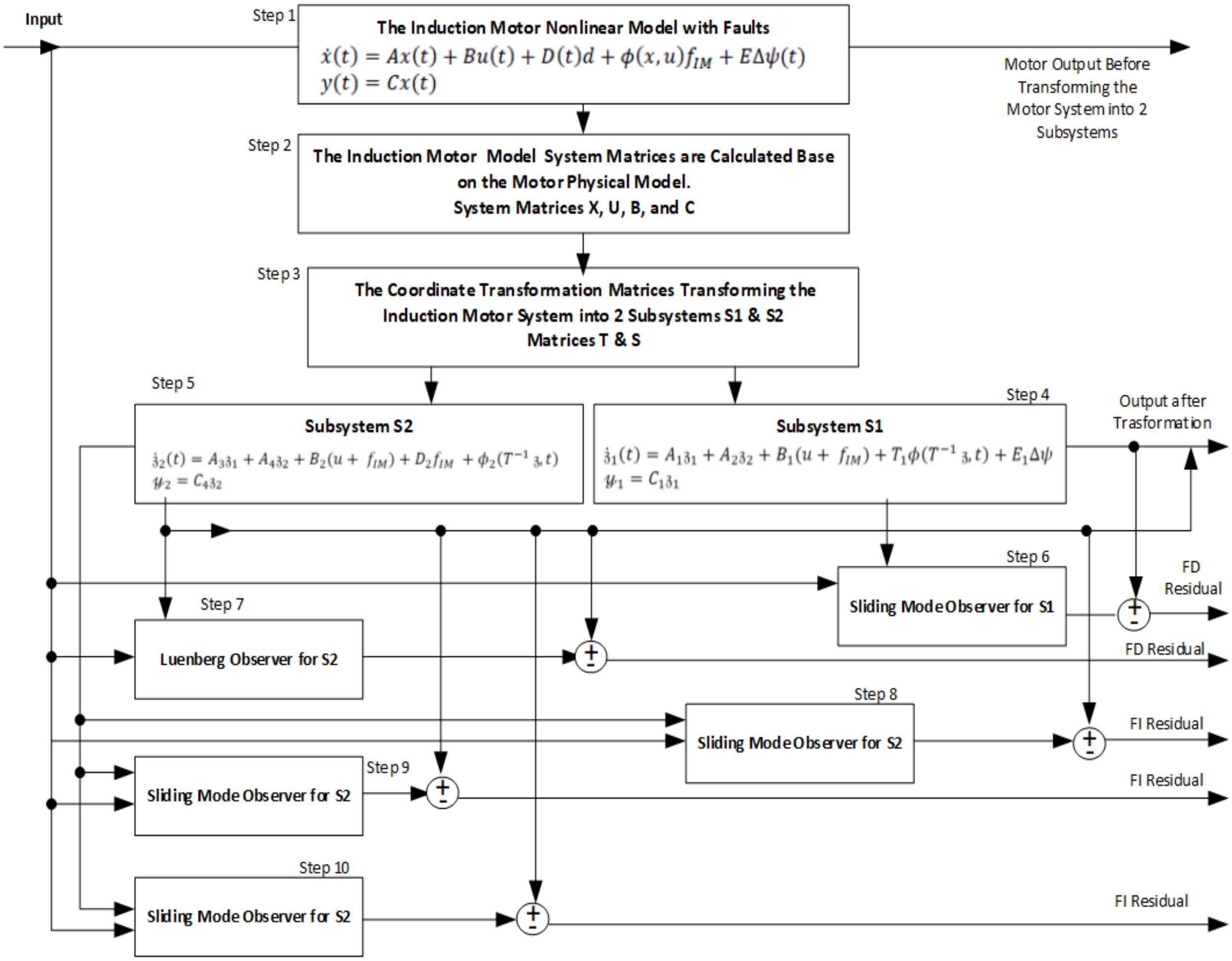

Input
Step 1
The Induction Motor Nonlinear Model with Faults
$\dot{x}(t) = Ax(t) + Bu(t) + D(t)d + \phi(x,u)f_{IM} + E\Delta\psi(t)$
$y(t) = Cx(t)$
Motor Output Before Transforming the Motor System into 2 Subsystems
Step 2
The Induction Motor Model System Matrices are Calculated Base on the Motor Physical Model.
System Matrices X, U, B, and C
Step 3
The Coordinate Transformation Matrices Transforming the Induction Motor System into 2 Subsystems S1 & S2
Matrices T & S
Step 5
Subsystem S2
$\dot{\mathfrak{z}}_2(t) = A_3\mathfrak{z}_1 + A_4\mathfrak{z}_2 + B_2(u + f_{IM}) + D_2f_{IM} + \phi_2(T^{-1}\mathfrak{z}, t)$
$y_2 = C_4\mathfrak{z}_2$
Step 4
Subsystem S1
$\dot{\mathfrak{z}}_1(t) = A_1\mathfrak{z}_1 + A_2\mathfrak{z}_2 + B_1(u + f_{IM}) + T_1\phi(T^{-1}\mathfrak{z}, t) + E_1\Delta\psi$
$y_1 = C_1\mathfrak{z}_1$
Output after Trasformation
Step 6
Sliding Mode Observer for S1
FD Residual
Step 7
Luenberg Observer for S2
FD Residual
Step 8
Sliding Mode Observer for S2
FI Residual
Step 9
Sliding Mode Observer for S2
FI Residual
Step 10
Sliding Mode Observer for S2
FI Residual

Figure 6.0 Major Steps Taken to Implement the Fault Diagnosis

The residuals are graphed and explained in Figures 6.1 to 6.5. The steps involved but not limited the following equations:

- Step 1: Refer to equations 93, 94, 162 and 163
- Step 2: Refer to equations 150 to 155; 157 to 161; 164 to 185
- Step 3: Refer to equations 106 to 109

- Steps 4-5: Refer to equations 110, 111d
- Step 6: Refer to equations 113,133
- Step 7: Refer to equations 115
- Steps 8-10: Refer to equations 136

Figures 6.1 to 6.6 show that depending on the threshold requirement for faults detection and isolating, the approach used in the dissertation detects and isolates the faults at the set threshold if the residual value crosses the set threshold at a particular time.

Once the fault is detected, if the Simulink simulation models more than one motor, the follow-up step is to identify which motor is faulty using the observers designed for fault isolation. The observer designed for the first fault has a corresponding residue analyzed with the threshold requirement for that motor fault. If the residual crosses the threshold line, it is noted at what time this has occurred. This event is illustrated in Figures 6.1 to 6.6. The same processes are implemented to isolate other faults in the motor models.

In Figure 6.1, the threshold value of the fault parameter is independent of time and is graphed in red at a magnitude of about 2.25; the residual from the fault model is graphed in blue. The picks of the residual never reach its set threshold at any time. Thus, no fault occurs, therefore, no fault is detected.

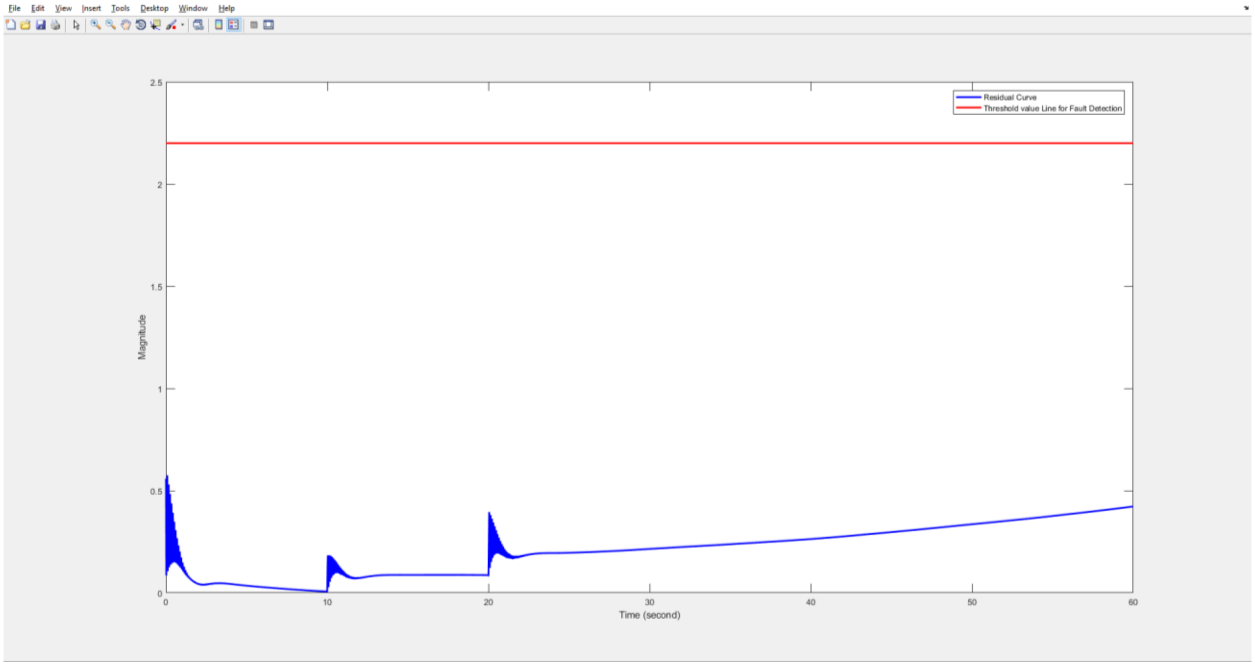


Figure 6.1 Fault Detection and Threshold1 vs Time

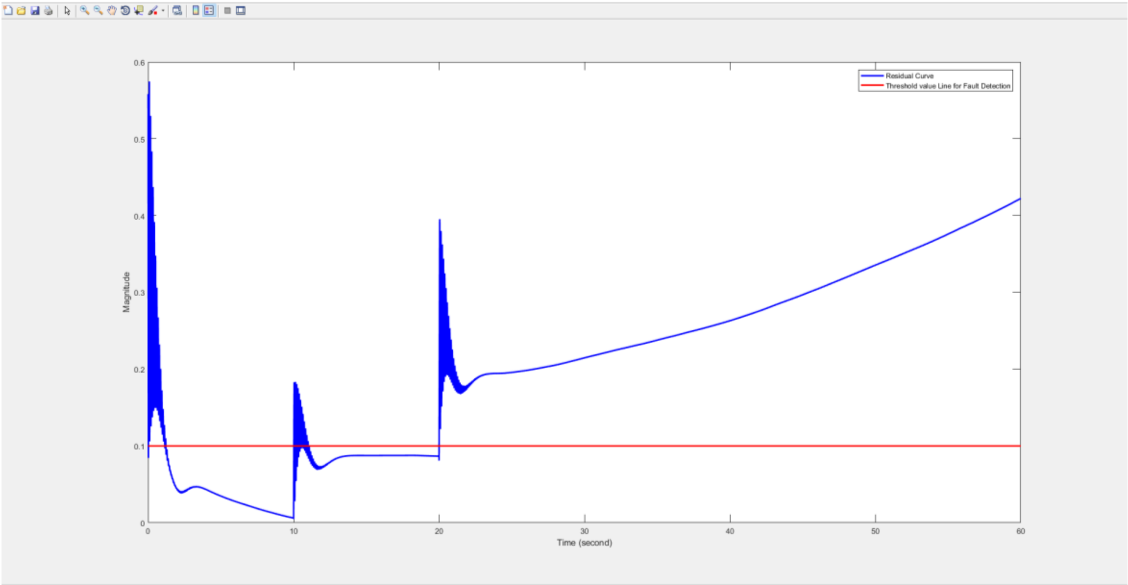


Figure 6.2 Fault Detection and Threshold2 vs Time

In Figure 6.2, the threshold value of the fault parameter is independent of time and is graphed in red at a magnitude of about 0.1; the residual from the fault model is graphed in blue. Three picks of the residual cross the set threshold at three different times. Thus, a fault is detected.

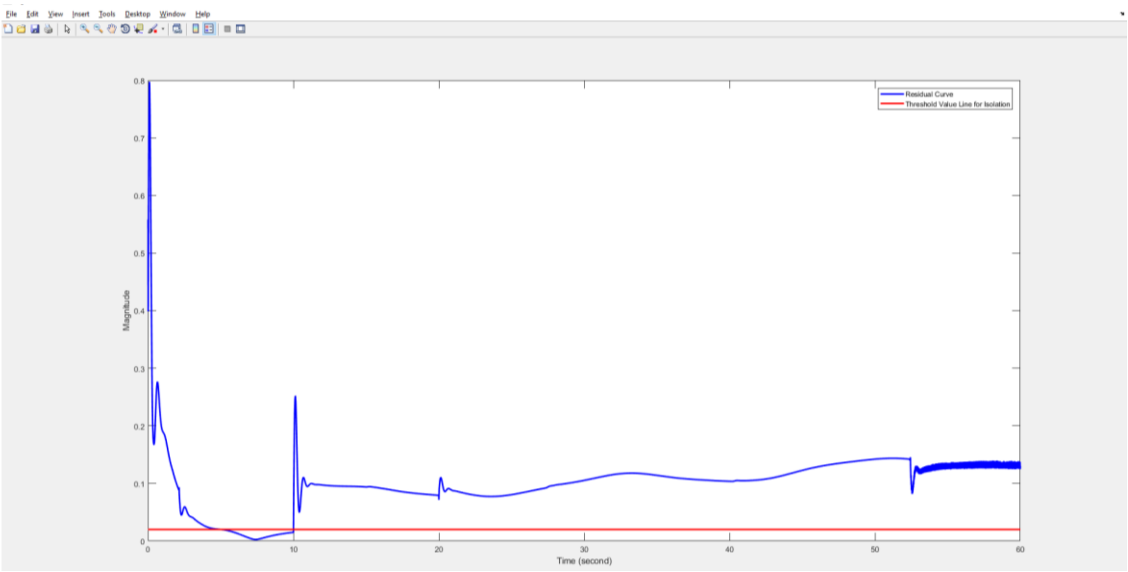


Figure 6.3 Fault 1 Fault Isolation and Threshold1 vs Time

In Figure 6.3 the fault isolation threshold value of the fault parameter is independent of time and is graphed in red is greater than zero. The residual from the fault model is graphed in blue. The first fault can be isolated.

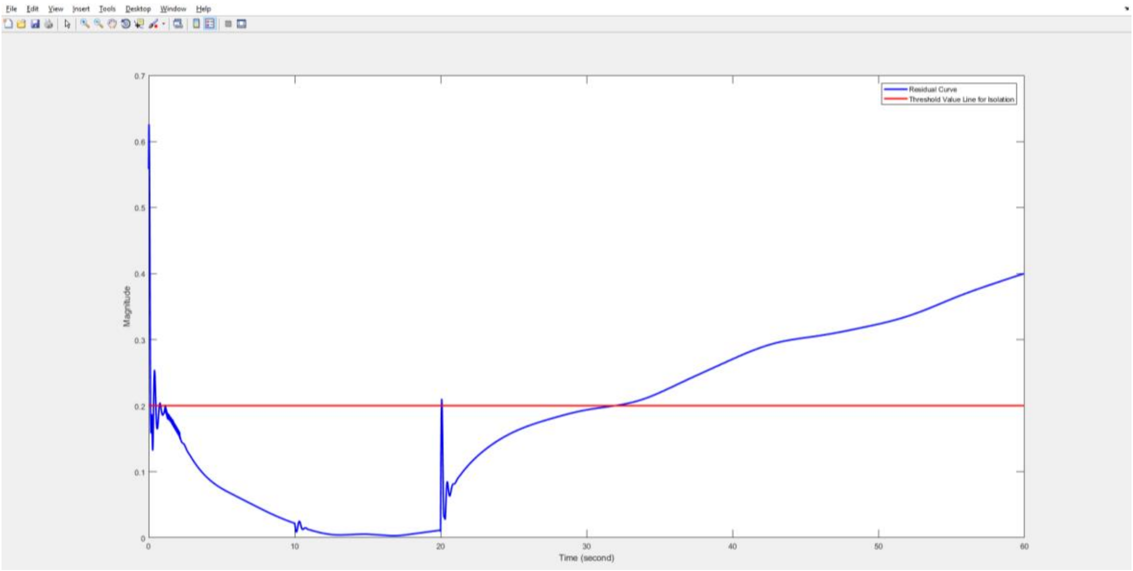


Figure 6.4 Fault 1 Fault Isolation and Threshold2 vs Time

Figure 6.4 shows the fault isolation threshold of the fault is independent of time and is graphed in red is greater than zero. The residual from the fault model is graphed in blue. The fault can be isolated at 20 and 30 seconds.

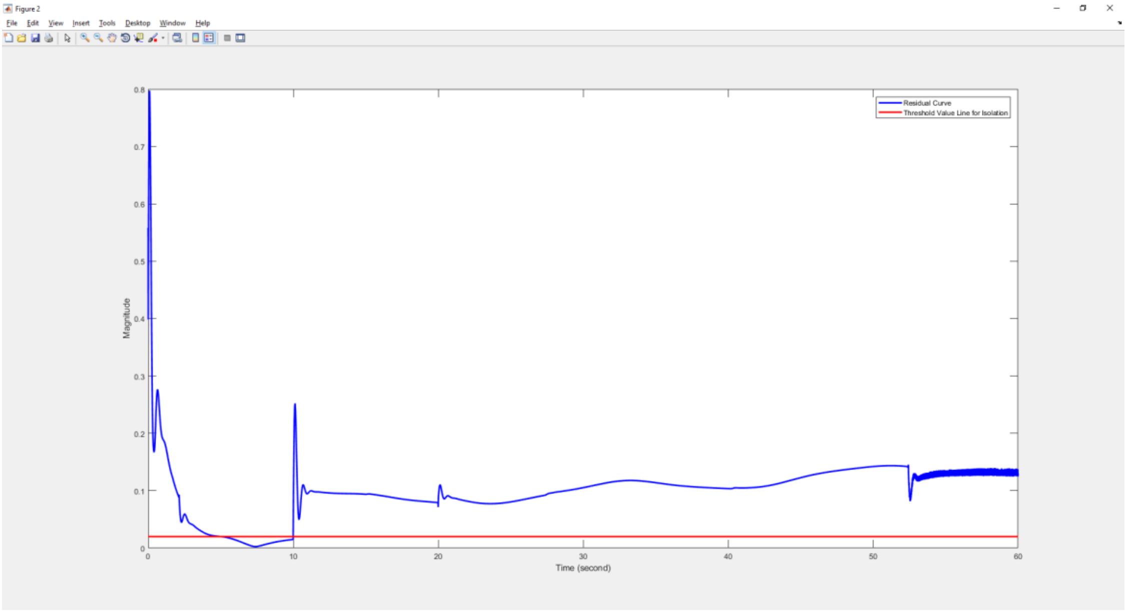


Figure 6.5 Fault 2 Fault Isolation and Threshold 1 vs Time

Figure 6.5 shows the fault isolation threshold of the fault is independent of time and is graphed in red is greater than zero. The residual from the fault model is graphed in blue. The fault can be isolated at 10 seconds.

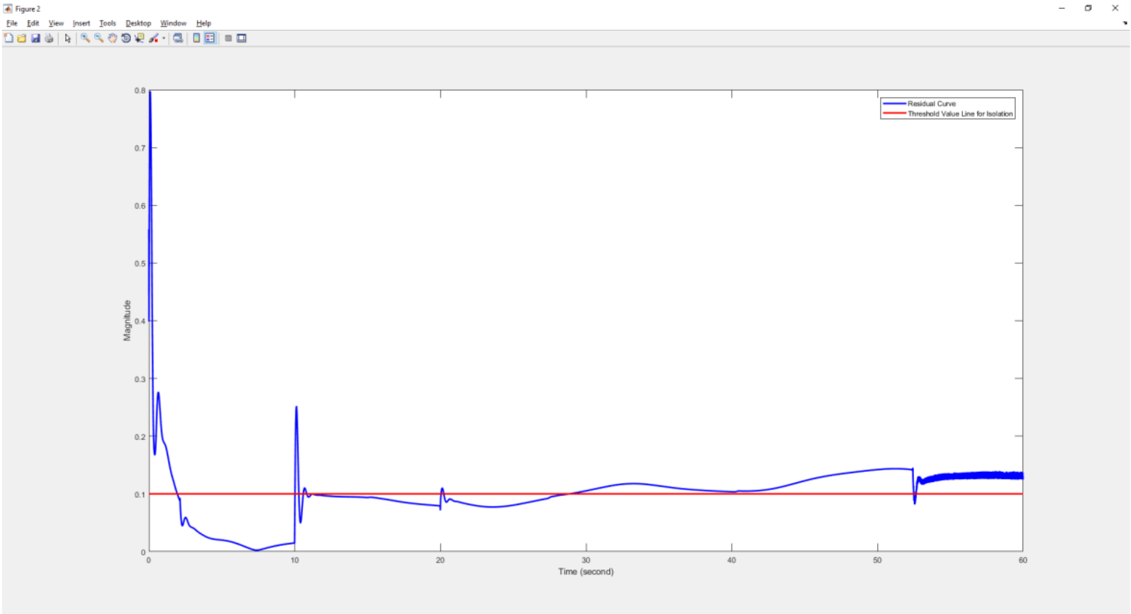


Figure 6.6 Fault 2 Fault Isolation and Threshold 2 vs Time

Figure 6.6 shows the fault isolation threshold of the fault is independent of time and is graphed in red is greater than zero. The residual from the fault model is graphed in blue. The fault can be isolated at 10, 20 and 30 seconds.

Chapter 7: Conclusion and Future Work

The "Objective, Motivation, State-of-the-Art and Background" in section 2.1 illustrates the author's objective and motivation for developing this method of fault diagnosis methodically. The State-of-the-Art reiterates and summarizes why the conventional methods used for classical fault diagnosis are unable to take on this challenge. Thus, since the author has been implementing the classical method using FMECA and Bayesian algorithm in the industry for many years and seen its performance, the need to develop an innovative hybrid method for fault diagnosis for the defense industry has become a necessity as stated throughout the dissertation. This method solves the challenge the classical approach cannot solve, and it has a wide field application that includes fault detection, fault isolation, fault accommodation, anti-tampering, system security, health monitoring, and prognosis in engineering systems. Furthermore, this approach serves to advance the Digital Transformation era through Model-Based System Engineering, Digital Twin and Hardware-in-the-Loop since the goals in the fault diagnosis align with those Digital Twin and Hardware-in-the-Loop and overlapping tasks may be leveraged.

The approach used in this dissertation to develop and implement an efficient fault diagnosis for incipient faults applies to abrupt and intermittent faults as well as to additive faults, multiplicative faults, uncertainties, disturbances, and noises. Parametric deviations such as discrepancies and suppress modeling errors are detectable. The induction motor system model is transformed into two systems: 1) real faults, discrepancies, disturbances, uncertainties and 2) those containing only the faults.

The fault diagnosis system's sensitivity and specificity are improved. This is done by separating the different types of faults, and; by transforming the induction motor state-space input matrix into a companion matrix and a transformation matrix. For readers who want to reproduce this work, all appropriated MATLAB codes are provided in the appendix. A well-versed reader in the topic will also note that the transformation matrix is not unique, and the induction motor state-space input companion matrix can be of the standard form.

Thus, the goal of this dissertation is technically achieved by designing fault diagnosis for systems that are susceptible to the diagnostic conditions named ReTest OK (RTOK) and Can Not Duplicate (CND) when tested. These conditions increase system life cycle costs. Designers of these systems are interested in the approach developed here as it reduces the number of these conditions.

The relevance of this dissertation to the aerospace industry is enormous. However, the Model-Based Fault Diagnosis in aerospace industry is still at its infancy. In this dissertation, only fault detection and fault isolation of a system are investigated and can be extended to any system. The author plans to further investigate model-based fault diagnosis in the aerospace industry to include:

- Fault estimation modeling
- Hardware like sensors into the system
- Estimation of system and system sensor faults simultaneously

The inclusion of sensors will include sensor fault detection and fault isolation as faults will also occur in sensors. In that situation the model-based fault diagnosis, the observers design are more complicated. In this situation, the observer is implemented using the following signals:

- Sensor Output
- System Excitation Signal
- System Model
- Model Sensor
- Proportional Integral-Proportional Integral Derivative (PI-PID) Compensator

References

[1] Electrical Machines, Drives and Power Systems, 6th Edition by Theodore Wildi.

[2] Principle of Electric Machines and Power Electronics, Second Edition by P.C. Sen, Professor of Electrical Engineering, Queens University Ontario Canada.

[3] Analysis of electric machinery, by Krause, P.C., McGraw-Hill book company, 1986.

[4] Modern Signals and Systems, by Kwakernaak, H., R. Sivan, Prentice-Hall, 1991.

[5] Control of electrical drives, by Leonhard, W., Springer-Verlag, 1990.

[6] Controlled AC drives, a successful transition from ideas to industrial practice, by Leonhard, W., Control Engineering Practice, Vol.4, No.7, 1996.

[7] Performance deterioration in indirect vector-controlled induction motor drives due to iron losses, by Levi, E., A. Boglietti, M. Lazzari, Proc.IEEE-PESC'95, Atlanta, 1995.

[8] Stator parameters influence of the field-oriented control tuning, by Loran, L., Proc. EPE'93, Brighton, 1993.

[9] Alternating Current Machines, by Say, M. G, Pitman Publishing, fourth edition, 1976.

[10] Electric Machines and drives' Addison-Wesley, by Siemon, G. R., 1992.

[11] Models for induction machines with magnetic saturation of the main flux path, by Sullivan, C.R., S. R. Sanders IEEE Trans. on industry applications, vol 31, no.4, 1995.

[12] Nonlinear field-oriented control in a rotor reference frame, by Coussens, P.J., A. P. Van den Bossche, J. A. Melkebeek, Proc. EPE Sevilla, 1995.

[13] The universal field-oriented controller, Doncker, R.W. De, D. Novotny, IEEE Trans. on Power Electronics, Vol.10, No.2, 1995.

[14] Field-oriented controllers with rotor deep bar compensation circuits, by Doncker, R.W. De, IEEE on industry applications, vol.28, no.5., 1992.

[15] Sensored Field Oriented Control of 3-Phase Induction Motors by Bilal Akin and Manish Bhardwaj.

[16] Applications of Monte Carlo Method in Science and Engineering by Edited by Shlomo Mark and Shaul Mordechai.

[17] Applied Matrix Algebra in the Statistical Science, by Basilevsky, A. (1983). North-Holland, New York.

[18] Numerical Analysis, ninth edition, by Richard L. Burden and J. Douglas Faires.

[19] Statistical Validation of Engineering and Scientific Models by Richard G. Hills Department of Mechanical Engineering New Mexico State University Las Cruces, New Mexico 88003.

[20] Kalman Filtering with Real-Time Applications by C. K. Chui and G. Chen. Springer Series in Information Sciences. Springer-Verlag, 2nd edition (1991).

[21] Elementary differential equations and boundary values problems, by Boyce, W.E., R.C. Diprima, John Wiley and Sons, New York, 1986.

[22] Decision with Bayesian network in the concurrent faults event by Sylvain Verron, Philippe Weber, Didier Theilliol, Teodor Tiplica, Abdessamad Kobi, et al., 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess'09), 2009, Barcelona, Spain, 2009.

[23] The Bayesian Choice - From Decision-Theoretic Foundations to Computational Implementation, by Christian P. Robert, Second Edition, Springer, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris cedex 16, France.

[24] Bayesian Core: The Complete Solution Manual, by Christian P. Robert and Jean-Michel Marin, Université Paris Dauphine, and CREST, INSEE, Paris, & Institut de et Modélisation de Montpellier, Université Montpellier 2, and CREST, INSEE, Paris.

[25] Bayesian Data Analysis, by Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari Donald B. Rubin, Third Edition, Chapman & Hall/CRC.

[26] Bayesian Filtering and Smoothing, by Simo Sarkka, Cambridge University Press.

[27] Bayesian Artificial Intelligence, by Kevin B. Korb, Ann E. Nicholson, Second Edition, Chapman & Hall/CRC.

[28] Accuracy issues for parameter estimation of field-oriented induction machine drives, by Borgard, D.E., G. Olsson, R. D. Lorenz, Proc. IEEE-IAS, 1994

[29] Robust Parameter and Signal Estimation in Induction Motors, by Borsting, H., PhD-Thesis, Aalborg University, Alborg, 1993.

[30] Estimation of physical parameters in induction motors, by Borsting,H., M. Knudsen, H. Rasmussen, P. Vadstrup Proc. SYSID'94, Copenhagen, 1994.

[31] Borsting, H., M. Knudsen, P. Vadstrup, "Standstill estimation of electrical parameters in induction motors using an optimal input signal," Proc. EPE, Sevilla, 1995.

[32] Estimation of complex parameters, by Bos, A. van den, Proc. SYSID'94, Copenhagen, 1994.

[33] Modelling, identification and simulation of dynamical systems, by Bosch, P.P.J. van den, A.C. van der Klauw, CRC Press Inc., 1994.

[34] Induction motor field-oriented control: a different viewpoint, by Bossche, A. P. M. van den, J. A. A. Melkebeek, Benelux Quarterly Journal on Automatic Control, Vol 31, no.3, 1990.

[35] Analysis and realization of a pulse width modulator based on voltage space vectors, by Broeck, H.W. van den, H. Skudelny, G. Stanke, Proc. IEEE-IAS, 1986.

[36] Off line parameter identification of an inverter-fed induction motor at standstill, by Biinte, A., H. Grotstollen, Proc. EPE'95, Sevilla, 1995.

[37] The Voltage/Current Model in Field-Oriented AC Drives at Very Low Flux Frequencies, PhD-Thesis, by Burgt, J.J.A. van der, Eindhoven University of Technology, 1996.

[38] Induction motor parameter estimation algorithm using spectral analysis, by Chai, H., P. P. Acarnley, lEEE Proc.-B, Vol.139, No.3, 1992.

[39] Robust recursive estimation of states and parameters of bilinear systems, by Dai, H., N. K. Sinha, Academic Press, Control and Dynamic Systems, vol.54, 1992.

[40] Parameter sensitivity of indirect universal field-oriented controllers, by Doncker, R.W. De, Proc. IEEE, Boston, 1991.

[41] Identification of continuous time systems- an application of Ljung’s corrected extended Kalman filter, by Gavel, D.T., S. G. Azevedo, Sixth IFAC symposium on Identification and Parameter estimation, Washington D.C., 1982.

[42] Recursive forgetting algorithms, by Parkum, J.E., N. Poulsen, J. Holst, Int. J. of Control, Vol. 55, No. 1, 1992.

[43] Robust recursive estimation of states and parameters of bilinear systems by Dai, H. and Sinha. N.K.

[44] Generalized Binary Noise Test signal concept for improved Identification Experiment Design, by Tulleken, H.J.A.F., Automatica, vol. 26, no. l, 1990.

[45] Selective forgetting in adaptive procedures, by Parkum, N. K. Poulsen, J. Holst, IFAC 11th Triennial World Congress, Vol. 2, 1990.

[46] IEEE Standard Test Procedure for Polyphase Induction Motors and Generators, IEEE Std 112™-2004 (Revision of IEEE Std 112-1996)

[47] Skin effect impact on induction motor parameters estimation using an output error identification method, by Gorter, R.J.A., A. Veltman, P. P. J. van den Bosch, Proc. IEEE-PESC 1994.

[48] Determination of physical parameters for dynamical processes, by Isermann, R., B. Freyermuth, A. Shumann, Proc. IFAC Conference on identification and system parameter estimation, Budapest, 1991.

[49] Application of the extended Kalman filter to parameters estimation of induction motors, by Loran, L., G. Laliberte, Proc. EPE'93, Brighton, 1993.

[50] Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems, by Ljung, L., IEEE Trans. on automatic control, Vol.24, no. l, 1979.

[51] Analysis of General Recursive Prediction Error Identification Algorithm, by Ljung, L., Automatica, Vol.17, no. l, 1981.

[52] System Identification Toolbox-User’s Guide, by Ljung, L., The Mathwork company, 1991.

[53] Matlab Users Guide, The Mathwork company, 1991.

[54] Parameter identification of an induction machine with regard to dependencies on saturation, by Klaes, N. R., Proc. IEEE-IAS'91, 1991.

[55] A sensitivity approach for estimation of physical parameters, Knudsen, K., Proc. SYSID'94, Copenhagen, 1994.

[56] Design and realization of a real-time environment on a DSP- PC system, Master-Thesis, by Mathijssen, S. G. A., Eindhoven University of Technology, Eindhoven, 1995.

[57] Digital Estimation and Control: A Unified Approach by Middleton, R. H., G.C. Goodwin, Prentice-Hall, 1990.

[58] Recursive identification algorithms, for parameter estimation of induction machines, Master-Thesis by Minderhout, M.G.F., Eindhoven University of Technology, Eindhoven, 1996.

[59] Parameter Identification of induction motor drives, by Moons, C., B. de Moor, Automatica, Vol. 31, 1995.

[60] The relationship between discrete time and continuous time linear estimation, by Ninness, B.M., G. C. Goodwin, Kluwer Academic Publisher, 1991.

[61] Vector Control of Induction Motor without shaft encoder, by Ohtani, T., N. Takada, K. Tanaka, IEEE Trans. on industry applications, volume 28, no.1, 1992.

[62] Selective forgetting in adaptive procedures, by Parkum, J.E., N. K. Poulsen, J. Holst IFAC 11th Triennial World Congress, Vol. 2, 1990.

[63] Recursive forgetting algorithms, by Parkum, J.E., N. Poulsen, J. Holst, Int. J. of Control, Vol. 55, No. 1, 1992.

[64] Inverter and motor model adaptation at stand-still using reference voltages and measured currents, by Rasmussen, H., M. Tonnes, M. Knudsen, Proc. EPE'95, Sevilla, 1995.

[65] Parameters and speed estimation for induction machines based on dynamic models, by Ribeiro, L.A.S., C. B. Jacobina, Proc. EPE', Sevilla, 1995.

[66] Modified least squares algorithm incorporating exponential resetting and forgetting, by Salgado, M. G. C. Goodwin, R. H. Middleton, International Journal of Control Vol.47, no.2, 1988.

[67] Adaptive Speed identification for vector control of induction motors without rotational transducers, by Schauder, C., IEEE Trans. on industry applications, vol.28, no.5, 1992.

[68] System identification, by Soderstrom, T., P. Stoica, Prentice Hall, 1989.

[69] Generalized Binary Noise Test signal concept for improved Identification Experiment Design, by Tulleken, H.J.A.F., Automatica, vol.26, no. l, 1990.

[70] Grey- box Modelling and Identification Topics, PhD-Thesis, by Tulleken, H.J.A.F., Delft University of Technology, Delft, 1992.

[71] Parameter Estimation, Condition Monitoring and Diagnosis of Electrical Machines, by Vas, P., Oxford Science Publications, 1993.

[72] An extended Kalman filter approach to rotor time constant measurement in PWM induction motor drives, by Zai, L., C. DeMacro, T. Lipo, IEEE Trans. on industry applications, Vol.28, no. l, 1992.

[73] Identification of multivariable industrial processes, for simulation, diagnosis and control, by Zhu, Y., T. Backx, Springer-Verlag, 1993.

[74] Parametric Uncertainty in System Identification, by Heinz Manfred Falkus, geboren te Oldenburg, ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven.

[75] Grey-box Modeling of a Small-scale Helicopter Using Physical Knowledge and Bayesian Techniques, by Fang Zhou, Li Ping Institute of Industrial Process Control Zhejiang University, 2008 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam.

[76] Linear State-Space Control Systems, by Robert L. Williams II, Douglas A. Lawrence, Ohio University, John Wiley & Sons, Inc.

[77] Fault Detection and Diagnosis in Engineering System by Janos J. Gertler.

[78] Parity space based FDI-scheme for vehicle lateral dynamics, Proc. of the IEEE International Conference on Control Applications, Toronto, 2005 by S. Schneider, N. Weinhold, S. X. Ding, and A. Rehm.

[79] An introduction to sensor signal validation in redundant measurement systems. IEEE Contr. Syst. Mag. 11(2), 44-49 (1991) by A. Ray and R. Luck.

[80] Generalized likelihood test for fdi in redundancy by K. C. Daly, E. Gai, and J. V. Harrison.

[81] A fault detection and isolation methodology – theory and application, by M. Desai and A. Ray, Proc. 1984 Amer. Contr. Conf. San Diego, USA (1984)

[82] Model-Based Fault Diagnosis Technique: Design schemes, algorithms and tools by S. X. Ding. Springer, 2008.

[83] Survey of model-based failure detection and isolation in complex plants, by J. J. Gertler, IEEE Control Systems Magazine, December 1988.

[84] A survey of design methods for failure detection in dynamic systems, by E. Y. Chow and A. S. Willsky, NASA STI/Recon Technical Report N, vol. 76, 1975.

[85] Analytical redundancy and the design of robust failure detection systems," IEEE Transactions on Automatic Control, October 1982.

[86] Fault Detection and Diagnosis in industrial Systems, by L. H. Chiang, E. Russell, and R. D. Braatz, Springer-Verlag, February 2001.

[87] Data-Driven Methods for Fault Detection and Diagnosis in Chemical Processes, by E. Russell, L. H. Chiang, and R. D. Braatz, (Advances in Industrial Control). Springer-Verlag.

[88] Detection of Abrupt Changes: Theory and Application. By 2000, M. Basseville and I. V. Nikiforov, Prentice-Hall, Inc, April 1993.

[89] Computer controlled systems, by Astrom, B. Wittenmark Prentice Hall, 1984.

[90] Observers for induction motor state and parameter estimation, by Atkinson, D.J., P. P. Acarnley, J. W. Finch, IEEE Trans. on industry applications, Vol.27, No.6, 1991.

[91] Simultaneous generation of sensitivity functions, Transfer function matrix approach, by Bingulac, S., J. H. Chow, J. R. Winkelman, Automatica, Vol.24, No.2, 1988.

[92] Control of AC machines, by Blaschke, F., A. Vandenput, Eindhoven University of Technology, EDT-report 96-E-296, 1996.

[93] Practical application of second order observers to electromechanical servo systems, by Geerts, J., J. Scannell, Journal on Automatic Control, Vol 31, no.3, 1990.

[94] Parametric Uncertainty in System Identification, by Falkus, H.M., PhD-Thesis, Eindhoven University of Technology, Eindhoven, 1994.

[95] An FFT-based novel approach to noninvasive speed measurement in induction motor drives, by Ferrah, A., K. J. Bradley, G. M. Asher, IEEE Trans. on instrumentation and measurement, vol. 41.

[96] Flux reconstruction, paper design, by Gils, P. van., Eindhoven University of Technology, Eindhoven,1995.

[97] Modelling and Control of a HERA joint, by Gorter, R.J.A., A. van Swieten, A. Veltman, P. P. J. van den Bosch, Proc. 12 IFAC symposium on automatic control in aerospace, Ottobrunn, 1992

[98] Observer-Based Field Orientation for both Zero and Very High-Speed Operation, by Jansen, P., C. Thompson, R. Lorenz, Proc. Power conversion conference, Yoko-hama, 1993, pp.432-437.

[99] A physically insightful approach to the design and accuracy assessment of flux observers for field-oriented induction machine drives, by Jansen, P., R. Lorenz, D. Novotny, IEEE Trans. on industrial applications, Vol.30, No.1, 1994.

[100] System Dynamics: A Unified Approach, Karnopp, D.C., D. L. Margolis, R. C. Rosenberg, Wiley & Sons, New York, 1990.

[101] High resolution velocity estimation for all digital AC servo drives, by Lorenz, R.D., K van Patten, Proc.IEEE-IAS'88, 1988.

[102] A simplified approach to continuous on-line tuning of field-oriented induction machine drives, by Lorenz, R.D., D. B. Lawson, IEEE Trans. on industry applications, vol.26, no.3, 1990.

[103] Future trends in power electronic controls of drive: robust, zero speed sensor less control and new standard approaches for field orientation, by Lorenz, R., Proc. International Power Electronics Conference IPEC'95, Yokohama, Vol. 1, 1995.

[104] Robust control of an induction machine, Master-Thesis by Uden, R. van, Eindhoven University of Technology, Eindhoven, 1995.

[105] Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques, by Silvio Simani, Cesare Fantuzzi and Ron J. Patton, Springer-Verlag Berlin Heidelberg New York, London Paris Tokyo, Hong Kong Barcelona, Budapest.

[106] Observers in Control Systems A Practical Guide, by George Ellis, Danaher Corporation, Academic Press, an imprint of Elsevier Science.

[107] Prognostic Algorithms for Condition Monitoring and Remaining Useful Life Estimation, by Shane Butler, A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy in the Faculty of Science and Engineering Department of Electronic Engineering, National University of Ireland, Maynooth.

[108] Residual Generation for Fault Diagnosis, by Erik Frisk, Linkoping Studies in Science and Technology. Dissertations No. 716.

[109] Robust Fault Diagnosis in Linear and Nonlinear Systems Based On Unknown Input, and Sliding Mode Functional Observer Methodologies, by Yi Xiong, a Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in the School of Engineering Science, Simon Fraser University.

[110] TI Spins Motors - Motor Drive and Control Solutions, by Texas Instruments, WWW.ti.com/motor.

[111] High Voltage Digital Motor Control Kit (R1.1) Quick Start Guide, by Texas Instruments, WWW.ti.com/motor.

[112] High Voltage Digital Motor Control Kit Quick Start Guide, by Texas Instruments, WWW.ti.com/motor.

[113] Scalar (V/f) Control of 3-Phase Induction Motors, by Texas Instruments, WWW.ti.com/motor.

[114] Sensored Field Oriented Control of 3-Phase Induction Motors, by Bilal Akin and Manish Bhardwaj, Texas Instruments, WWW.ti.com/motor.

[115] Sensorless Field Oriented Control of 3-Phase Induction Motors Using Control Law Accelerator (CLA), Manish Bhardwaj, Texas Instruments, WWW.ti.com/motor.

[116] Current Shunt Monitors, by Texas Instruments, WWW.ti.com/motor.

[117] Industrial Motor Drive Solution Guide, Industrial Automation Solution, by Texas Instruments, WWW.ti.com/motor.

[118] Digital Signal Controllers (DSCs), Data Manual, by Texas Instruments, WWW.ti.com/motor Literature Number: SPRS439M June 2007– Revised August 2012.

[119] Observers in Control Systems a Practical Guide, by George Ellis, Danaher Corporation, Academic Press.

[120] Parameter Estimation, Condition Monitoring, and Diagnosis of Electrical Machines, by Peter Vas, Department of Engineering, University of Aberdeen, Clarendon Press Oxford,.1993.

[121] Induction motor fault detection and isolation through unknown input observer, by Khalaf Salloum Gaeid and Hew Wooi Ping, Scientific Research and Essays Vol. 5(20), October 2010.

[122] Simulation and fault detection of three-phase induction motors, by B. Liang a, B.S. Payne, A. D. Ball b, S. D. Iwnicki, Department of Engineering and Technology, Manchester Metropolitan University, Manchester M1 5GD, UK School of Engineering, University of Manchester, Manchester M13 9PL, UK.

[123] Nonlinear Control Systems, by Alberto Isidori, Third Edition, Communications and Control Engineering, Springer.

[124] Nonlinear Systems, by Hassan K. Khalil, Third Edition, Prentice Hall 2002.

[125] Nonlinear Systems Analysis, by M. Vidyasagar, Second Edition, Prentice Hall, Englewood OHTs. New Jersey 07632

[126] Robust Residual Generation for Model-Based Fault Diagnosis of Dynamic Systems, by Jie Chen, Thesis Submitted in accordance with the requirements of the award of the Ph.D. Degree, University of York Department of Electronics, 1995.

[127] Fault Detection and Isolation in a Nuclear Reactor, by Asok Ray, Mukund Desai, and John Deyst, the Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

[128] Data Analysis Using Regression and Multilevel/Hierarchical Models, by Andrew Gelman, Jennifer Hill, Columbia University, Cambridge University Press.

[129] DOD Digital Modernization Strategy - DOD Information Resource Management Strategic Plan Y19-23.

[130] Feedback Systems - An Introduction for Scientists and Engineers, by Karl Johan Astrom, Automatic Control LTH, Lund University - Control, Dynamical Systems and Computation, University of California Santa Barbara - Richard M. Murray, Control and Dynamical Systems, California Institute of Technology Lyapunov Stability.

[131] NASA Systems Engineering Handbook, National Aeronautics and Space Administration NASA Headquarters Washington, D.C. 20546, December 2007 - NASA/SP-2007-6105 Rev1.

[132] Matrix Computations, 4th Edition, by Gene H. Golub, Department of Computer Science Stanford University, Charles F. Van Loan, Department of Computer Science Cornell University - The Johns Hopkins University Press.

[133] Introduction to Dynamic Systems, Theory, Models and Applications, By David G. Luenberd, Sandford University, John Wiley & Sons

[134] Numerical Recipes, The Art of Scientific Computing, Third Edition, by William H. Press Raymer Chair in Computer Sciences and Integrative Biology the University of Texas at Austin, Saul A. Teukolsky Hans A. Bethe Professor of Physics and Astrophysics Cornell University William.

[135] Fault-Diagnosis Applications - Model-Based Condition Monitoring:Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems, by Rolf Isermann – Springer.

[136] Detection Theory: A User's Guide, by Neel A. Macmillan University of Massachusetts And C. Douglas Creelman University of Toronto - Lawrence Erlbaum Associates, Publishers 2005 Mahwah, New Jersey.

[137] Linear Matrix Inequalities in System and Control Theory, by Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan, Society for Industrial and Applied Mathematics, SIAM Studies in Applied Mathematics Philadelphia.

[138] Inverse Problem Theory and Methods for Model Parameter Estimation, by Albert Tarantola, Institut de Physique du Globe de Paris Université de Paris 6 Paris, France - SIAM Society for Industrial and Applied Mathematics Philadelphia.

[139] Lipschitz Functions, Lorianne Ricco February 4, 2004

[140] Robust Model Based Fault Diagnosis for Dynamic Systems, by Chen, J. and Patton, R. J., Kluwer Academic, 1999

[141] Solving Applied Mathematical Problems with MATLAB, by Dingyü Xue, Yang Quan Chen

[142] System Testability Threshold Design Effectiveness Via Signal Detection Theory – Published in IEEE. Authors: Ken Pierre, PhD. Candidate, Dept. Electrical Engineering, Tampa, Florida, Dr. Wilfrido Moreno, PE and Dr. Chung Seop Jeong.

[143] Sliding mode observers for fault detection and isolation – Published in Automatica. Authors: C. Edwards, S. K. Spurgeon, and R. J. Patton.

[144] Simulink / MATLAB Dynamic Induction Motor Model for Use as A Teaching and Research Tool - International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-4, September 2013 – Author: Alec W. Leedy

[145] CBMS Conference on Fast Direct Solver, Dartmouth College, June 23 – June 27, 2014, Lecture 4: Structured Matrix Computations, Gunnar Martinsson The University of Colorado at Boulder – Research Support by DARPA and NSF

Appendix A: Experiment Setup, MATLAB Codes and Results

The equipment used in the experiment is listed below.

1) The Induction Motor Described in the Dissertation; 1a: Encoder in Motor
2) Clamp that locks up the motor shaft for the locked rotor experiment
3) TI High Voltage Digital Motor Control (DMC)
4) A Digital Meter
5) A Digital Meter
6) Variac Transformer (0 to 130 VAC)
7) Power Meter
8) The Dedicated HP Computer for the Experiment
9) Heater, 10) Computer, 11) Bookcase

The Heater 9, the Computer 10, and the Bookcase 11 are not related Experiment.

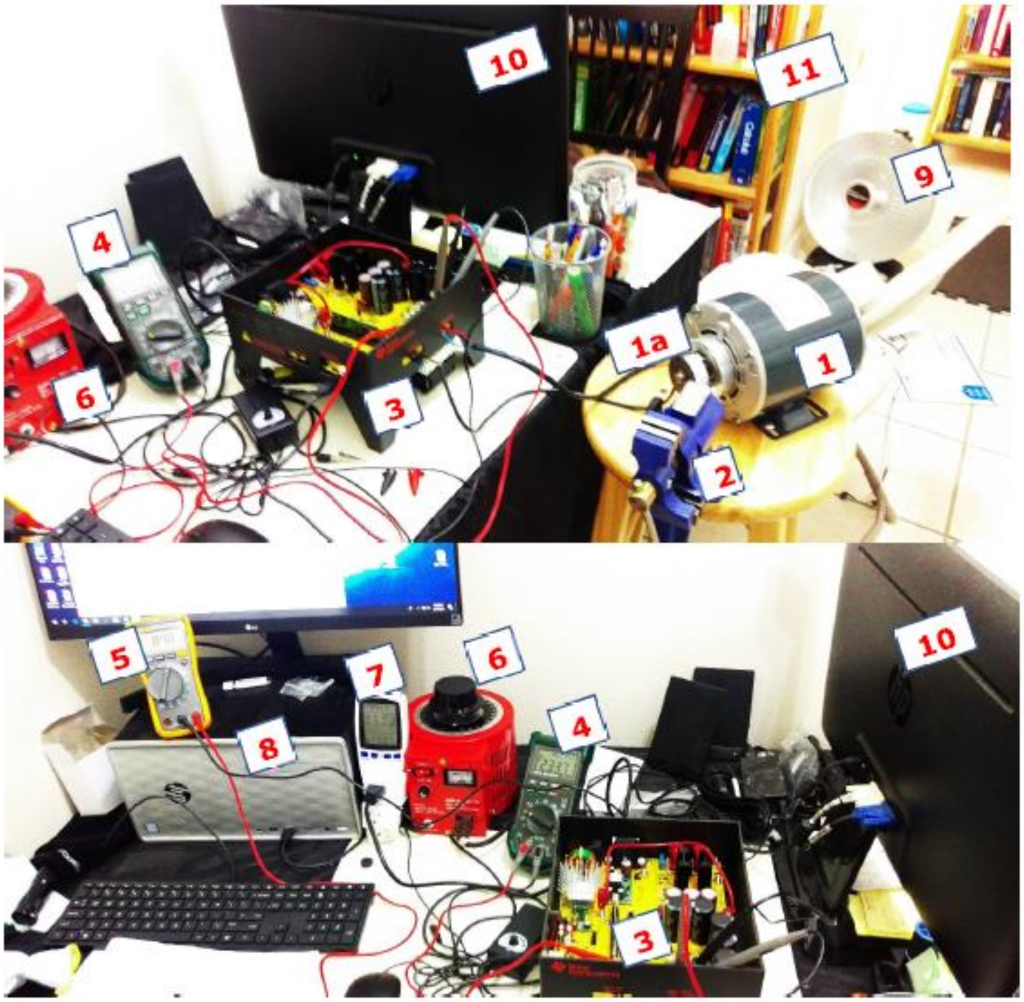


Figure A.1 The Home Lab Experiment Set Up

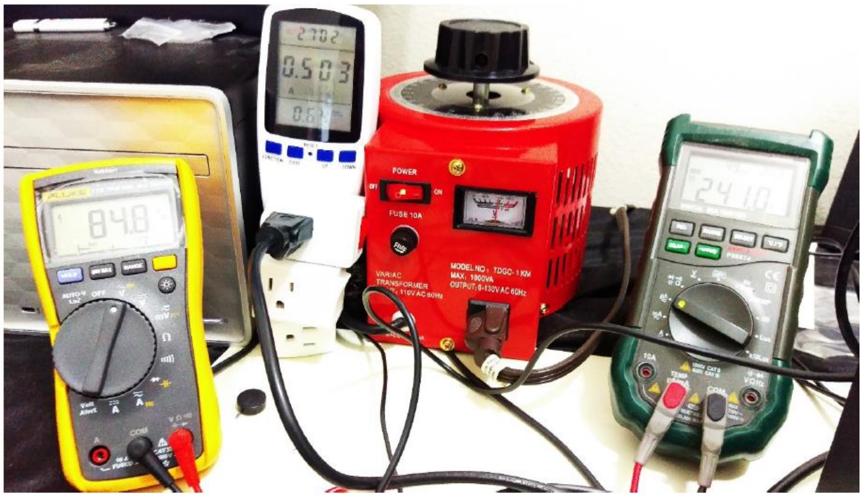
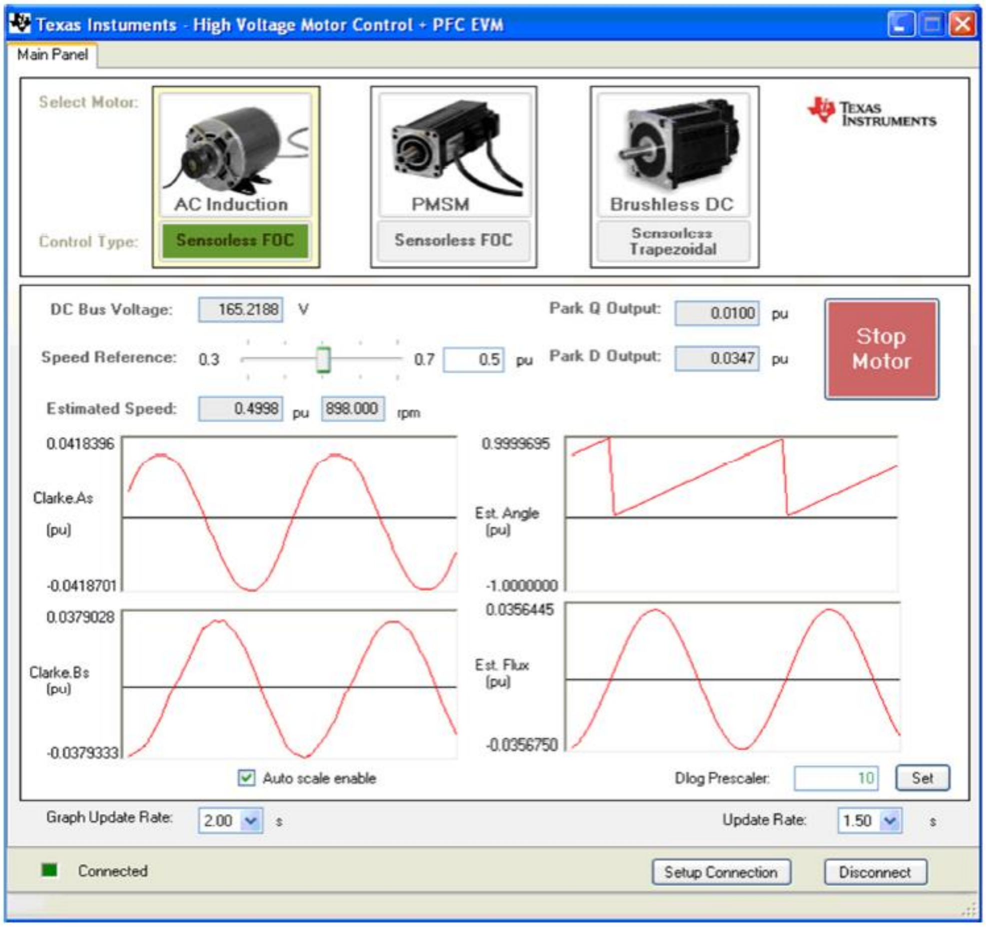


Figure A.2 A Closer View at Some Experiment in the Home Lab Set Up



Texas Instuments - High Voltage Motor Control + PFC EVM

Main Panel

Select Motor: AC Induction | PMSM | Brushless DC

Control Type: Sensorless FOC | Sensorless FOC | Sensorless Trapezoidal

DC Bus Voltage: 165.2188 V

Speed Reference: 0.3 — 0.7 0.5 pu

Estimated Speed: 0.4998 pu 898.000 rpm

Park Q Output: 0.0100 pu

Park D Output: 0.0347 pu

☑ Auto scale enable

Dlog Prescaler: 10

Graph Update Rate: 2.00 s

Update Rate: 1.50 s

🟩 Connected

Figure A.3 The TI GUI Showing a Few Results

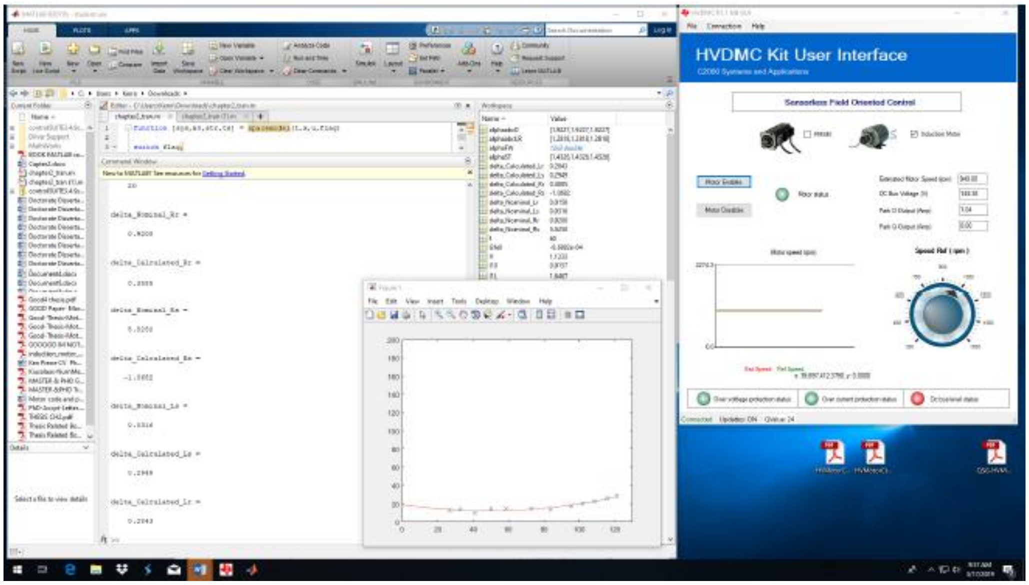
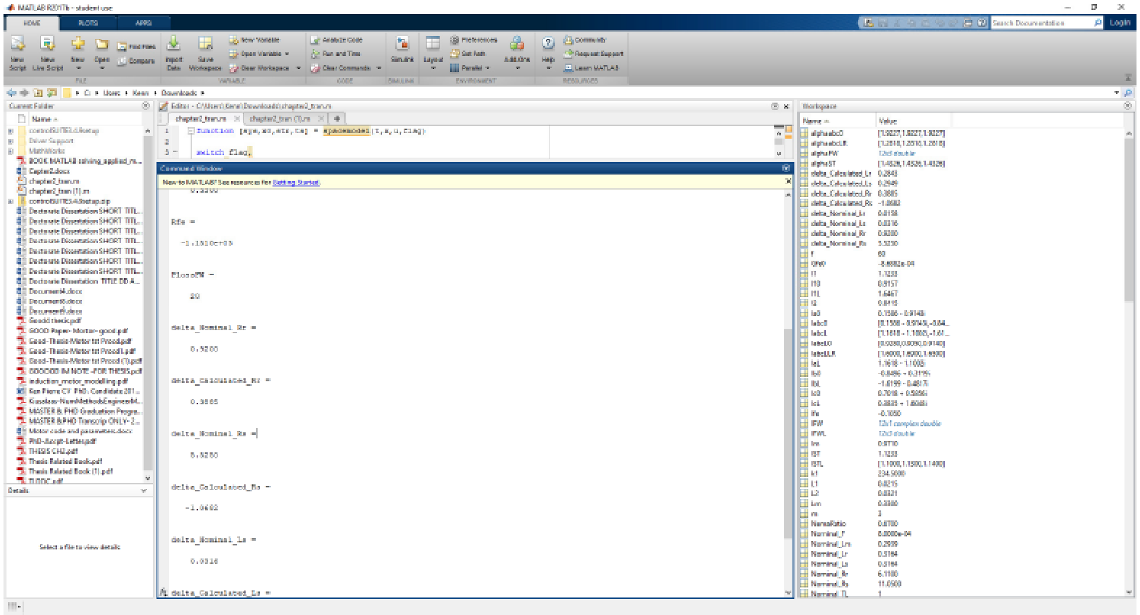


Figure A.4 The TI HVDMC Kit User Interface the Experiment


MATLAB R2017b - student use
HOME
PLOTS
APPS
Editor - C:\Users\Kena\Downloads\chapter2_tran.m
chapter2_tran.m
chapter2_tran (1).m
Current Folder
Command Window
New to MATLAB? See resources for Getting Started.
Workspace
Select a file to view details

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)

switch flag,
```

```
   0.3200


Rfe =

  -1.1810e+03


PlossFW =

    20


delta_Nominal_Rr =

    0.5200


delta_Calculated_Rr =

    0.3885


delta_Nominal_Rs =

    5.5250


delta_Calculated_Rs =

   -1.0682


delta_Nominal_Ls =

    0.0316


delta_Calculated_Ls =
```

Name	Value
alphaabc0	[1.9227,1.9227,1.9227]
alphaabcLR	[1.2818,1.2818,1.2818]
alphaFW	12x3 double
alphaST	[1.4526,1.4526,1.4526]
delta_Calculated_Lr	0.2843
delta_Calculated_Ls	0.2949
delta_Calculated_Rr	0.3885
delta_Calculated_Rs	-1.0682
delta_Nominal_Lr	0.0158
delta_Nominal_Ls	0.0316
delta_Nominal_Rr	0.5200
delta_Nominal_Rs	5.5250
f	60
Gfe0	-8.8882e-04
I1	1.1233
I10	0.9157
I1L	1.6467
I2	0.8415
Ia0	0.1586 - 0.9148i
Iabc0	[0.1586 - 0.9148i,-0.84…
IabcL	[1.1618 - 1.1002i,-1.61…
IabcL0	[0.9280,0.9090,0.9140]
IabcLLR	[1.6000,1.6900,1.6500]
IaL	1.1618 - 1.1002i
Ib0	-0.8496 + 0.3119i
IbL	-1.6199 - 0.4817i
Ic0	0.7018 + 0.5256i
IcL	0.2825 + 1.6068i
Ife	-0.1050
IFW	12x1 complex double
IFWL	12x3 double
Im	0.8718
IST	1.1233
ISTL	[1.1000,1.1500,1.1400]
k1	234.5000
L1	0.0215
L2	0.0321
Lm	0.2380
m	3
NameRatio	0.8700
Nominal_f	8.0000e-04
Nominal_Lm	0.2939
Nominal_Lr	0.5164
Nominal_Ls	0.5164
Nominal_Rr	6.1100
Nominal_Rs	11.0500
Nominal_TL	1

Figure A.5 MATLAB Script Results from the No Load, and Locked Rotor

Figure A.5 above illustrates the MATLAB script results from the No load, load and locked motor rotor /system identification experiment

Appendix B: MATLAB Script that Calculates Non-Measured Parameter Values from the Experiment Measurements

Run the following experiment scripts in the order given below.

Run the program below first.

```
%Read me Instructions
%Institute of Electrical and Electronics Engineers IEEE
%Standard Test Procedure for Polyphase Induction Motors and Generators
%used to derive the Test Procedure for these experiments

%This is a script uses the measured induction motor parameters from the
%experiment to calculate the non-measured parameters.

%Circuit model contains:
%Stator (Rs) Resistance
%Stator (Ls) Leakage Reactance
%Rotor (Rr) Resistance
%Rotor (Lr) Leakage Inductance
%Magnetizing Inductance
%Core Resistance

%The Calculation of model parameters is performed in the experiments by
%measuring the followings:
%rotor speed
%line-to-line voltage phasor
%line current phasor for No-Load Test
%line current phasor for Locked-Rotor Test
%line current phasor for Load-Point operating conditions
%All measurements are performed with a three-phase source derived from
my %private Engineering Lab in my house using the wall outlet connected to
the %utility grid (110 Vac; 60Hz).
%"%**" means measured, read off TI GUI, specification and/or input values
%National Electrical Manufacturers Association (NEMA) defines Design A, B,
%C, and D motors in NEMA MG-1-2003
```

```
clear all, clc;
% The Nominal Induction Motor Parameters and Test conditions
f = 60;                 %Stator frequency applied for No Load Test (Hz)
P = 4;                  %Number of machine poles 4 Pole-machine
m = 3;                  %Number of phases 3-phase excitation source
ns = (120*f/P);         %Synchronous speed for machine under test (rpm)
TF = 72;                %Room Temp (Degree Fahrenheit)
NemaRatio = 0.67;       %Ratio of X1 to X2 (X1/X2 = 0.67 for NEMA Design
B)
XmRatio = 1;            %Define a ratio (X1/Xm) for use in calculations
%Rr = 6.11;             %Rr=R2 Rotor Resistance (RR) (unit in ohm)
%ΔRr = 0.92;            %Bias Faults of Magnitudes (unit in ohm) at15%
%Rs = 11.05;            %Rs=R1 Stator resistance (SR) (unit in ohm)
%ΔRs = 5.525;           %Bias Faults of Magnitudes (unit in ohm) at 50%
%Lm = 0.29395;          %Mutual Inductance (MI) (unit in henry)
%Ls = 0.316423;         %Stator Inductance (SI) (unit in henry)
%ΔLs= 0.031642;         %Bias Faults of Magnitudes at 10%
%Lr = 0.316423;         %Rotor inductance (RI) in (unit in henry)
%ΔLr= 0.015821;         %Bias Faults of Magnitudes at 5%
%TL =1;                 %Torque Load (unit in N.m)
%ΔLr = 0.2;             %Bias Faults of Magnitudes at 20%
%F = 0.0008;            %Viscous friction coefficient unit in (N.m)/(rad/s)


%First Experiment- Stator Resistance, R1 the per phase DC resistance, in
ohms

%Stator winding resistance is measured with Digital Multi-Meter
%All three windings are measured, and their average is taken
%For WYE connection Stator the measurement between terminals is twice
%the resistance of each winding.

TC = (5/9)*(TF - 32);
Ta =TC;
ta = TC;               %Enter temperature of the coil at the time of
measurement; ambient temperature in “oC” (Degree Celsius)

tb = 40;              %Temperate at which resistance is reported; value on
nameplate in “oC” (Degree Celsius)

k1 = 234.5;            %factor for 100% IACS conductivity copper (inferred
temperature for zero resistance) or 225 for aluminum, based on a volume
conductivity of 62%.
Rab = 20.5;    %Measured value (ohm) per IEEE Std 112-2004 in home lab
```

```
Rbc =22;        %Measured value (ohm) per IEEE Std 112-2004 in home lab
Rca = 25.5;     %Measured value (ohm) per IEEE Std 112-2004 in home lab
R1 = (Rab/2 + Rbc/2 + Rca/2)/3; %Average stator resistance (ohm)


%NO LOAD TEST
%Second Experiment: No-Load and Locked Rotor tests finding Xm, X1, X2

%Running the motor at rated voltage and frequency with no connected load.
%What is measured: Stator terminal voltage, line current, phase and rotor
%speed, line-to-line voltage, current and time between them

VabcLL0 = 2*[104 104 103];    %[Vab Vbc Vca] %line-to-line voltage(Vroot-
mean-square) (rms)
IabcL0 = [0.929 0.905 0.914];  %[Ias Ibs Ics] %line current (Aroot-mean-
square) (rms)
tabc0 = (10^-3)*[5.2 5.2 5.2]; %[ta tb tc] %time measured between line-
to-line voltage and current peaks(s)

%The Rotor speed during No Load Test (rpm) is read off the TI GUI
nt0 = 1796;         % Rotor speed during No Load Test (rpm)

%Experiment Data
VabcLN0 = VabcLL0/sqrt(3);     % Line To Neutral  Voltage (LNV)
VabcM0 = sum(VabcLN0)/3;     %Average Line To Neutral voltage (ALNV)
I10 = sum(IabcL0)/3;              %Average Line current

%Calculation of the Angle between line-to-line voltage and line current in
%radiant
alphaabc0 = tabc0/(1/60)*2*pi; %Angle between line-to-line voltage and
line current in radiant
%Calculation of Angle between line-to-neutral voltage and line current in
%radiant
thetaabc0 = alphaabc0 - pi/6;
%Calculation of Average angle between line-to-neutral voltage and line
%current in radiant
theta0 = sum(thetaabc0)/3;
%Calculation of Average power factor
PF0 = cos(theta0);

%Reference Voltage Phasor Model
Va_zero = VabcLN0(1).*exp(i*0);
Va0 =Va_zero;
Vb_zero= VabcLN0(2).*exp(i*(-2*pi/3));
```

```
Vb0 =Vb_zero;
Vc_zero = VabcLN0(3).*exp(i*(+2*pi/3));
Vc0 =Vc_zero;


%Line-to-neutral Voltage Phasors Vabc Model
Vabc0 = [Va0 Vb0 Vc0];

%Line Current phasors Model
Ia_zero = IabcL0(1).*exp(i*(-thetaabc0(1)));
Ia0 =Ia_zero;
Ib_zero = IabcL0(2).*exp(i*(-thetaabc0(2) - 2*pi/3));
Ib0 =Ib_zero;
Ic_zero = IabcL0(3).*exp(i*(-thetaabc0(3) + 2*pi/3));
Ic0 =Ic_zero;
%Line current phasors Iabc Model
Iabc0 = [Ia0 Ib0 Ic0];

%Slip Calculation
SS0 = ns - nt0;  %Slip Speed (rpm)
s0 = SS0/ns;     %slip speed (pu)

%Power Calculation
S0NL = Vabc0 .* conj(Iabc0);  % This is per phase.
P0 = real(sum(S0NL)); % this is for three-phase.
Q0 = imag(sum(S0NL)); % This is for three-phase.


%LOCKED OR BLOCKED ROTOR TEST USING A CLAMP AS A MECHANICAL
ROTOR LOCKING DEVICE
%The motor is excited via Texas Instrument (TI) GUI at rated frequency;
however, the rotor is clamped; that is, it is not turning.
%The voltage is increased using the Variac Transformer until the line current
is at the rated value.

%The stator terminal voltage, line current, and phase are measured.
VabcLLLR = [36.2 35.5 36.1]; %[Vab Vbc Vca] %Line-to-line voltage (Vroot-
mean-square) (Vrms)
IabcLinetoLineR = [1.60 1.69 1.65]; %[Ias Ibs Ics] %Line current
magnitude (Aroot-mean-square) (rms)
IabcLLR =IabcLinetoLineR;
tabcLineR = (10^-3)*[3.4 3.4 3.4]; % Where the [ta tb tc] is
% The Time measured between Line To Line (LTL) voltage and current
peaks.
```

```
tabcLR =tabcLineR;
ntLR = 0; %**Rotor speed read off the TI GUI while Locked Rotor Test

%Experiment Data
VabcLNLR = VabcLLLR/sqrt(3); %Note -  the Line To Neutral Voltage (LNV)
V1L = sum(VabcLNLR)/3;  % Note - Average of the Voltage Line To Neutral
(AVLN)I1Line = sum(IabcLLR)/3;  % Note - Average of the Line Current
%(ALC)
I1L= I1Line;
alphaabcLR = tabcLR/(1/60)*2*pi; %Angle between line-to-line voltage and
line current in radiant
thetaanglebcLR = alphaabcLR - pi/6; %Angle between line-to-neutral
voltage and line current in radiant
thetaabcLR=thetaanglebcLR;
thetaangleLR = sum(thetaabcLR)/3; %Average angle between line-to-
neutral voltage and line current in radiant

thetaLR =thetaangleLR;
PFL = cos(thetaLR);               %Average power factor

%Reference Voltage Phasor Model
VaLine = VabcLNLR(1).*exp(i*0);
VaL=VaLine;
VbLine = VabcLNLR(2).*exp(i*(-2*pi/3));
VbL=VbLine;
VcLine = VabcLNLR(3).*exp(i*(+2*pi/3));
VcL=VcLine;
VabcL = [VaL VbL VcL];

%Line Current Phasors Model
IaLine = IabcLLR(1).*exp(i*(-thetaabcLR(1)));
IaL=IaLine;
IbLine = IabcLLR(2).*exp(i*(-thetaabcLR(2) - 2*pi/3));
IbL=IbLine;
IcLine = IabcLLR(3).*exp(i*(-thetaabcLR(3) + 2*pi/3));
IcL=IcLine;
IabcL = [IaL IbL IcL];

%Slip Calculation
SSLR = ns - ntLR; %Slip Speed (rpm)
sLR = SSLR/ns; %slip speed (pu)

%Real and Reactive Power Calculations
SLine = VabcL .* conj(IabcL);  %1-phase
```

```
SL= SLine;
PLine = real(sum(SL)); %3-phase
PL= PLine;
QLine = imag(sum(SL)); %3-phase
QL= QLine;

%Xm, X1, and X2 Iteration to conversion
%Setting X1, Xm and X2 initial values
X1intial= 2*pi*f*(0.001); %**(2*pi*f*Lohm)
X1 =X1intial;
Xmintial= X1/XmRatio; %(ohm)
Xm =Xmintial;
X2intial = X1/NemaRatio; %(ohm)
X2 =X2intial;

%The set initial values used for determination of convergence as
X1prev = X1;
Xmprev = Xm;
Xmc = Xm;

%The iteration processes
Xmc = m*VabcM0^2/(Q0-(m*I10^2*X1)) * 1/((1+X1/Xmc)^2);
Xm = Xmc; %(IEEE Std 112 equation 30)
X1Lc = QL/(m*I1L^2*((1+X1/X2+X1/Xm)/1)) * (X1/X2+X1/Xm); %(IEEE
Std 112 equation 31)
X1L=X1Lc;
X1 = 60/f*X1L;

%Calculating the delta between the previous and new value – is it > 0.1%
while or((or((X1>1.002*X1prev), (X1<0.999*X1prev))),
(or((Xm>1.001*Xmprev), (Xm<0.999*Xmprev))));
    X1prev = X1; %X1 Previous value is retained in X1
    Xmprev = Xm; % Xm1 Previous value is retained in X1m
Xm = 1*m*(VabcM0^2)/(Q0-(m*(I10^2)*X1)) * 1/((1+X1/Xm)^2); %New
value
X1L = QL/(1*m*I1L^2*(1+X1/X2+X1/Xm)) * (X1/X2+X1/Xm); %New
value
    X1 = 60/f*X1L;     %New value

end

%End the iterations and final values are saved
X1 = X1; %(ohm)
Xm = Xm; %(ohm)
```

```
X2 = X1/NemaRatio; %(ohm)

%Convert Reactance to Inductance
L1 = X1/(2*pi*f); %(Henry)
Lm = Xm/(2*pi*f); %(Henry)
L2 = X2/(2*pi*f); %(Henry)
```

Run the program below second.

```
%The Third Test
%No Load test at range of voltages to assess friction and windage

%Measured and Input voltages and currents at voltages from 125% rated
down
%to point at which current magnitude increases.
VFWLL=2*[104 104 104; 101 100 101; 94.0 93.7 94.1; 88.2 87.9 88.1;
82.7 82.2 82.8; 72.6 72.1 72.5; 63.4 62.7 63.4; 50.8 50.4 50.9; 43.4 43.0
43.5; 35.5 35.0 35.6; 28.5 28.3 28.5; 23.3 23.0 23.3;];
VFWLL = VFWLLmatrice;
%Where the rows are [Vab, Vbc, Vca], and the columns are different levels
of voltage.

IFWL = [0.928 0.922 0.927; 0.855 0.852 0.861; 0.786 0.790 0.786; 0.721
0.718 0.718; 0.668 0.666 0.662; 0.580 0.561 0.567; 0.503 0.502 0.491;
0.410 0.405 0.400; 0.360 0.355 0.349; 0.305 0.301 0.296; 0.285 0.283
0.275; 0.283 0.278 0.276;];
%Where the rows are [Ia, Ib, Ic].

tFW = (10^-3)*[5.2 5.1 5.1; 5.1 5.2 5.2; 5.1 5.1 5.1; 5.2 5.1 5.1; 5.1 5.1
5.2; 5.1 5.1 5.2; 5.0 5.0 5.0; 4.8 4.8 4.9; 4.6 4.7 4.7; 4.5 4.6 4.6; 3.8 3.9
3.9; 3.6 3.6 3.6; ];
%The time between (Vab, Ia), (Vbc, Ib) and (Vca, Ic)

VFWLN = VFWLL/sqrt(3);  %Line-to-Neutral voltage
VFW = sum(VFWLN,2)/3;   %Average Voltage Line-to-Neutral
IFW = sum(IFWL,2)/3;    %Average current, line current
alphaFW = tFW/(1/60)*2*pi; %Angle between line-to-line voltage and line
current in radiant
thetaFW = alphaFW - pi/6; % Angle (radiant) between line-to-neutral
voltage and line current in radiant
thetaFW = sum(thetaFW,2)/3; %Average angle(radiant) between line-to-
neutral voltage and line current
PFFW = cos(thetaFW); %Average power factor
```

```
%Input Power and Stator, I^2R loss Calculation
%Setting the Line-to-Neutral voltage phasor as reference (V)
VFW = VFW.*exp(i*0);
%Line Current phasor(A)Model
IFW = IFW.*exp(i*(-thetaFW));

%Input Power Model
SinFW = 3*(VFW .* conj(IFW)); %3-Phase (VA)
PinFW = real(SinFW); %3-Phase Input Power (PIP) (unit in Watt)

%The Stator Loss Model and Calculation
PsIR = 3*(abs(IFW).^2 .* R1); %3-Phase Stator Loss (PSL) (unit in Watt)

%Graphing;
%First, Subtract stator loss from input power
%Second, graph the resulting power vs. voltage
PM = PinFW - PsIR; %This is the Total Power Transferred across the airgap.

%The friction and windage loss are obtained by graphing the power versus
voltage.
figure(1), clf;
plot(abs(VFW), PM,'x');
% Graph the Average Voltage Line To Neutral (VLN) versus PM. PM is the
Power.
hold on;
p = polyfit(VFW,PM,2);
x = 0:360;
% plot(0:360,polyval(p,x));
%axis([0 360 0 360]);

% Input the power at V=0 from the graph.
PlossFW = input('What is the power at V=0. PFW = ');
if isempty(PlossFW);
PlossFW = 0; % Power loss due friction and windage is found at zero-voltage
intercept.
end;
clc

%Calculate Power due to Total Core Loss
%No Load Core Loss
%Ph0 equation assumes rotor current is zero (s=0, Zr=infinite)
Ph0 = PinFW - PsIR - PlossFW; %Core loss for applied voltage at No-Load
%Calculate Core Conductance for No-Load Friction Windage
```

```
Gfe0 = Ph0./(m*VFW.^2) * (1+X1/Xm)^2; %Core Conductance from No
Load configuration

Gfe0 = sum(Gfe0)/length(Gfe0); %Core conductance from Locked Rotor
configuration

%Calculate Core Resistance, Rfe
Rfe = 1/Gfe0; %Core Resistance

%Rotor Resistance, R2 Calculation by performing the full load test
%Full Load Slip Test is Performed. The Voltage Slip Test is Decreased
%The motor is turned at rated frequency, and rated voltage, then the load is
applied by partially clamping the rotor without locking the rotor

%Measured:
%the stator voltage
%the line current
%the phase angle and slip
%the calculations are performed per IEEE Std 112-2004 section 5.9.2.2.
%Average of several loads (partially clamping the rotor) are taken.
VSTLL = 2*[105 104 105]; % Note - Line To Line Voltage (LTL) at slip test
ISTL = [1.11 1.13 1.14]; % Note - Line Current (LC) at Slip Test
tST = (10^-3)*[3.7 3.7 3.7]; %Time between Line-to-Line voltage and line
current peak (Vab to Ia, etc.)
ntST = [1728]; %**Rotor speed during Slip Test (rev/per/minute) read of
the TI GUI


%Slip Test Condition
VSTLN = VSTLL/sqrt(3); %Line-to-Neutral voltage at slip test
VST = sum(VSTLN)/3; %Average Line-to-Neutral voltage during slip test (V)
IST = sum(ISTL)/3; %Average Line current during slip test (A)
alphaST = tST/(1/60)*2*pi; %Angle measured between line-to-line voltage
and line current in radiant
thetaST = alphaST - pi/6; %Angle between line-to-neutral voltage and line
current in radiant
thetaST = sum(thetaST)/3; %Average angle between line-to-neutral voltage
%and line current in radiant calculation
PFST = cos(thetaST); %Average power factor

%Slip Calculation
SSST = ns - ntST; %Slip Speed (rpm)
sST = SSST/ns; %slip speed (pu)
%R1 Adjusted for Temperature
```

```
R1 =  R1*(tb+k1)/(ta+k1); %Correct value for temperature (ohm)% Stator
resistance

%Rotor Winding Voltage V2 Reflected to Stator Calculation
V1 = VST; %Value for equation taken from standard 112
I1 = IST; %Value for equation taken from standard 112
theta1angle = thetaST; %Value for equation taken from standard 112
theta1 =theta1angle;
X1ctheta1 = X1*cos(theta1);
X1stheta1 = X1*sin(theta1);
R1cos = R1*cos(theta1);
R1sin = R1*sin(theta1);

V2 = sqrt((V1-I1*R1cos - X1stheta1))^2 + 1+ (I1*(R1sin
+X1ctheta1)))^2);

% Formula for “Theta2” Calculation
X1ctheta1 = X1*cos(theta1);
X1stheta1 = X1*sin(theta1);
theta2 = (atan((-I1*(R1sin - X1ctheta1)))/(V1 - I1*(R1cos -
X1stheta1))))/1);

%Ife and Ie Calculation
Im = V2/Xm;      %See IEEE Std 112-2004 use figure 2 to find: V2/Xm = Im
Ife = V2/Rfe ;

%I2, Z2 and R2 Calculations
I2 = sqrt((I1*cos(theta1) - Im*sin(theta2) – Ife*(Im/Im)*cos(theta2))^2 +
(-(I1*1)*sin(theta1) + Im*cos(theta2) + Ife*sin(theta2))^2);
Z2 = V2/I2;
R2 = sST*sqrt(Z2^2 - X2^2);

%Induction Motor Parameter Values Adjusted for Temperature, See IEEE Std
112-2004 equation 3 Display ('Machine Parameters:')

L1 % Stator-Leakage Inductance (H)
R1 % Stator-Resistance (ohm)
R2 % Rotor-Resistance (ohm)
L2 % Rotor-Leakage Inductance (H)
Lm % Magnetizing-Inductance (H)
Rfe %Core-Resistance (ohm)
PlossFW %Friction-and-Windage-Loss (W)
%END SCRIPT
```

After the scripts are run, the result from MATLAB in the Workspace is obtained and provided here as pictures.

This is the same script as above with fewer comments or no explanation, thus, more readable.

Run the script below first.

The new script starts below. All scripts start with “clear all, clc;”.

```
clear all, clc;

% The Nominal Induction Motor Parameters and Test conditions
f = 60;
P = 4;
m = 3;
ns = (120*f/P);
TF = 72;
NemaRatio = 0.67;
XmRatio = 1;


%First Experiment- Stator Resistance, R1 the per phase DC resistance, in
ohms

TC = (TF - 32)*5/9;
ta = TC;
tb = 40;
k1 = 234.5;

Rab = 20.5;
Rbc =22;
Rca = 25.5;
R1 = (Rab/2 + Rbc/2 + Rca/2)/3;

%Second Experiment: No-Load and Locked Rotor tests finding Xm, X1, X2
%NO LOAD TEST
VabcLL0 = 2*[104 104 103];
IabcL0 = [0.929 0.905 0.914];
```

```
tabc0 = (10^-3)*[5.2 5.2 5.2];

%The Rotor speed during No Load Test (rpm) is read off the TI GUI

nt0 = 1796;
%Experiment Data
VabcLN0 = VabcLL0/sqrt(3);
VabcM0 = sum(VabcLN0)/3;
I10 = sum(IabcL0)/3;

alphaabc0 = tabc0/(1/60)*2*pi;
thetaabc0 = alphaabc0 - pi/6;
theta0 = sum(thetaabc0)/3;
PF0 = cos(theta0);

Va_zero = VabcLN0(1).*exp(i*0);
Va0 =Va_zero;
Vb_zero= VabcLN0(2).*exp(i*(-2*pi/3));
Vb0 =Vb_zero;
Vc_zero = VabcLN0(3).*exp(i*(+2*pi/3));
Vc0 =Vc_zero;
Vabc0 = [Va0 Vb0 Vc0];

Ia_zero = IabcL0(1).*exp(i*(-thetaabc0(1)));
Ia0 =Ia_zero;
Ib_zero = IabcL0(2).*exp(i*(-thetaabc0(2) - 2*pi/3));
Ib0 =Ib_zero;
Ic_zero = IabcL0(3).*exp(i*(-thetaabc0(3) + 2*pi/3));
Ic0 =Ic_zero;
Iabc0 = [Ia0 Ib0 Ic0];

SS0 = ns - nt0;
s0 = SS0/ns;

S0NL = Vabc0 .* conj(Iabc0);
P0 = real(sum(S0NL));
Q0 = imag(sum(S0NL));


%LOCKED OR BLOCKED ROTOR TEST USING A CLAMP AS A MECHANICAL
ROTOR LOCKING DEVICE

VabcLLLR = [36.2 35.5 36.1]; %[Vab Vbc Vca]
IabcLinetoLineR = [1.60 1.69 1.65];
```

```
IabcLLR =IabcLinetoLineR;
tabcLineR = (10^-3)*[3.4 3.4 3.4]; %[ta tb tc]
 tabcLR =tabcLineR;
ntLR = 0;
%Experiment Data
VabcLNLR = VabcLLLR/sqrt(3);
V1L = sum(VabcLNLR)/3;
I1Line = sum(IabcLLR)/3;
I1L= I1Line;
alphaabcLR = tabcLR/(1/60)*2*pi;
thetaanglebcLR = alphaabcLR - pi/6;
thetaabcLR=thetaanglebcLR;
thetaangleLR = sum(thetaabcLR)/3;

thetaLR =thetaangleLR;

PFL = cos(thetaLR);
VaLine = VabcLNLR(1).*exp(i*0);
VaL=VaLine;
VbLine = VabcLNLR(2).*exp(i*(-2*pi/3));
VbL=VbLine;
VcLine = VabcLNLR(3).*exp(i*(+2*pi/3));
VcL=VcLine;
VabcL = [VaL VbL VcL] ;

%Line Current Phasors Model
IaLine = IabcLLR(1).*exp(i*(-thetaabcLR(1)));
IaL=IaLine;
IbLine = IabcLLR(2).*exp(i*(-thetaabcLR(2) - 2*pi/3));
IbL=IbLine;
IcLine = IabcLLR(3).*exp(i*(-thetaabcLR(3) + 2*pi/3));
IcL=IcLine;
IabcL = [IaL IbL IcL];

%Slip Calculation
SSLR = ns - ntLR; %Slip Speed (rpm)
sLR = SSLR/ns; %slip speed (pu)

%Real and Reactive Power Calculations
SLine = VabcL .* conj(IabcL);
SL= SLine;
PLine = real(sum(SL));
PL= PLine;
QLine = imag(sum(SL));
QL= QLine;
```

```
X1intial= 2*pi*f*(0.001);
X1 =X1intial;
Xmintial= X1/XmRatio;
Xm =Xmintial;
X2intial = X1/NemaRatio;
X2 =X2intial;

X1prev = X1;
Xmprev = Xm;
Xmc = Xm;

Xmc = m*VabcM0^2/(Q0-(m*I10^2*X1)) * 1/((1+X1/Xmc)^2);
Xm = Xmc;
%(IEEE Std 112 eqt 30)
X1Lc = QL/(m*I1L^2*(1+X1/X2+X1/Xm)) * (X1/X2+X1/Xm);
X1L=X1Lc;
X1 = 60/f*X1L;

while or((or((X1>1.002*X1prev), (X1<0.999*X1prev))),
(or((Xm>1.001*Xmprev), (Xm<0.999*Xmprev))));
    X1prev = X1; %X1 Previous value is retained in X1
    Xmprev = Xm; % Xm1 Previous value is retained in X1m
Xm = 1*m*(VabcM0^2)/(Q0-(m*(I10^2)*X1)) * 1/((1+X1/Xm)^2);
X1L = QL/(1*m*I1L^2*(1+X1/X2+X1/Xm)) * (X1/X2+X1/Xm);
X1 = 60/f*X1L;

end

X1 = X1; %(ohm)
Xm = Xm; %(ohm)
X2 = X1/NemaRatio; %(ohm)

L1 = X1/(2*pi*f); %(Henry)
Lm = Xm/(2*pi*f); %(Henry)
L2 = X2/(2*pi*f); %(Henry)
```

Run the script below second.

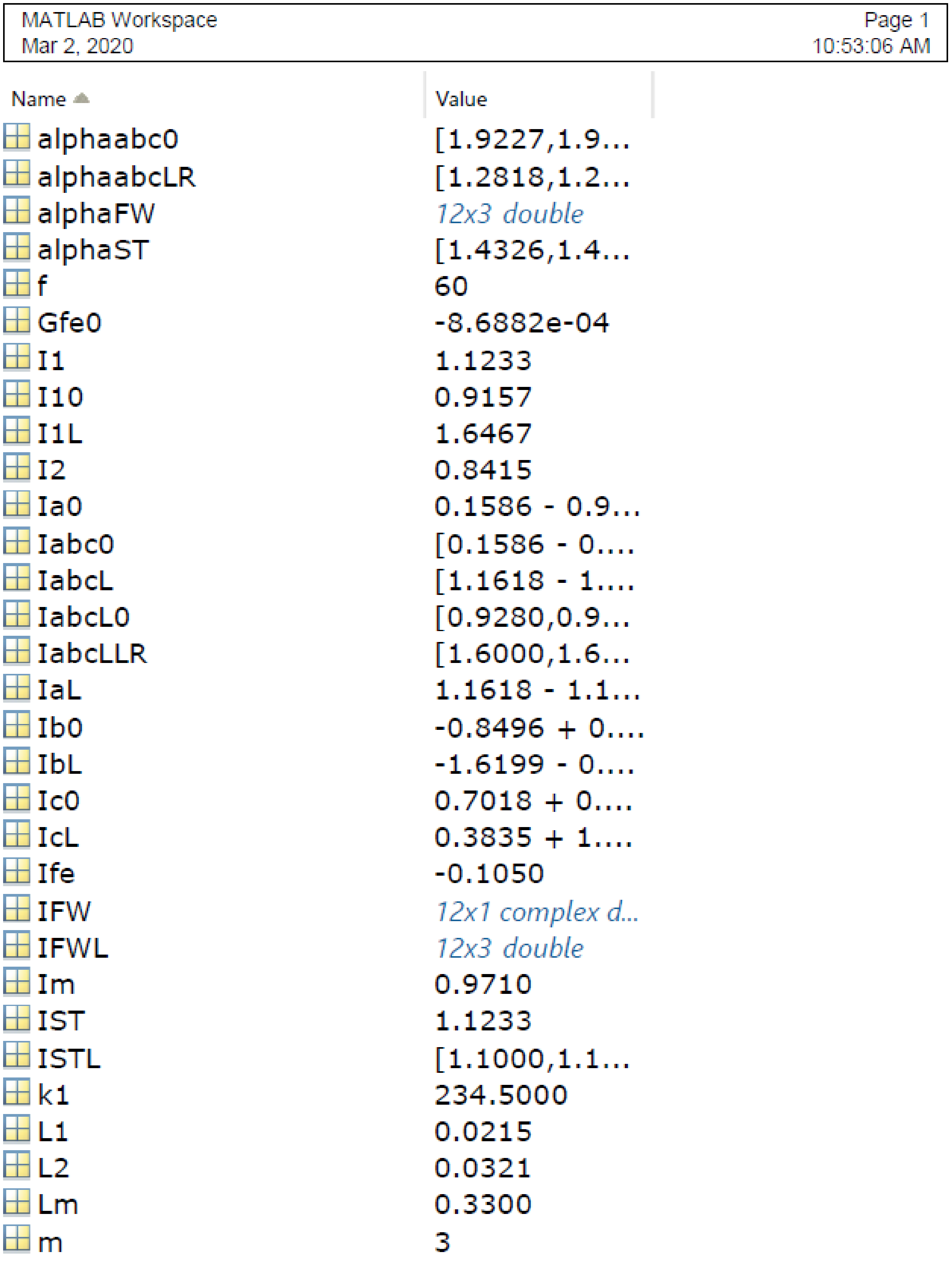
```
%The Third Test
%No Load test at range of voltages to assess friction and windage
%Measured and Input voltages and currents at voltages from 125% rated
down to point at which current magnitude increases.
```

```
VFWLL=2*[104 104 104; 101 100 101; 94.0 93.7 94.1; 88.2 87.9 88.1;
82.7 82.2 82.8; 72.6 72.1 72.5; 63.4 62.7 63.4; 50.8 50.4 50.9; 43.4 43.0
43.5; 35.5 35.0 35.6; 28.5 28.3 28.5; 23.3 23.0 23.3;];
VFWLL = VFWLLmatrice;
IFWL = [0.928 0.922 0.927; 0.855 0.852 0.861; 0.786 0.790 0.786; 0.721
0.718 0.718; 0.668 0.666 0.662; 0.580 0.561 0.567; 0.503 0.502 0.491;
0.410 0.405 0.400; 0.360 0.355 0.349; 0.305 0.301 0.296; 0.285 0.283
0.275; 0.283 0.278 0.276;];
%Where the rows are [Ia, Ib, Ic].

tFW = (10^-3)*[5.2 5.1 5.1; 5.1 5.2 5.2; 5.1 5.1 5.1; 5.2 5.1 5.1; 5.1 5.1
5.2; 5.1 5.1 5.2; 5.0 5.0 5.0; 4.8 4.8 4.9; 4.6 4.7 4.7; 4.5 4.6 4.6; 3.8 3.9
3.9; 3.6 3.6 3.6; ];

VFWLN = VFWLL/sqrt(3);
VFW = sum(VFWLN,2)/3;
IFW = sum(IFWL,2)/3;
alphaFW = tFW/(1/60)*2*pi;
thetaFW = alphaFW - pi/6;
thetaFW = sum(thetaFW,2)/3;
PFFW = cos(thetaFW); %Average power factor


VFW = VFW.*exp(i*0);
%Line Current phasor(A)Model
IFW = IFW.*exp(i*(-thetaFW));

SinFW = 3*(VFW .* conj(IFW)); %Note - 3-Phase (unit in VA)
PinFW = real(SinFW); % Note - 3-Phase Input Power(unit in W)

PsIR = 3*(abs(IFW).^2 .* R1); % Note - 3-Phase Stator Loss (unit in W)

PM = PinFW - PsIR;
figure(1), clf;
plot(abs(VFW), PM,'x');

hold on;
p = polyfit(VFW,PM,2);
x = 0:360;
% plot(0:360,polyval(p,x));
%axis([0 360 0 360]);
% Input the power at V=0 from the graph.
PlossFW = input('What is the power at V=0. PFW = ');
if isempty(PlossFW);
```

```
    PlossFW = 0;
end;
clc

Ph0 = PinFW - PsIR - PlossFW;

Gfe0 = Ph0./(m*VFW.^2) * (1+X1/Xm)^2;
Gfe0 = sum(Gfe0)/length(Gfe0);
Rfe = 1/Gfe0;

VSTLL = 2*[105 104 105];
ISTL = [1.11 1.13 1.14];
tST = (10^-3)*[3.7 3.7 3.7];
ntST = [1728];


%Slip Test Condition
VSTLN = VSTLL/sqrt(3);
VST = sum(VSTLN)/3;
IST = sum(ISTL)/3;

alphaST = tST/(1/60)*2*pi;
thetaST = alphaST - pi/6;
thetaST = sum(thetaST)/3;
PFST = cos(thetaST);

SSST = ns - ntST;
sST = SSST/ns;

%R1 Adjusted for Temperature
R1 =  R1*(tb+k1)/(ta+k1);

%Rotor Winding Voltage V2 Reflected to Stator Calculation
V1 = VST;
I1 = IST;
theta1angle = thetaST;
theta1 =theta1angle;

X1ctheta1 = X1*cos(theta1);
X1stheta1 = X1*sin(theta1);
R1cos = R1*cos(theta1);
R1sin = R1*sin(theta1);
V2 = sqrt((V1-I1*R1cos - X1stheta1))^2 + 1+ (I1*(R1sin
+X1ctheta1)))^2);
```

```
% Formula for “Theta2” Calculation
X1ctheta1 = X1*cos(theta1);
X1stheta1 = X1*sin(theta1);
theta2 = (atan((-I1*(R1sin - X1ctheta1)))/(V1 - I1*(R1cos -
X1stheta1))))/1);

%Ife and Ie Calculation
Im = V2/Xm;      %See IEEE Std 112-2004 use figure 2 to find: V2/Xm = Im
Ife = V2/Rfe ;

%I2, Z2 and R2 Calculations
I2 = sqrt((I1*cos(theta1) - Im*sin(theta2) – Ife*(Im/Im)*cos(theta2))^2 +
(-(I1*1)*sin(theta1) + Im*cos(theta2) + Ife*sin(theta2))^2);
Z2 = V2/I2;

R2 = sST*sqrt(Z2^2 - X2^2);

%Induction Motor Parameter Values Adjusted for Temperature, See IEEE Std
112-2004 equation (3) Displays ('Machine Parameters:')
```

[L1 R1 R2 L2 Lm Rfe PlossFW]^{T}

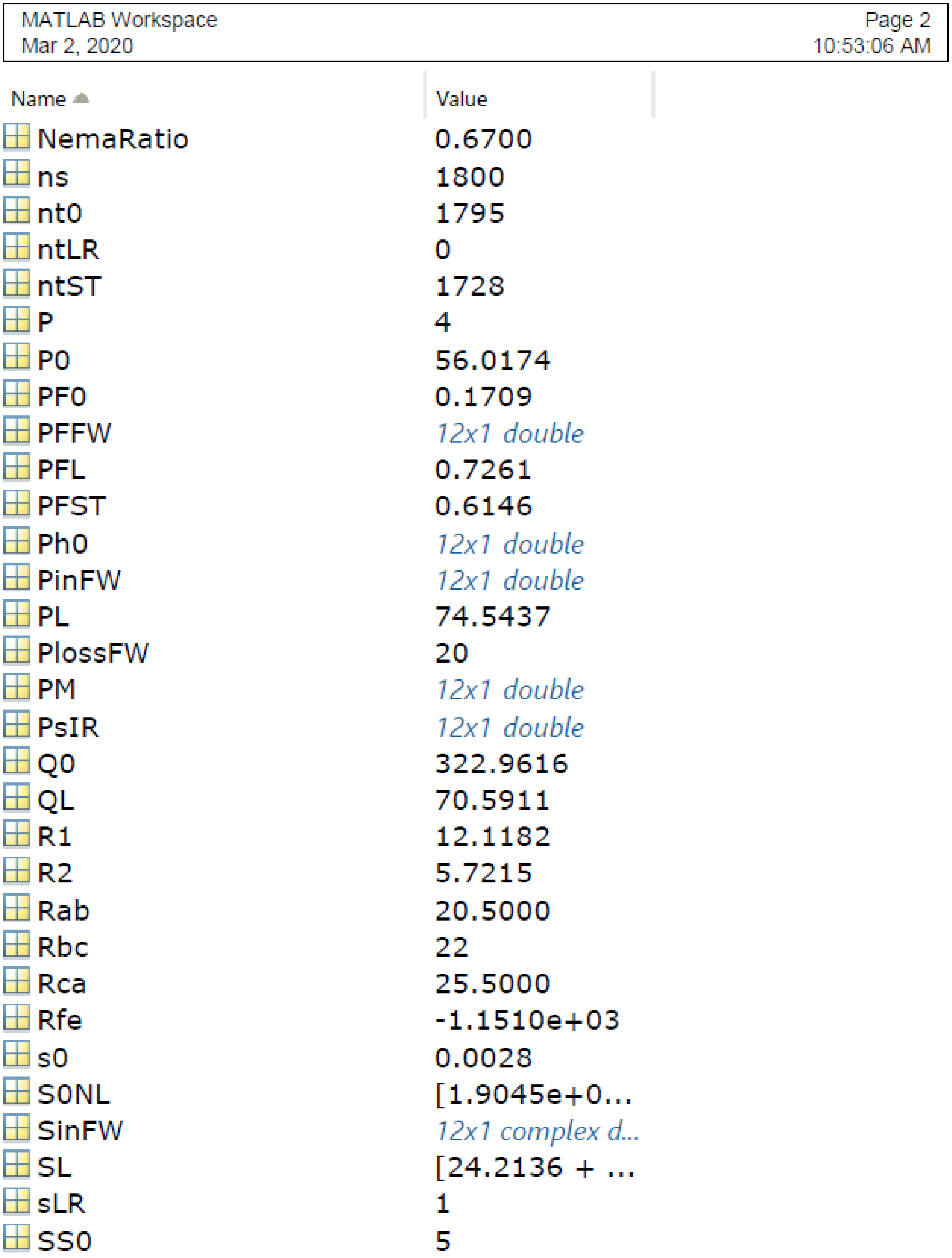


MATLAB Workspace
Mar 2, 2020

Page 1
10:53:06 AM

Name	Value
alphaabc0	[1.9227,1.9...
alphaabcLR	[1.2818,1.2...
alphaFW	*12x3 double*
alphaST	[1.4326,1.4...
f	60
Gfe0	-8.6882e-04
I1	1.1233
I10	0.9157
I1L	1.6467
I2	0.8415
Ia0	0.1586 - 0.9...
Iabc0	[0.1586 - 0....
IabcL	[1.1618 - 1....
IabcL0	[0.9280,0.9...
IabcLLR	[1.6000,1.6...
IaL	1.1618 - 1.1...
Ib0	-0.8496 + 0....
IbL	-1.6199 - 0....
Ic0	0.7018 + 0....
IcL	0.3835 + 1....
Ife	-0.1050
IFW	*12x1 complex d...*
IFWL	*12x3 double*
Im	0.9710
IST	1.1233
ISTL	[1.1000,1.1...
k1	234.5000
L1	0.0215
L2	0.0321
Lm	0.3300
m	3

Figure B.1 MATLAB Workspace Non-Measured and Measured Parameters

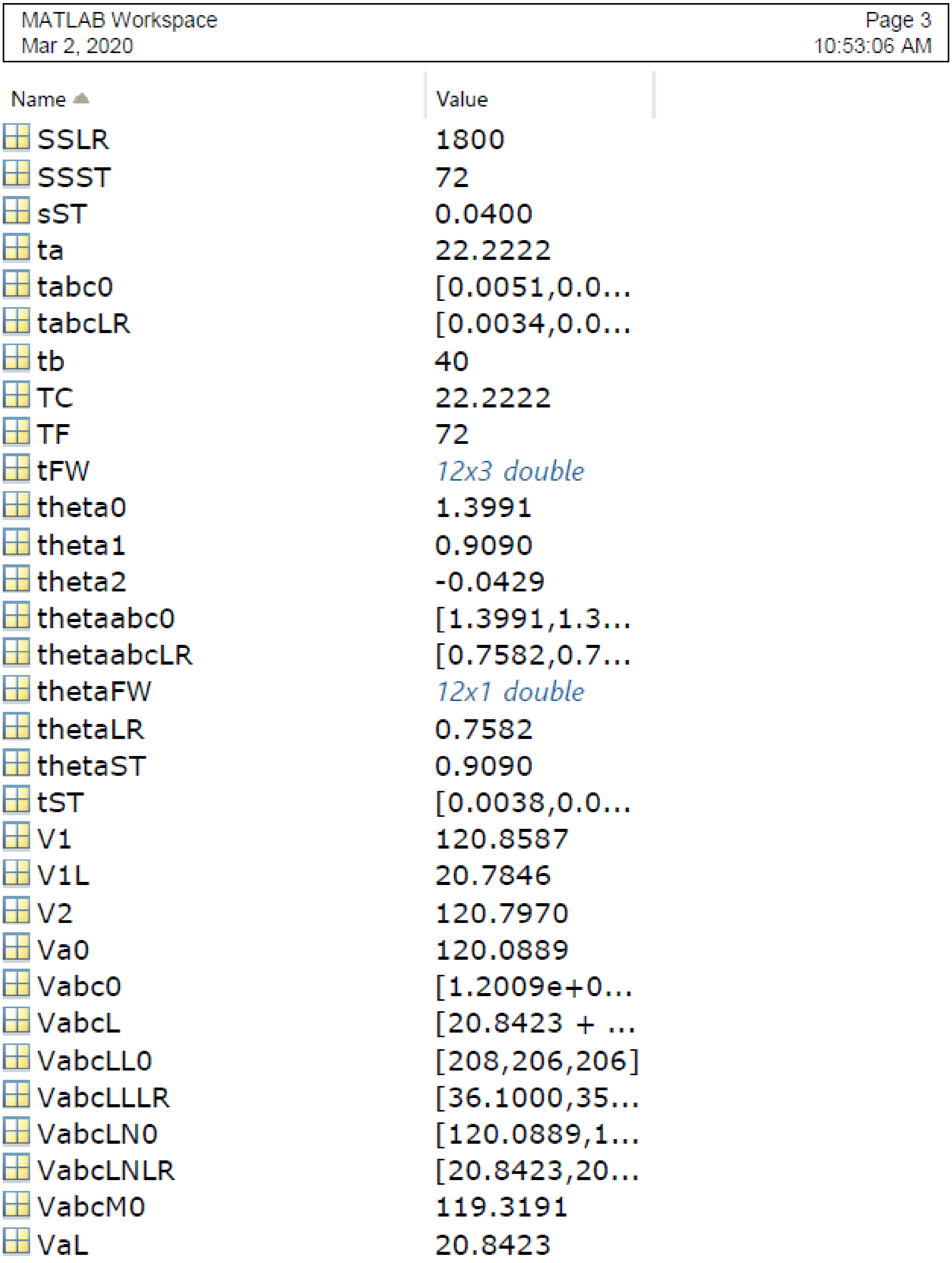


MATLAB Workspace
Mar 2, 2020

Page 2
10:53:06 AM

Name	Value
NemaRatio	0.6700
ns	1800
nt0	1795
ntLR	0
ntST	1728
P	4
P0	56.0174
PF0	0.1709
PFFW	*12x1 double*
PFL	0.7261
PFST	0.6146
Ph0	*12x1 double*
PinFW	*12x1 double*
PL	74.5437
PlossFW	20
PM	*12x1 double*
PsIR	*12x1 double*
Q0	322.9616
QL	70.5911
R1	12.1182
R2	5.7215
Rab	20.5000
Rbc	22
Rca	25.5000
Rfe	-1.1510e+03
s0	0.0028
S0NL	[1.9045e+0...
SinFW	*12x1 complex d...*
SL	[24.2136 + ...
sLR	1
SS0	5

Figure B.2 MATLAB Workspace Non-Measured/Measured Parameters

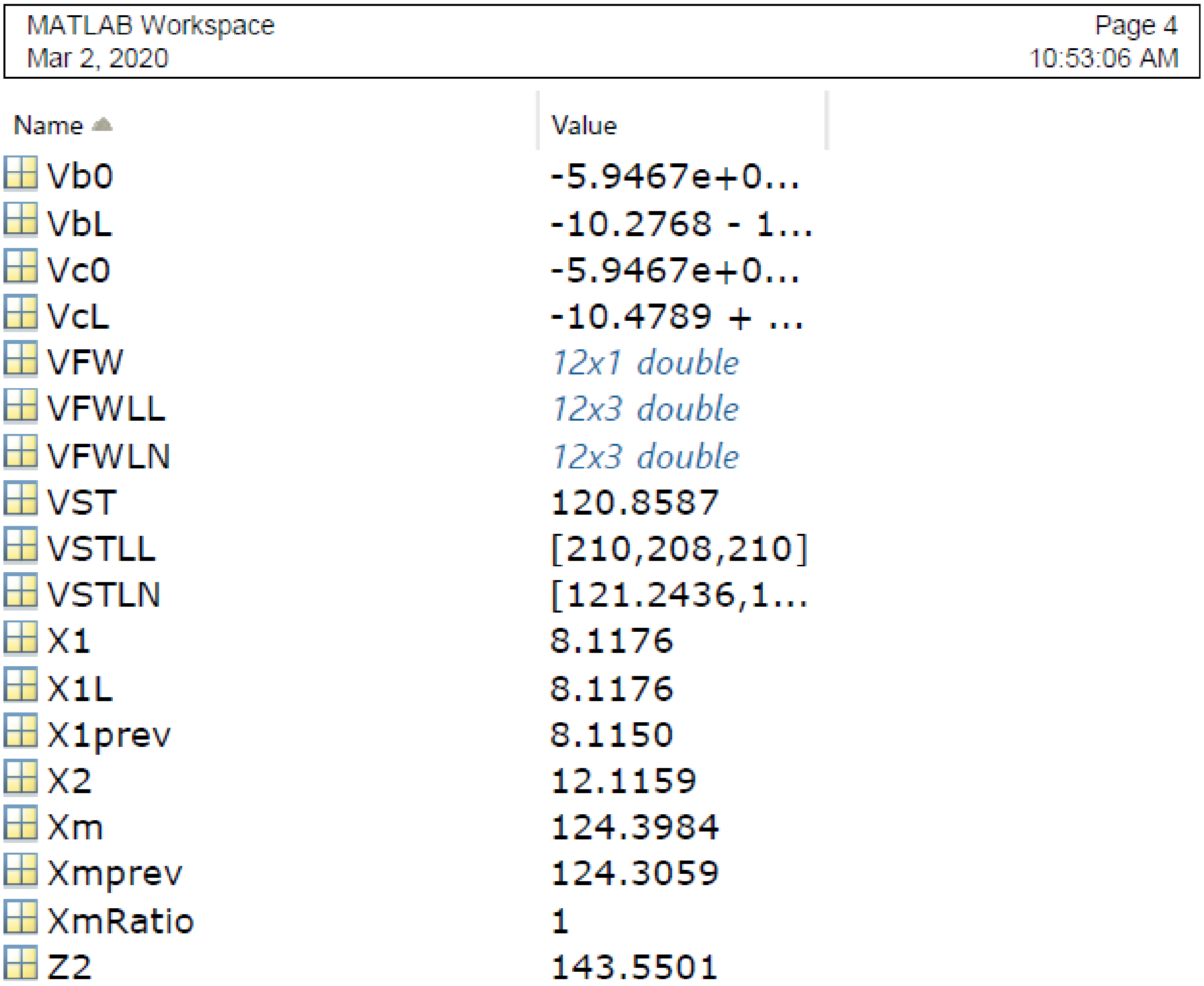


MATLAB Workspace
Mar 2, 2020

Page 3
10:53:06 AM

Name	Value
SSLR	1800
SSST	72
sST	0.0400
ta	22.2222
tabc0	[0.0051,0.0...
tabcLR	[0.0034,0.0...
tb	40
TC	22.2222
TF	72
tFW	*12x3 double*
theta0	1.3991
theta1	0.9090
theta2	-0.0429
thetaabc0	[1.3991,1.3...
thetaabcLR	[0.7582,0.7...
thetaFW	*12x1 double*
thetaLR	0.7582
thetaST	0.9090
tST	[0.0038,0.0...
V1	120.8587
V1L	20.7846
V2	120.7970
Va0	120.0889
Vabc0	[1.2009e+0...
VabcL	[20.8423 + ...
VabcLL0	[208,206,206]
VabcLLLR	[36.1000,35...
VabcLN0	[120.0889,1...
VabcLNLR	[20.8423,20...
VabcM0	119.3191
VaL	20.8423

Figure B.3 MATLAB Workspace Non-Measured/Measured Parameters



MATLAB Workspace
Mar 2, 2020

Page 4
10:53:06 AM

Name	Value
Vb0	-5.9467e+0...
VbL	-10.2768 - 1...
Vc0	-5.9467e+0...
VcL	-10.4789 + ...
VFW	*12x1 double*
VFWLL	*12x3 double*
VFWLN	*12x3 double*
VST	120.8587
VSTLL	[210,208,210]
VSTLN	[121.2436,1...
X1	8.1176
X1L	8.1176
X1prev	8.1150
X2	12.1159
Xm	124.3984
Xmprev	124.3059
XmRatio	1
Z2	143.5501

Figure B.4 MATLAB Workspace Non-Measured/Measured Parameters

Appendix C: MATLAB Script for State Space Inputs and Output Transformations

This MATLAB Code calculates. the state space inputs transformation matrix.

```
%This code will calculate the state transformation matrix and a transformed
matrix.

clear all
clc

%Input the obtained Matrix A

A=[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0; 0.0, -0.154, -0.04, 1.54, 0, -0.744, -
0.032;
0,  0.249, -1, -5.2,  0,   0.337,  -1.12; 0.0386,  -0.996,  0, -2.117,  0, 0.02,
0.0; 0.0, 0.50, 0.0, 0.0, -4.0, 0.0, 0.0; 0.0, 0.0, 0.0, 0.0, 0.0, -20.0, 0.0;
0.0, 0.0, 0.0,  0.0, 0.0, 0.0,  -25.0 ];

while (1), a=floor(2*rand(7,1));
T=sym([a  A*a  A^2*a  A^3*a  A^4*a  A^5*a A^6*a]);
if rank(T)== 7, break; end, end
T,
A1=inv(T)*A*T
T1 = inv(T*fliplr(eye(7))),
A2 = inv(T1)*A*T1,
T2 = inv(T1*fliplr(eye(7)))',
A3 = inv(T2)*A*T1,
T3 = inv(T2*fliplr(eye(7)))',
A4 = inv(T3)*A*T1,
T4 = inv(T3*fliplr(eye(7)))',

%The Nonsingular State Transformation Matrix (NSTM).
```

This program result yields the input transformed matrix T and the following matrices T1, T2, T3, T4, A1, A2, A3 and A4.

The following code will calculate the output transformation matrix and a transformed matrix.

```
%Code Begins
clear all
clc

%Input the obtained Matrix A

C=[0, -0.451, -0.41, 1.55, 0, -0.44, -0.052; 0, 1, -1, -4.55, 0, 0.474, -1.14;
1,  0, 0, 0,  0,   0,  0; 0,  0,  0, 0, 1,  0, 0; 0, 0, 0, 0, 0, 0, 0;
0.0, 0.0, 0.0, 0.0, 0.0, 10.1, 0; 0, 0, 0,  0, 0, 0,  1];

while (1), c=floor(2*rand(7,1));
S=sym([c  C*c  C^2*c  C^3*c  C^4*c  C^5*c C^6*c]);
if rank(S)== 7, break; end, end
S,
C1=inv(S)*C*S
S1 = inv(S*fliplr(eye(7))),
C2 = inv(S1)*C*S1,
S2 = inv(S1*fliplr(eye(7)))’,
C3 = inv(S2)*C*S1,
S3 = inv(S2*fliplr(eye(7)))’,
C4 = inv(S3)*C*S1,
S4 = inv(S3*fliplr(eye(7)))’,

%End of Code
```

The output transformation matrix is not unique, there are many. This program result yields the input transformed matrix T and the following matrices S1, S2, S3, S4, C1, C2, C3 and C4.

This MATLAB code calculates the matrix products “T*A/T

```
%Code Begins
>> A= [0, 0, 1, 0, 0, 0, 0; 0, -0.154, -0.04, 1.54, 0, -0.744, -0.032 ;
0, 0.249, -1, -5.2, 0, 0.337, -1.12; 0.0386,  -0.996,  0, -2.117,  0, 0.02, 0;
0, 0.5, 0, 0, -4, 0, 0; 0, 0, 0, 0, 0, -20, 0; 0, 0, 0,  0, 0, 0,  -25];

T= [0.9, 0.16, 0.04, -1.6, 0, 0.844, 0.03; -1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0;
0, 0, 1, 0, 0, 0, 0 ; 0, 0, 0, 1, 0, 0, 0 ; -1, 0, 0, 0, 1, 0, 0;
0, 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 0, 1];

(T*A)* inv(T)

ans =

    1.4313    1.3499    0.7963    5.7157    0.0000  -18.2256   -0.8429
   -0.1453   -0.1308   -1.0342    1.3075   -0.0000   -0.6214   -0.0276
    0.2349    0.2114   -1.0094   -4.8242    0.0000    0.1387   -1.1270
   -0.9032   -0.8515    0.0361   -3.5621   -0.0000    0.7823    0.0271
   -3.3019    1.0283   -0.8679   -5.2830   -4.0000    2.7868    0.0991
        0         0          0          0          0       -20.001   0
        0         0          0          0          0          0         -25.0000
```

The MATLAB matrix calculation of S*C/T is given below.

```
%Code Begins

>> C= [0, -0.4510, -0.410, 1.55, 0, -0.440, -0.0520; 0, 1, -1, -4.550, 0,
0.4740, -1.140; 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0; 0.0, 0.0, 0.0, 0.0, 1.0, 0.0,
0.0];

S= [1.0, 0.0, -1.0, 0.0; - 1.0, 1.0, -0.50, 0; 1, 0, 0, 0; 1, 0, -0.90, 1.0];

T= [ 0.90, 0.160, 0.040, -1.6, 0, 0.8441, 0.030; -1, 1, 0, 0, 0, 0, 0;
0, 0, 1, 0, 0, 0, 0; 0, 0, 0, 1, 0, 0, 0; -1, 0, 0, 0, 1, 0, 0;
0, 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 0, 1];

(S*C) * inv(T)


ans =

   -1.3689   -0.2320   -0.3552   -0.6402   -0.0000    0.7153   -0.0109
    0.8972    1.3075   -0.6259   -4.6645    0.0000    0.1568   -1.1149
   -0.4255   -0.3829   -0.3930    0.8692   -0.0000   -0.0809   -0.0392
   -0.3311   -0.3980   -0.3968    1.0202    1.0000   -0.1605   -0.0421
```

The MATLAB matrix calculation of T*B is below.

```
%Code Begins

>> T= [ 0.9, 0.16, 0.04, -1.6, 0, 0.844, 0.03; -1, 1, 0, 0, 0, 0, 0; 0, 0, 1, 0, 0, 0, 0;
0, 0, 0, 1, 0, 0, 0; -1, 0, 0, 0, 1, 0, 0;  0, 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 0, 1];

B= [0, 0; 0, 0; 0, 0; 0, 0; 0, 0; 7, 0; 0, 19];


(T*B)

ans =

   5.9080    0.5700
        0         0
        0         0
        0         0
        0         0
   7.000          0
        0    19.000
```

The MATLAB matrix calculation of T*E is below.

```
%Code Begins

>> T= [ 0.9, 0.16, 0.04, -1.6, 0, 0.844, 0.03; -1, 1, 0, 0, 0, 0, 0;
0, 0, 1, 0, 0, 0, 0; 0, 0, 0, 1, 0, 0, 0; -1, 0, 0, 0, 1, 0, 0;
0, 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 0, 1];

E =[1; 1; 0; 0; 1; 0; 0];


(T*E)

ans =
```

$[1.0600, 0, 0, 0, 0, 0, 0]^T$

Appendix D: MATLAB Code for the Fault Diagnosis and Simulink Functional Block Diagram and all Required Functions

The MATLAB Fault Diagnosis and Simulink Functional Block Diagram begins.

```
%Code begins

clear all
clc

% Given the Induction Motor System Matrices.
A= [0, 0, 1, 0, 0, 0, 0; 0, -0.1, -0.04, 1.5, 0, -0.7, -0.03;
    0, 0.2, -1, -5.2, 0, 0.3, -1.1; 0.03, -0.9, 0, -2.1, 0, 0.02, 0;
 0, 0.5, 0, 0, -4, 0, 0; 0, 0, 0, 0, 0, -20, 0; 0, 0, 0, 0, 0, 0, -25];

B= [0, 0; 0, 0; 0, 0; 0, 0; 0, 0; 7, 0; 0, 19];

C= [0, -0.15, -0.04, 1.54, 0 -0.7, -0.03; 0, 0.2, -1, -5.2, 0, 0.3, -1.1; 1, 0,
0, 0, 0, 0, 0; 0, 0, 0, 0, 1, 0, 0];

E = [1; 1; 0; 0; 1; 0; 0];

% Calculating the System Matrices Dimensions “n”, “m”
% “p” and “r”
n=size(A,1);
m=size(B,2);
p=size(C,1);
r=size(E,2);

% Use of Calculated Transformation Matrix T.
T = [0.84, 0.23, 0.04, -1.5, 0, 0.80, 0.03; -1, 1, 0, 0, 0, 0, 0;
    0, 0,1, 0, 0, 0, 0; 0, 0, 0, 1, 0, 0, 0; -1, 0, 0, 0, 1, 0, 0;
    0, 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 0, 1];

% Use of Calculated Output Transformation Matrix S.
```

```
S = [1, 0, -0.8, 0; -1.4, 1, -0.5, 0; 1, 0, 0.2, 0; 1, 0, -0.8,1];

T1=T(1,:);
T2=T(2:7,:);
S1=S(1,:);
S2=S(2:4,:);

% Calculation of Transformed System Matrices in the new
% coordinate.

A=T*A*inv(T);
B=T*B;
E=T*E;
C=S*C*inv(T);

% Calculation of Subsystem Matrices, which are in the new
% coordinate.

A1=A(1:r,1:r);
A2=A(1:r,r+1:n);
A3=A(r+1:n,1:r);
A4=A(r+1:n,r+1:n);

% Calculation of Subsystem Matrices, which are in the new
% Coordinate.


B1=B(1:r,:);
B2=B(r+1:n,:);

% Calculation of Subsystem Matrix, which are in the new
% coordinate.

E1=E(1:r,:);

% Calculation of Subsystem Matrices, which are in the new
% coordinate.

C1=C(1:r,1:r);
C4=C(r+1:p,r+1:n);

% As written below “lf1”, “f2”, are constant.
lf1=1;
f2 =0;
```

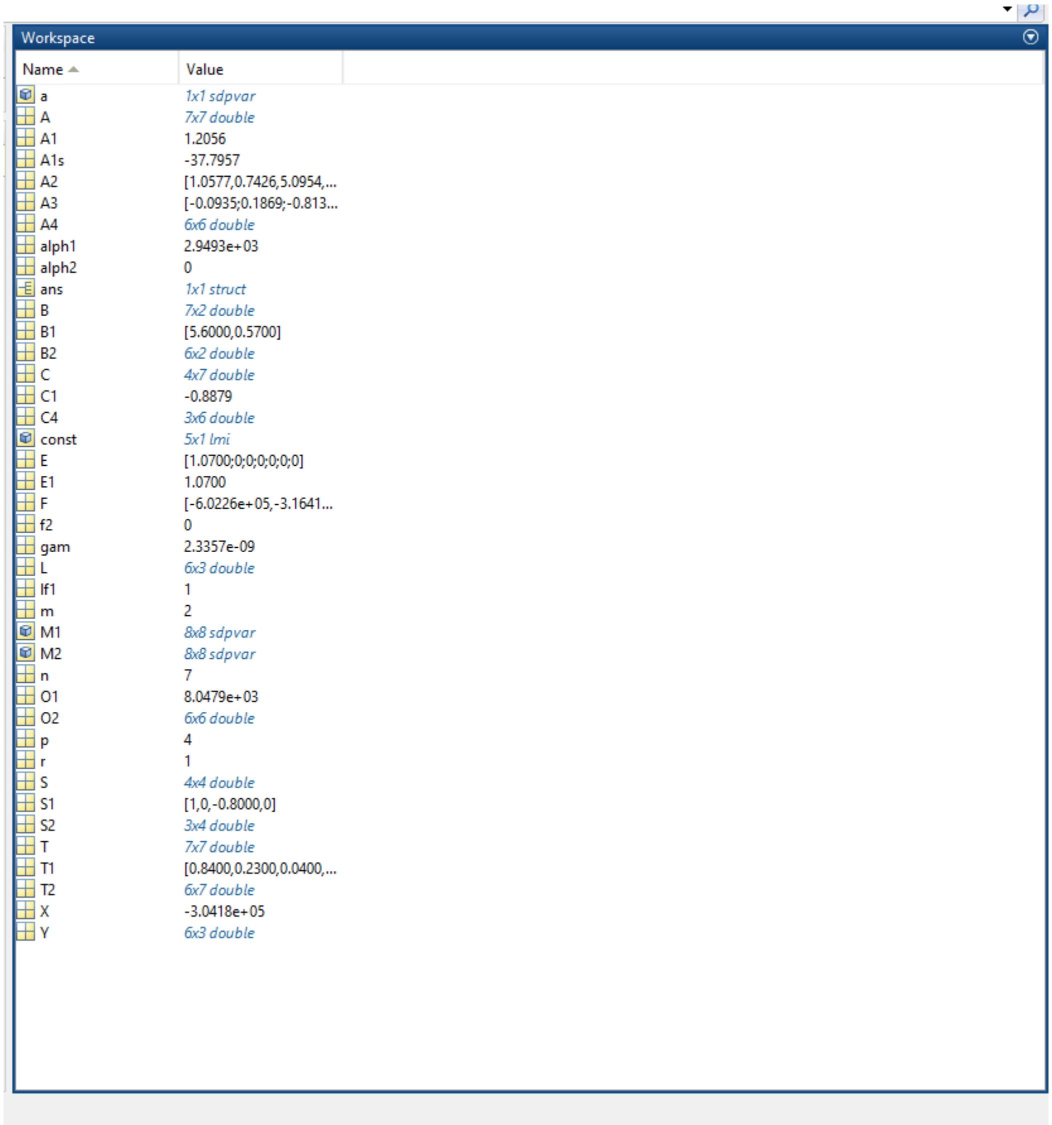
```
% “alph2” a is scalar.
alph2=0;
% Set the variables “O1= $O_1$”, “O2= $O_2$”, “$X = \left(O_1 A_1^{stable}\right)$”
% and “$Y = (O_2 L)$”, the matrix “F”, the scalar “alph1”, “gam”

%Since this is an optimization problem, with
%decision variables are dealt with. I download YALMIP to my home lab
%computer using this command
%Start script:
%urlwrite('https://github.com/yalmip/yalmip/archive/master.
%zip',%'yalmip.zip');
%unzip('yalmip.zip','yalmip')
%addpath(genpath([pwd filesep 'yalmip']));
%savepath
%End script
%and test the installation using the command “yamiptest”

%Decision variables in YALMIP are represented by the command %“sdpvar”
thus:

O1=sdpvar(r, r);
O2=sdpvar(n-r, n-r);
X=sdpvar(r,r);
Y=sdpvar(n-r, p-r);
F=sdpvar(m, p-r);
alph1=sdpvar(1,1);
gam=sdpvar(1,1);

a=alph1*lf1^2*(norm(inv(T)))^2;

 % The First Linear Matrix Inequality (LMI) Calculation

M1=[X+X' O1 O1*A2
     O1 -alph1 zeros(r,n-r)
     A2'*O1 zeros(n-r,r) A4'*O2+O2*A4-C4'*Y'-Y*C4+a*eye(n-r)];

% The Second Linear Matrix Inequality (LMI) Calculation

M2= [-gam*eye(n-r) (B2'*O2-F*C4)'
      B2'*O2-F*C4 -gam*eye(m)];

const = [M1<0,M2<0,O1>0,O2>0,alph1>0];
solvesdp(const,gam);
```

```
% “O1” and “O2” dummy variables.
% “Y” and “F” are used for convenience
% Recall “$Y = (O_2L)$”, “F” is a Matrix, “alph1” and “gam” are scalar
O1=double(O1)
O2=double(O2)
F=double(F)
X=double(X);
Y=double(Y);
A1s=inv(O1)*X
L=inv(O2)*Y
alph1=double(alph1)
gam=double(gam)

zh=[inv(C1)*u(3);[x(2);x(3);x(4);x(5);x(6);x(7)]];
xh=inv(T)*zh;
xh3=xh(3,:);
f=[sin(xh3);sin(xh3);0;0;sin(xh3);0;0];
f1=T1*f;
f2=T2*f;

% proposed SMO for subsystem 1
e1=inv(C1)*(u(3)-C1*x(1));

if e1
   nu1=0;
else
   nu1=3*O1*e1/(norm(O1*e1)+0.01);
end
```

The Results from MATLAB Code are below. This is the solver for linear objective minimization under LMI constraints. The iterations best objective value so far is below.

```
* switching to QR
    1
    2              0.238668
    3              0.049671
    4              0.028506
    5              0.015189
    6              0.015189
    7             2.268839e-03
    8             8.712340e-04
    9             2.662916e-04
   10              1.345352e-04
   11              1.711341e-05
   12              1.711341e-05
   13              1.711341e-05
   14              1.670530e-06
   15              1.670530e-06
   16              1.670530e-06
   17              8.591267e-08
   18              8.591267e-08
   19              8.591267e-08
   20              2.269219e-08
   21              2.269219e-08
   22              5.689903e-09
   23              5.689903e-09
   24              5.689903e-09
   25              5.689903e-09
   26              2.335687e-09
   27              2.335687e-09
   28              2.335687e-09
   29              2.335687e-09
   30              2.335687e-09
   31              2.335687e-09
   32              2.335687e-09
   33              2.335687e-09
   34              2.335687e-09
   35              2.335687e-09
   36              2.335687e-09

 Result:  feasible solution
         best objective value: 2.335687e-09
         f-radius saturation:  1.324% of R =  1.00e+09
 Termination due to SLOW PROGRESS:
         the objective was decreased by less than
         0.100% during the last 10 iterations.

##################################################
You are using LMILAB. Please don't use LMILAB with YALMIP
https://yalmip.github.io/solver/lmilab/

Install a better SDP solver
https://yalmip.github.io/allsolvers/
```

```
To get rid of this message, edit calllmilab.m
(but don't expect support when things do not work,
YALMIP + LMILAB => No support)
##############################################

O1 =

   8.0479e+03

O2 =

   1.0e+06 *

    0.7729    0.0122   -0.3307   -0.6721    0.0343    0.0143
    0.0122    0.5725    0.1577    0.2221    0.0995    0.0698
   -0.3307    0.1577    1.2283    0.5359   -0.0699    0.0564
   -0.6721    0.2221    0.5359    1.1852    0.0087    0.0064
    0.0343    0.0995   -0.0699    0.0087    0.2109    0.1047
    0.0143    0.0698    0.0564    0.0064    0.1047    0.0743


F =

   1.0e+06 *

   -0.6023   -3.1641    0.0612
   -1.1930   -4.9997    0.1213


A1s =

  -37.7957

L =

    2.3819    5.9244    1.4400
   -2.7703  -11.8487   -1.2303
   -3.8537  -11.8087    0.3798
    4.6140   13.3747   -3.0116
  -21.0300  -41.8140   -6.1198
   55.0637  151.3668   24.4748

alph1 =
   2.9493e+03
gam =

   2.3357e-09
```



Workspace

Name	Value
a	*1x1 sdpvar*
A	*7x7 double*
A1	1.2056
A1s	-37.7957
A2	[1.0577,0.7426,5.0954,...
A3	[-0.0935;0.1869;-0.813...
A4	*6x6 double*
alph1	2.9493e+03
alph2	0
ans	*1x1 struct*
B	*7x2 double*
B1	[5.6000,0.5700]
B2	*6x2 double*
C	*4x7 double*
C1	-0.8879
C4	*3x6 double*
const	*5x1 lmi*
E	[1.0700;0;0;0;0;0;0]
E1	1.0700
F	[-6.0226e+05,-3.1641...
f2	0
gam	2.3357e-09
L	*6x3 double*
lf1	1
m	2
M1	*8x8 sdpvar*
M2	*8x8 sdpvar*
n	7
O1	8.0479e+03
O2	*6x6 double*
p	4
r	1
S	*4x4 double*
S1	[1,0,-0.8000,0]
S2	*3x4 double*
T	*7x7 double*
T1	[0.8400,0.2300,0.0400,...
T2	*6x7 double*
X	-3.0418e+05
Y	*6x3 double*

Figure D.1 The MATLAB Workspace for the Ran Script

Run the MATLAB Simulink KP_Motor2 as in the figure below.

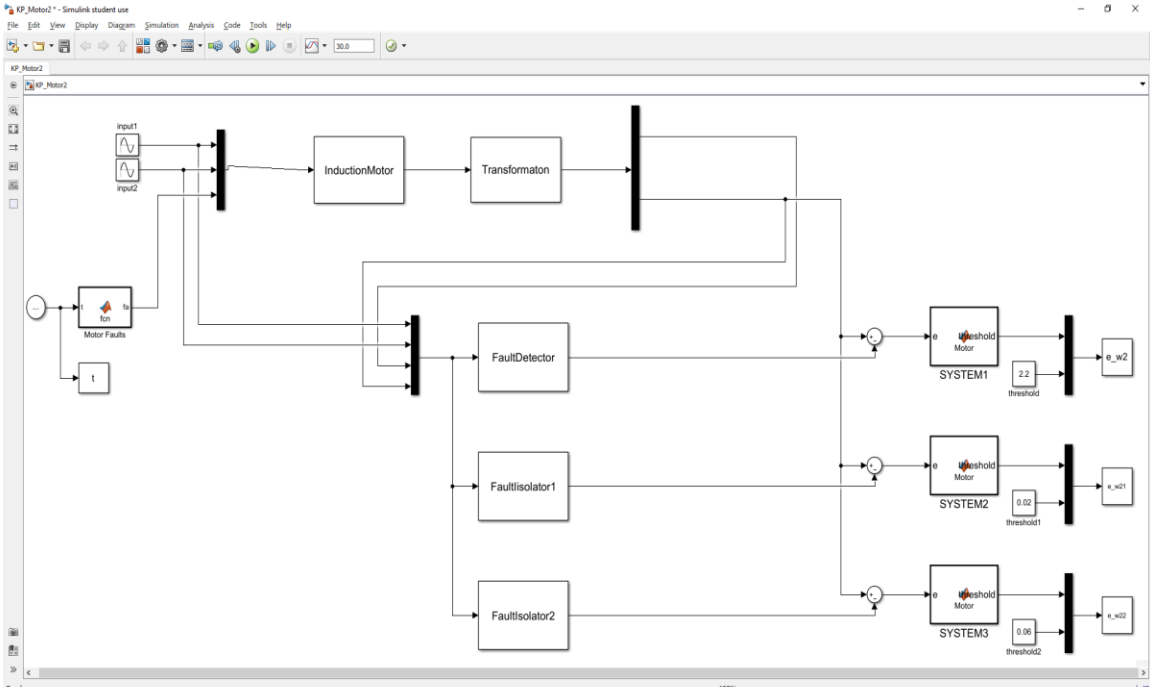


Figure D.2 MATLAB Simulink Functional Block Diagram

KP_Motor2 Simulink calls for a series of short functions named below. Functions 6, 7 and 8 codes are provided below. The codes for functions 1, 2, 3, 4 and 5 are available upon request for interested readers.

- InductionMotor.m
- Transformation.m
- FaultIsolator1.m
- FaultIsolator2.m
- MototFault.m
- SYSTEM1.m
- SYSTEM2.m
- SYSTEM3.m

SYSTEM1.m

```
%Function Code Begins

function threshold = Motor(e)
threshold =norm(e);

%Function Code Ends
```

SYSTEM2.m

```
%Function Code Begins

function threshold = Motor(e)
threshold =norm(e);

%Function Code Ends
```

SYSTEM3.m

```
%Function Code Begins

function threshold = Motor(e)
threshold =norm(e);

%Function Code Ends
```

After running the “MATLAB Simulink KP_Motor2” the FDFI_graph.m is run. This FDFI_graph.m MATLAB code graphs the residual, threshold as a function of time to visualize the detected and isolated faults in the motor.

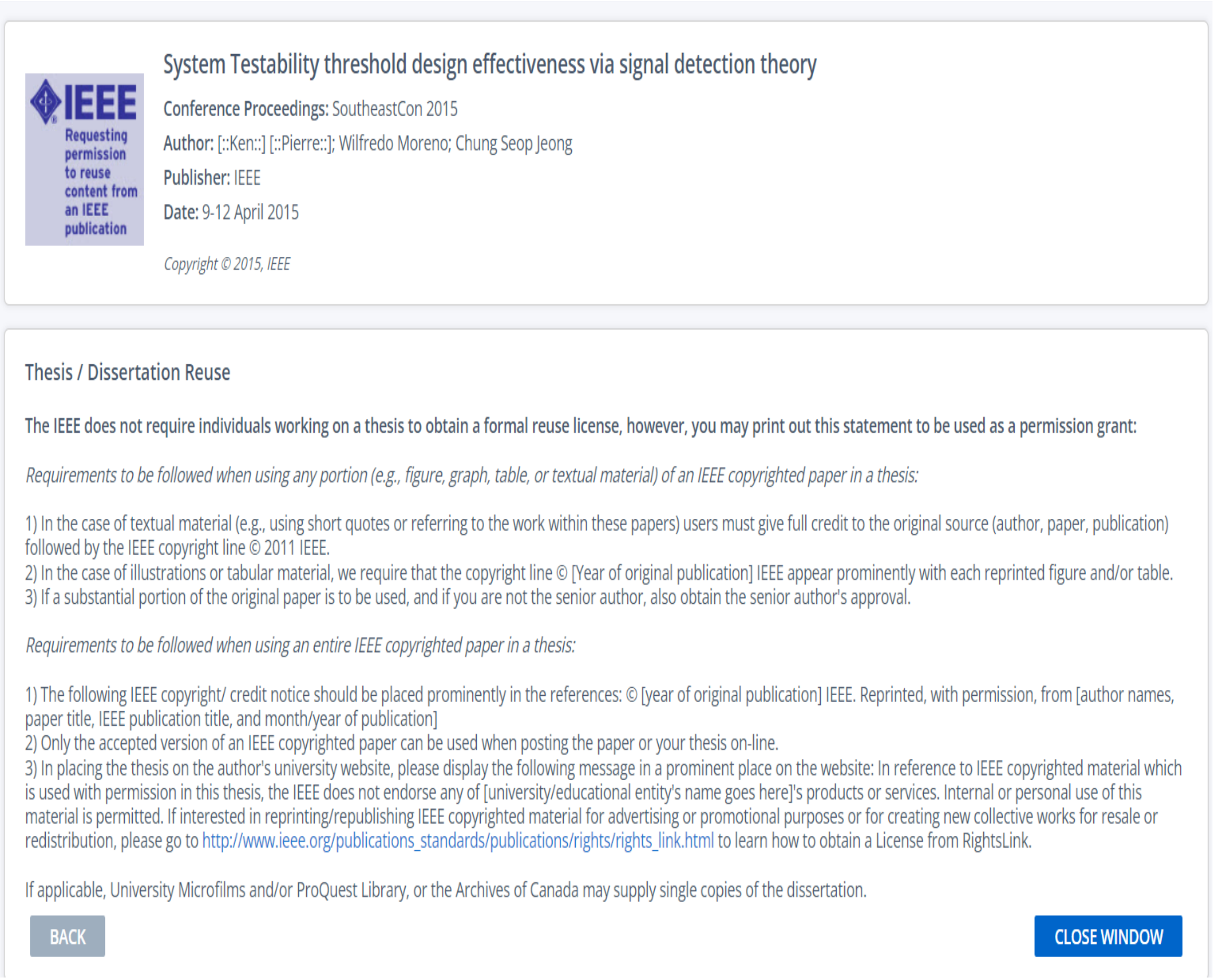
Appendix E: MATLAB Code Graphing Residuals and Thresholds

This section graphs the residual, threshold as a function of time.

```
% To Visualize the Detected and Isolated Faults.

close all;

figure(1) %
plot(t,e_w2(:,1),'b',t,e_w2(:,2),'--r','linewidth',2); % The graph command
xlabel('Time (Second)'); % The x axis label
ylabel('Residual Magnitude'); %The y axis label
legend('Residual Curve', 'Threshold for Motor Fault Detection');% The Curve
legend

figure(2) %
plot(t,e_w21(:,1),'b',t,e_w21(:,2),'--r','linewidth',2); % The graph command
xlabel('Time (Second)'); % The x axis label
ylabel('Residual Magnitude'); %The y axis label
legend('Residual Curve', 'Threshold for Motor Fault Isolation'); % The Curve
legend


figure(3)  %
plot(t,e_w22(:,1),'b',t,e_w22(:,2),'--r','linewidth',2); % The graph command
xlabel('Time (Second)'); % The x axis label
ylabel('Residual Magnitude'); %The y axis label
legend('Residual Curve', 'Threshold for Motor Fault Isolation');% The Curve
legend
```

Appendix F: Copyright Permissions

The referenced paper [142] by the author in this Dissertation is the Senior author that writes the paper. No Copyright permission is needed per IEEE rules. See IEEE Copyright guidance below.


IEEE
Requesting permission to reuse content from an IEEE publication

System Testability threshold design effectiveness via signal detection theory

Conference Proceedings: SoutheastCon 2015

Author: [::Ken::] [::Pierre::]; Wilfredo Moreno; Chung Seop Jeong

Publisher: IEEE

Date: 9-12 April 2015

Copyright © 2015, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

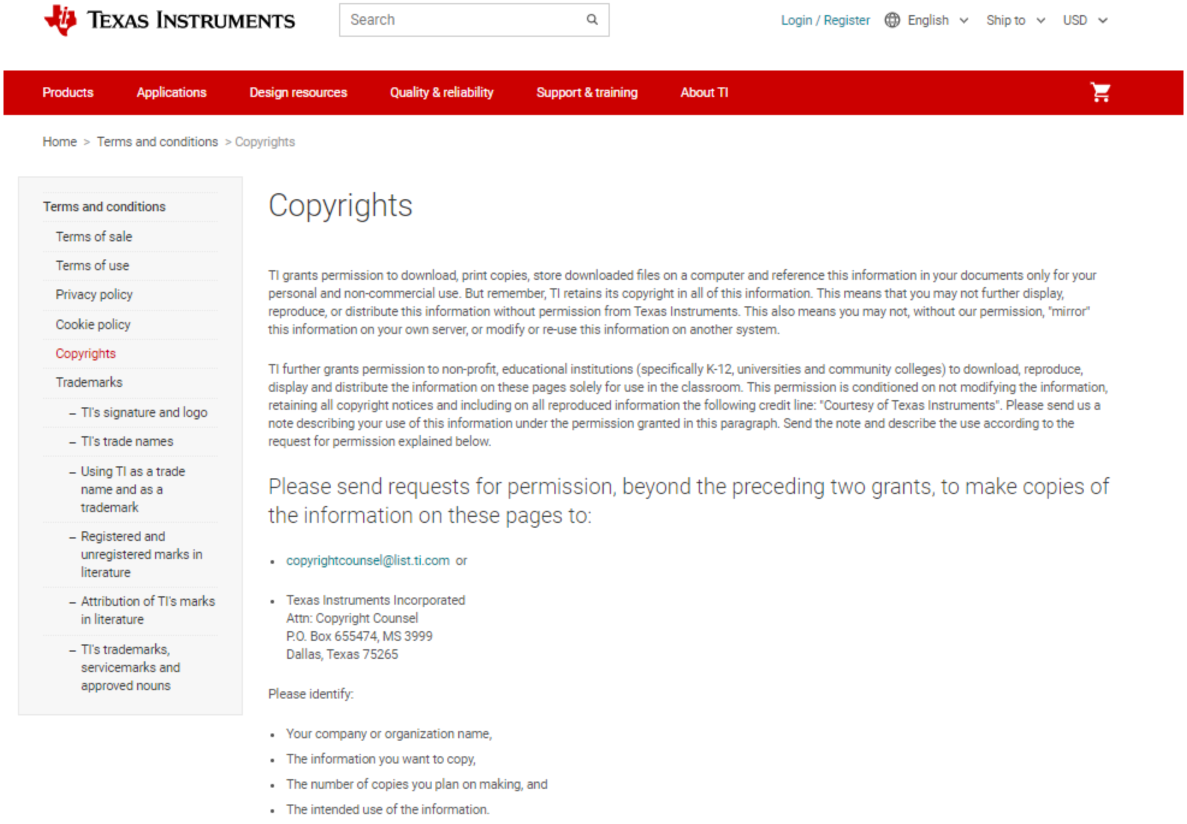
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

The permission below is for the of all referenced TI pictures and other TI items referenced in this dissertation. The Induction Motor, the software and most controller board are bought from TI.


Texas Instruments
Search
Login / Register
English
Ship to
USD
Products
Applications
Design resources
Quality & reliability
Support & training
About TI
Home > Terms and conditions > Copyrights
Terms and conditions
Terms of sale
Terms of use
Privacy policy
Cookie policy
Copyrights
Trademarks
– TI's signature and logo
– TI's trade names
– Using TI as a trade name and as a trademark
– Registered and unregistered marks in literature
– Attribution of TI's marks in literature
– TI's trademarks, servicemarks and approved nouns

Copyrights

TI grants permission to download, print copies, store downloaded files on a computer and reference this information in your documents only for your personal and non-commercial use. But remember, TI retains its copyright in all of this information. This means that you may not further display, reproduce, or distribute this information without permission from Texas Instruments. This also means you may not, without our permission, "mirror" this information on your own server, or modify or re-use this information on another system.

TI further grants permission to non-profit, educational institutions (specifically K-12, universities and community colleges) to download, reproduce, display and distribute the information on these pages solely for use in the classroom. This permission is conditioned on not modifying the information, retaining all copyright notices and including on all reproduced information the following credit line: "Courtesy of Texas Instruments". Please send us a note describing your use of this information under the permission granted in this paragraph. Send the note and describe the use according to the request for permission explained below.

Please send requests for permission, beyond the preceding two grants, to make copies of the information on these pages to:

- copyrightcounsel@list.ti.com or
- Texas Instruments Incorporated
  Attn: Copyright Counsel
  P.O. Box 655474, MS 3999
  Dallas, Texas 75265

Please identify:

- Your company or organization name,
- The information you want to copy,
- The number of copies you plan on making, and
- The intended use of the information.

https://www.ti.com/legal/terms-conditions/copyright.html

Appendix G: YALMIP Installation

YALMIP is a solver compatible with MATLAB. More information and installation are available in the solver manual. The required paths must be added. to your MATLAB installation that may already have solvers available that YALMIP can interface.

Six steps towards a successful installation[1]
Uninstall old versions of YALMIP
unzip yalmip.zip. This should create the structure

- /yalmip
- /yalmip/@sdpvar
- /yalmip/extras
- /yalmip/solvers
- /yalmip/modules
- /yalmip/operators

Copy these paths in your MATLAB path

- /yalmip

[1] For detail visit YALMIP site. "https://yalmip.github.io/" The YALMIP installation here is for ease of access.

- /yalmip/extras
- /yalmip/solvers
- /yalmip/modules
- /yalmip/modules/parametric
- /yalmip/modules/moment
- /yalmip/modules/global
- /yalmip/modules/robust
- /yalmip/modules/sos
- /yalmip/operators

This can be done either using the Graphic User Interface (GUI) or using the command "addpath" as in:

```
addpath(genpath('yourcurrentlocation/yalmip'))
```

- Verify you have the right solvers in your Matlab path.
- Shut down and restart Matlab, and or type "clear classes".
- Run the yalmiptest.m file and everything should work (as long as you have the necessary solvers).

About the Author

Kenelt Pierre is an Electrical Engineering Ph.D. Candidate at the University of South Florida in Florida. He completed the doctoral courses in RF-Microwave engineering and control engineering. He has been at Lockheed Martin for fifteen years, where he is a Senior Staff System Engineer working on Reliability Engineering, Testing, and Fault Diagnosis. He worked at Bell Labs, British Aerospace Engineering, Northrop Grumman, Jabil Circuit, and the Department of Commerce Patent and Trademark Offices. He obtained a Bachelor of Science in Electrical Engineering (BSEE) at the City University of New York (CUNY) and a Master of Science in Electrical Engineering (MSEE) Microelectronics at USF. He wrote his MSEE thesis in "Microelectronic Devices Performance and Ruggedness" at Bell Labs. Ken wrote three papers while completing the doctoral RF-MW course work.

The papers are titled as follows: 1) "System Testability Threshold Design Effectiveness Via Signal Detection Theory," 2) "The Foundations of Numerical Techniques in Electromagnetics Underlying Physics, Computational Analysis, and Design Aspects" 3) "Outline of Techniques to Assess Ruggedness and Lifetime of Gallium Nitride GaN RFMW Power Amplifier under High Voltage Standing Wave Ratio VSWR and other Harsh Environments." He presented some of his papers at IEEE conferences.