

The Islamic University of Gaza
Deanship of Research and
graduate students
Faculty of Engineering
Master of Electrical Engineering



الجامعة الإسلامية بغزة
عمادة البحث العلمي
والدراسات العليا
كلية الهندسة
ماجستير الهندسة الكهربائية

Fault Tolerant Control System Design In Electrical Machines

تصميم نظام التحكم في التسامح مع الخطأ في الآلات الكهربائية

By

Yousef S. S. Almutayeb

Supervised by

Dr. Moayed Almobaied

Asst. Professor in Electrical Engineering

**A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Electrical Engineering**

June -2020

Abstract

Fault Tolerant Control System (FTCS) have become an important area of research for improving safety, reliability and efficiency of modern control systems. In this thesis, the Fault Detection and Diagnosis (FDD) method well known for improving system reliability and reducing maintenance cost is studied.

Detecting and diagnosing faults methodologies depend on the process and the type of data available. Hence, these methodologies can be divided into two types: model-based methods and data-based methods. The work in this thesis focused on the investigation of the use of Luenberger observer technique, one of model-based methods. The selected method was implemented and experimentally evaluated.

Electrical motors become more popular and more useful especially in the industry control applications. The method was applied on two types of electrical motors: Brushless DC motor (BLDC) and Permanent Magnet DC motor (PMDC).

The aim of this thesis is to study faults detection of electrical motors, particularly faults detected in the speed sensor of electrical motors.

The Luenberger observer technique was used for fault detection purpose in speed sensor based on residual signal. The residual signal is the difference between the measured signal from real process and the estimated signal from the model. It was used in the system as a fault indicator.

The proposed technique was simulated and evaluated using the MATLAB / Simulink environment and was also implemented experimentally (in real time) on the motors used in this study.

The obtained results from real-time experiment showed close match with those from simulation thus proving the accuracy and reliability of the proposed methodology for fault detection in the motor speed sensor.

ملخص الدراسة

أصبح نظام التحكم في تحمل مع الأعطال (FTCS) مجالاً مهماً للبحث لتحسين السلامة والموثوقية والكفاءة لأنظمة التحكم الحديثة. في هذه الأطروحة ، تمت دراسة طريقة الكشف عن الأخطاء وتشخيصها (FDD) المعروفة جيداً لتحسين موثوقية النظام وتقليل تكلفة الصيانة.

تعتمد منهجيات اكتشاف وتشخيص الأعطال على العملية ونوع البيانات المتاحة. وبالتالي، يمكن تقسيم هذه المنهجيات إلى نوعين: الطرق القائمة على النموذج والطرق القائمة على البيانات. ركّز العمل في هذه الأطروحة على التحقيق في استخدام تقنية مراقب لينبرغر (Luenberger) ، إحدى الطرق القائمة على النموذج. تم تنفيذ الطريقة المختارة وتقييمها تجريبياً.

أصبحت المحركات الكهربائية أكثر شيوعاً وأكثر فائدة خاصة في تطبيقات التحكم في الصناعة. تم تطبيق الطريقة على نوعين من المحركات الكهربائية: محرك تيار مستمر عديمة الفرش الكربونية ومحرك تيار مستمر ذو إثارة دائمة. الهدف من هذه الأطروحة هو دراسة كشف الأعطال للمحركات الكهربائية ، وخاصة الأعطال المكتشفة في مستشعر السرعة للمحركات الكهربائية.

تم استخدام تقنية مراقب لينبرغر (Luenberger) لغرض الكشف عن الخطأ في مستشعر السرعة بناءً على الإشارة المتبقية وهي الفرق بين الإشارة المقاسة من العملية الحقيقية والإشارة المقدّرة من النموذج. تم استخدامه في النظام كمؤشر خطأ.

تمت محاكاة التقنية المقترحة وتقييمها باستخدام بيئة MATLAB / Simulink وتم تنفيذها أيضاً تجريبياً (في الوقت الفعلي) على المحركات المستخدمة في هذه الدراسة.

أظهرت النتائج التي تم الحصول عليها من تجربة الوقت الحقيقي تطابقاً وثيقاً مع نتائج المحاكاة مما يثبت دقة وموثوقية المنهجية المقترحة لاكتشاف الأعطال في مستشعر سرعة المحرك.

Table of Contents

Declaration	I
Abstract.....	III
ملخص الدراسة.....	IV
Dedication	V
Acknowledgment.....	VI
Table of Contents	VII
List of Tables	X
List of Figures.....	XI
List of Abbreviations	XIV
Chapter 1 Introduction	1
1.1 Background and Motivation	2
1.2 Problem statement.....	3
1.3 Thesis Objectives	3
1.4 Contributions	4
1.5 Literature review	4
1.6 Organization of the Thesis	5
Chapter 2 Mathematical Models of BLDC and PMDC Motors.....	7
2.1 Introduction of Electric motors	8
2.2 Principle Of Operation for BLDC motor	9
2.3 BLDC motor mathematical model.....	11
2.3.1 Background information of the MOONS' BLDC motor	11
2.3.2 Electrical equations	13
2.3.3 Mechanical equations	15
2.4 Transfer function of BLDC motor	16
2.5 State space model of the MOONS' BLDC motor	17
2.5.1 Controllability and Observability of BLDC motor	18
2.6 PMDC motor mathematical model	18
2.6.1 State space model of PMDC motor	20
Chapter 3 Fault Tolerant Control System (FTCS).....	22
3.1 Introduction.....	22
3.2 Fundamental Definitions:	24

3.3 Types of faults	25
3.3.1 (Nature Fault) Based on the location of fault	25
3.3.2 Time dependent fault.....	26
3.3.3 Model of fault	27
3.4 Fault detection methods	28
3.4.1 Model-Based Methods	29
3.4.1.1 Quantitative model based method	29
3.4.2 State Estimation Methods.....	29
3.4.3 Luenberger Observer Method	29
Chapter 4 Hardware and Software Design & Implementation.....	34
4.1 Introduction.....	35
4.2 Design and implementation for PMDC motor using Simulink	36
4.3 Design and implementation for BLDC motor using Simulink.....	38
4.4 Luenberger observer design using Simulink	45
4.5 Experimental setup for PMDC motor	46
4.6 Details of Hardware implementation	47
4.6.1 Motor controller (Arduino Mega 2650)	47
4.6.2 L298N H Bridge Dual Motor Drive	48
4.6.3 Optical encoder.....	49
4.7 Hardware/ Practical Implementation	50
4.7.1 Experimental setup for BLDC motor	51
4.7.2 Details of Hardware implementation	52
4.7.2.1 Motor controller (Arduino Mega 2650) and interface circuit	52
4.7.2.2 Inverter circuit design.....	53
4.7.2.3 Hall effect Sensors.....	55
Chapter 5 Simulation and Experimental Results	59
5.1 Introduction.....	60
5.2 Simulation and experimental results for PMDC motor	63
5.2.1 Results of simulation without fault for PMDC motor	63
5.2.2 Experimental results without fault for PMDC motor.....	64
5.2.3 Simulation results of abrupt fault for PMDC motor.....	66
5.2.4 Experimental results of abrupt fault for PMDC motor.....	67

5.2.5 Simulation results of incipient fault for PMDC motor.....	68
5.2.6 Experimental results of incipient fault for PMDC motor.....	69
5.2.7 Simulation results of intermittent fault for PMDC motor	70
5.2.8 Experimental results of intermittent fault for PMDC motor	72
5.2.9 Simulation results of sensor fault for PMDC motor.....	73
5.2.10 Experimental results of sensor fault for PMDC motor.....	74
5.3 Simulation and practical results for BLDC motor	75
5.3.1 Simulation results without fault for BLDC motor.....	75
5.3.2 Experimental results without fault for BLDC motor.....	76
5.3.3 Simulation results of abrupt fault for BLDC motor	79
5.3.4 Simulation results of incipient fault for BLDC motor	80
5.3.5 Simulation results of intermittent fault for BLDC motor	81
5.3.6 Simulation results of sensor fault for BLDC motor	83
5.3.7 Experimental results of sensor fault for BLDC motor	84
Chapter 6 Conclusion and Future Work	85
6.1 General conclusion	86
6.2 Future work.....	86
References.....	88
Appendices.....	92
Appendix A.....	92
Appendix B	95
Appendix C	98
Appendix D.....	100

List of Tables

Table (2.1) : Shows the differences between BLDC motor and DC motor.....	9
Table (2.2): MOSFET switching conditions.....	11
Table (2.3): 42BL30L2-5 MOONS' BLDC motor specifications.....	13
Table (2.4): YA-070 PMDC motor specifications.....	19
Table (4.1): Arduino Mega 2560 specification.....	48
Table (4.2): Technical specifications of components and driving parameters.	55

List of Figures

Figure (2.1): The differences between BLDC motor and PMDC motor (Mukkar, 2017).	9
Figure (2.2): Switching sequence of three-phase delta winding connection(Ridwan,et al., 2016)	10
Figure (2.3): Equivalent circuit of delta winding connection for BLDC motor. (Chan,et al. , 2015).....	12
Figure (2.4): 42BL30L2-5 MOONS' BLDC motor.....	12
Figure (2.5): One commutation step of a three-phase BLDC motor.	13
Figure (2.6): PMDC motor equivalent circuit.	19
Figure (2.7): YA-070 PMDC motor.	19
Figure (3.1): Classification of (FTC).....	23
Figure (3.2): Block diagram of the parts Active fault-tolerant control (ATFC).....	24
Figure (3.3): Types of faults.	25
Figure (3.4): Types of locations faults.....	25
Figure (3.5): Time behaviour of faults.....	26
Figure (3.6): Additive fault model.	27
Figure (3.7): Multiplicative fault model.	27
Figure (3.8): Methods of fault detection.	28
Figure (3.9): Block diagram of the quantitative model method.	29
Figure (3.10): Functional diagram of real system and observer model.	31
Figure(3.11): Flowchart of Luenberger observer for fault detection.....	32
Figure (3.12): Speed Luenberger observer for MOONS' BLDC.	32
Figure (3.13): Speed Luenberger observer for YA-070 DC motor.	33
Figure (4.1): Simulink Support Package for Arduino Hardware.....	35
Figure (4.2): Simulink support for Arduino sensors.....	36
Figure (4.3): Simulink model of PWM output.	36
Figure (4.4): Simulink design to find speed.	37
Figure (4.5): Tachometer to measure speed in RPM.	37
Figure (4.6): Overall Simulink design for PMDC motor.....	37
Figure (4.7): PWM generation.....	38
Figure (4.8): Internal Simulink design of PWM.....	38
Figure (4.9): Internal Simulink design of dead time.....	39
Figure (4.10): Complementary PWM pulses with dead time	39
Figure (4.11): Design of Hall-effect sensors in display.....	40
Figure (4.12): Hall sensor signals.	40
Figure (4.13): MOSFET switch	41
Figure (4.14): Decoder block design.	41
Figure (4.15): Internal Simulink design for decoder block.	41
Figure (4.16): Gates block design.....	42
Figure (4.17): Internal Simulink design for gates block.....	42
Figure (4.18): Outer Simulink design to find speed in RPM.....	43
Figure (4.19): Internal Simulink design to find speed in RPM.	43
Figure (4.20): Measured speed output of MOONS' BLDC motor in RPM.....	44
Figure (4.21): Overall Simulink design for BLDC motor.	44
Figure (4.22): The Luenberger observer block.....	45
Figure (4.23): Block parameters of Luenberger observer.....	45

Figure (4.24): Schematic diagram of the equipment setup.	46
Figure (4.25): Photograph of equipment used.	47
Figure (4.26): Arduino Mega 2650 board (motor controller).	48
Figure (4.27): Motor driver (L298N H-bridge).	49
Figure (4.28): Optical encoder signals.	49
Figure (4.29): H9700 optical incremental encoder.	50
Figure (4.30): Optical sensor output (Scale:2 V/div).	50
Figure (4.31): Schematic diagram of hardware implementation.	51
Figure (4.32): Photo of the experimental setup of the BLDC motor.	52
Figure (4.33): Simplified diagram of the motor controller.	52
Figure (4. 34): Photo of the PCB board.	53
Figure (4.35): Internal block diagram of the IR2110.	54
Figure (4.36): Electronic block diagram of the IR2110 Driver (Franceschet,et al., 2017).	54
Figure (4.37): Photo of the PCB board.	55
Figure (4.38): Hall effect principle.	56
Figure (4.39): Block diagram of the hall sensor circuit.	56
Figure (4.40): Photo of the PCB board.	57
Figure (4.41): Testing circuit for hall sensor.	57
Figure (4.42): Hall sensor output (Scale:2 V/div).	58
Figure (5.1): Simulation design of Luenberger observer for PMDC motor.	60
Figure (5.2): Simulation design of Luenberger observer for BLDC motor.	60
Figure (5.3): Simulink design (real-time) of a Luenberger observer for PMDC motor.	61
Figure (5.4): Simulink design (real-time) of a Luenberger observer for BLDC motor.	61
Figure (5.5): Simulation of output speed without fault.	63
Figure (5.6): Simulation of estimated output speed without fault.	63
Figure (5.7): Simulation of residual output with no fault for PMDC motor.	64
Figure (5.8): The measured speed of PMDC motor.	64
Figure (5.9): The estimated speed of PMDC motor.	65
Figure (5.10): Residual output without any fault, upper threshold 0.0157, lower threshold value was - 0.009.	65
Figure (5.11): Simulation of output and estimated speeds with abrupt fault.	66
Figure (5.12): Simulation of residual output with abrupt fault.	66
Figure (5.13): Measured and estimated speeds of PMDC motor with Abrupt fault.	67
Figure (5.14): Residual output with Abrupt fault.	68
Figure (5.15): Simulation of output and estimated speeds with incipient fault.	68
Figure (5.16): Simulation of residual output with incipient fault.	69
Figure (5.17): Measured and estimated speeds of PMDC with incipient fault.	69
Figure (5.18): Residual output with incipient fault.	70
Figure (5.19): Simulation of output and estimated speeds with intermittent fault. ...	71
Figure (5.20): Simulation of residual output with intermittent fault.	71
Figure (5.21): Measured and estimated speeds of PMDC with intermittent fault.	72
Figure (5.22): Residual output with intermittent fault.	72
Figure (5 .23): Simulation of output and estimated speeds of PMDC motor with sensor fault.	73
Figure (5 .24): Simulation of residual output with sensor fault.	73

Figure (5.25): Measured and estimated speed with sensor fault.....	74
Figure (5.26): Residual output with sensor fault.	74
Figure (5.27): Simulation of output speed without fault.	75
Figure (5.28): Simulation of estimated speed without fault.	76
Figure (5.29): Simulation of residual output with no fault.	76
Figure (5.30): The measured speed of the BLDC motor.	77
Figure (5.31): The estimated speed of the BLDC motor.	77
Figure (5.32): Residual output with upper threshold value is 0.05 and lower threshold value is - 0.01848.....	78
Figure (5.33): Residual output with upper threshold value is 0.0425 and lower threshold value is - 0.01848.....	78
Figure (5.34): Simulation of the output and estimated speeds of the BLDC motor with Abrupt fault.....	79
Figure (5.35): Simulation of residual output with Abrupt fault.....	80
Figure (5.36): Simulation of the output and estimated speeds of the BLDC motor with incipient fault.....	81
Figure (5.37): Simulation of residual output with incipient fault.....	81
Figure (5.38): Simulation of the output and estimated speeds of the BLDC motor with intermittent fault.	82
Figure (5.39): Simulation of residual output with intermittent fault.	82
Figure (5.40): Simulation of the output and estimated speeds of BLDC motor with sensor fault.....	83
Figure (5.41): Simulation of residual output with sensor fault.....	83
Figure (5.42): Measured and estimated speeds with sensor fault.	84
Figure (5.43): Residual output with sensor fault.	84
Figure(A.1): YA-070/071 PMDC motor.	93
Figure (C.1): Schematic of the inverter circuit.	98
Figure (C.2): Schematic of the IR2110 Driver circuit.	98
Figure (C.3): Schematic of the hall sensor circuit.	99
Figure (C.4): Schematic of the L298N H-bridge circuit (motor driver).....	99

Chapter 1

Introduction

1.1 Background and Motivation

Fault Tolerant Control System (FTCS) is a class of highly advanced control functions designed in a unified framework to provide high levels of system safety. There are two types of FTCS: the passive type and active type (Gaeid, K. S., & Ping, H. W., 2011). Fault detection and diagnosis (FDD) was studied in this thesis which is of the active fault tolerant control type, as will be explained in Chapter 3. Over the past twenty years, Fault Tolerance Control (FTC) has attracted a lot of attention in most industrial applications. In a wide range of industrial applications, there is a growing need to improve the reliability and safety of electrical systems. Examples of industrial applications are electrical motors, aircraft systems, and industrial production lines. Electrical motors are an essential component in many electrical systems, where the operating conditions of electric motors are important for the reliability of the entire system (Liu, L, 2006). Fault detection and diagnosis (FDD) improves the reliability and safety of electric motors (ALKAYA, 2012). Therefore, reliable FDD technology is needed for electrical motors, research in this area has attracted much attention. A sudden fault of a system in these examples (motors) can lead to costly downtime, damage to surrounding equipment or even danger to humans. Since several faults deteriorate relatively slowly, the faults can be detected at the early stage which can increase safety, prevent system failures, product damage and extend the life of the equipment (Miljković, 2011). Faults that occur in electrical motors can be classified in two types: electrical and mechanical faults. Electrical faults include faults caused by a short circuit in the stator winding, rotor breakdown, a fault in the inverter and a broken end ring. Mechanical faults include faults caused by bearing damage, gearbox fault, shaft misalignment and rotor eccentricity (Abed, 2015); (Liu, L, 2006). Sensors are important components for the operation of motors. Sensor faults represent a major challenge when investigating a fault detection and diagnosis of motors. As a general rule, two methods of monitoring and detecting faults in electric motors are distinguished:

1- Model based method: based on the behaviours comparison between the real plant (BLDC motor or PMDC motor) and the mathematical model of the system. This method uses the residual signal, which is the difference between real

process (measured signal) and the model (estimated signal). Residual generation can be accomplished in various ways: Parity equations, observer based generation and parameter estimation (Gaeid, K. S., & Ping, H. W., 2011). Observer based and parameter estimation methods are the most frequently used methods for fault detection (Isermann, R., & Balle, P., 1997).

2- Data based method: here, a dynamic model of an actual process is not necessary. The fault detection strategy is entirely based on measured physical quantities which are analyzed to extract the fault signatures for detection and diagnosis (Isermann, 2006).

1.2 Problem statement

Sensor faults of electrical motors can lead to decrease the performance of system , so that detection of these faults is very important . This thesis presents the fault detection method based on the sensor model. Observer-based method is chosen to fault detection and diagnostics for motors which are based on processing and analysis. The proposed method is aimed at detecting speed sensor fault on Brushless direct current (BLDC) motor and permanent DC motor (PMDC) using a Luenberger observer

1.3 Thesis Objectives

The main objective of this dissertation is to focus on model based sensor fault detection methods and experimental application to BLDC motor and PMDC Motor (real-time implementation).

Specific objectives include:

- 1- Describe the terminology used in detecting and diagnosing fault types and classifying them.
- 2- Prove the feasibility of implementing fault detection algorithms for BLDC and PMDC motors.
- 3- Design and implementation for BLDC motor using MATLAB Simulink.
- 4- Apply of an observer based fault detection method on electromechanical system.

5 - Detect speed sensor's faults of BLDC motor and PMDC motor.

1.4 Contributions

The main important contributions of this thesis can be summarized as follows:

- a) Modelling the BLDC motor and applying Luenberger observer design for detecting speed Sensors faults.
- b) Design and practical implementation of electronic hardware driver for the BLDC motor.
- c) BLDC motor and PMDC motor was successfully completed at Simulink MATLAB 2018.
- d) Fault detection was performed in real time for BLDC motor and PMDC motor.

1.5 Literature review

This section provides an overview of the published literature on fault detection and diagnosis for BLDC and PMDC motors. There are many studies and research on the subject of fault detection and diagnosis of BLDC and PMDC motors.

Zandi and Poshtan presented bearing fault detection of the BLDC motor, using hall effect sensors and wavelet transform (Zandi, O., & Poshtan, J., 2018).

A new methodology for active fault tolerant control with simultaneous faults of the actuator and sensor has been developed by **Allous and Zanzouri** (Allous, M., & Zanzouri, N., 2017). The work strategy combines a Luenberger observer and bond graph linear fractional transformation model .

Skóra presents the results of experimental studies related to the reproduction of the angular position of the rotor used after detecting the fault of the rotor position sensor (Skóra, 2017).

Zhang and Feng presents a new fast fault diagnosis method for Hall effect sensor based BLDC motor drives (Zhang, Q., & Feng, M., 2017).

Model-based sensor Fault detection approach has been developed by Eissa, et al, The proposed approach used Luenberger observer in order to give an appropriate method in order to detect the speed sensor faults in BLDC motor (Eissa , et al, 2015).

Eissa , et al. presented an improvement for Luenberger observer using fuzzy logic for detecting sensor faults of BLDC motor (Eissa, et al, 2015).

Eissa , et al. developed a model for estimating the state of the system and then detecting the valid fault and that's by using Unknown Input (UI) Propotional Integral (PI) observer.(Eissa, et al., 2015).

Sova, et al. presented the algorithm that allows the operation of a BLDC motor drive with one faulty hall position sensor (Sova, et al., 2015).

This paper provides many simulations on how the presented method reacts for various types of faults.

Fazal , et al. develop and test robust methodology for the Fault Tolerant Control (FTC) of a permanent magnet DC motor in the presence of different types of actuator faults (Fazal,et al., 2015).

Alkaya and Eker, presented online sensor FDD based on the model-based method using a Luenberger observer and experimental application on DC motor (Alkaya,et al., 2014).

Dobra, et al. focuses on the implementation details of fault detection and diagnosis based on parameters estimation of BLDC motor mathematical model (Dobra, et al., 2014).

Saoudi, **et al.** proposed a fault detection using a Luenberger observer based on Bond Graph model . This method was applied for the construction of a full order observer in the case of linear systems (Saoudi,et al., 2013).

Through the previous literary study, real-time implementation of sensor fault detection (FD) for electrical motors were not enough. Therefore, the focus of this thesis was on the real-time implementation of the electric motors used in the experiments, which are the BLDC motor and the PMDC motor.

1.6 Organization of the Thesis

This thesis consists of six chapters. The following is a brief overview of each chapter: The first chapter, this chapter, contains background and motivation, problem statement, thesis Objectives, contributions, literature review of the thesis and organization of the thesis. The mathematical model of BLDC and PMDC motors and the properties of process control systems are described in chapter two.

Chapter three discusses basic principles of the Fault Tolerant Control System (FTCS), detailed discussion about the methods of fault detection and Luenberger observer method will be introduced and studied in the thesis.

The fourth chapter presents the design and implementation of the BLDC motor and PMDC motor using Simulink, presents the experimental setup and prerequisites, as well as provides technical information about the Arduino Mega 2560 and the driver used in the experiments.

In chapter five, Luenberger observer is implemented for fault detection methods. Several experimental tests are performed for BLDC motor and PMDC motor to explain the performance of proposed method, and also present the simulation and results.

The final chapter provides concluding remarks and discusses areas for future work. Besides, all references used in the thesis and sections of the Appendix are presented.

Chapter 2

Mathematical Models of

BLDC and PMDC

Motors

2.1 Introduction of Electric motors

All electromechanical applications require energy conversion, and that makes electric motors basic components in these applications. There are many applications of electric motors such as power tools, household appliances, electric cars and industrial production equipment. These motors are reliable, highly efficient in energy consumption and were continuously improved since their invention. Electrical motors are classified into two types: Direct Current (DC) motors and Alternating Current (AC) motors, depending on the supply voltage. In DC motors, the current will be flowing through the collector depending on brushes for commutating this current on the rotor. BLDC motors are one type of synchronous machine whose characteristics are similar to those of a DC brush motor (Kara, 2017). Currently, BLDC motors are becoming an increasingly popular research area for researchers around the world and that's due to their high efficiency, high reliability, compact size and low maintenance, making them used most in electrical and mechanical applications. Hence, there is no loss of excitation, a simple design is implemented and can be operated reliably. therefore, according to the advantages of BLDC motors. It can be used in a wide range of appliances such as robotics, electric and hybrid vehicles, refrigerators, washing machines, high-end pumps, fans and other devices (Sakunthala,et al., 2017). The BLDC motors have permanent rotor magnets. The rotor position can be detected using Hall effect sensors or optical encoders. The back electromotive Force (BEMF) of a BLDC has a trapezoidal wave which is similar to a square wave. The BLDC motor has three-phase windings on the stator (with n number of poles), and the rotor is equipped with permanent magnets, which are located in the center of the motor with a bearing (Ridwan,et al., 2016). The main difference between PMDC motors and BLDC motors is the brushes. Additional energy will be consumed due to the friction of the brushes when mechanically commute the motor. BLDC motors are more reliable than PMDC motors and are better suited for speed control (Kara, 2017). Figure (2.1) shows the difference between BLDC motor and PMDC motor .Other differences are given in Table (2.1) (Neto, 2017).

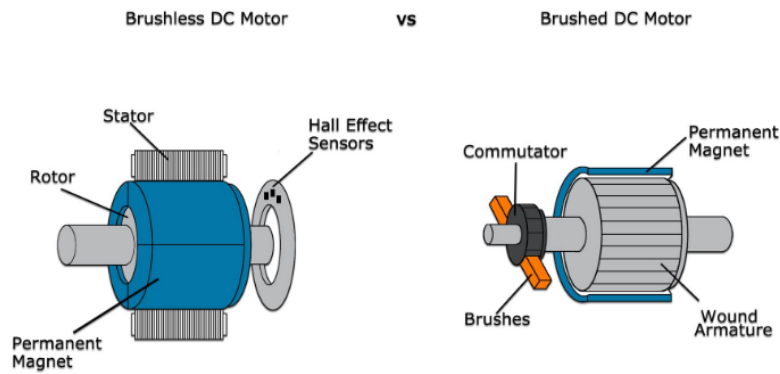


Figure (2.1): The differences between BLDC motor and PMDC motor (Mukkar, 2017).

Table (2.1) : Shows the differences between BLDC motor and DC motor.

Feature	BLDC motor	DC motor
Commutation	Electronic commutation	Brushed commutation
Maintenance	Low maintenance	High maintenance
Life	Long life	Short life
Speed / Torque	Allow operation at all speeds with rated load	At higher speeds, brush friction increases and reduce torque
Efficiency	High	Moderate
Speed range	Higher-No mechanical limitation due to contact	Lower –Mechanical limitation due to brushes

Here, in this thesis, two types of motors were discussed: the BLDC motor and the PMDC motor. More information on these motors will be in the following sections.

2.2 Principle Of Operation for BLDC motor

The BLDC motor operates according to a six-step commutating sequence. Accordingly, A hall sensors will excite each phase of the BLDC motor. In all operating modes of the BLDC motor, at any given time, only two phases are excited, and the third is not excited. The position of the rotor is determined by Hall sensors, which gives an exact position of the rotor , which helps the controller to supply the appropriate gate signal to the inverter switches. There is an inverter unit that is supplied to the BLDC motor consists of six - power switches (N-channel

MOSFETs). Two switches will be turned on during each conduction period and by commutation sequence, one will be a positive switch and the other is negative.

The switching regions will be divided into six modes (modes 1 to 6) due to the complete electric cycle of 360° , Each mode uses a different pair of switches. For example, the first mode used the MOSFET 1 which connected to phase A and MOSFET 4 which is connected to phase B.

Six modes at intervals of 60° will be make up according to six switching, (Lee,et al., 2015). Figure (2.2) shows the six-step commutation sequence of BLDC motor drive and Table (2.2) summarize the sequential modes of BLDC motor drive as given below.

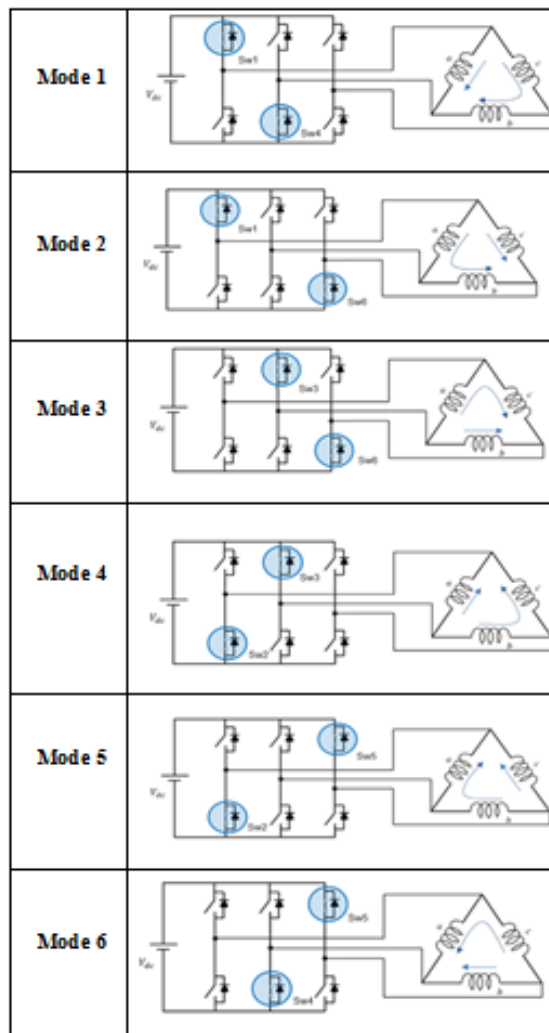


Figure (2.2): Switching sequence of three-phase delta winding connection(Ridwan,et al., 2016) .

Table (2.2): MOSFET switching conditions

# pin	Pos. sensor			Phase current						Active switches
	H1	H2	H3	A High Drive	A Low Drive	B High Drive	B Low Drive	C High Drive	C Low Drive	
0° – 60°	1	0	1	1/PWM	0	0	1/PWM	0	0	Sw1,Sw4
60° – 120°	1	0	0	1/PWM	0	0	0	0	1/PWM	Sw1,Sw6
120° – 180°	1	1	0	0	0	1/PWM	0	0	1/PWM	Sw3,Sw6
180° – 240°	0	1	0	0	1/PWM	1/PWM	0	0	0	Sw2,Sw3
240° – 300°	0	1	1	0	1/PWM	0	0	1/PWM	0	Sw2,Sw5
300° – 360°	0	0	1	0	0	0	1/PWM	1/PWM	0	Sw4,Sw5

2.3 BLDC motor mathematical model

Modeling of BLDC motors can be obtained similarly to DC motors, which have two types of elements: electrical and mechanical equations (Ridwan, et al., 2016). With the intention of simplifying equations and overall model, the following assumptions are made (Xia, C. L., 2012) :

- A - All windings of the stator phase are symmetrical and concentrated, therefore, the stator resistance, inductance are constant and equal, and the mutual inductance of each inductor is ignored.
- B – Neglecting the core saturation effect.
- C – Armature reaction not taken into account.

2.3.1 Background information of the MOONS' BLDC motor

BLDC motors are usually connected as a star (Y) connection. However, A delta (Δ) winding connection is used for BLDC motor of MOONS' Company. Figure (2.3) shows the equivalent circuit of a delta connected . Six pole for MOONS' BLDC motor modeling procedure has been introduced. Hall effect sensors are placed at 120° electrical apart. For every 60° one of the hall sensors changes it's state.

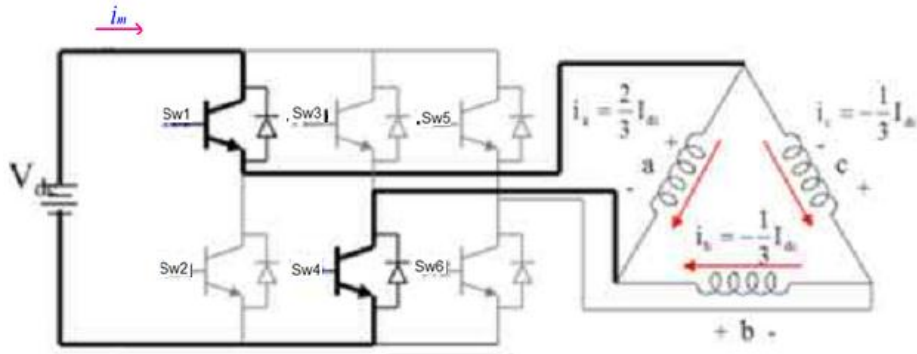


Figure (2.3): Equivalent circuit of delta winding connection for BLDC motor. (Chan,et al. , 2015)

Figure (2.4) shows the used 30 watt BLDC motor of serial no **42BL30L2-5** which has been provided from MOONS' Company for Industrial motors. Table (2.3) gives the parameters specifications which are taken from motor's datasheet in the website (<https://www.moonsindustries.eu>), where the data sheet is presented in Appendix A.



Figure (2.4): 42BL30L2-5 MOONS' BLDC motor.

Table (2.3): 42BL30L2-5 MOONS' BLDC motor specifications.

S.	Motor Model	42BL30L2-5
1	Rated drive voltage U_N	24 V
2	Hall effect angle	Electrical Angle
3	Poles number	6
4	Pair of poles	3
5	Phase number	3
6	Rated rotational speed n_N	4000 RPM // 418.88 rad/sec
7	Rated winding current I_N	1.67 A
8	Peak winding current I_p	3.34 A
9	Ph/ph resistance R_a	1.34 Ohm
10	Ph/ph inductance L_a	0.00115 H
11	Rotor Inertia J	0.0388e-4 Kg.m ²
12	Friction coefficient B	1.718e-4 Nm/rad/sec
13	Back EMF constant K_b	0.0281 Vrms/rad/sec
14	Torque constant K_t	0.043 Nm/Arms

2.3.2 Electrical equations

The electrical equation for a BLDC motor whose windings are connected to a delta is slightly different from motors whose windings are connected to a star (Bulán, M., 2014). The motor parameter conversion table was used where the table is presented in Appendix A. To simplify the problem, one step will be modeled. Figure (2.5) shows one step will be modeled, one terminal is connected to the supply voltage, the second terminal is connected to the ground, and the third is disconnected. All three phases then make a series -parallel connection. One Phase is connected in parallel with two series phases. For example, a series phases B and C are connected in parallel with phase A (Chan,et al. , 2015)

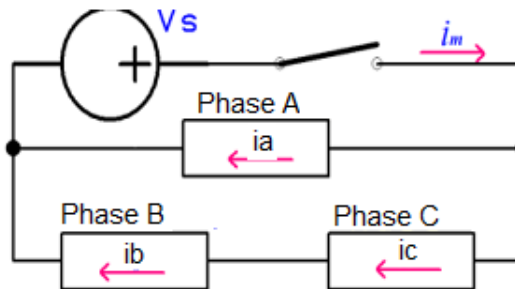


Figure (2.5): One commutation step of a three-phase BLDC motor.

The electrical model of the BLDC motor is expressed as follows (Bulán, M., 2014):

$$V_S = I_a R_a + L_a \frac{di_a}{dt} + e_a \quad (2.1)$$

$$V_S = I_b R_b + L_b \frac{di_b}{dt} - e_b + I_c R_c + L_c \frac{di_c}{dt} - e_c \quad (2.2)$$

$$e_a + e_b + e_c = 0 \quad (2.3)$$

Where V_S is the supply voltage [V], $i_{a,b,c}$ is the current in the phase [A], $R_{a,b,c}$ is the phase resistances, [Ω], $L_{a,b,c}$ is the phase inductance [H], $e_{a,b,c}$ is the Back-EMF on the phase.

Suppose that i_m is the input current of the source, then the phase current will become:

$$i_a = \frac{2}{3} i_m \quad (2.4)$$

$$i_b = i_c = -\frac{1}{3} i_m \quad (2.5)$$

The negative sign of the phase currents of B and C indicates that they flow in the opposite direction to the reference presented in Figure (2.3) .

In the three-phase BLDC motor, the BEMF is related to a function of the rotor position and the BEMF of each phase has a 120° phase angle difference, so the equation for each phase must be as follows:

$$e_a = K_\omega f(\theta_e) \omega \quad (2.6)$$

$$e_b = K_\omega f\left(\theta_e - 2\frac{\pi}{3}\right) \omega \quad (2.7)$$

$$e_c = K_\omega f\left(\theta_e + 2\frac{\pi}{3}\right) \omega \quad (2.8)$$

Where K_ω is BEMF constant of one phase [v/rad. s^{-1}], θ_e electrical rotor angle [$^\circ$ el.] and ω is rotor speed [rad. s^{-1}]

2.3.3 Mechanical equations

The equation of the mechanical properties is (Tibor, et al. , 2011):

$$T_e = T_l + J \frac{d\omega}{dt} + B \omega \quad (2.9)$$

Where T_e is an electromagnetic torque [N.m], T_l is a load torque [N.m], B is a viscous friction constant [N.m.s], J rotor's moment of inertia [kg.m²], ω is an angular speed [rad/s]

The torque of one phase for example phase A can be expressed by the equation:

$$T_a = \frac{P_a}{\omega} = \frac{e_a i_a}{\omega} \quad (2.10)$$

Where : T_a is the torque of phase A [N.m], P_a is the power of phase A [W]

The total output torque can be represented as a summation of the torque of each phase. The following equation represents the total output torque:

$$T = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega} \quad (2.11)$$

Where T is the total torque output [N.m] .

The relationship between the angle of the electric rotor and the mechanical angle of the rotor angle is then (Tibor, et al. , 2011):

$$\theta_m = \frac{2}{p} \theta_e \quad (2.12)$$

Where θ_e is the electrical rotor angle [^o el.] θ_m mechanical rotor angle [^o el.], p is the number of pole.

2.4 Transfer function of BLDC motor

The transfer function using the ratio of the angular velocity, ω_m to source voltage, V_s (Oguntoyinbo, O. , 2009) expressed as:

$$T.F = \frac{\omega_m}{V_s} = \frac{\frac{1}{k_e}}{\tau_m \cdot \tau_e \cdot s^2 + \tau_m \cdot s + 1} \quad (2.13)$$

For BLDC motor the mechanical time and electrical time constants are very important modeling parameters. The mechanical time constant (τ_m) of a BLDC motor can be given in equation (2.13) :

$$\tau_m = \frac{3.R \cdot J}{K_e \cdot k_t} \quad (2.14)$$

And the Electrical time constant (τ_e) :

$$\tau_e = \frac{L}{3.R} \quad (2.15)$$

We used the parameters in Table (2.3) and equation (2.13) to obtain the mathematical model of the MOONS' BLDC motor. And that's will obtain the motor model by calculating values of τ_e , τ_m and k_e (Oguntoyinbo, O. , 2009) , (Copt,et al., 2017).

From equation (2.14) the mechanical time constant (τ_m) is :

$$\tau_m = \frac{3.R \cdot J}{K_e \cdot k_e} = \frac{3 \cdot 1.34 \cdot 0.0388 \cdot 10^{-4}}{0.0281 \cdot 0.043} = 0.0129 \text{ sec}$$

And from equation (2.15) the electrical time constant (τ_e) :is

$$\tau_e = \frac{L}{3.R} = \frac{1.1 \cdot 10^{-3}}{3 \cdot 1.34} = 2.8606e^{-4} \text{ sec}$$

Therefore, transfer function becomes:

$$G(s) = T.F = \frac{35.587}{3.690 \cdot 10^{-6} S^2 + 0.0129S + 1}$$

2.5 State space model of the MOONS' BLDC motor

The state space model used for designing the observer of fault detection for BLDC motor. General form for state-space model is defined as follow (Aghaei, et al., 2018) :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.16)$$

$$y(t) = Cx(t) \quad (2.17)$$

where $u(t)$ is the input, $x(t)$ is the state, and the output is $y(t)$.

There are three constant matrices which defined the plant. These matrices are A is the state matrix, B is the input matrix, and C is the output matrix. The state vector $x(t)$, has been selected such that:

$$x_1(t) = \omega(t) = \frac{d\theta(t)}{dt} \quad (2.18)$$

$$x_2(t) = i(t) \quad (2.19)$$

$$\dot{x} = \begin{pmatrix} \dot{\omega} \\ \dot{i} \end{pmatrix} = \begin{pmatrix} \frac{-B}{J} & \frac{K_t}{J} \\ -\frac{K_e}{La} & -\frac{Ra}{La} \end{pmatrix} x + \begin{pmatrix} 0 \\ \frac{1}{La} \end{pmatrix} U \quad (2.20)$$

$$y = \omega = (1 \quad 0) x \quad (2.21)$$

where : $\omega(t)$ is the angular velocity and $i(t)$ is the current of the motor .

Therefore, A, B and C Matrices are obtained as indicated below: (Ridwan, M., & Yuniarto, M. N., 2016)

$$A = \begin{pmatrix} \frac{-B}{J} & \frac{K_t}{J} \\ -\frac{K_e}{La} & -\frac{Ra}{La} \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{1}{La} \end{pmatrix}, C = (1 \quad 0) \quad (2.21)$$

Using the MATLAB m-file (see Appendix B) for giving A, B and C Matrices

$$A = \begin{pmatrix} -44.28 & 1.108e+04 \\ -24.43 & -1165 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 869.6 \end{pmatrix}, C = (1 \quad 0)$$

2.5.1 Controllability and Observability of BLDC motor

The system is said to be controllable if there is a control input that transfers any state of the system to zero in finite time. It can be demonstrated that an Linear time-invariant (LTI) system is controllable if, and only if, its controllability matrix, presented in equation (2.22), has full rank (Sellami, L., 2014).

$$\text{controllability} = [B \mid AB \mid A^2B \mid \dots \mid A^{n-1}B] \quad (2.22)$$

The system is said to be observable if there the initial state $x(t_0)$ can be determined from the system output, $y(t)$, over some finite time $t_0 \leq t \leq t_f$. It can be demonstrated that an LTI system is observable if, and only if, its Observability matrix, presented in equation (2.23), has full rank.

$$\text{Observability} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdot \\ \cdot \\ CA^{n-1} \end{bmatrix} \quad (2.23)$$

The system is observable and controllable, this was done using the MATLAB m file (see Appendix B).

2.6 PMDC motor mathematical model

The PMDC motor was introduced as a speed control target. The parameters of the PMDC motor model type (YA-070) are calculated practically (Hassan, et al., 2017). Typically, PMDC motors models consist of two parts: electrical and mechanical equations. The electrical equation shows the main parameters: armature resistance (R_a) and inductance (L_a). On the other side, the main parameters of the mechanical equation are the inertia rotor (J) and the friction coefficient (B).

Equivalent circuit of PMDC motor is shown Figure (2.6), that uses the armature voltage control method.

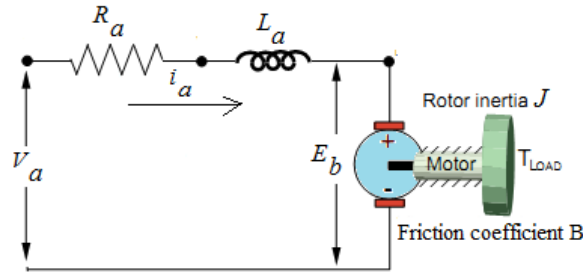


Figure (2.6): PMDC motor equivalent circuit.

The targeted plant is PMDC motor (Model: YA-070), as shown in Figure (2.7), which has the parameters are shown Table (2.4). The motor datasheet, which is found (<https://www.yoycart.com/Product/26695000175/>) is presented in Appendix A



Figure (2.7): YA-070 PMDC motor.

Table (2.4): YA-070 PMDC motor specifications.

S.	Motor model	YA-070
1	Rated drive voltage	24 V
6	Rated rotational speed	3000 RPM
7	Rated current	200 mA
9	Resistance R_a	7 Ω
10	Inductance L_a	0.008436 H
11	Rotor of inertia J	2.2097e-04 Kg.m ²
12	Friction coefficient B	1.65e-04 N .m/rad/sec
13	Back EMF constant K_b	0.094 V/rad/sec
14	Torque constant K_t	0.094 N .m/A

The equations below represent the characteristics of the PMDC motor, because the back EMF E_b is proportional to speed $\omega(t)$ directly, hence

$$E_b(t) = K_b \frac{d\theta(t)}{dt} = K_b \omega(t) \quad (2.24)$$

Using the KCL voltage law we can obtain

$$V_a = R_a I_a(t) + L_a \frac{dI_a}{dt} + E_b(t) \quad (2.25)$$

From Newton's law, the torque can get

$$T_m(t) = J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} = K_t I_a(t) \quad (2.26)$$

Where : R_a Resistance of armature (Ω), L_a Inductance of armature (H), I_a Current of armature (A), V_a Input voltage (V), E_b Back electromotive force (V), T_m Motor torque (N. m), ω Angular velocity of rotor (rad/sec), J Rotor inertia ($kg.m^2$), B Friction constant (N. m/rad/sec), K_b EMF constant (V/rad/sec), K_t Torque constant (Nm/A)

The final speed transfer functions for PMDC motor relative to the input voltage can be written as follows (Hassan,et al., 2017):

$$\frac{\omega(s)}{V_a(s)} = \frac{k_t}{L_a J S^2 + (R_a J + L_a B) S + (R_a B + k_t k_b)} \quad (2.27)$$

A mathematical model of a PMDC motor based on the parameters in Table (2.4) is obtained using the above equation. Then the PMDC motor speed transfer function becomes:

$$T.F = \frac{0.094}{1.864e^{-6}S^2 + 0.001548S + 0.00991}$$

2.6.1 State space model of PMDC motor

The state space model used for designing the observer of fault detection for PMDC motor, therefor, the transfer function in equation (2.27) can be represented in state space and expressed as:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.28)$$

$$y(t) = Cx(t) \quad (2.29)$$

The state vector $x(t)$ is selected as :

$$x_1(t) = \omega(t) = \frac{d\theta(t)}{dt} \quad (2.30)$$

$$x_2(t) = i(t) \quad (2.31)$$

The state transition, input, and observation matrices for the model are:

$$\dot{x} = \begin{pmatrix} \frac{dx_1(t)}{dt} \\ \frac{dx_2(t)}{dt} \end{pmatrix} = \begin{pmatrix} \frac{-B}{J} & \frac{K_t}{J} \\ -\frac{K_e}{L_a} & -\frac{R_a}{L_a} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_a} \end{pmatrix} u(t) \quad (2.32)$$

$$y(t) = [1 \quad 0] \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \quad (2.33)$$

The input and output matrices of system for the PMDC motor model are:

$$A = \begin{pmatrix} \frac{-B}{J} & \frac{K_t}{J} \\ -\frac{K_e}{L_a} & -\frac{R_a}{L_a} \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{1}{L_a} \end{pmatrix}, C = (1 \quad 0) \quad (2.34)$$

Using the MATLAB m-file (see Appendix B), input and output matrix values for the PMDC motor model are :

$$A = \begin{pmatrix} -0.7467 & 425.4 \\ -11.14 & -829.8 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 118.5 \end{pmatrix}, C = (1 \quad 0)$$

2.6.2 Controllability and Observability of PMDC motor

The system is observable and controllable, this was done using the MATLAB m. file (see Appendix B).

Chapter 3

Fault Tolerant Control System (FTCS)

3.1 Introduction

Fault Tolerant Control System (FTCS) can be defined as the ability to accommodate the faults of system components automatically, and is capable to maintain and fix the overall system stability, in order to increase reliability, safety

and automation level of modern engineering systems (Zhang, 2006). There are two classification of Fault Tolerant Control (FTC) which are Passive FTCS (PFTC) and Active FTCS (AFTC) . Figure (3.1) shows the FTC classification (Zhang, 2006).

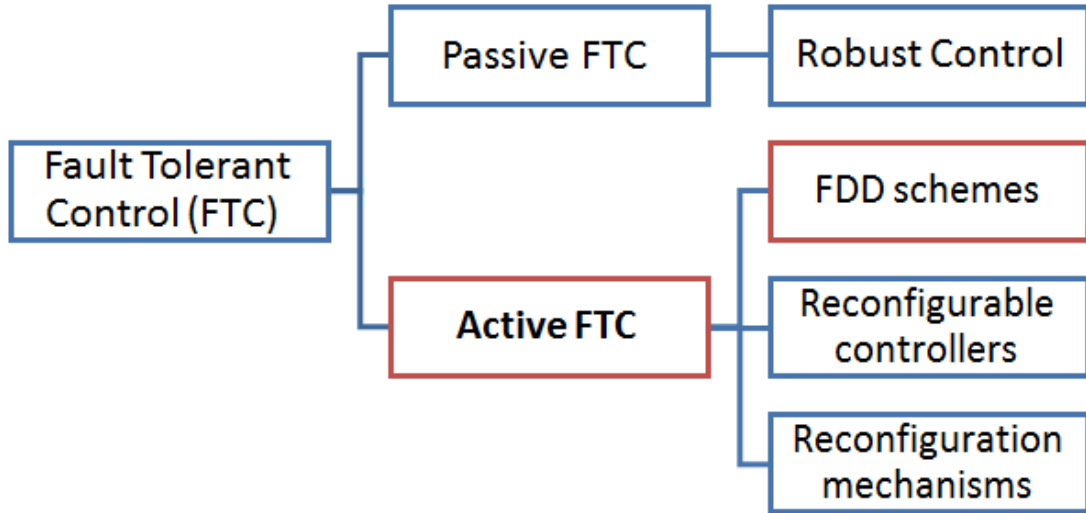


Figure (3.1): Classification of (FTC).

Passive FTCS (PFTCS) are systems designed to operate with a specific class of component failures without the need for online fault information which has the expected fault tolerance properties and fixed controller parameters. (Zhang, 2006).

Active FTCS (AFTCS) are systems that can reconfigure the control law online and in real time to adapt to component faults, which have the properties of explicit fault detection and diagnosis (FDD) methods, real-time decision making and reconfiguration of the controller, taking into account unforeseen anticipations or faults and acceptable degraded performance in the presence of faults (Zhang, 2006). There are three parts to AFTCS that are FDD, reconfigurable controllers and reconfiguration mechanisms. Parts of the AFTCS are shown in Figure (3.2).

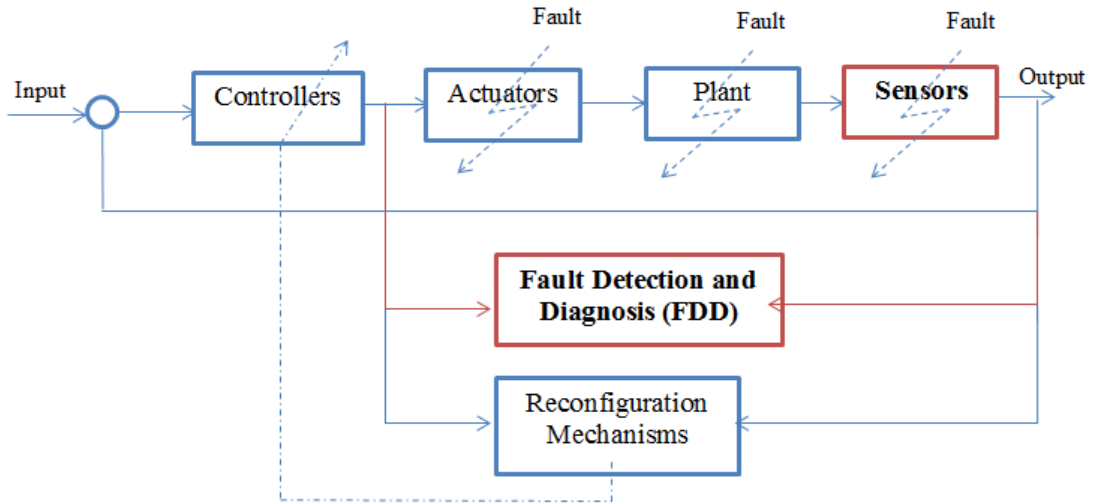


Figure (3.2): Block diagram of the parts Active fault-tolerant control (ATFC).

In this thesis, a partial FDD of the AFTC was studied and the fault was applied to the speed sensor of the BLDC and PMDC motors.

3.2 Fundamental Definitions:

There are some definitions in the field of Fault Detection suggested by IFAC (International Federation of Automatic Control) Technical Committee SAFEPROCESS and it is as follows (Isermann, 2006), (Nyberg,et al., 1999):

Fault: "unacceptable deviation of at least one characteristic property".

Failure: Constantly interrupting the ability of the system to perform the required function under certain operating conditions.

Fault Detection: Determination of faults present in the system and detection time.

Fault Isolation: Determination of the type, location and time of fault detection. Follows fault detection.

Fault Identification: Determining the size and temporary behavior of the error.. Follows fault isolation.

Fault Diagnosis: Determining the type, size, location and time of detection of a fault. Follows fault detection. Includes fault isolation and identification.

3.3 Types of faults

There are three types of faults, the first type of faults is nature fault that based on the physical location where the fault occurred, the second type is based on the time that the fault happened and the last type is based on the model of fault (Monteiro, D., 2015) . Types of faults are shown in Figure (3.3) .

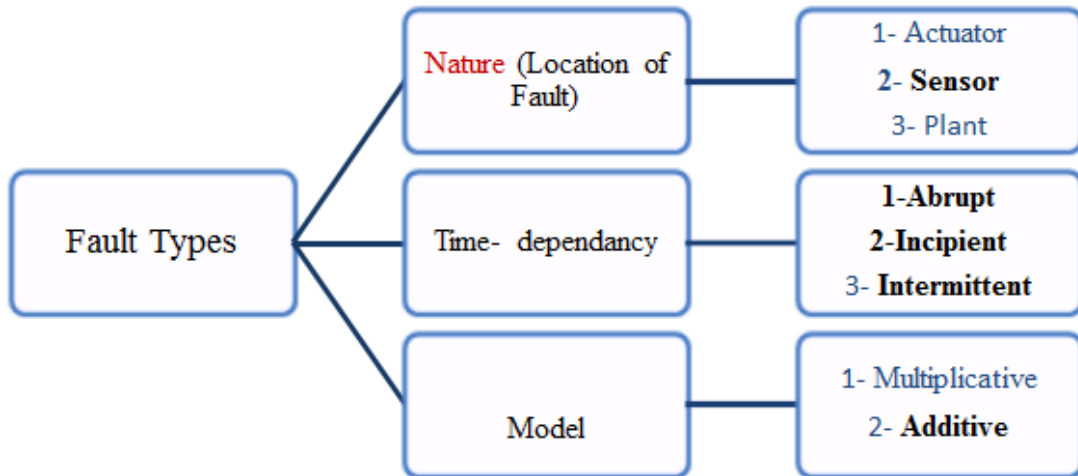


Figure (3.3): Types of faults.

In this research project there were three types of fault detection used. Sensor fault was chosen from the nature fault type while from the time dependent fault the following three types where used: 1- Abrupt type, 2- incipient and 3- intermittent. In the model dependent type, the additive fault was used. These faults have been applied to the motors(BLDC and PMDC) used in this thesis

3.3.1 (Nature Fault) Based on the location of fault

There are three types of faults based on the location of fault, which are actuator faults, plant faults and sensor faults which shown in Figure (3.4).

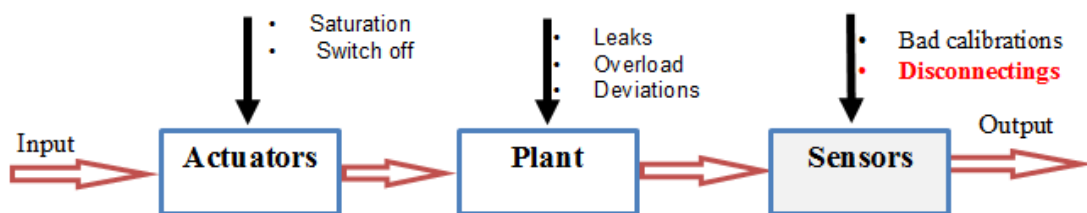


Figure (3.4): Types of locations faults.

Actuator faults : can be represented as the partial or total loss of control system. A complete fault of actuator can be occurred, for example, due to a broken, cut, or burnt wiring, or the presence of an external body in the actuator (Kanev, 2004). A partially faulty actuator produces only part of a normal actuation. This can be caused, for example, by hydraulic or pneumatic leakage, increased resistance or a drop in supply voltage (Kanev, 2004).

Plant faults: refer to faults which result in changes to the parameters of the system dynamics.

Sensor faults : "represent an incorrect reading from the sensors". This can be caused, for example, by bad calibrations and disconnecting the speed sensor (Kanev, 2004). The sensor fault in electrical motors is the subject of the thesis

3.3.2 Time dependent fault

Three types of faults based on the time that the fault happened which are abrupt fault, incipient fault, and intermittent fault (Kanev, 2004) as shown in Figure (3.5).

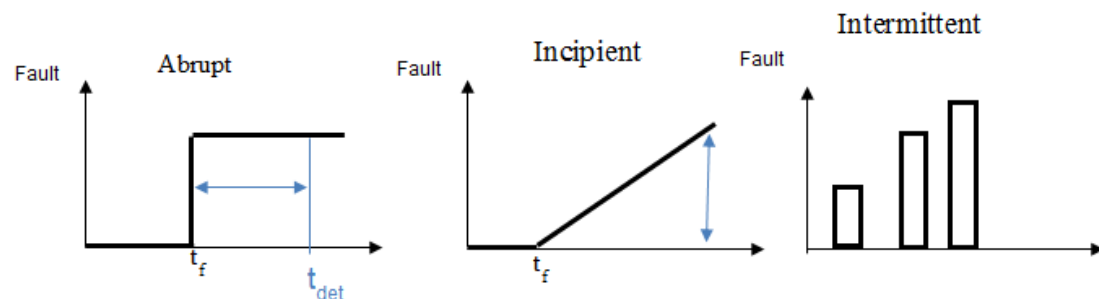


Figure (3.5): Time behaviour of faults.

The definitions of time dependent fault are (Silmon, J. A., 2009) :

Abrupt fault : that appears suddenly and without any indications. This can be caused, for example, short-circuits or melt-down

Incipient fault : created gradually over a period of time

Intermittent fault : a fault that appear and disappear over and over again. This can be caused, for example by loose wires.

3.3.3 Model of fault

There are two types of faults based on the model of fault which are additive faults and multiplicative faults (Isermann, R., 2005). The additive fault, variable $y_u(t)$ is changed by addition of fault $f(t)$ as shown in Figure (3.6).

$$y(t) = y_u(t) + f(t) \quad (3.1)$$

Where $y_u(t)$ is the measured signal and $f(t)$ is the constant value.

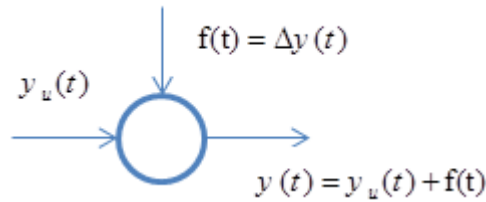


Figure (3.6): Additive fault model.

Multiplicative fault has a more complex effect on the system by the product of the variable $u(t)$ by Δp , as shown in Figure (3.7).

$$y(t) = pu(t) + u(t)\Delta p \quad (3.2)$$

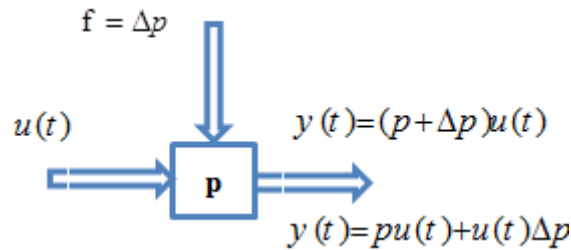


Figure (3.7): Multiplicative fault model.

The most typical additive faults are sensor offsets, while multiplicative failures are changes in parameters in the system (Miljković, 2011).

3.4 Fault detection methods

Fault detection methods can be represented into two types: a model based methods and data based methods (Gaeid,et al., 2011) as shown in Figure (3.8). The method used in this study is demonstrated in Figure (3.8) distinguished by dashed line.

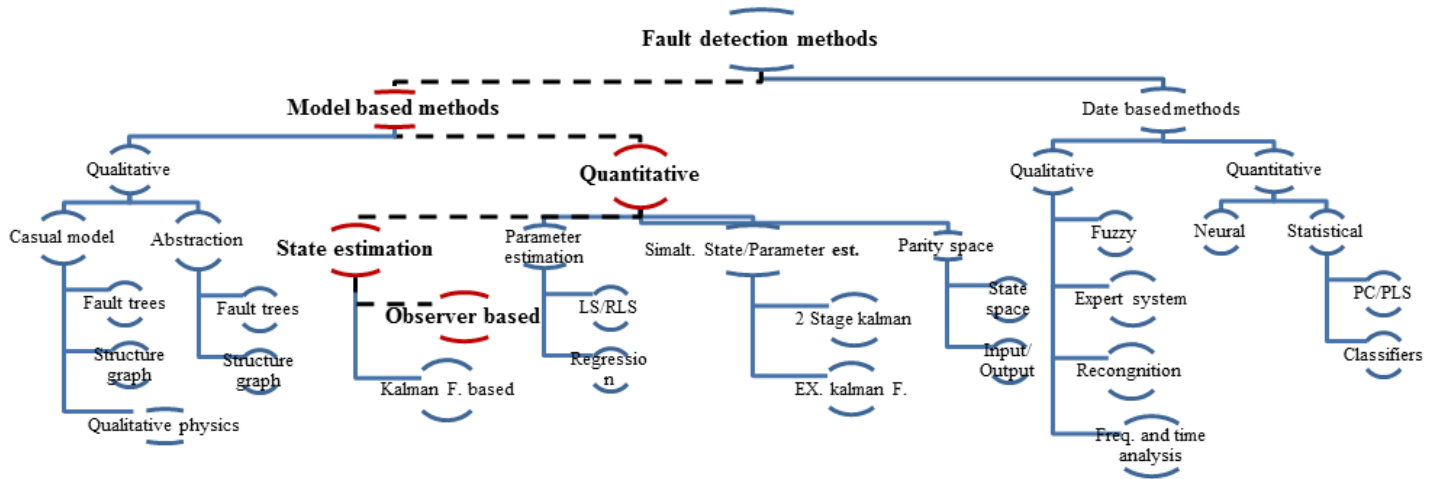


Figure (3.8): Methods of fault detection.

Fault detection methods are usually performed in the following steps:

- A- Building a mathematical model that estimate the output from the from the obtained Input.
- B - Calculate residuals by comparing the measured outputs and estimated outputs from (A).
- C- Making decisions based on calculated residuals which can be evaluated by simple threshold.

3.4.1 Model-Based Methods

3.4.1.1 Quantitative model based method

The basic idea of methods based on quantitative models is to create a mathematical model which expresses the behavior of a controlled object based on a fundamental understanding of the process. These methods usually include input and output parameters. The model generates an estimated output from the input parameters, and this estimated output is then compared to the measured output. (ALKAYA, 2012). A block diagram of the method based on a quantitative model is shown in Figure (3.9)

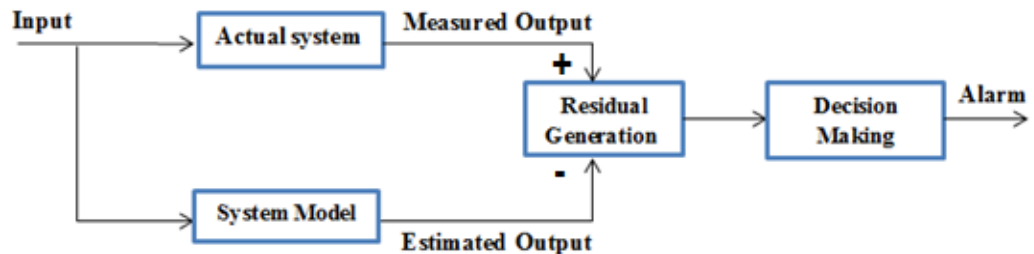


Figure (3.9): Block diagram of the quantitative model method.

Typical methods used in the quantitative model method include: state estimation , parity space and parameter estimation method

3.4.2 State Estimation Methods

The basic idea of the state estimation methods is to estimate the states or outputs of the system from the measurements by using either Observer (Luenberger's observers) in a deterministic setting or Kalman's filters in a stochastic case (ALKAYA, 2012). Luenberger observer method was used in this thesis .

3.4.3 Luenberger Observer Method

The basic idea of this method is to use an observer (Luenberger observer) to observe the state of the system from the measured data. Then, by comparing the observed states to the measured states of the monitored system as the objective of generating the residual, the residue is then obtained by calculating the difference between the outputs of the observer and the measured signals (Sellami, L., 2014).

The Luenberger observer design for fault detection is based on the state space model.
For a given linear state space model:

$$\dot{X}(t) = Ax(t) + Bu(t) \quad (3.3)$$

$$y(t) = Cx(t) \quad (3.4)$$

where $u(t)$ is the input, $x(t)$ is the state, $y(t)$ is the output and A , B and C are known constant matrices. The state vector, $x(t)$, was chosen so:

$$x_1(t) = \omega(t) = \frac{d\theta(t)}{dt} \quad (3.5)$$

$$x_2(t) = i(t) \quad (3.6)$$

The observer can be designed as follows to provide the system observable:

$$\dot{\hat{X}}(t) = A \hat{x}(t) + Bu(t) + Le(t) \quad (3.7)$$

$$\hat{y}(t) = C \hat{x}(t) \quad (3.8)$$

$$e(t) = y(t) - \hat{y}(t) \quad (3.9)$$

where $\hat{x}(t)$ is the estimated system state, $\hat{y}(t)$ is the estimated output, L is the observer gain, and $e(t)$ is the output error (difference between the actual measured output $y(t)$ and its estimate $\hat{y}(t)$).

Replacing (3.9) in (3.7):

$$\dot{\hat{X}}(t) = [A - LC] \hat{x}(t) + Bu(t) + Ly(t) \quad (3.10)$$

The estimation error is given by:

$$e(t) = X(t) - \hat{X}(t) \quad (3.11)$$

With taking the first derivate of $e(t)$ we can get:

$$\dot{e}(t) = \frac{d}{dt}(x(t) - \hat{X}(t)) = Ax(t) + Bu(t) - A \hat{x}(t) - Bu(t) - LC(x(t) - \hat{X}(t)) \quad (3.12)$$

The dynamic of the error is given by:

$$\dot{e}(t) = (A - LC)(X(t) - \hat{X}(t)) \quad (3.13)$$

Accordingly, observer error dynamics are defined by placing the eigenvalues of matrix $(A - LC)$ in order to impose faster observer dynamics compared with system dynamics. The functional diagram of the actual system and Luenberger observer model used for fault detection methods is shown in Figure (3.10).

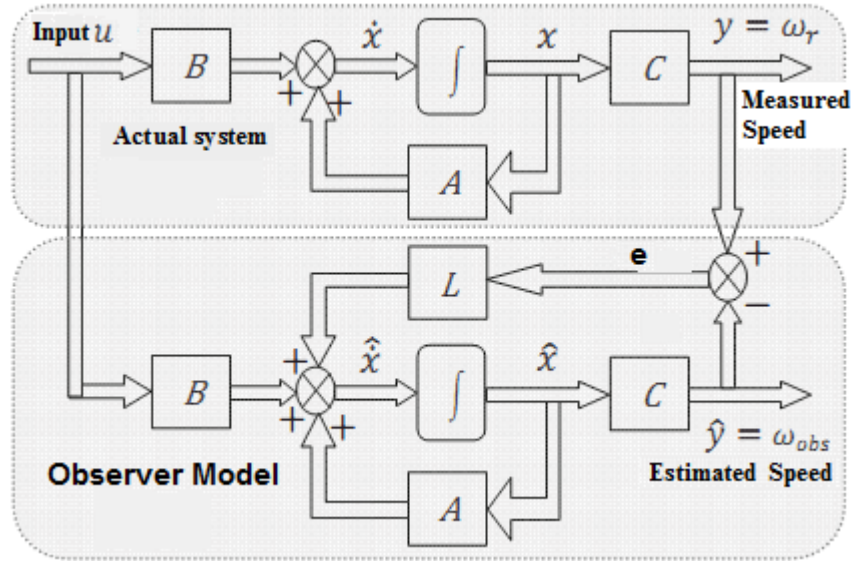
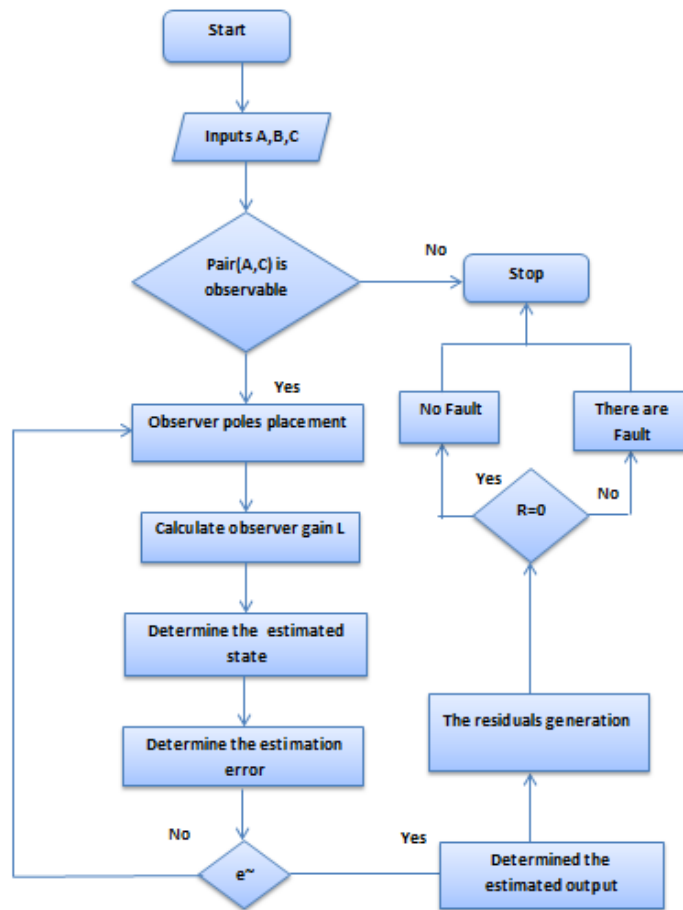


Figure (3.10): Functional diagram of real system and observer model.

Observed and measured speeds are used to generate residual $r(t)$ in order to detect the speed sensor fault. The residual $r(t)$ is given by:

$$r(t) = y(t) - \hat{y}(t) \quad (3.14)$$

The residual value will be checked automatically in order to carry out the fault detection. If the residual is equal or tends to zero, there will be no fault, otherwise a fault has occurred. The flowchart describing the algorithm used to create the Luenberger observer to detect faults is illustrated in Figure (3.11) (Eissa, et al., 2015). This flowchart shows the work of the project from start to finish.



Figure(3.11): Flowchart of Luenberger observer for fault detection.

In this research we designed Speed Luenberger observer for BLDC and PMDC motors are shown in Figure (3.12) and Figure (3.13).

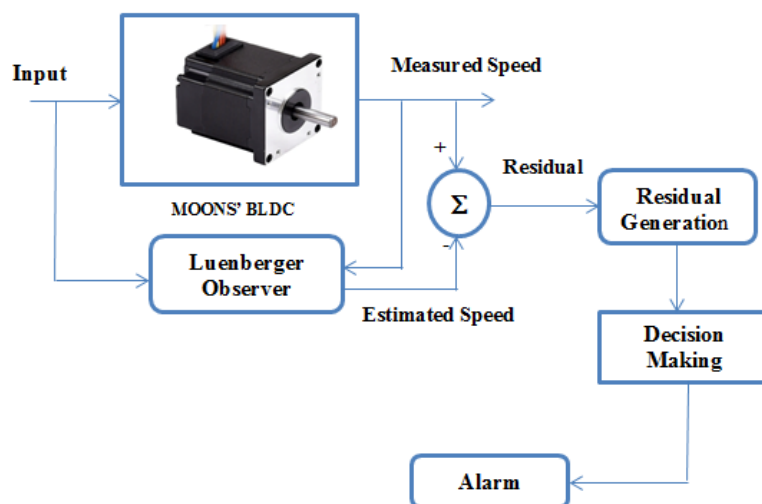


Figure (3.12): Speed Luenberger observer for MOONS' BLDC.

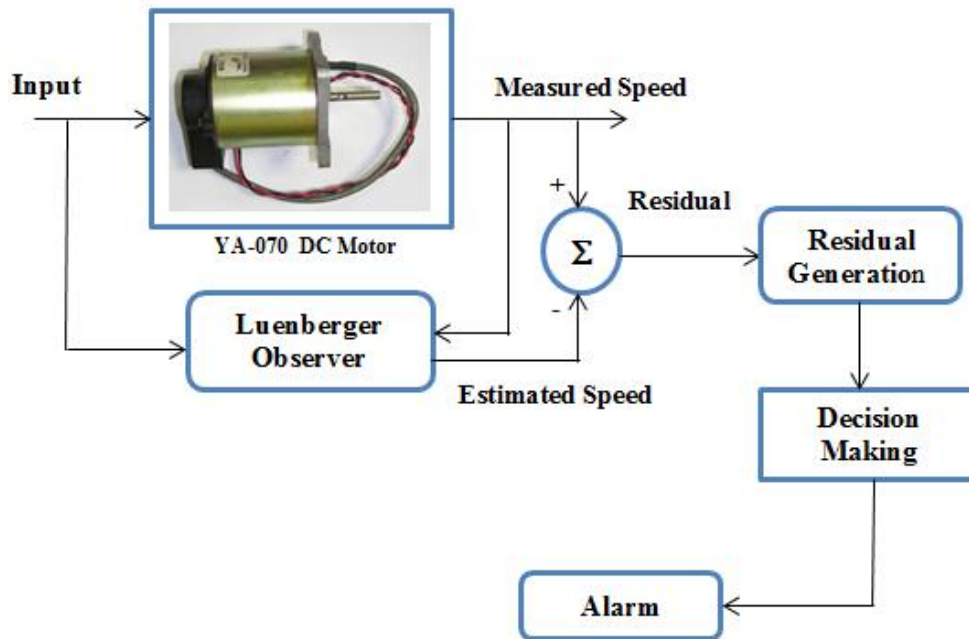


Figure (3.13): Speed Luenberger observer for YA-070 DC motor.

Chapter 4

Hardware and Software Design & Implementation

4.1 Introduction

The experiment contains various software and hardware components which are described in this chapter. The experimental system consists of two motors, the BLDC motor and the PMDC motor. Drive circuits for motors are designed on the Proteus 8.6 software. Luenberger observer was also designed using Simulink, which is a special toolbox of the MATLAB numerical calculation software. The interface can be programmed with Simulink. The interfacing is done via the Arduino Mega 2560 board. Simulink Support Package for Arduino Hardware allows you to create and run Simulink models on Arduino Mega 2560 board (Sadun, et al., 2015). Figure (4.1) shows the Simulink Support Package for Arduino Hardware.

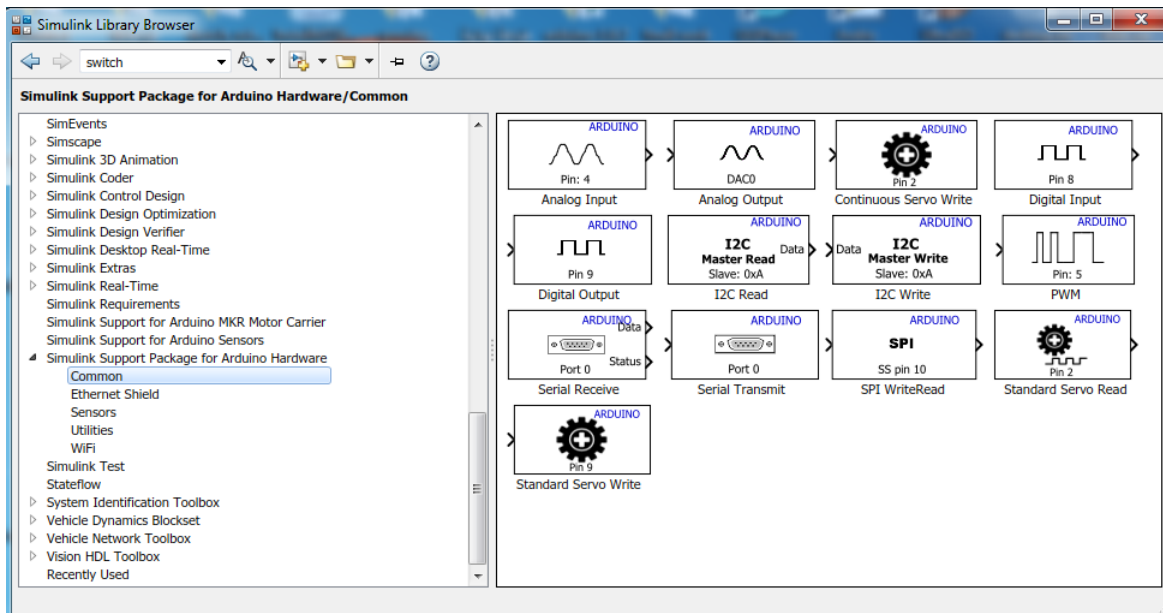


Figure (4.1): Simulink Support Package for Arduino Hardware.

Simulink Support Package includes a library of Simulink blocks for accessing Arduino sensors for example tachometer sensor is shown in Figure (4.2). Toolbox allows you to configure your external hardware devices, read data in Simulink environments for immediate analysis and send data. The measurement data are read from the PC by an Arduino Mega 2560, which can also process the data in real time.

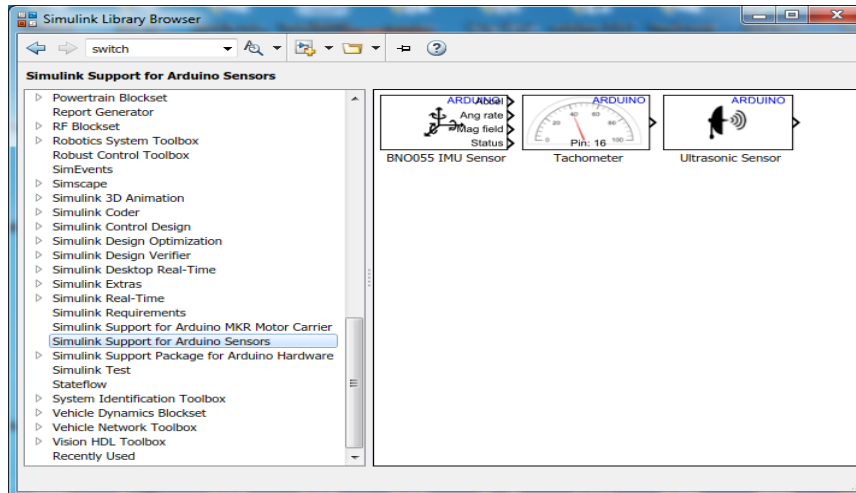


Figure (4.2): Simulink support for Arduino sensors.

4.2 Design and implementation for PMDC motor using Simulink

We created a model for an armature-controlled PMDC motor based on its step response. The speed command module generates a step signal with a range 0-255 . Figure (4.3) shows the Simulink (design) model of PMDC motor . This model explains how to use Simulink blocks to use PWM output signals to control the PMDC motor . The PWM signal is sent through Pin 5 of the Arduino Mega 2560 board.

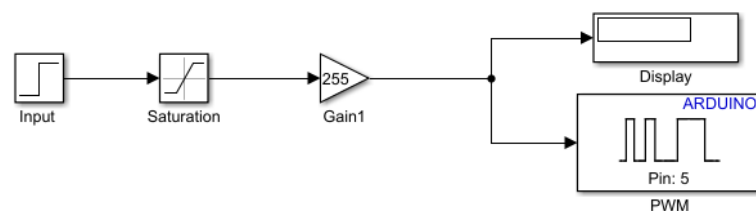


Figure (4.3): Simulink model of PWM output.

The PMDC motor speed is determined by optical encoder. The encoder is the AB phase incremental encoder, each circle outputs 200 pulses. We used the pulse output of channel A to plot the speed curve and the output channel B to measure speed in RPM. The encoder pulses are counted on the Arduino Mega 2560 board via two of the board's digital inputs(pin 2, pin 3) (Tang,et al., 2017).The logic for estimating the motor's speed based on encoder counts is implemented within Simulink. Figure (4.4) shows the Simulink design to find speed curve.

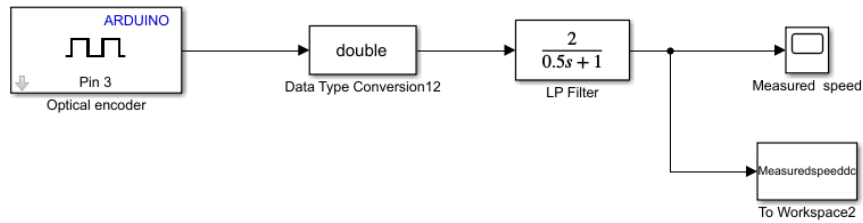


Figure (4.4): Simulink design to find speed.

The tachometer in the Simulink support package for Arduino hardware / sensors was used to measure motor speed in RPM through digital pin 2 of the Arduino Mega 2560, as shown in Figure (4.5).

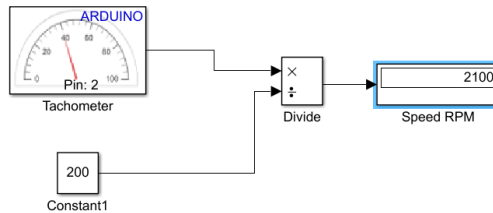


Figure (4.5): Tachometer to measure speed in RPM.

The general construction of Simulink design is shown in Figure (4.6), allows you to successfully run the PMDC YA-070 motor and control the speed of the PMDC motor.

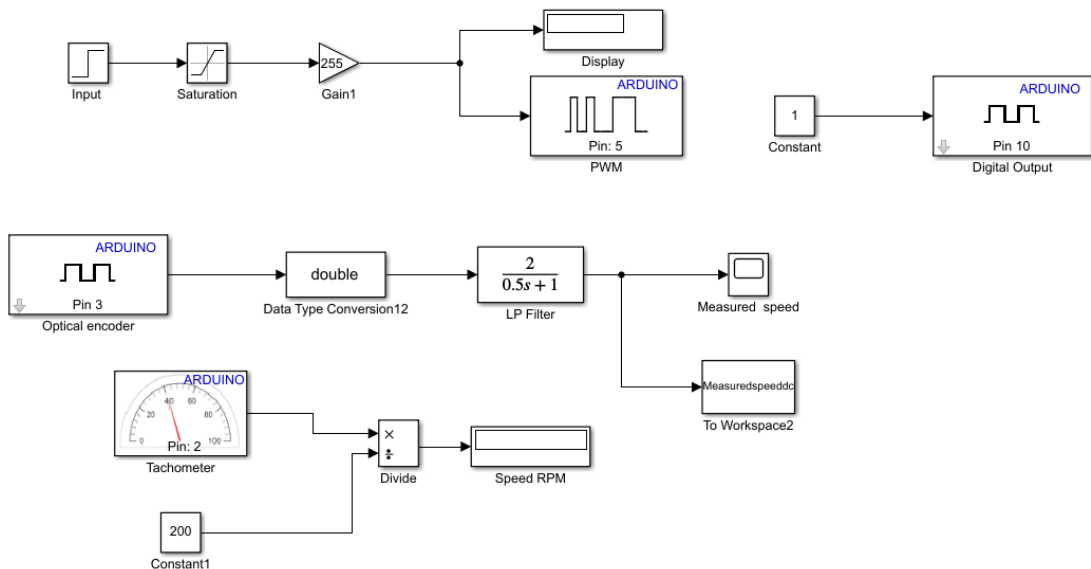


Figure (4.6): Overall Simulink design for PMDC motor.

4.3 Design and implementation for BLDC motor using Simulink

The BLDC motor's Simulink design includes phase signal generation because BLDC motors only operate when the input voltage phase changes. A three-phase BLDC motor requires a signal with three different phase delays. These three-phase signals are generated using a PWM generation block. The PWM signals generated in Simulink are programmed for Arduino Mega 2560. The programmed Arduino Mega 2650 transmits the PWM output to the BLDC motor drive. A PWM generation block is shown in Figure (4.7) that generates PWM signals and transmits these signals to digital pins(32,34,36,38,40,42) in Arduino Mega 2560 .

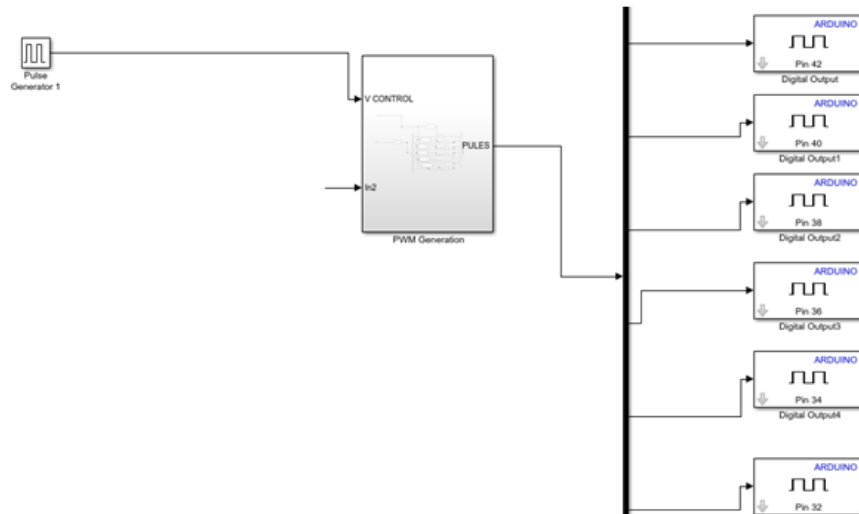


Figure (4.7): PWM generation.

The Internal Simulink design of PWM generation is presented in Figure (4.8).

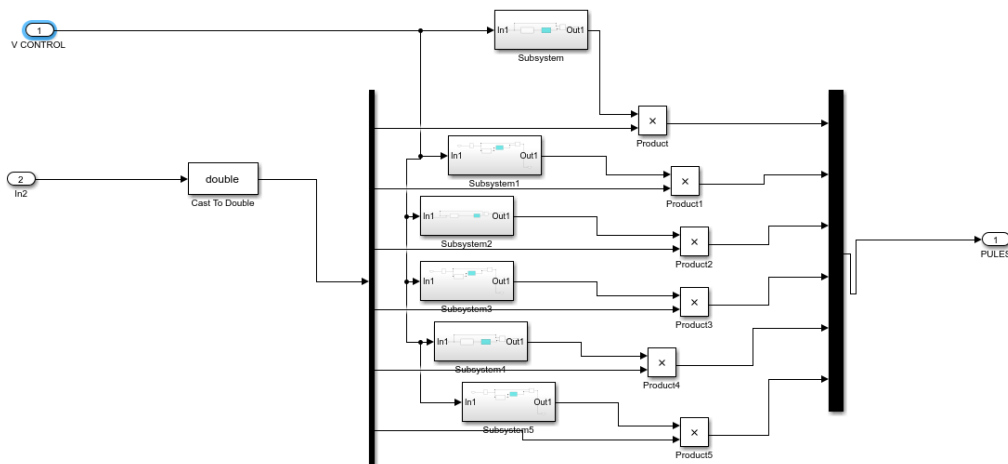


Figure (4.8): Internal Simulink design of PWM.

All generations of PWM signals are similar, but the phase will change at the output of each of pins (32,34,36,38,40,42) for the PWM signals generated as shown in Figure (4.7). Inside the PWM block, a dead time block was integrated in. It's function was to allow the inverter arm switches not to operate together, ensuring that a key can open before another close by changing the block's parameters (Agrawal, et al. , 2016). The full parameterization of this block is presented in Figure (4.9). The dead time between the pulses given to two complimentary switches is shown in Figure (4.10).

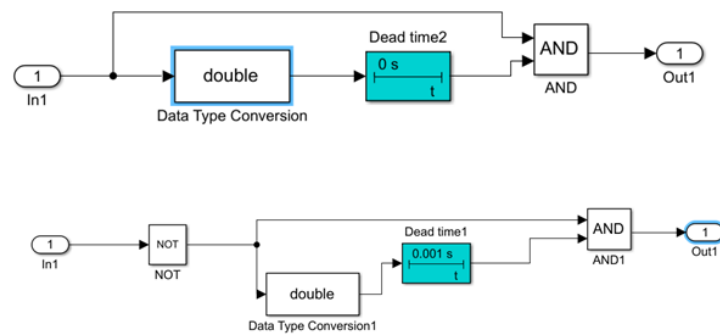


Figure (4.9): Internal Simulink design of dead time.

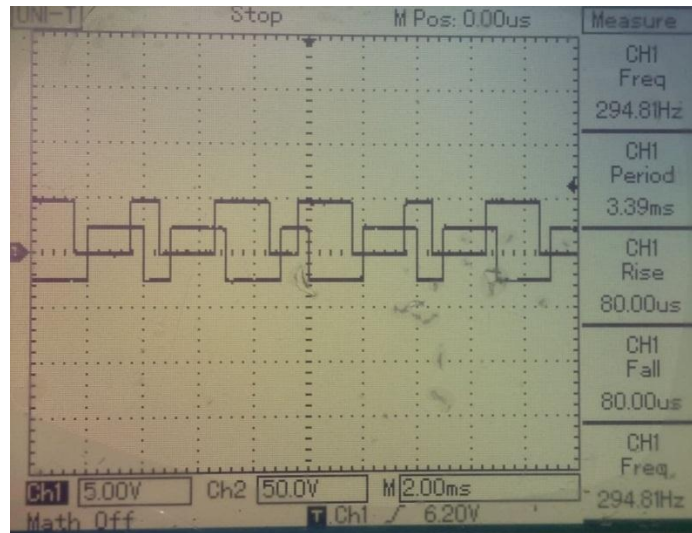


Figure (4.10): Complementary PWM pulses with dead time

To gain control over the position and speed of the BLDC motor, we need Hall sensors or an EMF input signal to determine the position and speed of the rotor. In this thesis, the Hall sensors were used to calculate the speed of the rotor. Figure (4.11) shows the software function designed to receive the input of Hall effect sensors from the Arduino Mega pins (3, 4, 5).

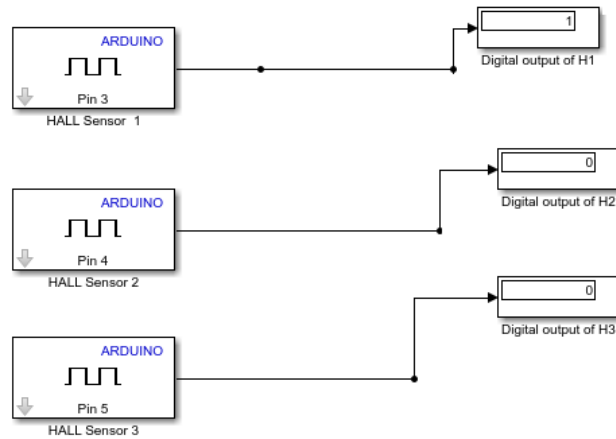


Figure (4.11): Design of Hall-effect sensors in display.

Three signals from the Hall effect sensor were received from the BLDC motor using the Arduino Mega are shown in Figure (4.12).

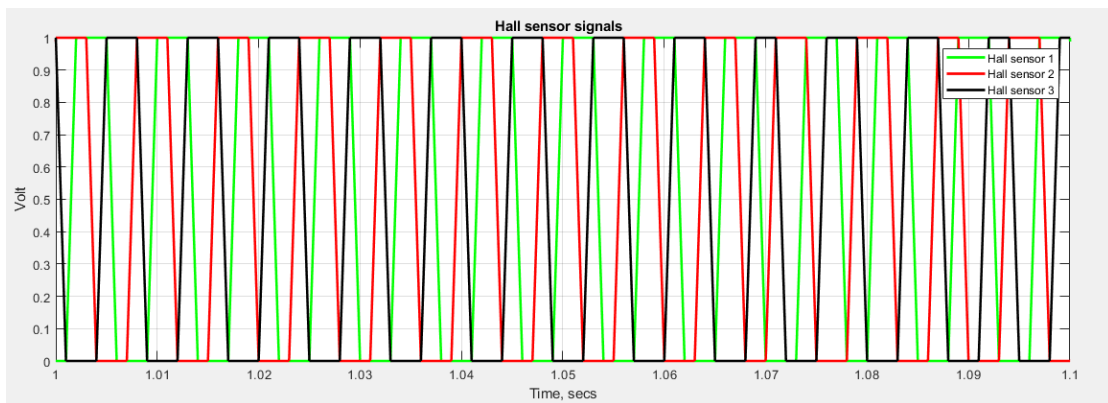


Figure (4.12): Hall sensor signals.

To control the speed and position of the BLDC motor, the BLDC motor must rotate relative to the inputs of the Hall effect sensor. Indirectly, it is the phase which must change compared to the Hall sensor inputs (Megalingam, et al., 2019). We know that related phase signals are only activated for a specific pair of MOSFETs. Figure (4.13) shows the pairs of MOSFET switches. Therefore, we need to match the input

of the Hall sensor with the states of the MOSFET pair in order to get the corresponding phase signals. Typically, the Hall effect sensor input circuit consists of 101, 100 , 110, 010, 011, 001 (see Appendix A). The matching of the input and switching conditions of the Hall effect sensor to control the speed and position of the BLDC motor is presented in table (2.2) of chapter two.

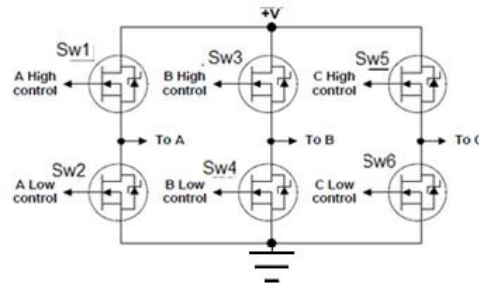


Figure (4.13): MOSFET switch

This matching is achieved using two internal logic blocks, the first block is a decoder block, which receives the input signals of the Hall sensor and performs calculations. The decoder block is shown in Figure (4.14) and the internal Simulink design for decoder block is shown in Figure (4.15) .

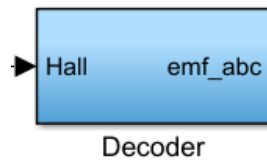


Figure (4.14): Decoder block design.

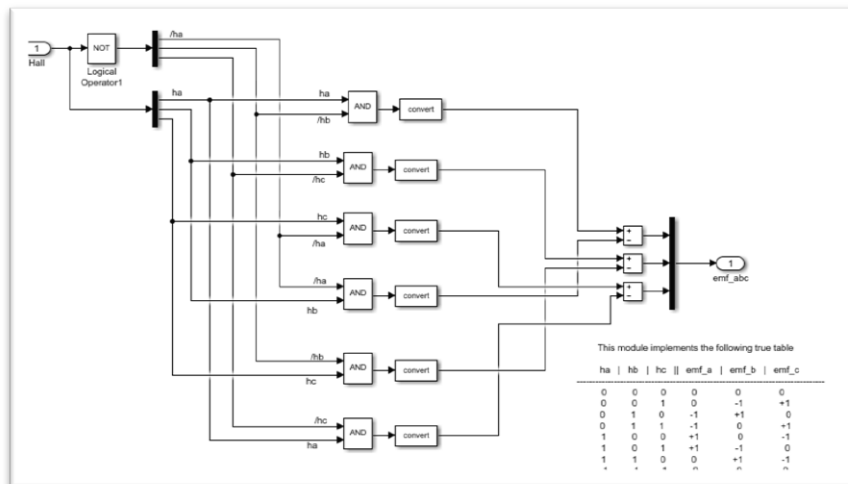


Figure (4.15): Internal Simulink design for decoder block.

The calculated outputs from the decoder block go to the comparison logic circuit for selecting the PWM signal in the second block, which is the gate block as shown in Figure (4.16) and the internal Simulink design for gates block is shown in Figure (4.17) .

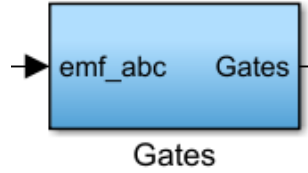


Figure (4.16): Gates block design.

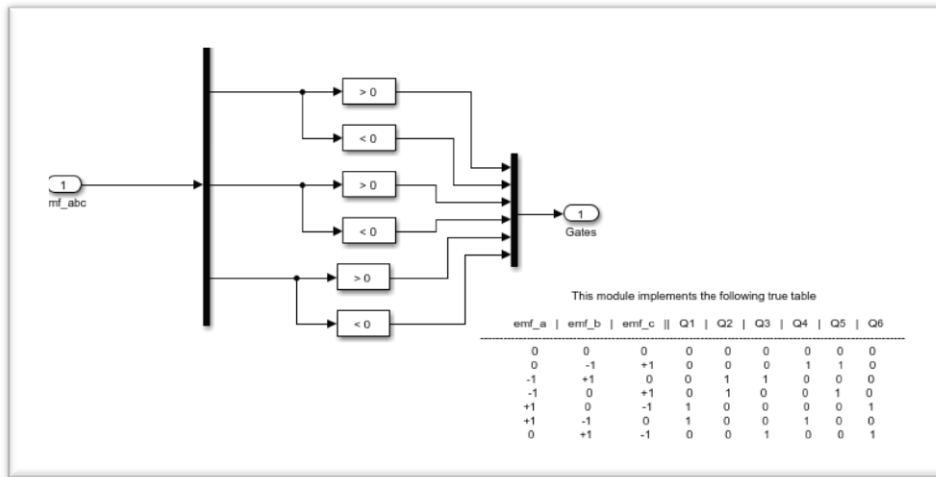


Figure (4.17): Internal Simulink design for gates block.

A Hall sensor input used to calculate the BLDC motor speed in RPM. The following equations were used for Simulink block design (Hall to RPM) to read the speed of BLDC motor. The angular velocity of the rotor can be written as:

$$\omega_{Hall} = \frac{d\theta_m}{dt} \quad (4.1)$$

Where : ω_{Hall} rotor angular velocity calculated by Hall sensors,

which in combination with the electro-mechanical equation (2.12) in chapter two gives:

$$\omega_{Hall} = \frac{2}{p} \frac{d\theta_e}{dt} \quad (4.2)$$

The Hall sensor will change the state of each electric 60° and by measuring the time between each state change, we can get the angular velocity. Converting the angular velocity to RPM gives the final equation (Johansson, M., 2017):

$$\omega_{Hall-RPM} = \frac{60}{360} \cdot \frac{2}{P} \cdot \frac{60}{\Delta T_s} \quad (4.3)$$

Where: $\omega_{Hall-RPM}$ rotor speed calculated by Hall sensors in RPM, ΔT_s difference between the last two known time samples generated by Hall state switching.

The Simulink design for finding speed of BLDC motor is shown in Figure (4.18).

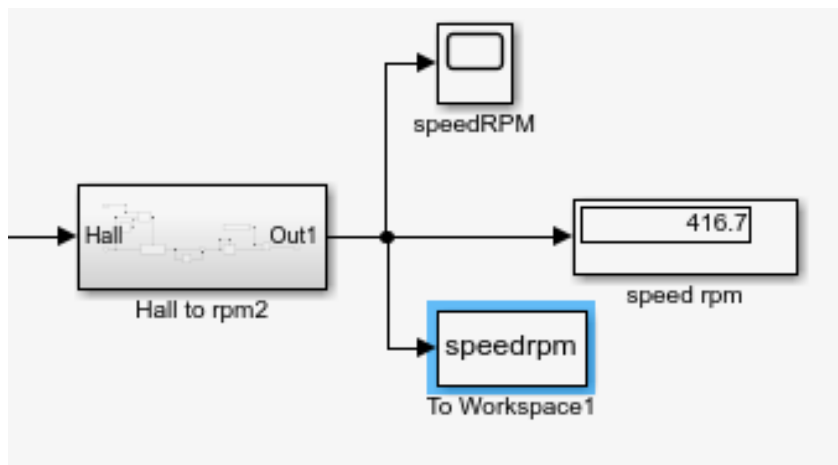


Figure (4.18): Outer Simulink design to find speed in RPM.

Simulink internal design for determining the speed in RPM is illustrated in Figure (4.19).

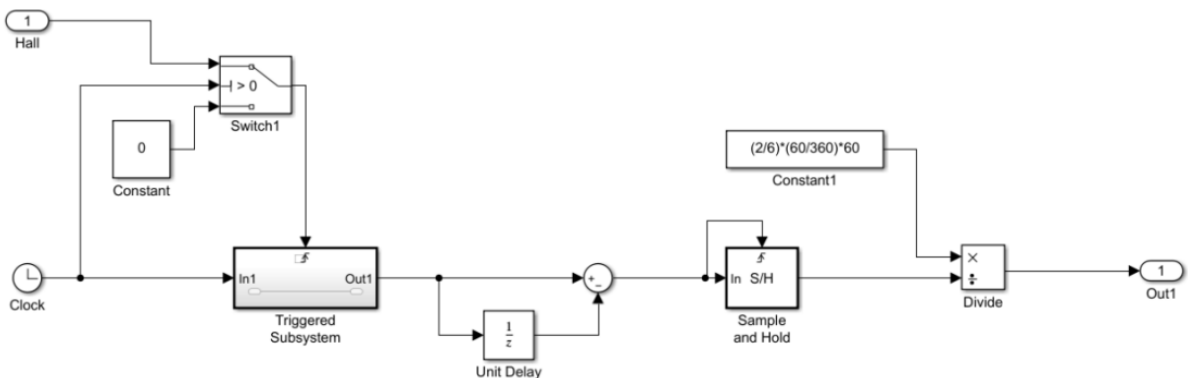


Figure (4.19): Internal Simulink design to find speed in RPM.

The measured output speed of the MOONS BLDC motor in RPM is shown in Figure (4.20)

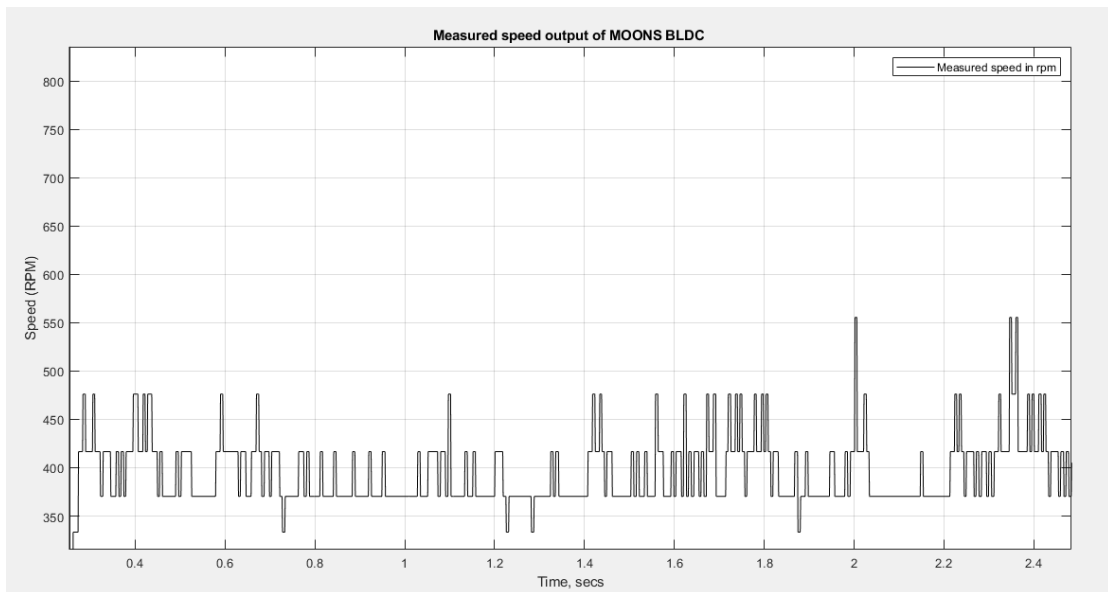


Figure (4.20): Measured speed output of MOONS' BLDC motor in RPM.

The overall Simulink design as shown in Figure (4.21), allows the BLDC motor to be successfully started and the BLDC motor speed controlled.

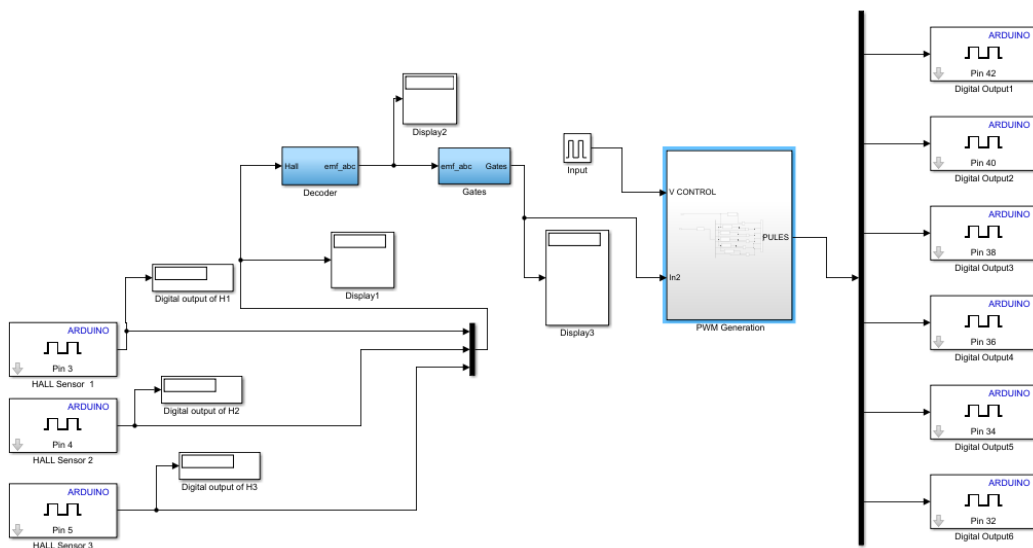


Figure (4.21): Overall Simulink design for BLDC motor.

4.4 Luenberger observer design using Simulink

The observer of motors used in this thesis are designed by Luenberger observer block in Simulink, as shown in Figure (4.22).

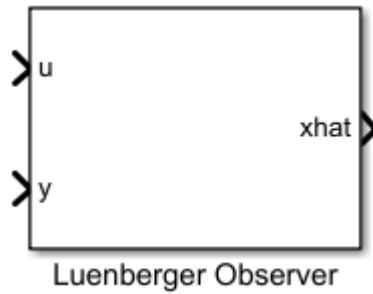


Figure (4.22): The Luenberger observer block.

The Luenberger observation block can be designed either as a continuous time system or as a discrete time system (Sellami, L., 2014). In this thesis, the design of the parameters was chosen as a continuous time system, as shown in Figure (4.23)

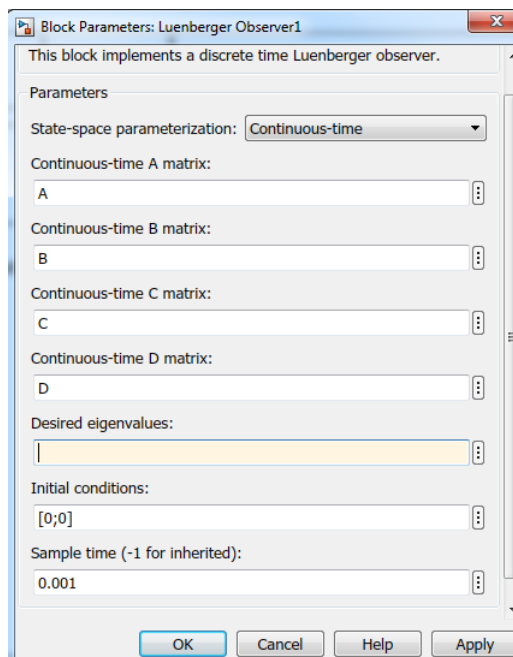


Figure (4.23): Block parameters of Luenberger observer.

The desired eigenvalues of the system (motors used in the thesis) were calculated by using the MATLAB m-file (see Appendix B) for BLDC motor and by using the MATLAB m-file (see Appendix B) for PMDC motor (Nise, 2007). The observer's eigenvalues must be chosen to be about ten times faster than the system eigenvalues (Pal, D., 2016). Then, a block of Luenberger observation parameters is added to the Simulink model of the Arduino program for real-time implementation via Arduino Mega 2560.

4.5 Experimental setup for PMDC motor

The control system consists of a PMDC motor with a sensor (optical encoder), an Arduino mega 2560 controller, PMDC motor drive (L298N H-Bridge) and DC power supply. A schematic diagram of the equipment setup for this experiment is shown in Figure (4.24) and corresponding photo of the Experimental setup is shown in Figure (4.25).

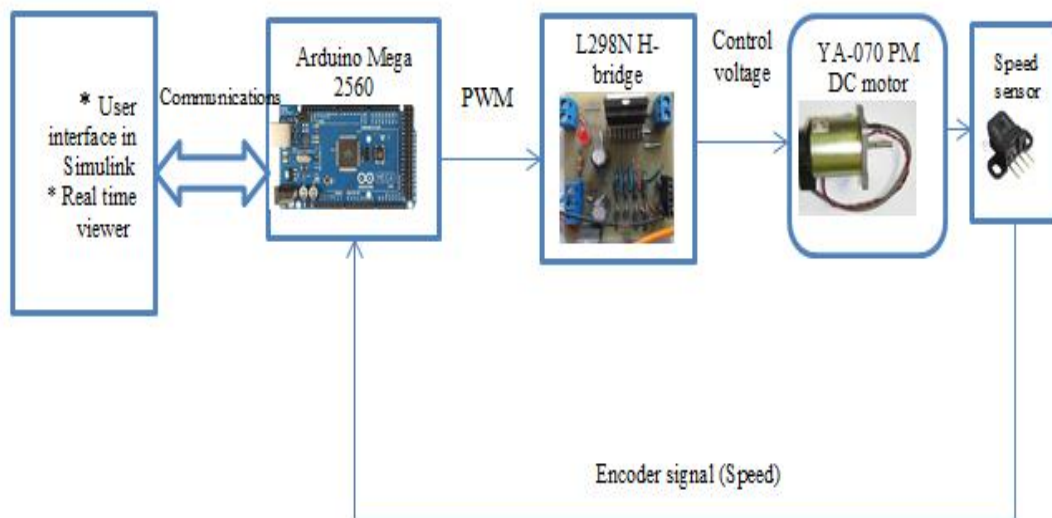


Figure (4.24): Schematic diagram of the equipment setup.

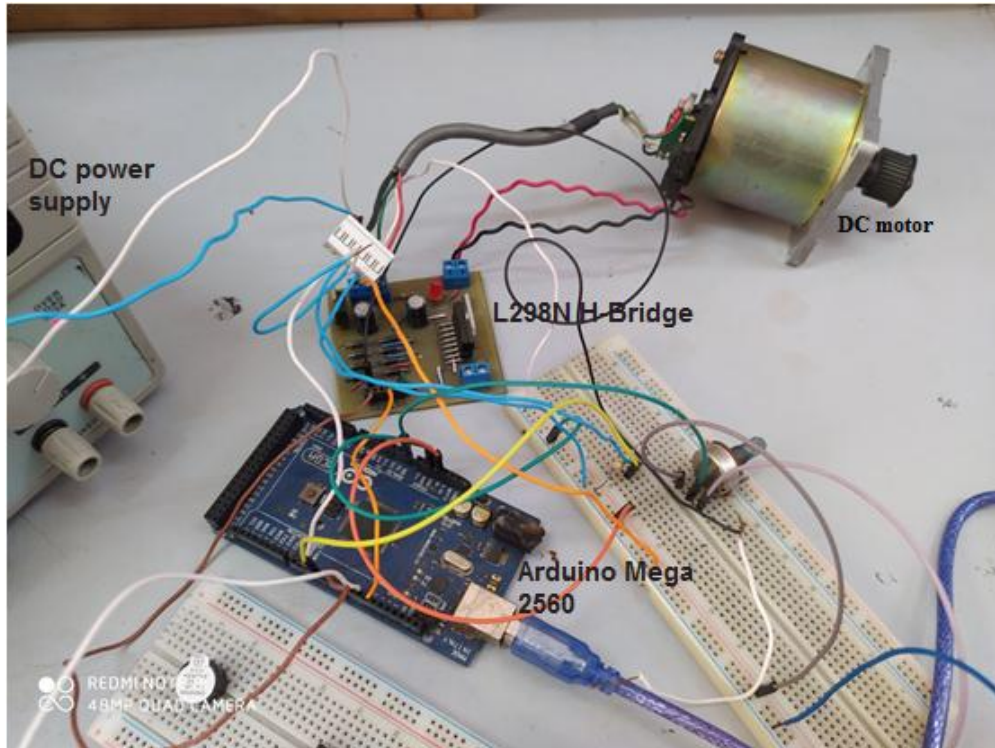


Figure (4.25): Photograph of equipment used.

4.6 Details of Hardware implementation

4.6.1 Motor controller (Arduino Mega 2560)

The Arduino Mega 2560 is a microcontroller board based on ATmega 2560 as shown in Figure (4.26) . It has 54 digital input/output pins (of which 15 can be used as PWM output), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Other specification are given in Table (4.1),which found in (Aqeel, 2018).

Table (4.1): Arduino Mega 2560 specification.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage	7-12V
USB Port	Yes
DC Power Jack	Yes
Current Rating Per I/O Pin	20mA
Current Drawn from Chip	50mA
Digital I/O Pins	54
PWM	15
Analog Pins (Can be used as Digital Pins)	16(Out of Digital I/O Pins)
Flash Memory	256KB
SRAM	8KB
EEPROM	4KB
Crystal Oscillator	16 MHz
LED	Yes/Attached with Digital Pin 13
Wi-Fi	No
Shield Compatibility	Yes



Figure (4.26): Arduino Mega 2560 board (motor controller).

4.6.2 L298N H Bridge Dual Motor Drive

The L298N is a dual H-bridge motor driver board that used to control the speed and direction of two DC motors (Dejan, 2017), which its datasheet available in Appendix D. H-bridges are circuits used to control the transmission of large currents with low currents. The motor drive circuit is design by using Proteus 8.6 software .The practical implementation of the PCB for the motor drive circuit shown in Figure (4.27). Appendix C contain a complete design of the circuit.

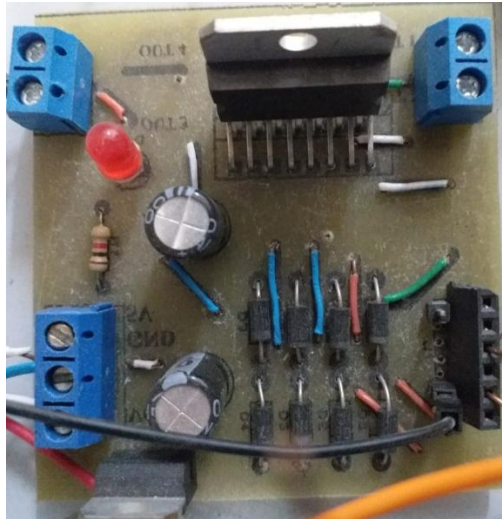


Figure (4.27): Motor driver (L298N H-bridge).

4.6.3 Optical encoder

The feedback used here is an optical incremental encoder, which is a linear or rotational electromechanical device with two output signals A and B that give out pulses when the device moves. More Details about optical encoder can be found in (Creative Robotics Ltd). Figure (4.28) shows the output signals of optical encoders .The sensor provides incremental position feedback, which can be extrapolated to accurate speed or position information.

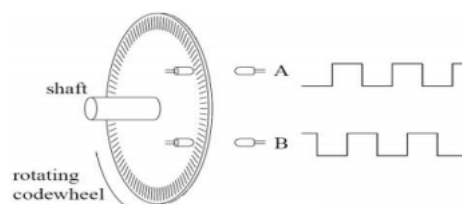


Figure (4.28): Optical encoder signals.

The optical incremental encoder used in the YA-070 PMDC motor is the H9700, as shown in Figure (4.29), where the data sheet is presented in Appendix D.



Figure (4.29): H9700 optical incremental encoder.

A Digital Storage Oscilloscope has been used to store the waveform of optical incremental encoder signal pulses. The sampled speed signal is shown in Figure (4.30). In this study, the speed sensor signal is sampled as an analog signal using a low-pass filter.

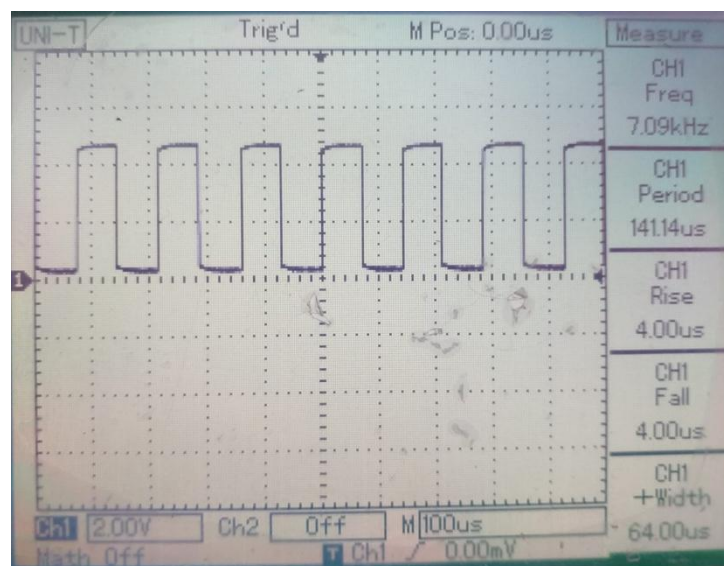


Figure (4.30): Optical sensor output (Scale:2 V/div).

4.7 Hardware/ Practical Implementation

This section takes up the working principle of all the fundamental devices of our thesis, and after each explanation, a description of the device, circuit or integrated circuit that we have chosen to use will be included and illustrated with real images.

4.7.1 Experimental setup for BLDC motor

The measurement data is read into the PC by Arduino Mega 2560, which can also process data in real time. The interface can be programmed with Simulink, which is a special toolbox of the numerical calculation software MATLAB . The experimental setup consists of seven parts as shown in Figure (4.32) , which are MOONS' BLDC motor, Power supply with three DC channel, Arduino Mega 2560, invert circuit, driver circuit, Hall sensor circuit and the host computer. The host computer is used to execute control algorithms and is connected to the BLDC motor. The motor speed is determined by the Arduino Mega 2650 microcontroller. Schematic diagram of the hardware implementation is illustrated in Figure (4.31) and Figure (4.32) shows the corresponding photo of the Experimental setup.

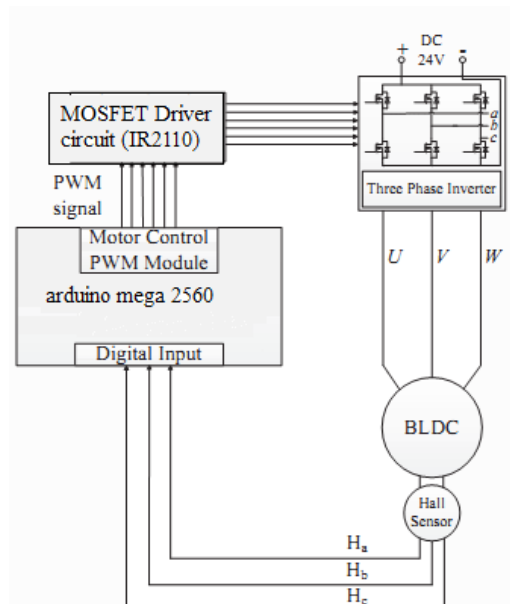


Figure (4.31): Schematic diagram of hardware implementation.

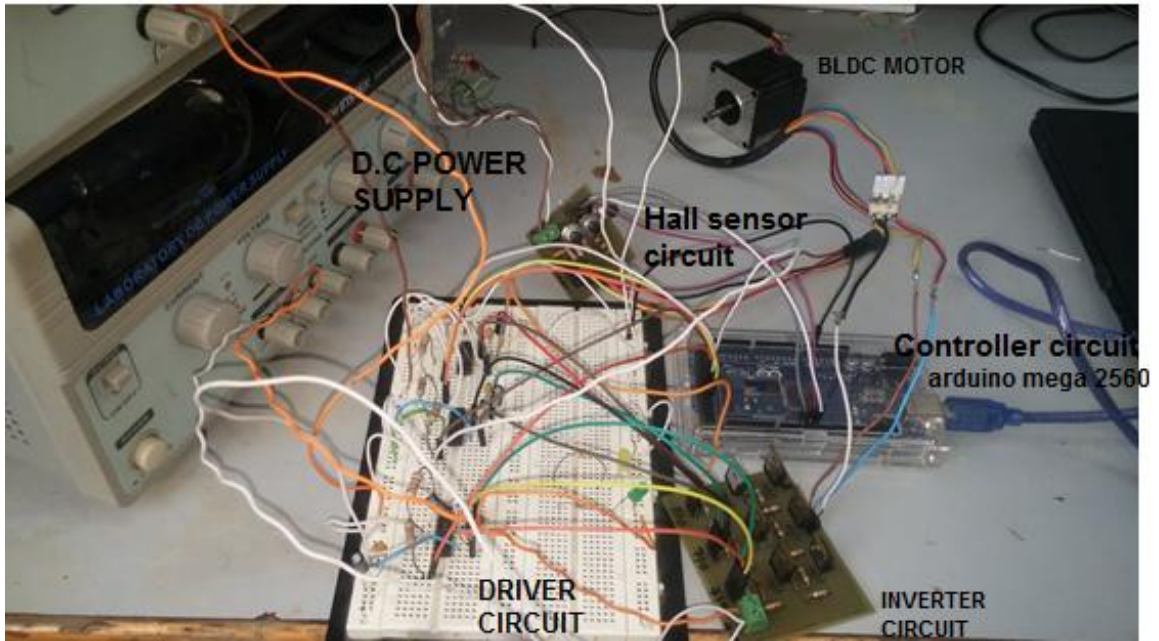


Figure (4.32): Photo of the experimental setup of the BLDC motor.

4.7.2 Details of Hardware implementation

4.7.2.1 Motor controller (Arduino Mega 2650) and interface circuit

Arduino Mega 2560 is a hardware interface between computer, sensors and actuators. To collect real input or output data from the physical system, the Arduino Mega 2650 is programmed using the Arduino block set developed by MATLAB on Simulink to send the PWM signal to the six switches of the three phase inverter. Then, the inverter is also connected to phases of motor (U, V and W) and receive speed measurement from the Hall sensors (H_1 , H_2 , H_3). A simplified block diagram of the motor controller is shown in Figure (4.33).

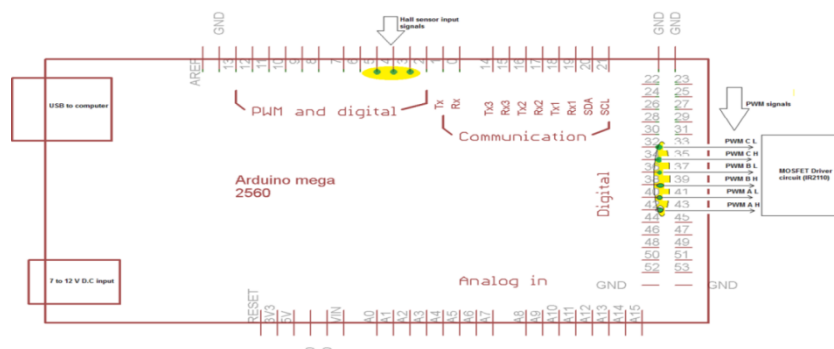


Figure (4.33): Simplified diagram of the motor controller.

4.7.2.2 Inverter circuit design

Inverters are power electronic circuits that transfer power from a direct current (DC) source to an alternating current (AC) source. The inverter output has three alternating voltage signals with a delay of 120° . The inverter has a DC link that will be converted by switching MOSFET's into AC voltage. BLDC motor is driven by three phase inverter. In our system, we need to use a three-phase inverter with six switches (N-channel MOSFET's) which are divided into three up/down arms. A inverter circuit diagram shown in Figure (4.13), the upper arm includes upper bridge switches (A+ , B+ ,and C+). In addition, the lower arm includes lower bridge switches (A-,B-,and C-).

The inverter circuit is design by using Proteus 8.6 software . In this circuit, the MOSFET (IRF740 N) is used, which its datasheet available in Appendix D . Figure (4.34) shows the practical implementation of printed circuit board (PCB) for the inverter circuit and appendix C contain a complete design of the circuit.

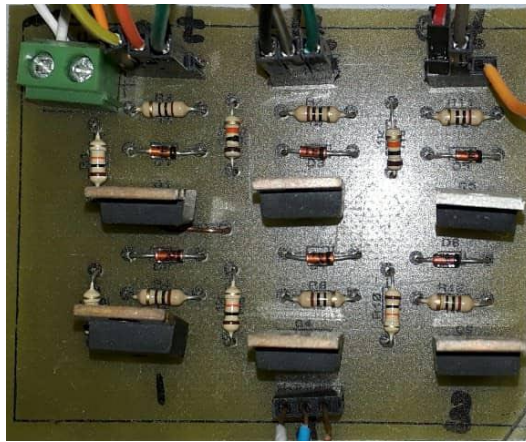


Figure (4. 34): Photo of the PCB board.

A short circuit will be occurred when the bridge switch with one arm up and down turned on at the same time, which will seriously damage the circuit. Therefore, in accordance with the principle of complementary switching of the up and down bridge, there are six types of switching patterns. However, this solution rises high side gate driving problem. A high side n-channel switch needs special gate driving technique. The solutions to the problem are TLP250 isolated gate driving ,

transformer isolated gate driving, and bootstrap technique (Balogh, L., 2001). In this thesis the last technique is chosen for its simplicity. An integrated circuit, IR2110, is chosen which its datasheet available in Appendix D. Also, the IR2110 driver itself helps to handle with this problem. As shown in Figure (4.35) an internal delay is added on the structure of the IR2110 in order to avoid the simultaneous conduction of the switchers.

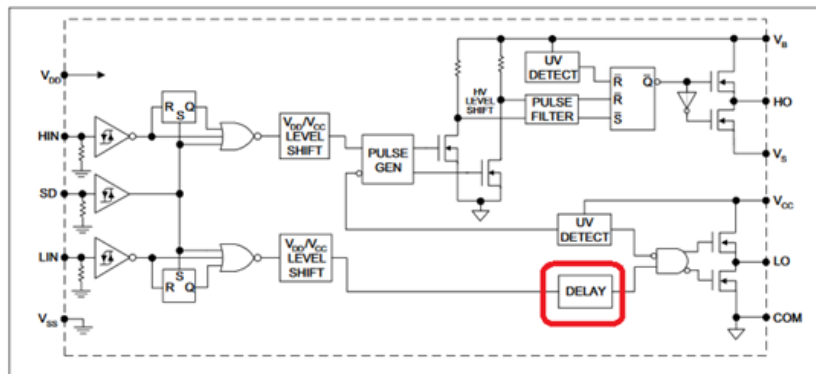


Figure (4.35): Internal block diagram of the IR2110.

Electronic block diagram of the IR2110 drive circuit is shown in Figure (4.36). The driver circuit is repeated 3 times (each driver triggers complete inverter arm). More Details can be found in (Mahbub, 2013).

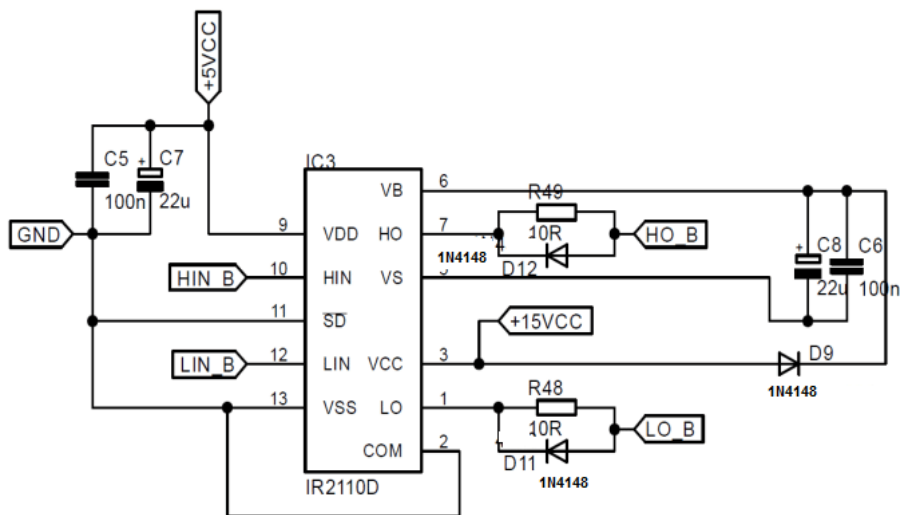


Figure (4.36): Electronic block diagram of the IR2110 Driver (Franceschet, et al., 2017).

The drive circuit is design by using Proteus 8.6 software. Figure (4.37) shows the practical implementation of PCB for IR2110 driver circuit and appendix C include a complete design of the circuit. The technical specifications of components and driving parameters are presented in Table (4.2)

Table (4.2): Technical specifications of components and driving parameters.

Component/Parameter	Specification
Input voltage	24 V
Controller	Arduino Mega 2560
Inverter drive	IR2110
MOSFET	IRF740
Switching frequency	500 H_z



Figure (4.37): Photo of the PCB board.

4.7.2.3 Hall effect Sensors

Hall effect sensors (H_1 , H_2 , H_3) are also called (Hall sensors) "is a transducer which convert the change in magnetic field into a change in output voltage". Figure (4.38) shows the Hall effect principle.

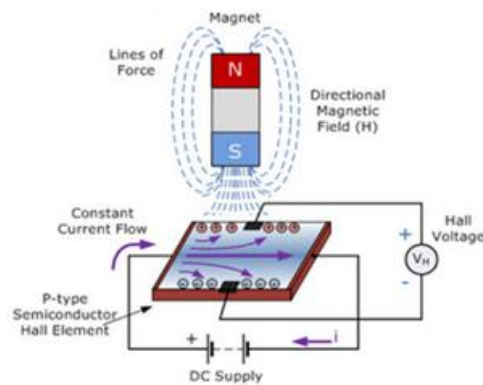


Figure (4.38): Hall effect principle.

Hall effect sensors are used to determine the position of the rotor and speed detection that been built into the stator itself. This will send a digital signal (high or low), which will detect the position of the rotor (Poovizhi,et al., 2017). Hall effect sensors are placed at 120° electrical apart. For every 60° , one of the Hall sensors changes its state. The output of Hall Effect sensor is fed to the controller (Arduino Mega 2650) for triggering of MOSFET circuit, which requires a 5 V supply for its sensor. The connection of Hall sensor circuit is shown in Figure (4.39).

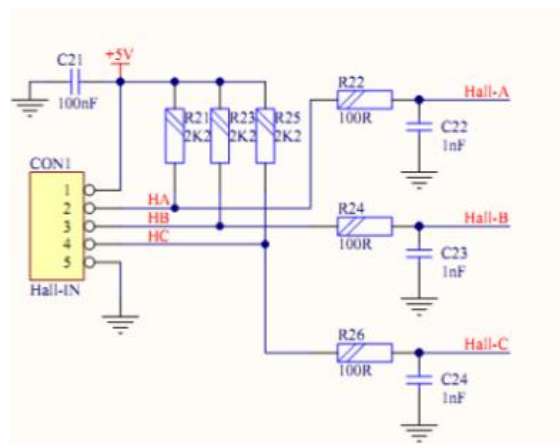


Figure (4.39): Block diagram of the hall sensor circuit.

The Hall effect sensor circuit is designed using Proteus 8.6 software. Figure (4.40) shows the practical implementation of PCB for the Hall effect sensor circuit and Appendix C includes a complete design of the circuit. A normally closed switch has been added to the circuit, in order to make a fault on the Hall effect sensor.



Figure (4.40): Photo of the PCB board.

The BLDC motor speed sensor was tested, this is done by connecting the Hall effect sensor circuit to the Simulink models on the Arduino Mega 2560 board, then connect the digital pin (32,34,36,38,40,42) to the six LEDs. Figure (4.41) shows the testing circuit for Hall sensor. Then we manually move the BLDC motor step by step according to the Table (2.2) at each step two LED lights are lit.

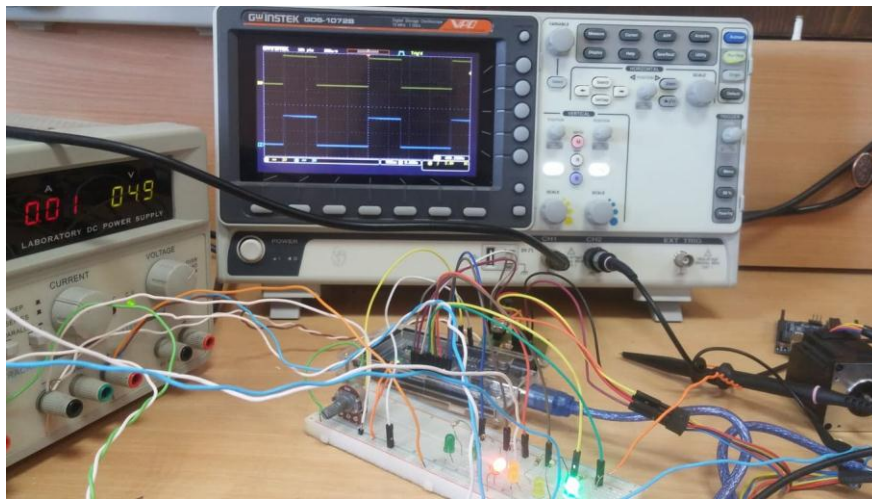


Figure (4.41): Testing circuit for hall sensor.

A Digital Storage Oscilloscope has been used to store the waveforms of Hall sensor signals. One of Hall sensor output signals is shown in Figure (4.42).

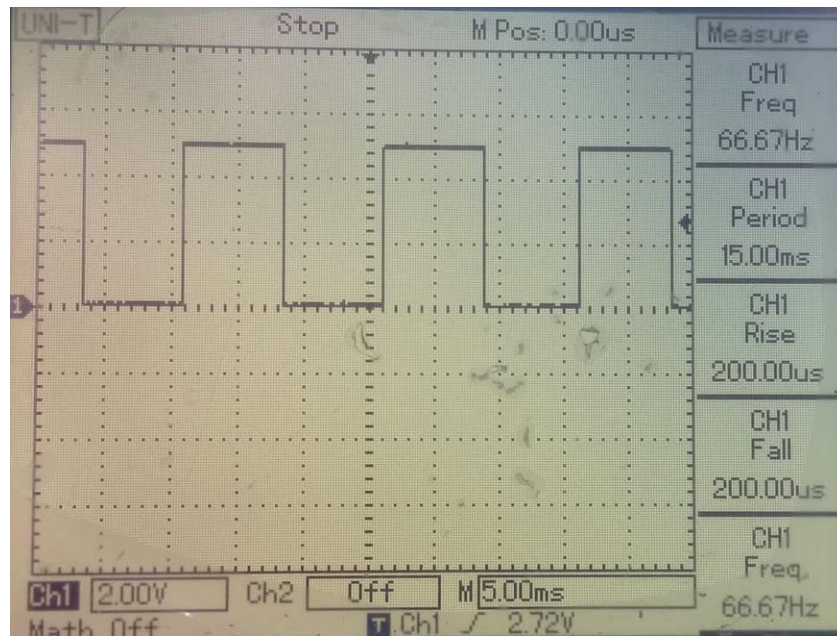


Figure (4.42): Hall sensor output (Scale:2 V/div).

Chapter 5

Simulation and Experimental Results

5.1 Introduction

The accuracy of the Luenberger observer for the PMDC motor and the BLDC motor was verified by using simulation and an experimental system, the simulation was performed on the MATLAB program by Simulink and all kinds of faults were tested on the speed of the motors. Implemented fault detection based on Luenberger observer for PMDC motor and BLDC motor by simulation in Simulink as shown in Figure (5.1) and in Figure (5.2)

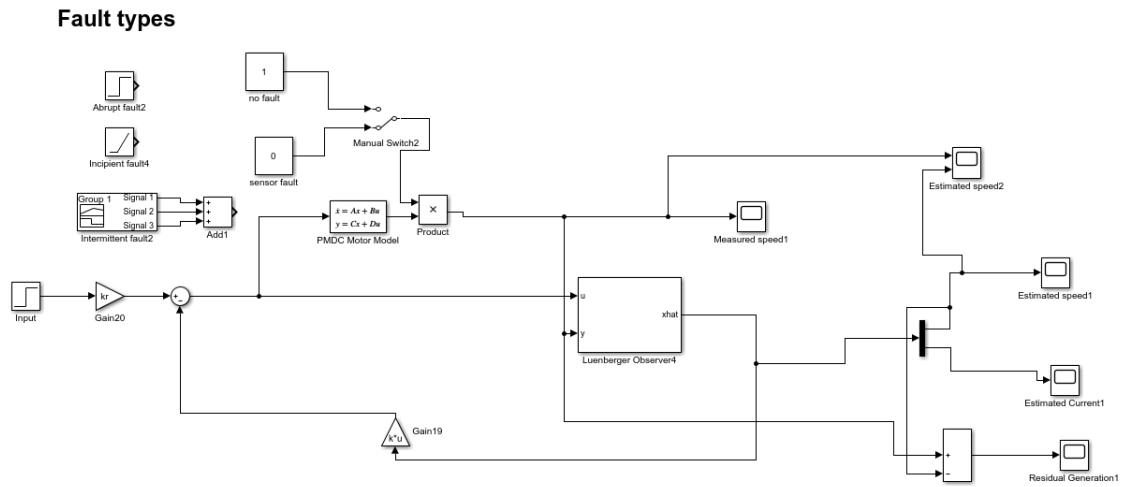


Figure (5.1): Simulation design of Luenberger observer for PMDC motor.

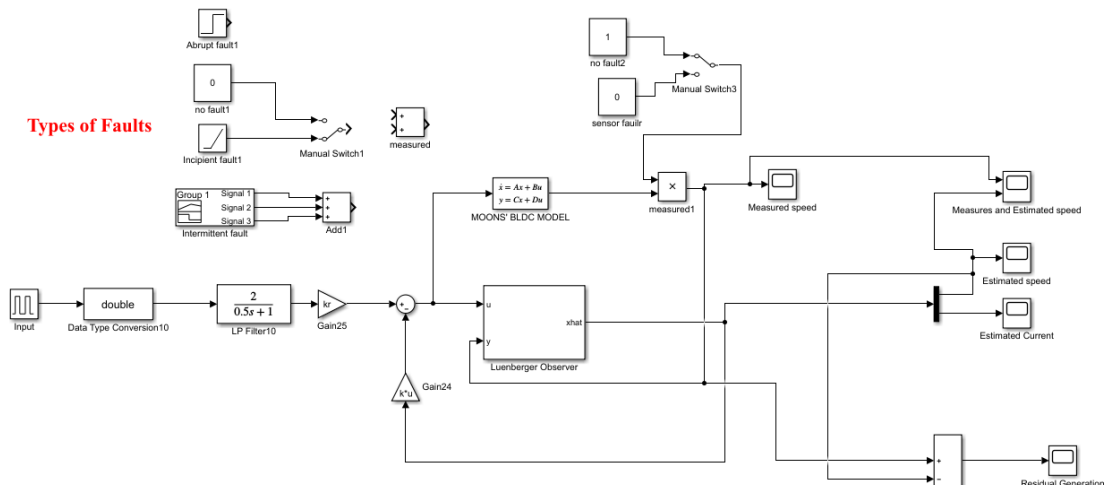


Figure (5.2): Simulation design of Luenberger observer for BLDC motor.

In the experimental system, all experiments were performed in real time. All experiments were carried out at a sampling time of $T_s = 1ms$. For the PMDC motor(YA-070), all experiments are carried out with an input voltage of 1 V (corresponds to 2100 RPM). For the MOONS' BLDC motor, all experiments are carried out with an input voltage of 5V (corresponds to 416 RPM). PMDC motor and BLDC motor models are created in MATLAB / Simulink environment. A Simulink design of the fault detection method based on the Luenberger observer for the PMDC motor and the BLDC motor as represented in Figure (5.3) and in Figure (5.4).

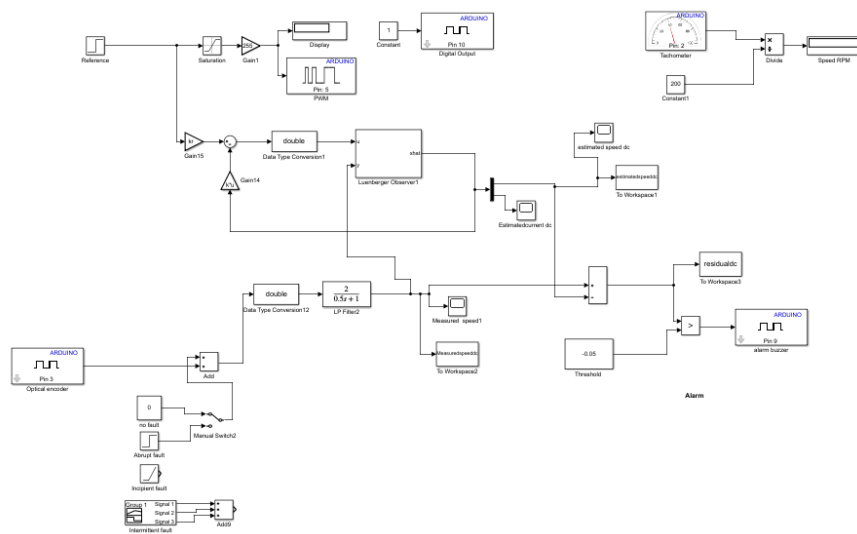


Figure (5.3): Simulink design (real-time) of a Luenberger observer for PMDC motor.

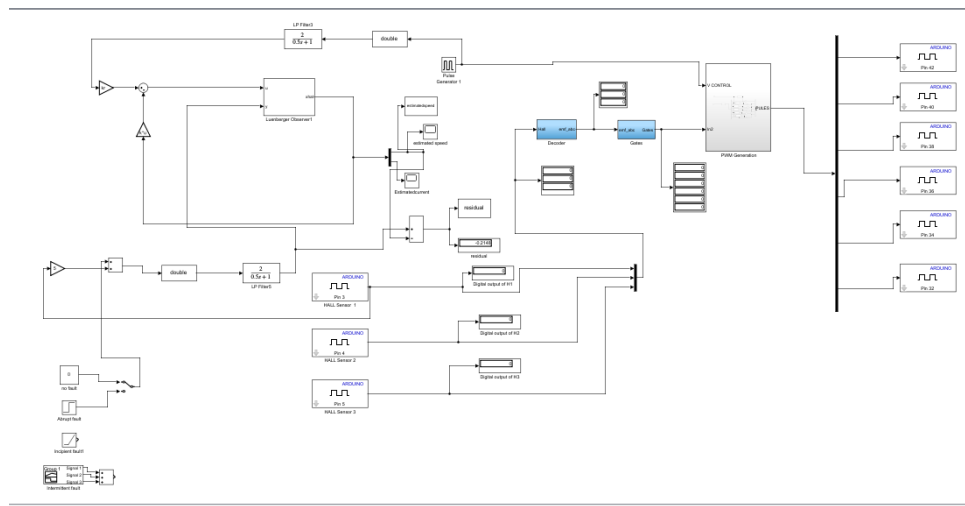


Figure (5.4): Simulink design (real-time) of a Luenberger observer for BLDC motor.

At the time the model was constructed, entering input data into the Luenberger observer will generate the residuals and by monitoring the corresponding estimated outputs of the model and compare it with the measured values. Then faults are considered residuals. Fault detection thresholds are defined according to the maximum values achieved by residuals over several experiments. There are two types of thresholds, one of which is fixed threshold and the other is adaptive threshold (Frank, et al., 2000). Fixed Type threshold was used in this thesis where more than one test was applied on the motors at fault free state in order to determine the values of both the upper and lower thresholds. As soon as the residual exceeds a specific threshold, the alarm is triggered to denote a fault. A buzzer device was used as an alarm.

In this thesis the sensor faults in motors (PMDC and BLDC motors) was tested .The kinds of faults discussed in this thesis are as follows (Farmen, et al. , 2018):

Additive fault: "occurs due to the change of internal temperature or calibration issues, the speed sensor output has contains an additional constant value such as intermittent and abrupt fault" (ALKAYA, 2012). The abrupt fault model is designed as a step function in Simulink. The intermittent fault model is designed as a series of pulses of different amplitudes in Simulink.

Multiplicative fault: this may be due to the fact that a multiplier coefficient is applied to the sensor, for example, an incipient fault. The incipient fault model is designed as a ramp function in Simulink (ALKAYA, 2012).

Sensor fault: this is a disastrous fault, The sensor fails at a certain point in time and generates a constant zero after the appearance of the fault (ALKAYA, 2012). This fault was practically implemented by separating the speed sensor from the motors for a short period, then reconnecting it again, or by cutting off the power to the motors. Sensor fault is implemented as online (Real time) .

All the faults were implemented in the motors speed sensor and the results were successful. In addition, the alarm was activated at each fault. In this chapter, we will first show the results of the work in the form of simulation then show the results

of the practical work, we will see that the practical results are very close to the results of simulation obtained on the motors used.

5.2 Simulation and experimental results for PMDC motor

5.2.1 Results of simulation without fault for PMDC motor

This experiment was performed as a simulation, and the simulation was performed as if there was no fault in the PMDC motor speed as shown in Figure (5.1). The simulation of output speed and estimated speed is shown in Figure (5.5) and Figure (5.6). Figure (5.7) shows the simulation of the residual output.

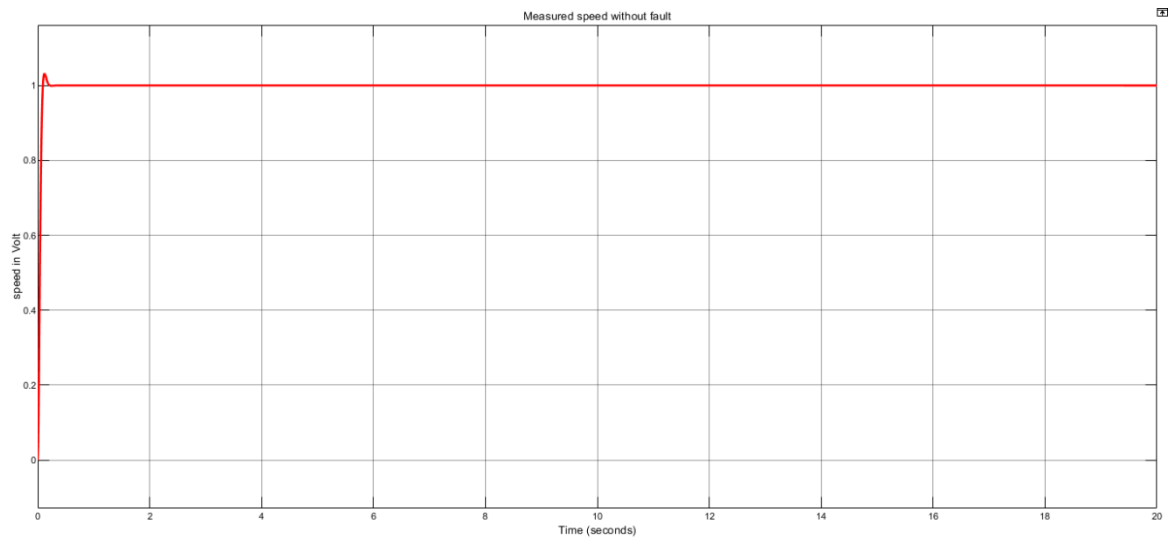


Figure (5.5): Simulation of output speed without fault.

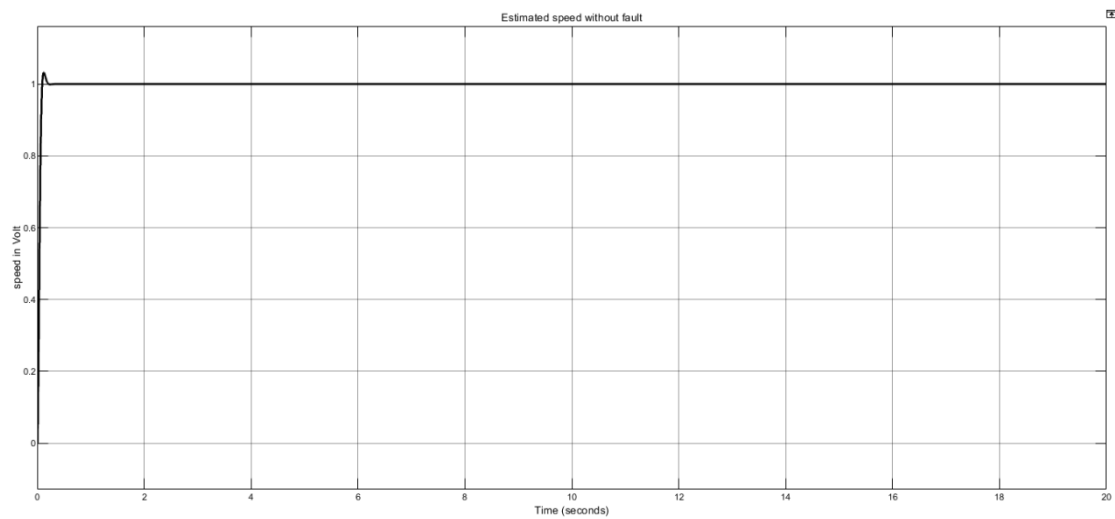


Figure (5.6): Simulation of estimated output speed without fault.

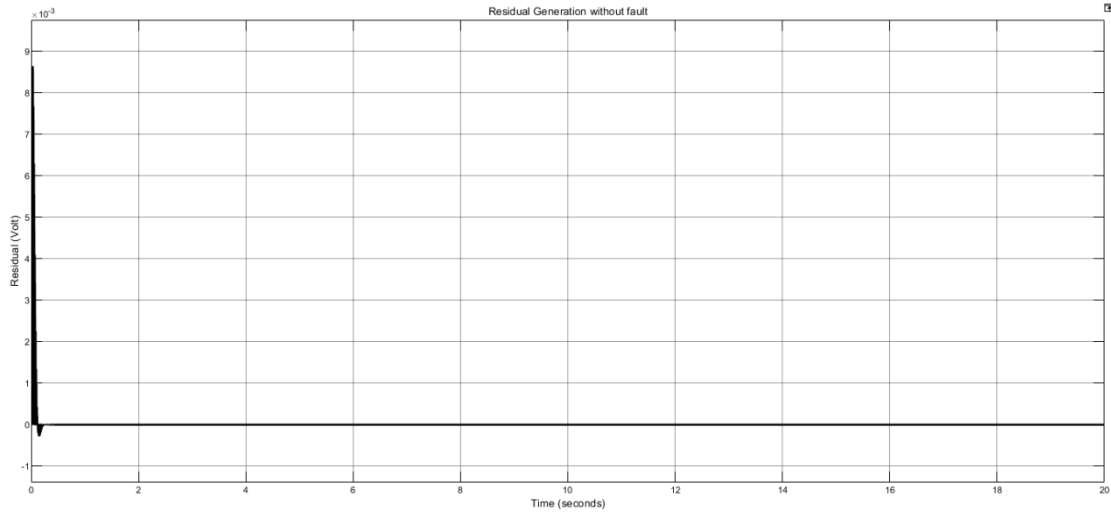


Figure (5.7): Simulation of residual output with no fault for PMDC motor.

We notice from the previous figure that the residual generation is almost zero because the experiment was performed as a simulation in Simulink.

5.2.2 Experimental results without fault for PMDC motor

This experiment was carried out without any fault in the speed sensor of the PMDC motor as illustrated in Figure (5.3), Figures (5.8) and (5.9) shows the measured and estimated speed of PMDC motor respectively.

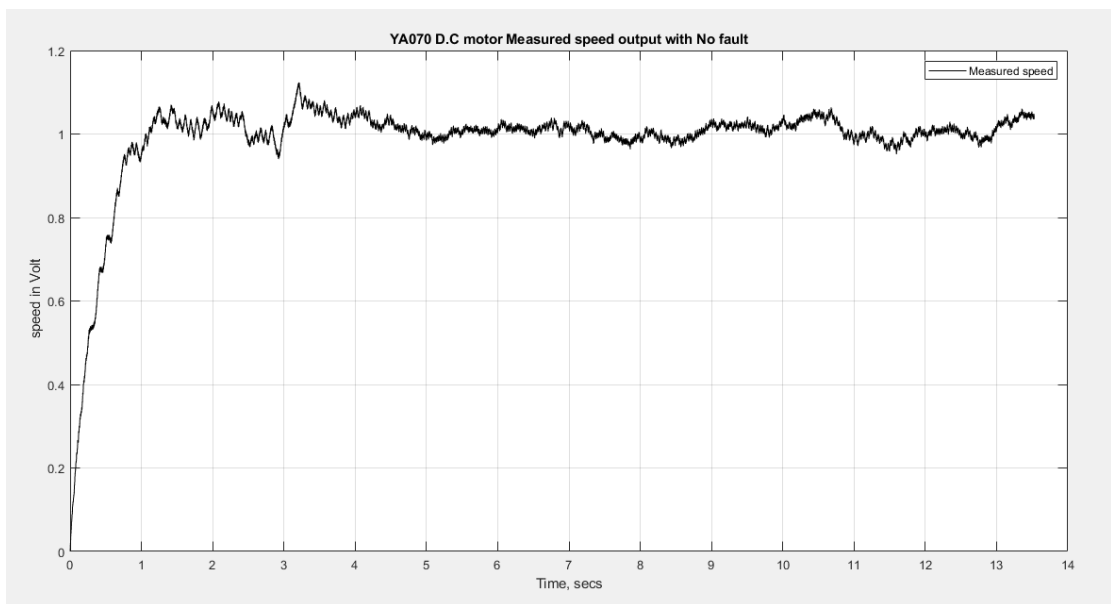


Figure (5.8): The measured speed of PMDC motor.

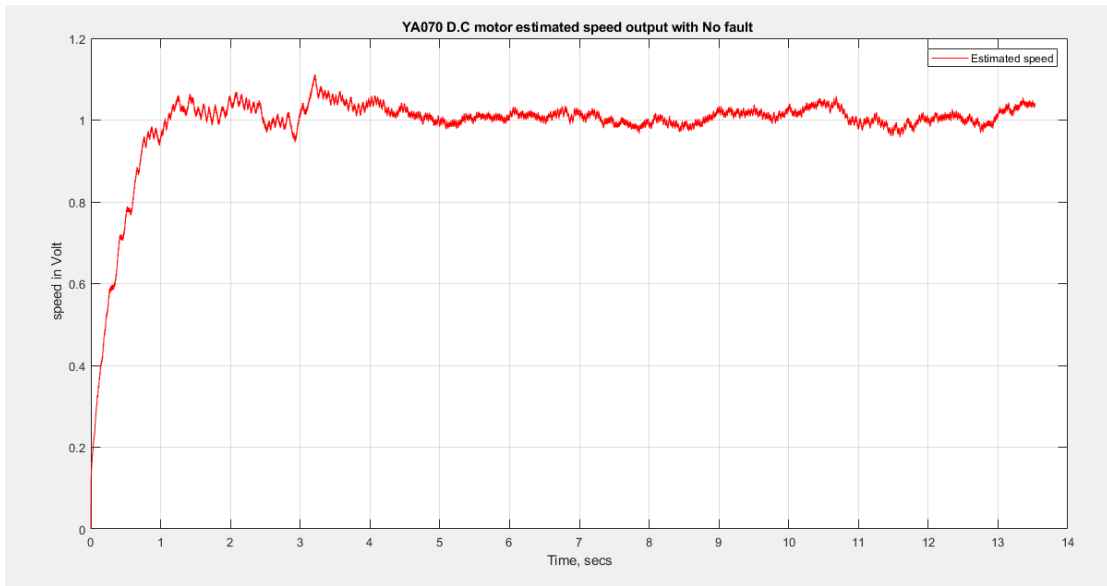


Figure (5.9): The estimated speed of PMDC motor.

More than a test have been experimentally applied on the PMDC motor while it is fault free in order to determine the threshold value from the residual generation . At the test applied the upper threshold value was 0.0157 while the lower threshold value was - 0.009 as shown in the Figure (5.10). It is important to know the threshold values, so it is necessary to develop an alarm when faults occur in the motor speed.

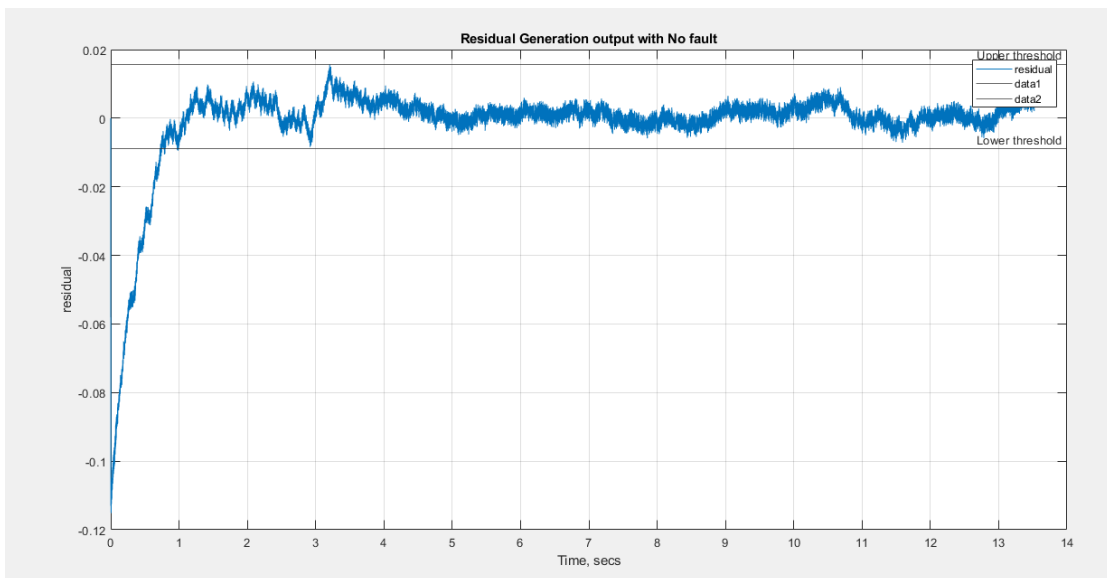


Figure (5.10): Residual output without any fault, upper threshold 0.0157, lower threshold value was - 0.009.

From the previous figure (5.10), note that the residual generation is not equal to zero due to the noise that presented in the PMDC motor and errors in the

parameters of observer design. Therefore, we chose a higher threshold value of 0.0157 and a lower value of - 0.009 in order to avoid false alarms for the residual value. These threshold values were adopted for PMDC motor tests to implement the alarm.

5.2.3 Simulation results of abrupt fault for PMDC motor

The simulation was carried out as if there was an abrupt fault in the speed sensor of the PMDC motor, as illustrated in Figure (5.1). At 14 seconds of the measured motor speed sensor an abrupt fault was implemented, The output and estimated speeds as a simulation are shown in Figure (5.11), and residual output as a simulation is illustrated in Figure (5.12)

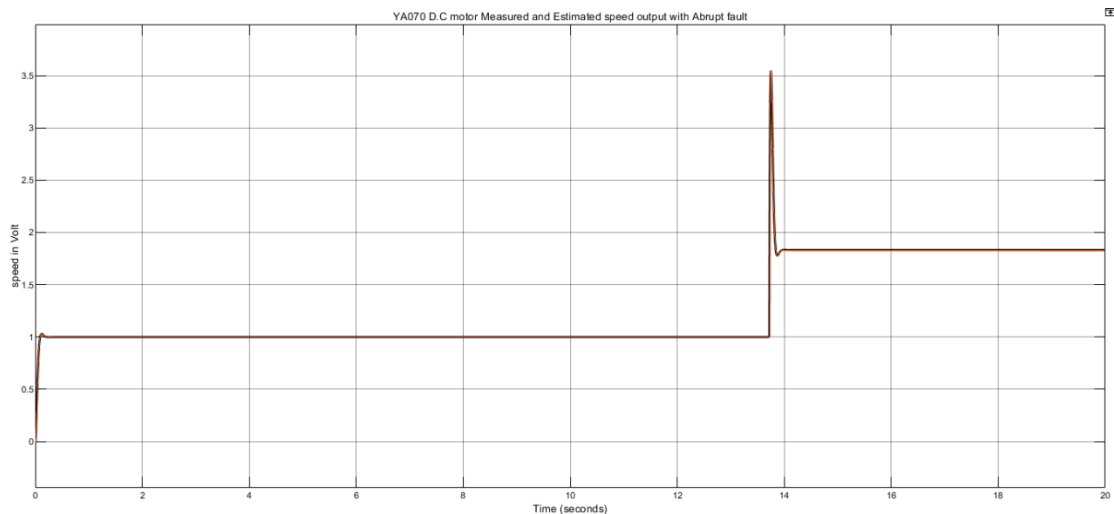


Figure (5.11): Simulation of output and estimated speeds with abrupt fault.

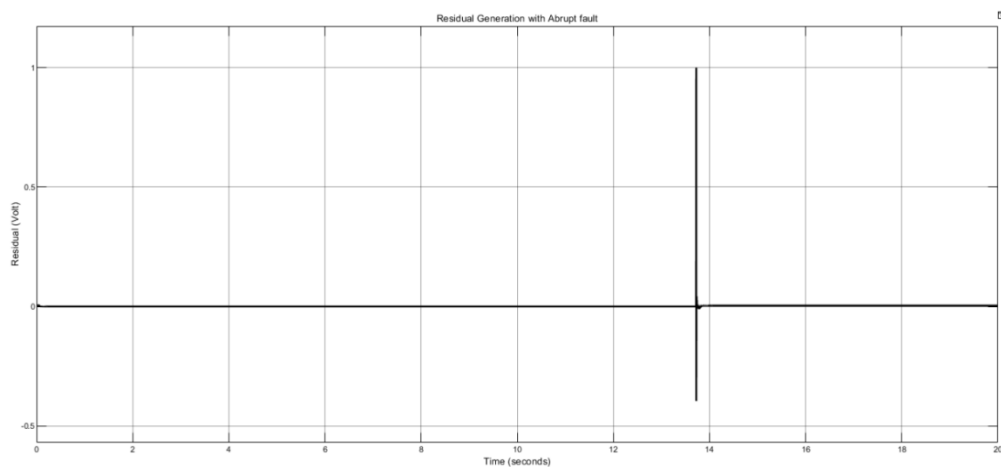


Figure (5.12): Simulation of residual output with abrupt fault.

From Figure (5.12), we notice that the residual value at 14 seconds exceeds the threshold value, the experiment was performed as a simulation in Simulink.

5.2.4 Experimental results of abrupt fault for PMDC motor

The abrupt fault was introduced during this experiment and applied to the speed sensor for the PMDC motor, as shown in Figure (5.3). This fault is implemented by adding a constant value to the sensor reading. At 3 seconds of the measured motor speed sensor an abrupt fault was implemented. Figure (5.13) shows responding of the measured and estimated speeds of PMDC motor. and Figure (5.14) illustrate the residual output, we can see that the fault detection was carried out successfully at the time of its appearance.

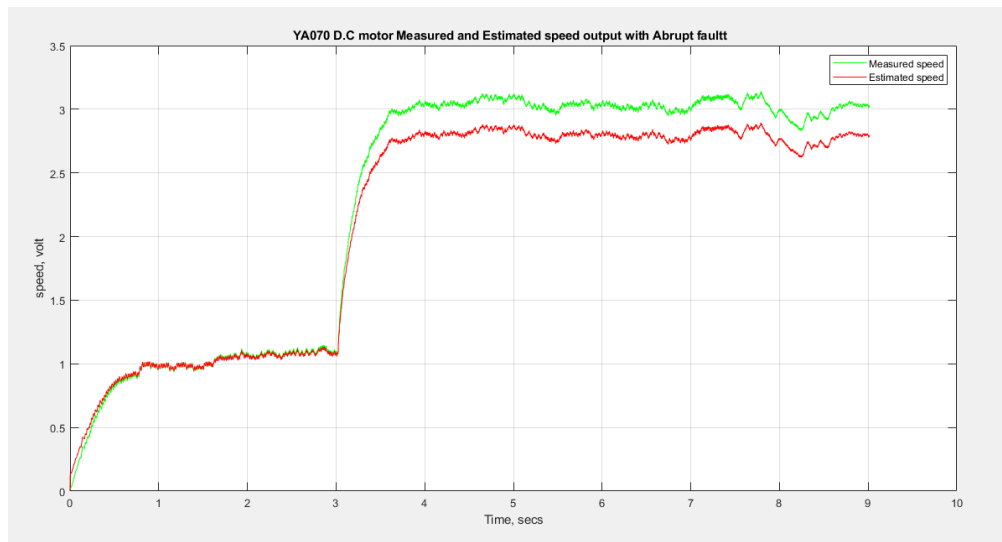


Figure (5.13): Measured and estimated speeds of PMDC motor with Abrupt fault.

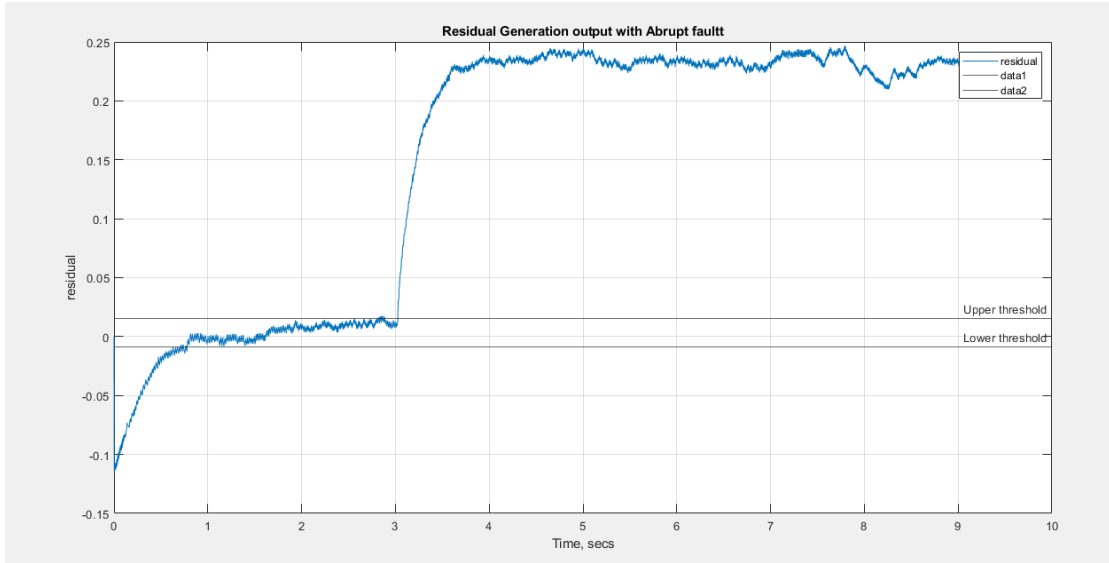


Figure (5.14): Residual output with Abrupt fault.

From the previous figure, we note that the residual value exceeds the threshold value, which indicates a fault in the sensor. At this moment, the alarm is triggered.

5.2.5 Simulation results of incipient fault for PMDC motor

The simulation was performed as if there was an incipient fault in the speed of the PMDC motor, as illustrated in Figure (5.1). At 11 seconds an incipient fault was implemented to the output speed. The output and estimated PMDC motor speeds as a simulation are shown in Figure (5.15), and the residual output as a simulation is shown in Figure (5.16)

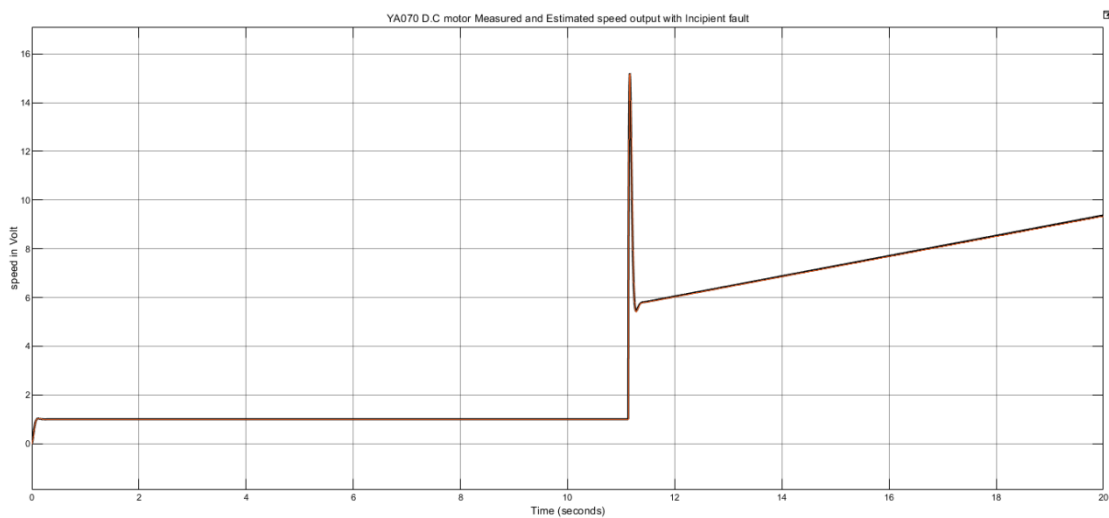


Figure (5.15): Simulation of output and estimated speeds with incipient fault.

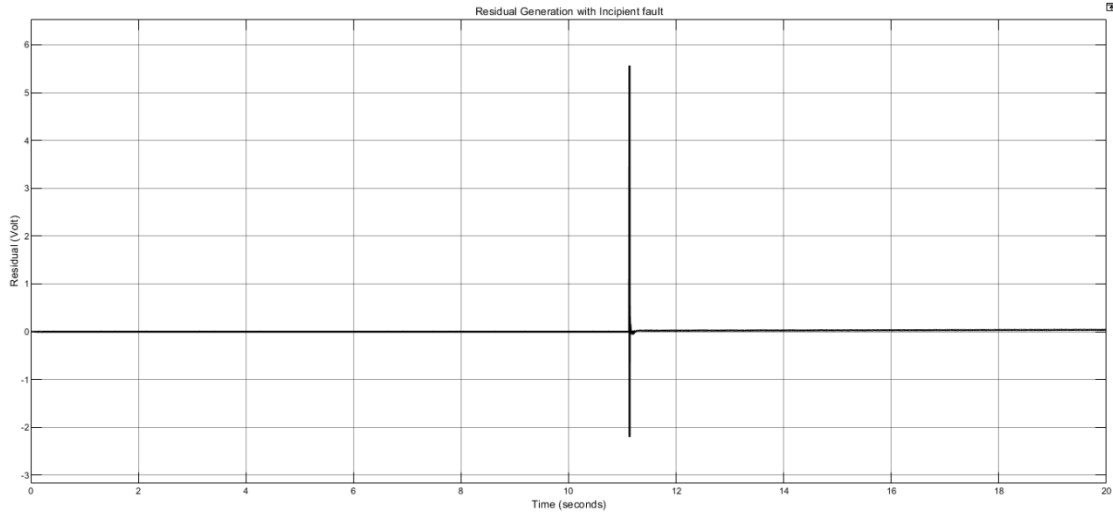


Figure (5.16): Simulation of residual output with incipient fault.

From the previous figure we see that the residual value increases above the threshold value at 11 seconds, that is, the time when the fault occurred.

5.2.6 Experimental results of incipient fault for PMDC motor

The incipient fault was introduced during this experiment and Figure (5.3) shows the its application to the speed sensor and was implemented to the output speed at 2 seconds. Figure (5.17) shows the responding of the measured and estimated speeds, and Figure (5.18) illustrate the residual output, we can see that the fault detection was carried out successfully at the time of its appearance .

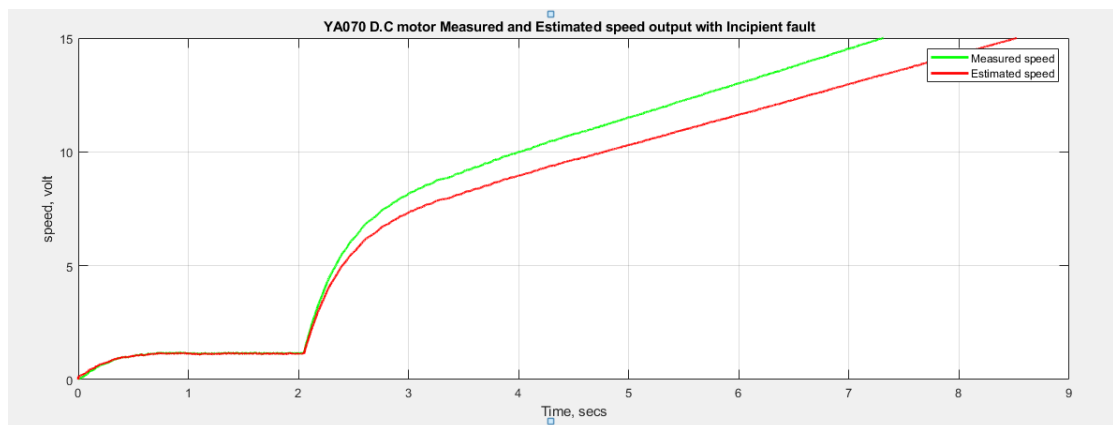


Figure (5.17): Measured and estimated speeds of PMDC with incipient fault.

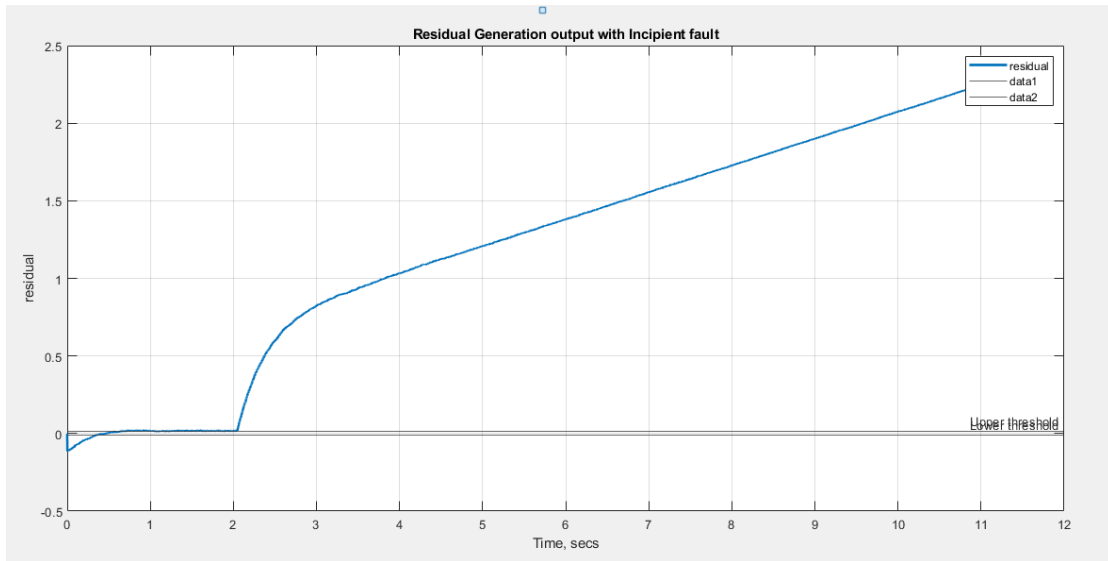


Figure (5.18): Residual output with incipient fault.

From the previous figure we see that the residual value gradually increases above the threshold value for 2 seconds, that is, the time when the fault occurred, which indicates a fault of the speed sensor.

5.2.7 Simulation results of intermittent fault for PMDC motor

This experiment was presented as a simulation, by applying an intermittent fault in the speed of the PMDC motor, as shown in Figure (5.1). After 5, 8, and 11 seconds with an amplitude of 1, 1.5, and 2, respectively an intermittent fault was applied to the output speed, each time for 1 second. The results were quite similar to the practical results in the absence of a fault. The output and estimated speeds of the PMDC motor as a simulation are shown in Figure (5.19), and the residual output as a simulation is shown in Figure (5.20).

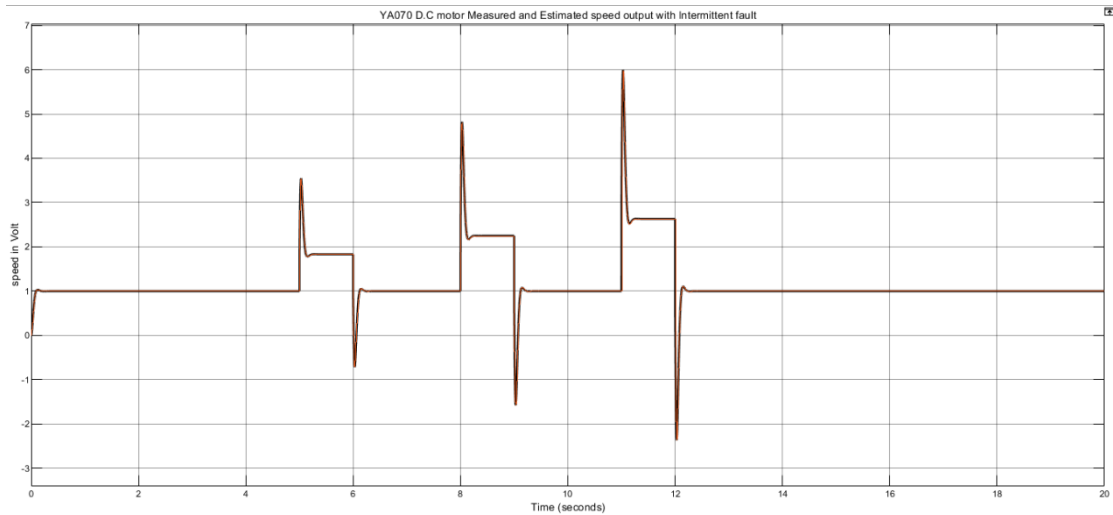


Figure (5.19): Simulation of output and estimated speeds with intermittent fault.

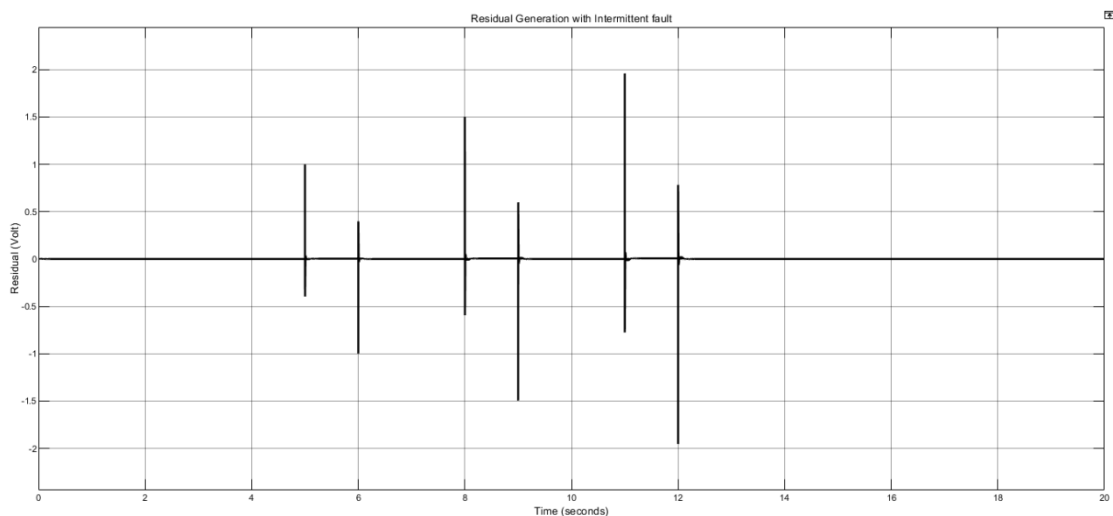


Figure (5.20): Simulation of residual output with intermittent fault.

From the previous figure, we notice that the value of the residuals exceeds the threshold value.

5.2.8 Experimental results of intermittent fault for PMDC motor

Figure (5.3) shows that an intermittent fault was introduced during this experiment and applied to the PMDC motor speed sensor. It is implemented by adding a constant value to the sensor reading in periodic periods. Intermittent fault was applied to the output speed after 5, 8, and 11 seconds with an amplitude of 1, 1.5, and 2, respectively, each time for 1 second. Figure (5.21) shows the response of the measured and estimated speeds of PMDC motor, and Figure (5.22) shows residual output, we see that the detection of the fault was successfully performed during its beginning of the appearance time and the end of the same time.

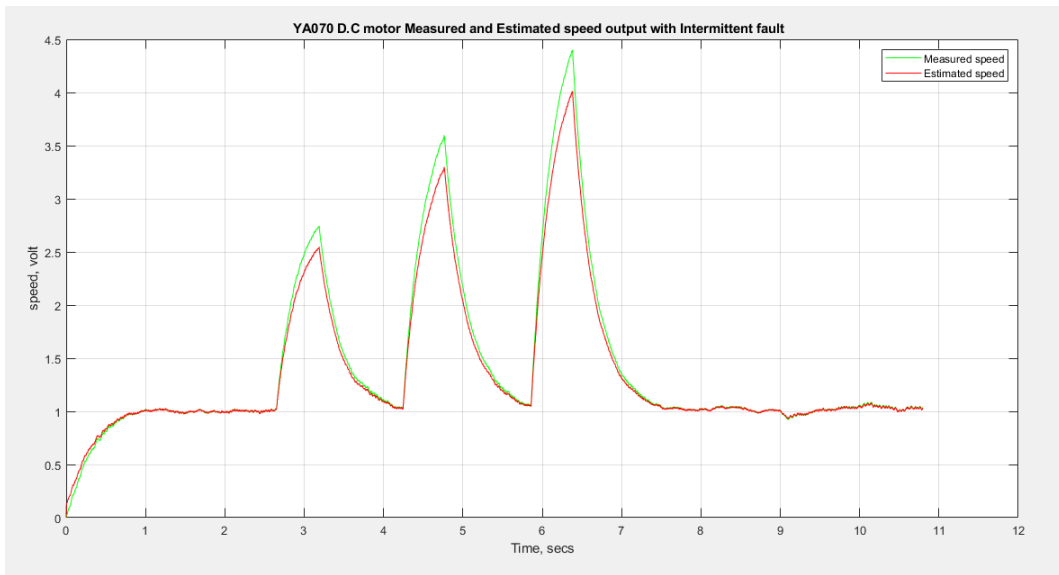


Figure (5.21): Measured and estimated speeds of PMDC with intermittent fault.

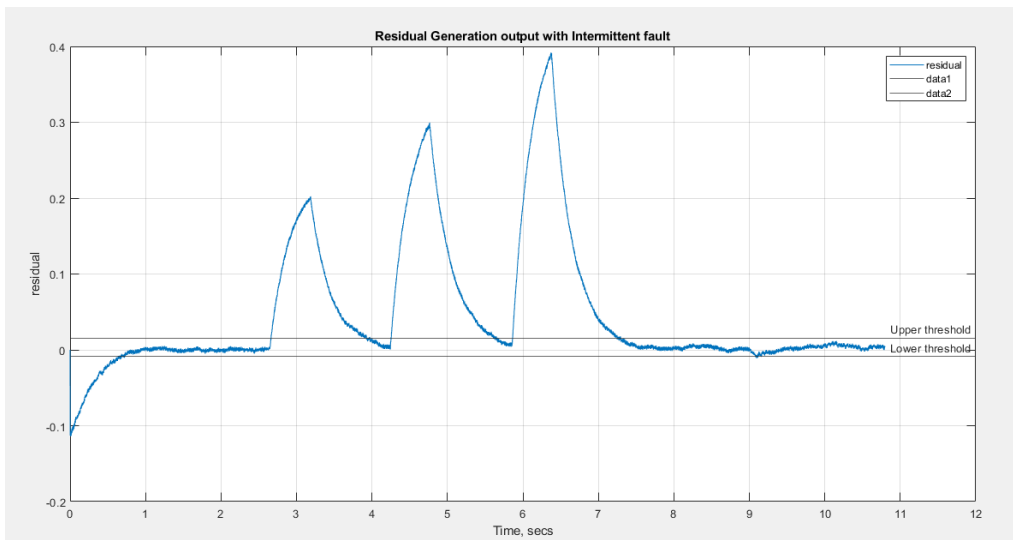


Figure (5.22): Residual output with intermittent fault.

As can be seen from the above figure, we notice that the residuals value exceeds the threshold value.

5.2.9 Simulation results of sensor fault for PMDC motor

The sensor fault simulation was performed in Simulink by multiplying the speed signal by zero. The output and estimated PMDC motor speeds as a simulation are shown in Figure (5.23). The residual simulation is shown in Figure (5.24).

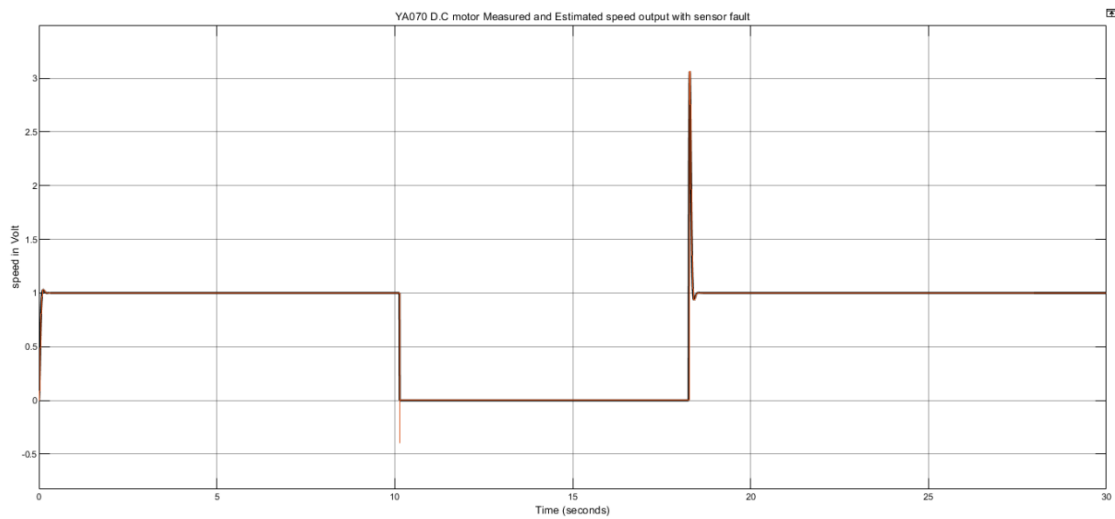


Figure (5 .23): Simulation of output and estimated speeds of PMDC motor with sensor fault.

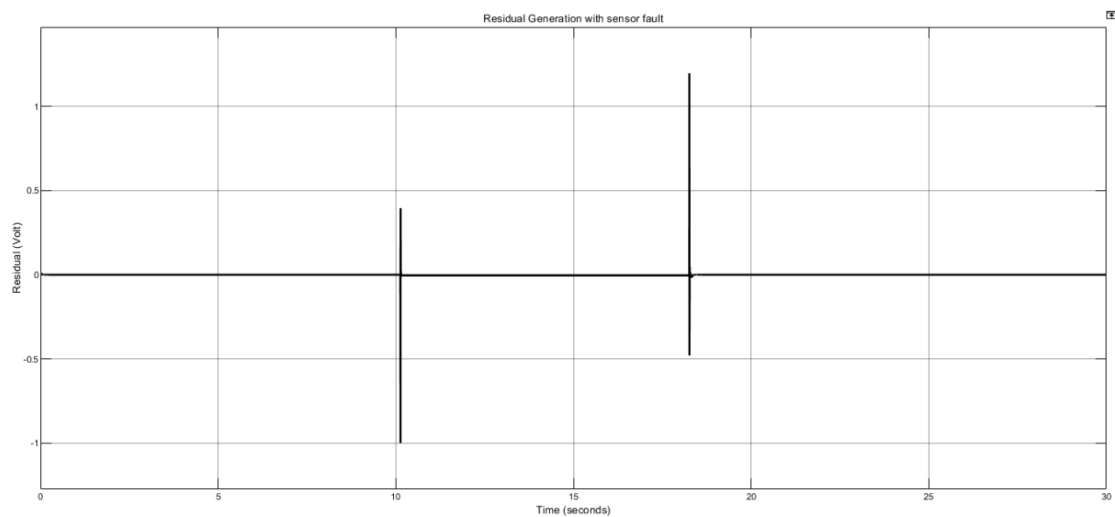


Figure (5 .24): Simulation of residual output with sensor fault.

From the figure above, it is noted that the value of the residuals falls below the threshold value at the time of the applied fault.

5.2.10 Experimental results of sensor fault for PMDC motor

A sensor fault occurs when the speed sensor is disconnected from the PMDC motor when the time is 3 seconds. Then the sensor reconnects after 1 second. This fault was implemented online (in real time), as shown in Figure (5.3). Figure (5.25) shows the responding of the measured and estimated speeds, and Figure (5.26) shows residual generation output. We can see that the fault detection was carried out successfully.

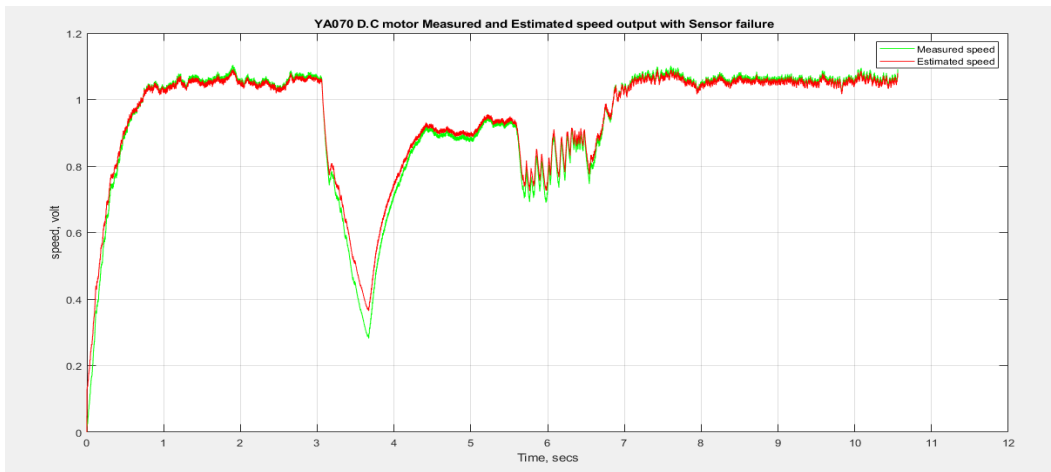


Figure (5.25): Measured and estimated speed with sensor fault

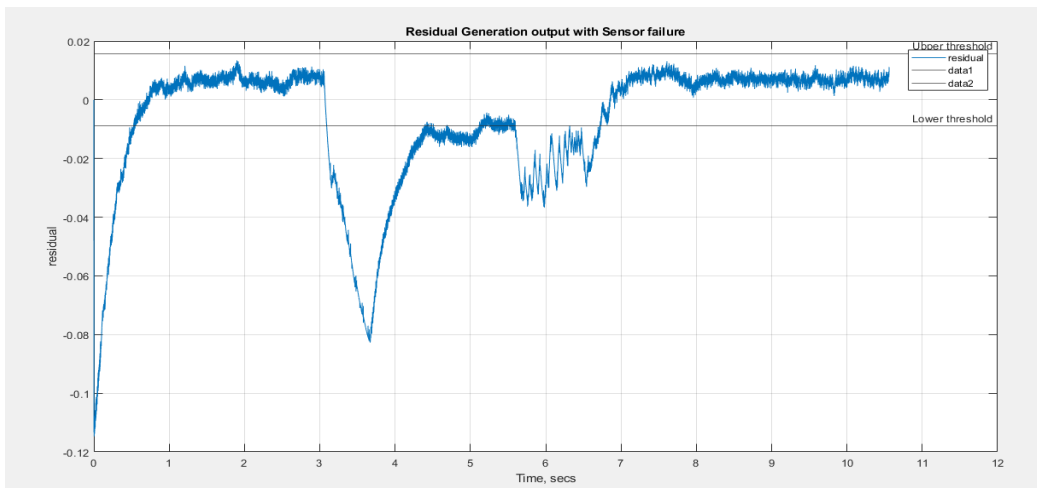


Figure (5.26): Residual output with sensor fault.

From the figure above, it is noted that the value of the residuals falls below the threshold value at the time of the applied fault. At this moment, the alarm is triggered.

5.3 Simulation and practical results for BLDC motor

Faults such as abrupt fault , an intermittent fault and incipient fault in the speed sensor of the BLDC motor have been implemented as simulations in Simulink, as illustrated in Figure (5.2). The speed sensor fault resulting from disconnecting the sensor for a specified period and then reconnecting it was implemented in real time and the results were successful. In addition, the alarm is activated when this fault occurs.

5.3.1 Simulation results without fault for BLDC motor

The simulation was performed as if there was no fault in the BLDC motor speed. The output speed and estimated speed as a simulation is shown in Figure (5.27) and Figure (5.28). Figure (5.29) shows the simulation of the residual output.

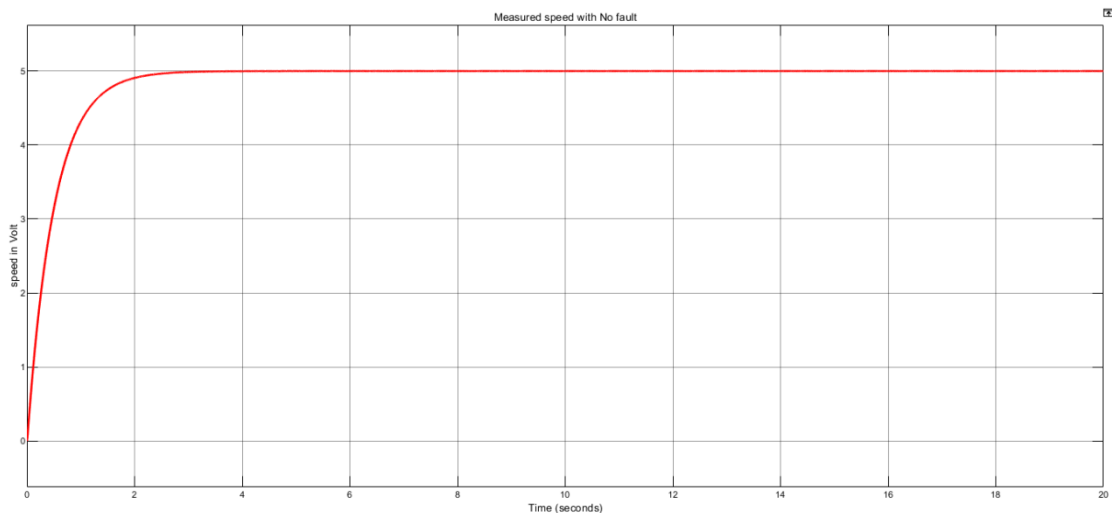


Figure (5.27): Simulation of output speed without fault.

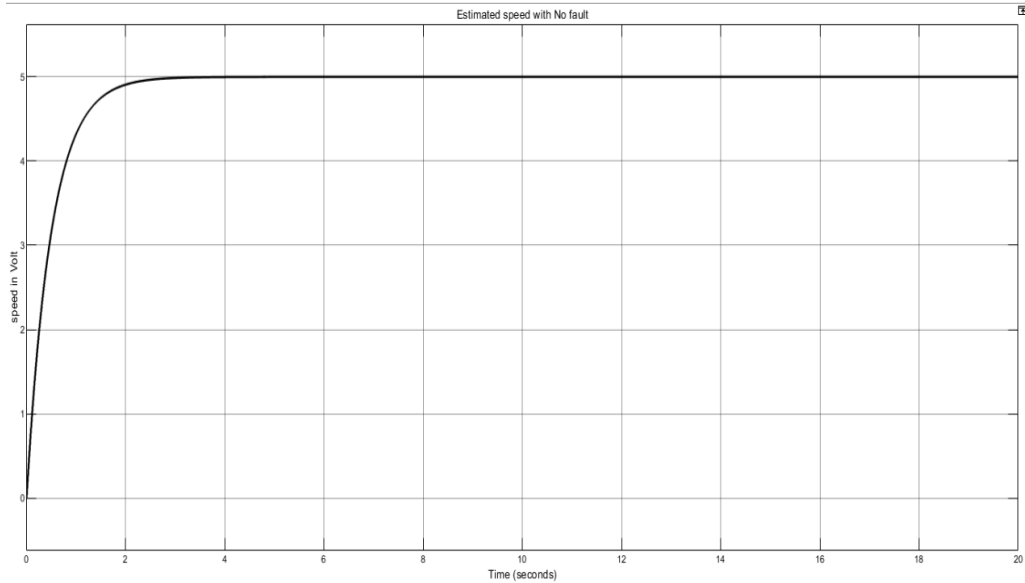


Figure (5.28): Simulation of estimated speed without fault.

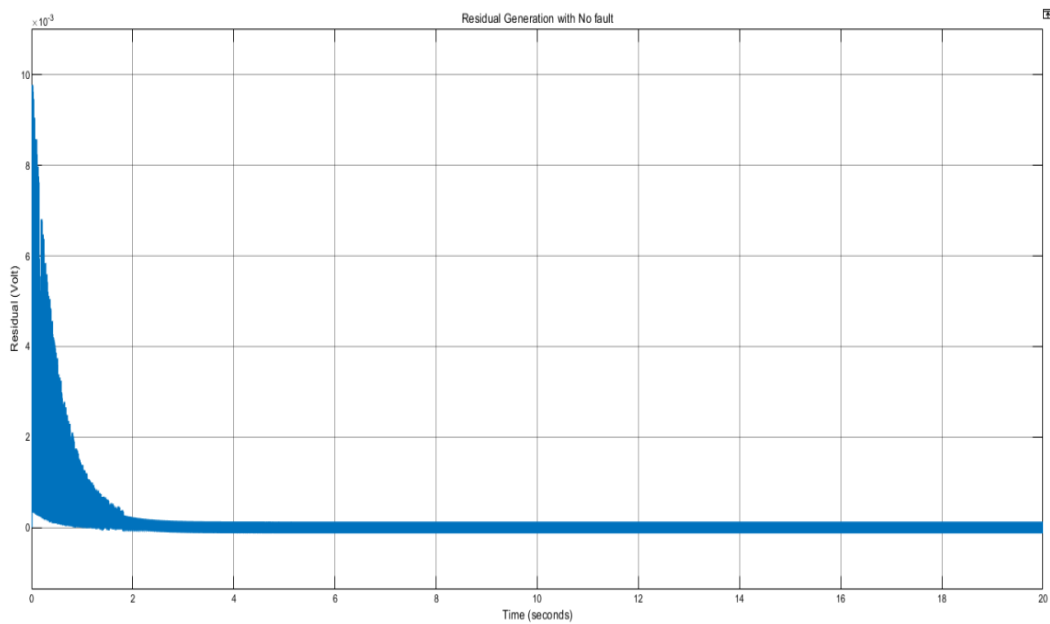


Figure (5.29): Simulation of residual output with no fault.

5.3.2 Experimental results without fault for BLDC motor

This experiment was performed with no fault in the speed sensor for BLDC motor as shown in Figure (5.4). Figures (5.30) and (5.31) shows the measured and estimated output speeds.

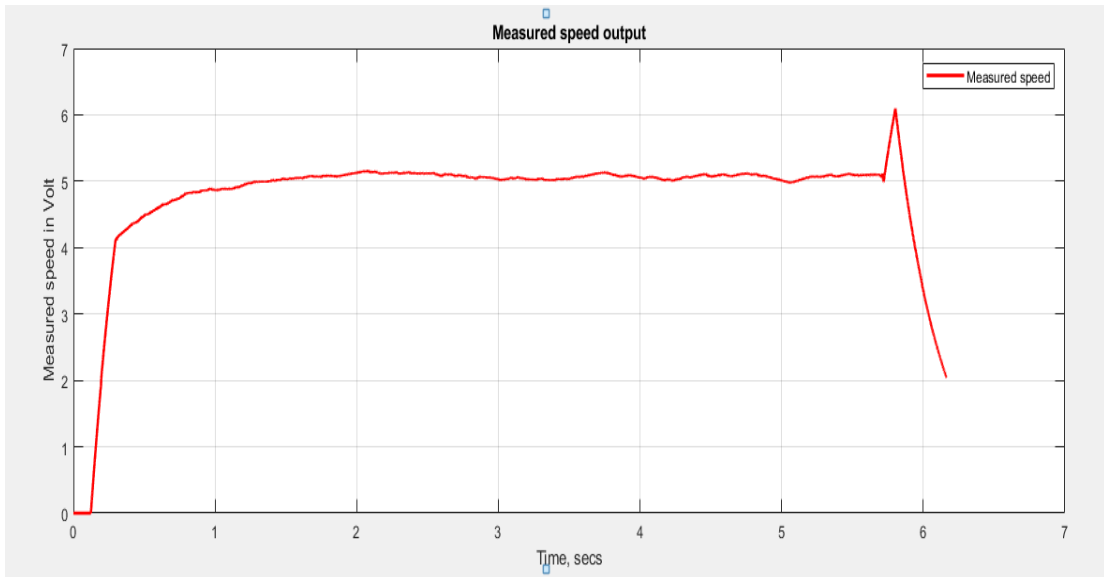


Figure (5.30): The measured speed of the BLDC motor.

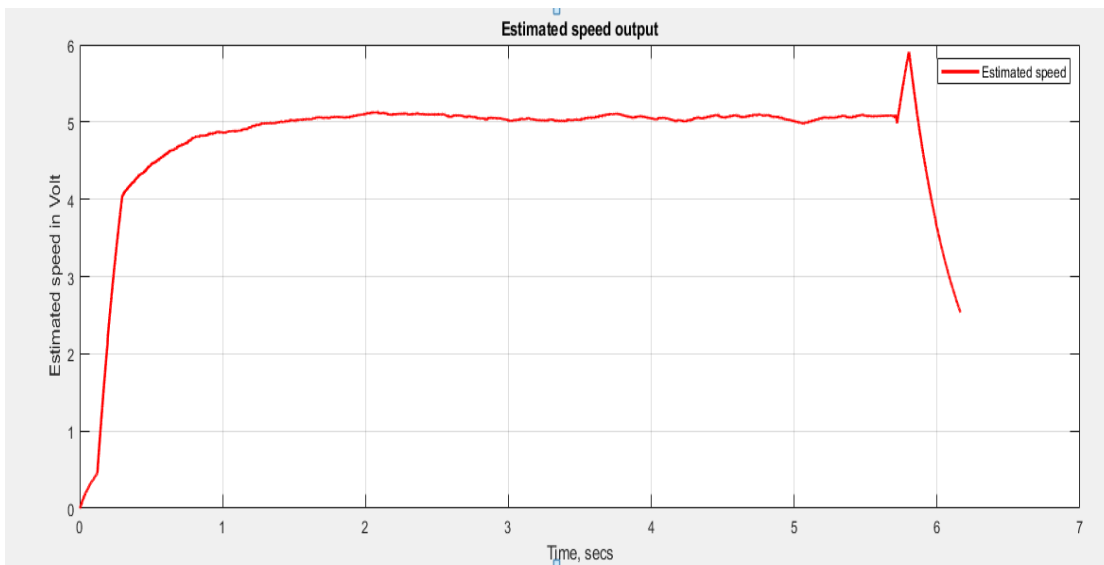


Figure (5.31): The estimated speed of the BLDC motor.

The results were quite similar to the simulation results in the absence of an fault. More than a test have been experimentally applied on the BLDC motor while it is fault free in order to determine the threshold value from the residual generation . At the first test applied the upper threshold value was 0.05 while the lower threshold value was - 0.01848 as shown in the Figure (5.32) and at the second test applied the upper threshold value was 0.0425 while the lower threshold value was - 0.01848 as shown in the Figure (5.33). It is important to know the threshold values , therefore to design the alarm when there are faults in the motor speed.

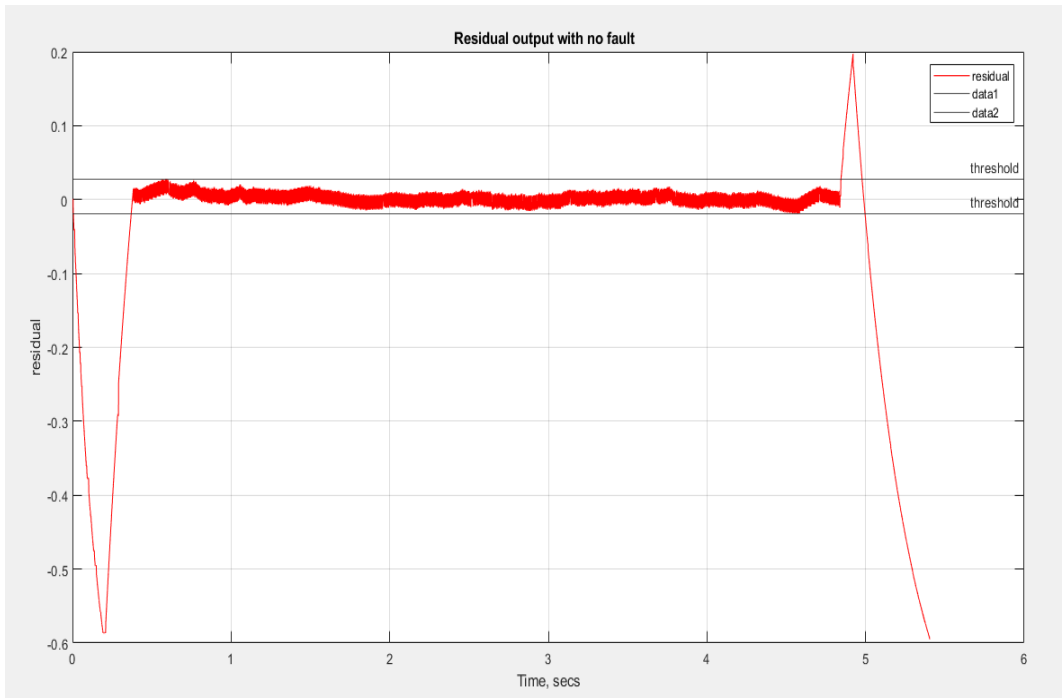


Figure (5.32): Residual output with upper threshold value is 0.05 and lower threshold value is - 0.01848.

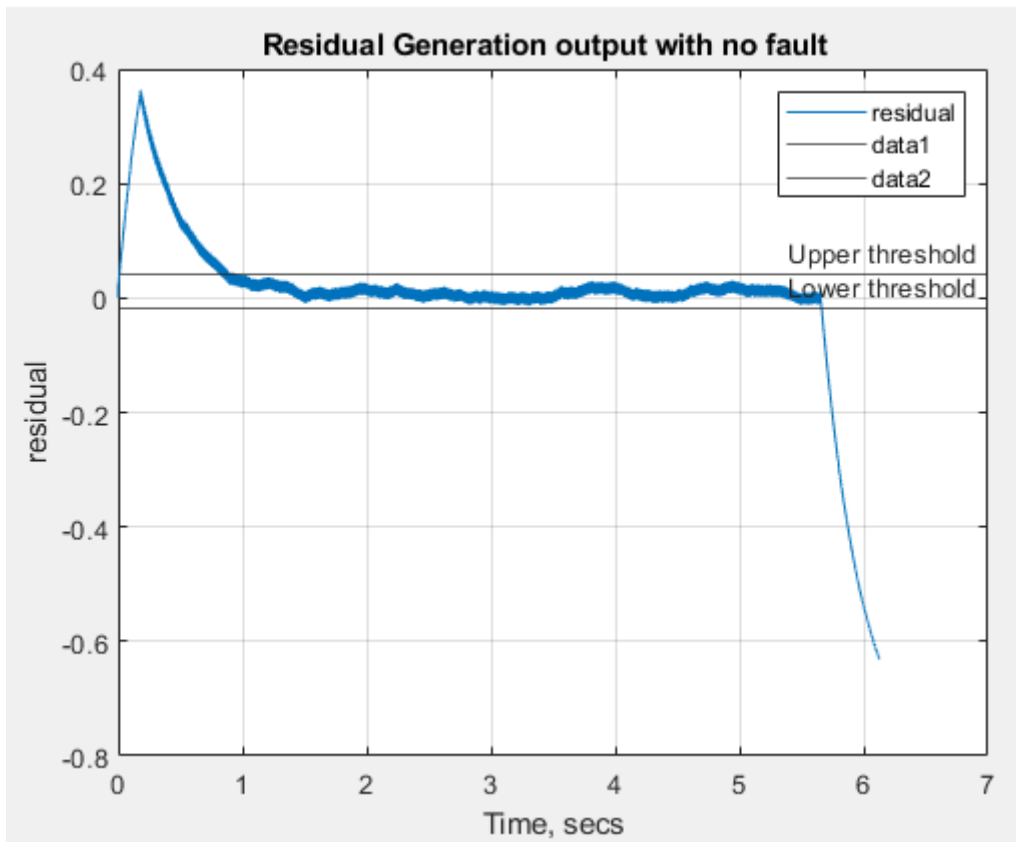


Figure (5.33): Residual output with upper threshold value is 0.0425 and lower threshold value is - 0.01848.

From the previous figure (5.33), note that the residual generation is not equal to zero due to the noise that presented in the BLDC motor and errors in the parameters of observer design. Therefore, we chose a higher threshold value of 0.05 and a lower value of - 0.01848 in order to avoid false alarms for the residual value. These threshold values were adopted for BLDC motor tests to implement the alarm.

5.3.3 Simulation results of abrupt fault for BLDC motor

The abrupt fault was introduced during this experiment as a simulation and applied to the BLDC motor speed sensor, as shown in Figure (5.2). This fault is implemented by adding a constant value to the sensor reading.. This fault didn't work in real time for BLDC motor. This fault was executed as a simulation in Simulink. An abrupt fault was implemented on the measured motor speed sensor at 8 seconds. The simulation of the output and estimated BLDC motor speeds are represented in Figure (5.34) and the simulation of the residual output is shown in Figure (5.35), we can see that the fault detection was carried out successfully at the time of its appearance.

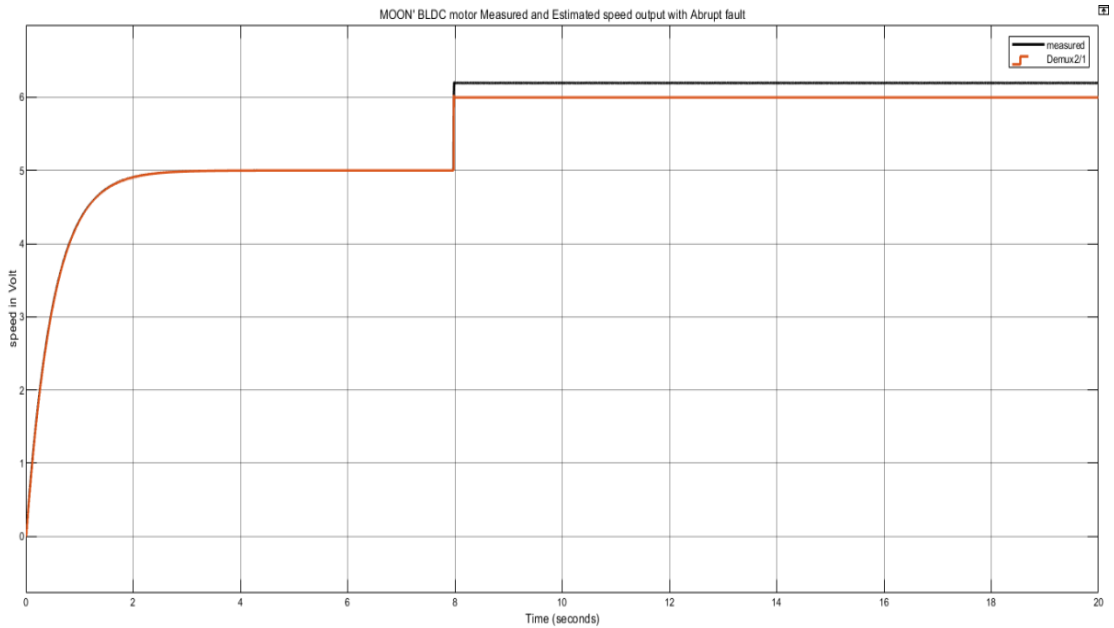


Figure (5.34): Simulation of the output and estimated speeds of the BLDC motor with Abrupt fault.

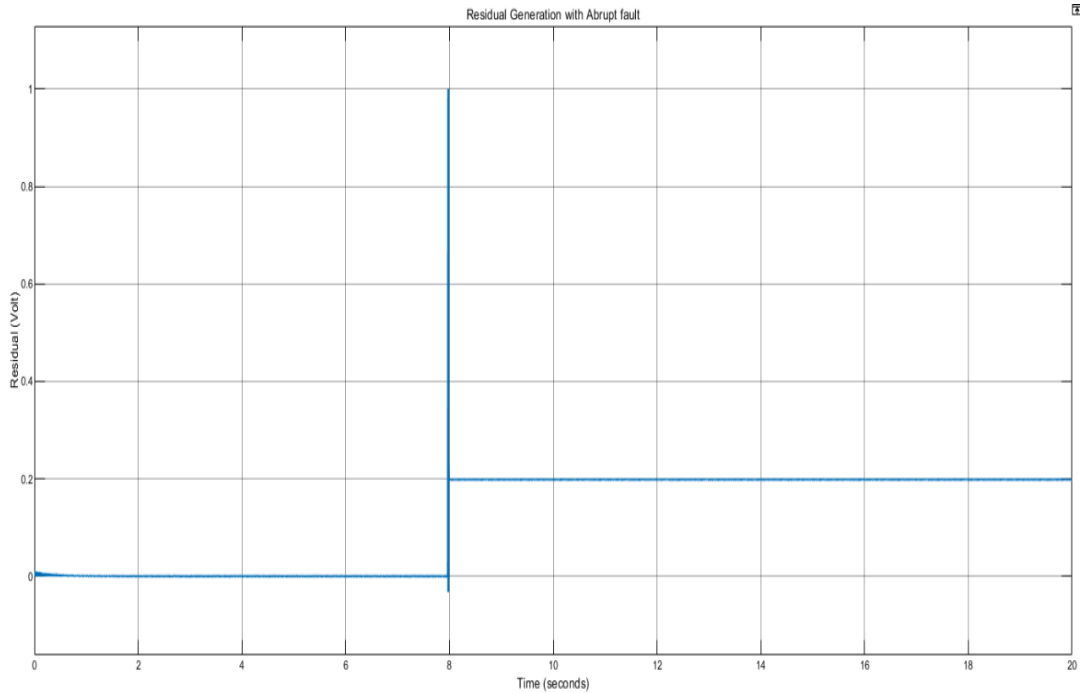


Figure (5.35): Simulation of residual output with Abrupt fault.

From the previous figure, we note that the residual value exceeds the threshold value, which indicates a fault in the speed sensor.

5.3.4 Simulation results of incipient fault for BLDC motor

The incipient fault was introduced during this experiment as a simulation and applied to the BLDC motor speed sensor, as shown in Figure (5.2). This fault does not work in real time for the BLDC motor and was executed as a simulation in Simulink. It was implemented to the output speed at 12 seconds. The simulation of the output and estimated BLDC motor speeds are represented in Figure (5.36) and the simulation of the residual output is shown in figure (5.37), we can see that the fault detection was carried out successfully at the time of its appearance .

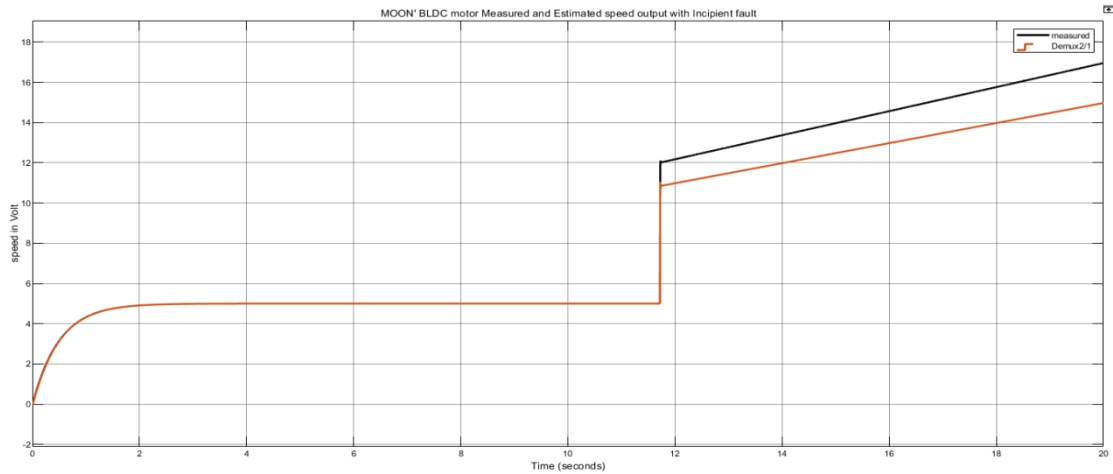


Figure (5.36): Simulation of the output and estimated speeds of the BLDC motor with incipient fault.

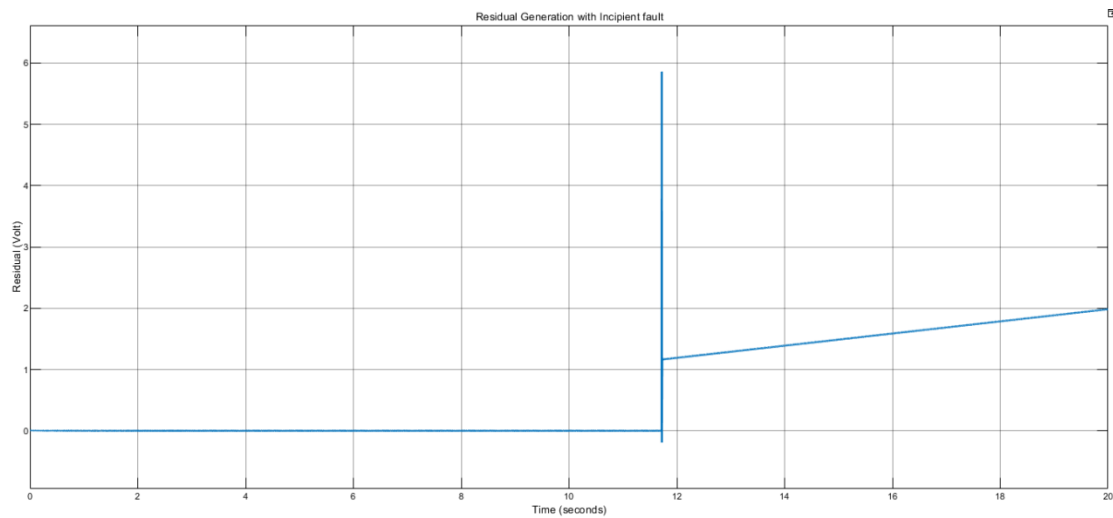


Figure (5.37): Simulation of residual output with incipient fault.

From the figure above, it can be seen that the residual value exceeds the threshold value at the start of 12 seconds, which indicates a fault in the speed sensor

5.3.5 Simulation results of intermittent fault for BLDC motor

Figure (5.2) shows that an intermittent fault was introduced during this experiment as a simulation and applied to the BLDC motor speed sensor. It is implemented by adding a constant value to the sensor reading in periodic periods. This fault didn't work in real time for BLDC motor. This fault was executed as a simulation in Simulink.

Intermittent fault was applied to the output speed after 5, 8, and 11 seconds with an amplitude of 1, 1.5, and 2, respectively, each time for 1 second. The simulation of the output and estimated BLDC motor speeds are shown in Figure (5.38) and the simulation of the residual output is shown in Figure (5.39), we see that the detection of the fault was successfully performed during its beginning of the appearance time and the end of the same time.

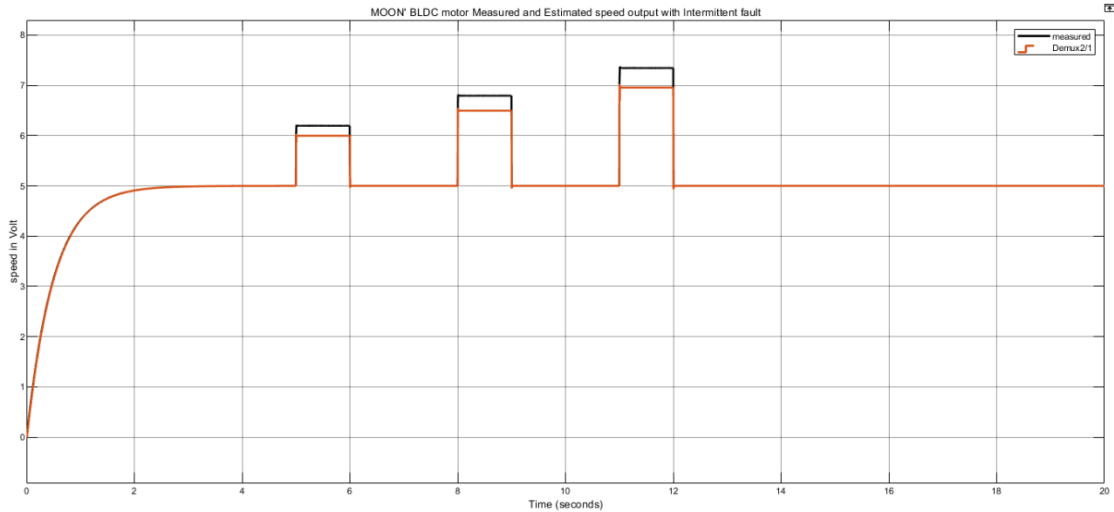


Figure (5.38): Simulation of the output and estimated speeds of the BLDC motor with intermittent fault.

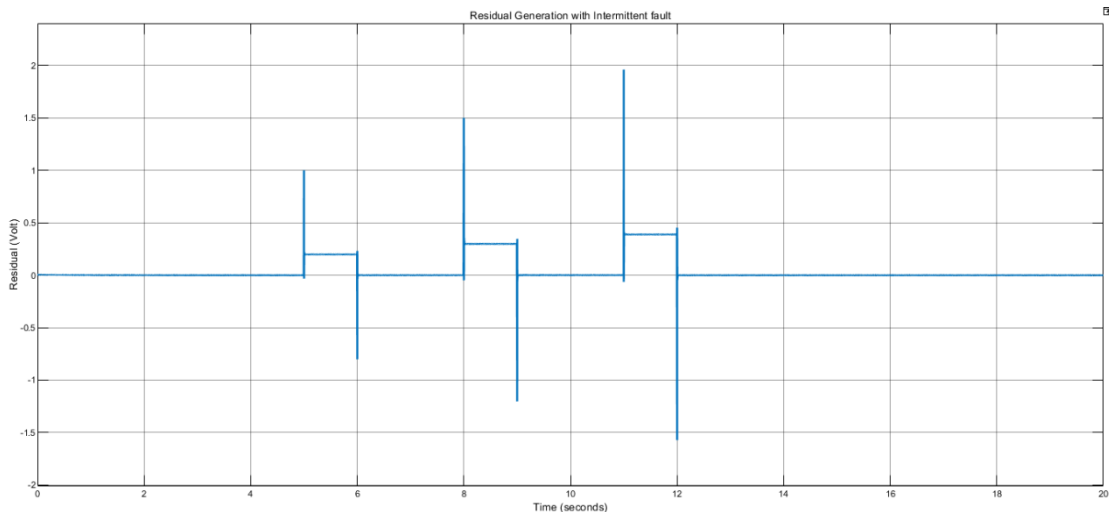


Figure (5.39): Simulation of residual output with intermittent fault.

From the previous figure, we note that the residual value exceeds the threshold value.

5.3.6 Simulation results of sensor fault for BLDC motor

The sensor fault simulation was performed in Simulink by multiplying the speed signal by zero. The output and estimated speeds as a simulation are illustrated in Figure (5.40) and Simulation of the residual output is shown in Figure (5.41)

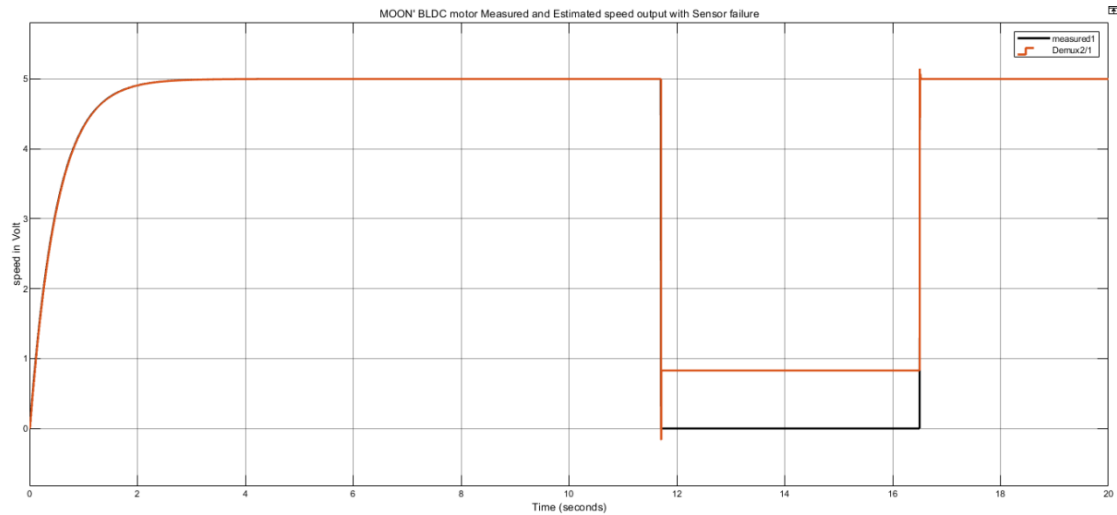


Figure (5.40): Simulation of the output and estimated speeds of BLDC motor with sensor fault.

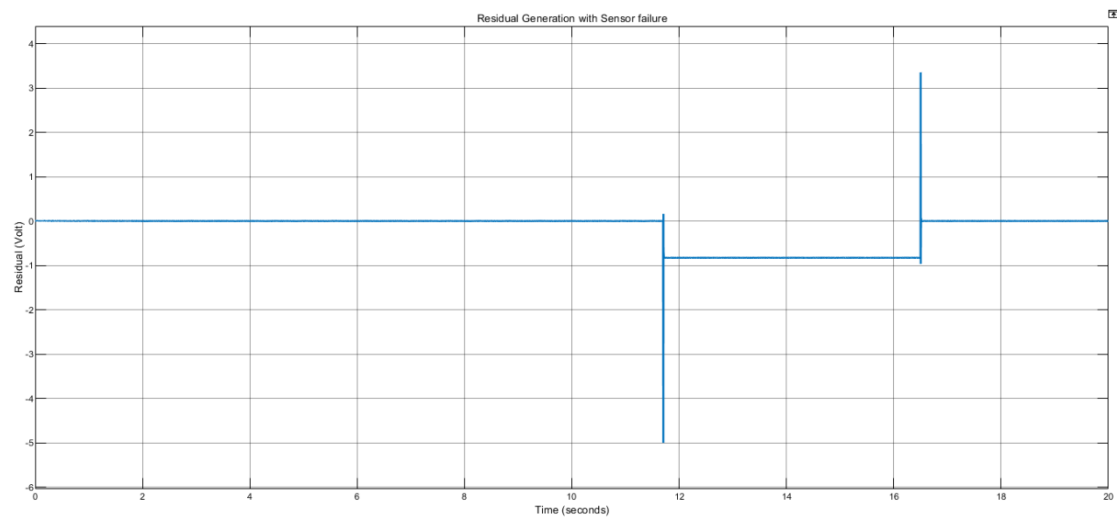


Figure (5.41): Simulation of residual output with sensor fault.

From the figure above, it is noted that the value of the residuals falls below the threshold value at the time of the applied fault.

5.3.7 Experimental results of sensor fault for BLDC motor

A sensor fault occurs when the speed sensor is disconnected from the BLDC motor when the time is two seconds. Then the sensor reconnects after one second. This fault was implemented online (in real time), as shown in Figure (5.4). Figure (5.42) shows the responding of the measured and estimated speeds and Figure (5.43) shows the residual generation output. We can see that the fault detection was carried out successfully.

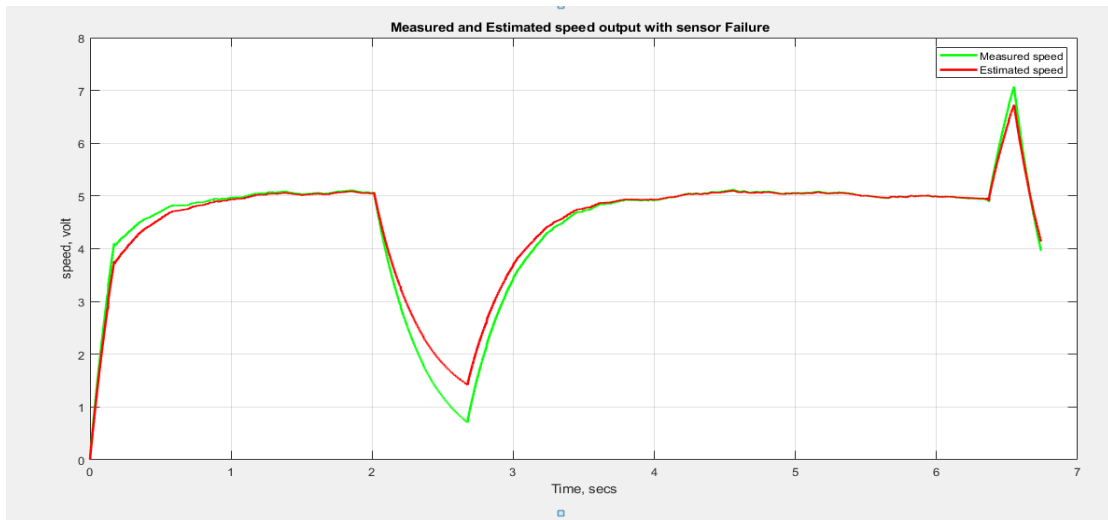


Figure (5.42): Measured and estimated speeds with sensor fault.

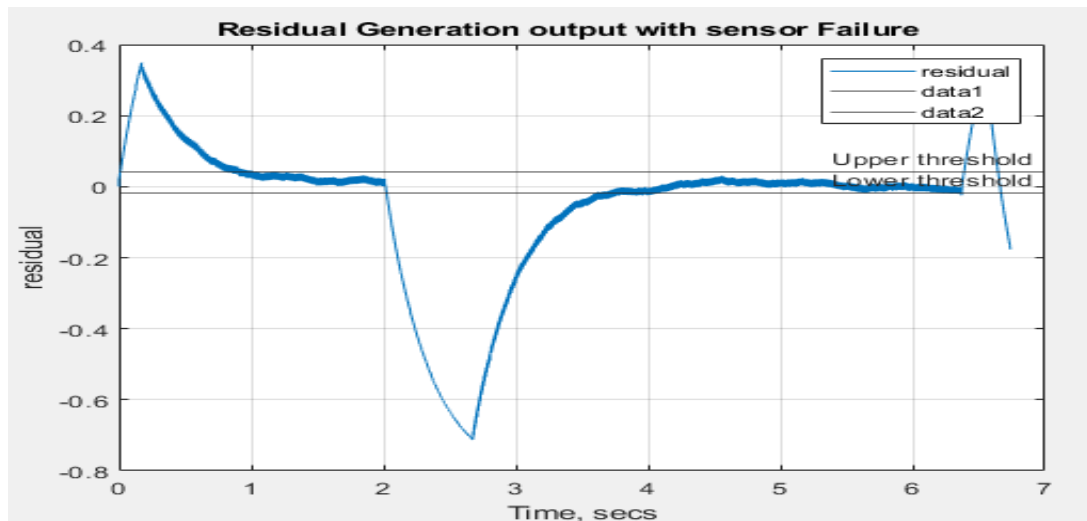


Figure (5.43): Residual output with sensor fault.

From the figure above, it can be seen that the residual value falls below the threshold value at the time of application of the fault and that will indicate a fault in the sensor. At this time, the alarm is triggered to indicate the presence of the fault.

Chapter 6

Conclusion and Future Work

6.1 General conclusion

Fault Detection and Diagnostics (FDD) methods play an effective role in improving the reliability, maintenance, and security of control systems. This thesis included the study of observer-based methods applied to the detection of faults in electrical motors. Electrical motors are one of the most commonly used devices in our time and are used in many applications. In this thesis, Simulink was used to design a controller driver system for BLDC motor speed and PMDC motor speed. The Luenberger observer method for fault detection is demonstrated for electrical motors (BLDC and PMDC motors). A sensor fault detection scheme for the BLDC motor and the PMDC motor is presented. The sensor fault was tested to prove the benefits and workability of the proposed method. The ability of residuals to detect different types of sensor failures is demonstrated, such as intermittent fault, abrupt fault and incipient fault. The method for condition monitoring used in this thesis is thresholding. A fixed threshold (simple threshold algorithms) is used for the residual signal. A buzzer device was used as an alarm for the speed sensor faults used in the research. This research has experimentally demonstrated the implementation of fault detection in electric motors (PMDC motor and BLDC motor). The Practical results showing the accuracy and benefits of the error detection method in real time.

6.2 Future work

Here, a study is conducted to detect a speed sensor fault on a BLDC motor using Luenberger observer. Another topic that could be a future work is an fault in one of the Hall sensors signal (H_1, H_2, H_3) for BLDC motors. At the end of the study, a fault in the H_1 signal was made through a normally closed switch in the hall sensor circuit which leaves a pace for any future work to continue the work on H1 and make new study on H2 and H3.

Accordingly, any researcher can complete the subject of detection faults signal Hall effect sensor for BLDC motors(one faulty hall effect sensor).Although the importance of the method used in this thesis to detect faults in the motor speed sensor, much research remains to be done in the future. These research topics can also be studied in the future, as follows:

- A. Another method of calculating the residual generation can also be obtained from the Kalman filter.
- B. Adaptive threshold can be used for residual signal.

References

- Abed, W. (2015). Robust Fault Analysis for Permanent Magnet DC Motor in Safety Critical Applications.
- Aghaee, M., & Jalali, A. A. (2018, May). BLDC Motor Speed Control Based on MPC Sliding Mode Multi-Loop Control Strategy–Implementation on Matlab and Arduino Software. *In Electrical Engineering (ICEE), Iranian Conference on* (pp. 795-800). *IEEE*.
- Agrawal, K., Gandhi, A., Shah, M. T., & Gojiya, M. V. . (2016, December). Design, analysis and realization of SVPWM using embedded code generation technique for a three phase, two level inverter. *In 2016 International Conference on Electrical Power and Energy Systems (ICEPES)* (pp. 377-382). *IEEE*.
- ALKAYA, A. (2012). Novel data driven-based fault detection for electromechanical and process control systems. (*Doctoral dissertation, Adana, Turkey: Cukurova University, 2012: 38-43*).
- Alkaya, A., & Eker, I. (2014). Luenberger observer-based sensor fault detection: online application to DC motor. *Turkish Journal of Electrical Engineering & Computer Sciences*, 22(2), 363-370.
- Allous, M., & Zanzouri, N. (2017, January). Fault tolerant control of system with simultaneous actuator and sensor faults. *In 2017 International Conference on Control, Automation and Diagnosis (ICCAD)* (pp. 230-235). *IEEE*.
- Aqeel, A. (2018, June 30). <http://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html>. Retrieved from Theen Gineering projects (TGP).
- Balogh, L. (2001, May). Design and application guide for high speed MOSFET gate drive circuits. *In Power Supply Design Seminar SEM-1400, Topic (Vol. 2). brushed-vs-brushless-dc-motors*. (n.d.). Retrieved from <https://microlinearactuator.com>.
- Bulán, M. (2014). Simulation and Hardware Modeling of a Real BLDC Motor Applicable in Medicine.
- Chan, H. L., & Woo, K. T. . (2015). Closed loop speed control of miniature brushless DC motors. *Journal of Automation and Control Engineering*, 3(4).
- Copt, F., Araujo, D. M., Koechli, C., & Perriard, Y. (2017, October). Current control strategy for dynamic winding reconfiguration of a brushless dc motor. *In 2017 IEEE Energy Conversion Congress and Exposition (ECCE)* (pp. 3577-3582). *IEEE*.
- Creative Robotics Ltd. (n.d.). <http://www.creative-robotics.com/quadrature-intro>. Retrieved from Creative Robotics Ltd.
- Dejan. (2017). <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>. Retrieved from How To Mechatronics.
- Dobra, P., Dobra, M., Moga, D., Sita, V. I., Dobra, P., Trusca, M., ... & Munteanu, R. A. . (2014, May). Model based fault detection for electrical drives with BLDC motor. *In 2014 IEEE International Conference on Automation, Quality and Testing, Robotics* (pp. 1-5). *IEEE*.

- Eissa, M. A., Ahmed, M. S., Darwish, R. R., & Bassiuny, A. M. (2015, December). Model-based sensor fault detection to brushless DC motor using Luenberger observer. *In 2015 7th International Conference on Modelling, Identification and Control (ICMIC) (pp. 1-6). IEEE.*
- Eissa, M. A., Ahmed, M. S., Darwish, R. R., & Bassiuny, A. M. (2015, December). Improved fuzzy Luenberger observer-based fault detection for BLDC motor. *In 2015 Tenth International Conference on Computer Engineering & Systems (ICCES) (pp. 167-174). IEEE.*
- Eissa, M. A., Ahmed, M. S., Darwish, R. R., & Bassiuny, A. M. (2015, December). Unknown Inputs PI observer-based sensor fault detection technique for BLDC motor. *In 2015 7th International Conference on Modelling, Identification and Control (ICMIC) (pp. 1-4). IEEE.*
- Farmen, T., & Zarchi, M. . (2018). Fault Diagnosis of an Electrical Winch based on Structural Analysis . (Master's thesis, Universitetet i Agder; University of Agder).
- Fazal, Q., Liaquat, M., & Naz, N. (2015, December). Robust fault tolerant control of a DC motor in the presence of actuator faults. *n 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (pp. 301-333). IEEE.*
- Franceschet, A. B., Scortegagna, R. G., Januário, M., & Hoffmann, K. (2017). Three-phase frequency inverter with SPWM modulation: Simulation and implementation with TMS320F28335. *Scientific Initiation Seminar, Integrated Teaching, Research and Extension Seminar and University Exhibition.*
- Frank, P. M., Ding, S. X., & Marcu, T. (2000). Model-based fault diagnosis in technical processes. *Transactions of the Institute of Measurement and Control, 22(1), 57-101.*
- Gaeid, K. S., & Ping, H. W. (2011). Fault tolerant control of induction motor. *Modern Applied Science, 5(4), p. 83.*
- Gaeid, K. S., & Ping, H. W. (2011). Wavelet fault diagnosis and tolerant of induction motor: A review. *International Journal of the Physical Sciences, 6(3), pp. 358-376.*
- Gaeid, K. S., & Ping, H. W. (2011). Wavelet fault diagnosis and tolerant of induction motor: A review. *International Journal of the Physical Sciences, 6(3), 358-376.*
- Hassan, A. A., Al-Shamaa, N. K., & Abdalla, K. K. (2017). Comparative Study of Conventional and Optimal PID Tuned Methods for PMDCM Speed Control. *International Journal of Applied Engineering Research, 12(24), 15999-16007.* <https://www.moonsindustries.eu/index.php/brushless-dc-motors/42bl-series.html>. (n.d.). Retrieved from MOONS' Industries.
- <https://www.yoycart.com/Product/26695000175/>. (n.d.). Retrieved from Yoycart.
- Isermann, R. (2005). Model-based fault-detection and diagnosis—status and applications. *Annual Reviews in control, 29(1), 71-85.*
- Isermann, R. (2006). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance.* Springer Science & Business Media.
- Isermann, R., & Balle, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control engineering practice, 5(5), pp. 709-719.*

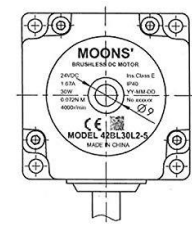
- Johansson, M. (2017). Evaluation of Sensor Solutions & Motor Speed Control Methods for BLDCM/PMSM in Aerospace Applications.
- Kanev, S. K. (2004). *Robust fault-tolerant control (pp. 0842-0842)*. FEBO-DRUK.
- Kara, S. M. (2017). Brushless DC motor control under varying load conditions. (*Master's thesis, Izmir Institute of Technology*).
- Lee, T. Y., Seo, M. K., Kim, Y. J., & Jung, S. Y. (2015). Design and Torque Ripple Analysis of Brush-less DC Motor According to Delta Winding Connection. *Journal of Magnetism*, 20(2), 166-175.
- Liu, L. (2006). Robust fault detection and diagnosis for permanent magnet synchronous motors.
- Mahbub, S. T. (2013, January 20). <http://tahmidmc.blogspot.com/2013/01/using-high-low-side-driver-ir2110-with.html>. Retrieved from Tahmid's blog.
- Megalingam, R. K.; Vadivel, S. R. R.; Pula, B. T.; Sathi, S. R.; & Gupta, U. S. C. (2019, April). Motor Control Design for Position Measurement and Speed Control. In *2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0405-0409)*. IEEE.
- Miljković, D. (2011, May). Fault detection methods: A literature survey. In *2011 Proceedings of the 34th international convention MIPRO (pp. 750-755)*. IEEE.
- Monteiro, D. (2015). Fault detection and isolation for linear dynamical systems.
- Mukkar, M. M. (2017, November 29). <http://www.thisisthi.com/index.php/the-difference-between-ac-and-dc-electric-motors-mohnish-mukkar/>. Retrieved from Thisisthi.
- Neto, G. F. (2017, August 9). <http://microlinearactuator.com/brushed-vs-brushless-dc-motors/>. Retrieved from Morai Motion.
- Nise, N. S. (2007). *CONTROL SYSTEMS ENGINEERING, (With CD)*. John Wiley & Sons.
- Nyberg, M., & Nyberg, C. M. (1999). Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications. *PhDthesis*.
- Oguntoyinbo, O. . (2009). PID control of brushless DC motor and robot trajectory planning simulation with MATLAB®/SIMULINK®.
- Pal, D. (2016). Observers design and implementation for DC motor. *Int. J. Sci. Res. IJSR*, 3.
- Poovizhi, M., Kumaran, M. S., Ragul, P., Priyadarshini, L. I., & Logambal, R. (2017, March). Investigation of mathematical modelling of brushless dc motor (BLDC) drives by using MATLAB-SIMULINK. In *2017 International Conference on Power and Embedded Drive Control (ICPEDC) (pp. 178-183)*. IEEE.
- Ridwan, M., & Yuniarto, M. N. (2016, July). Electrical equivalent circuit based modeling and analysis of brushless direct current (BLDC) motor. In *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA) (pp. 471-478)*. IEEE.
- Sadun, A. S., Jalani, J., Sukor, J. A., & Pahat, B. (2015). A comparative study on the position control method of dc servo motor with position feedback by using arduino.
- Sakunthala, S., Kiranmayi, R., & Mandadi, P. N. (2017, August). A study on industrial motor drives: Comparison and applications of PMSM and BLDC

- motor drives. . In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 537-540). *IEEE*.
- Saoudi, G., El Harabi, R., & Abdelkrim, M. N. (2013, March). Graphical linear observers for fault detection: The dc motor case study. In *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)* (pp. 1-8). *IEEE*.
- Sellami, L. (2014). Simulink model of a full state observer for a DC motor position, speed, and current. In *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV)* (p. 1). *The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*.
- Silmon, J. A. (2009). Operational industrial fault detection and diagnosis: railway actuator case studies. (*Doctoral dissertation, University of Birmingham*).
- Skóra, M. (2017, June). Operation of PM BLDC motor drives with faulty rotor position sensor. In *2017 International Symposium on Electrical Machines (SME)* (pp. 1-6). *IEEE*.
- Sova, V., Chalupa, J., & Grepl, R. (2015, September). Fault tolerant BLDC motor control for Hall sensors failure. In *2015 21st International Conference on Automation and Computing (ICAC)* (pp. 1-6). *IEEE*.
- Tang, W. J., Liu, Z. T., & Wang, Q. (2017, July). Dc motor speed control based on system identification and pid auto tuning. In *2017 36th Chinese Control Conference (CCC)* (pp. 6420-6423). *IEEE*.
- Tibor, B., Fedak, V., & Durovský, F. . (2011, June). Modeling and simulation of the BLDC motor in MATLAB GUI. In *2011 IEEE International Symposium on Industrial Electronics* (pp. 1403-1407). *IEEE*.
- www.k.com. (n.d.).
- Xia, C. L. (2012). *ermanent magnet brushless DC motor drives and controls*. John Wiley & Sons.
- Zandi, O., & Poshtan, J. (2018, May). Brushless DC motor bearing fault detection using Hall effect sensors and a two-stage wavelet transform. In *Electrical Engineering (ICEE), Iranian Conference on* (pp. 827-833). *IEEE*.
- Zhang, Q., & Feng, M. (2017, August). A new fault diagnosis method for hall signals in brushless DC motor drives. In *2017 20th International Conference on Electrical Machines and Systems (ICEMS)* (pp. 1-4). *IEEE*.
- Zhang, Y. (2006). *FP9-1: Fault Tolerant Control Systems. Course PPT, 1*.

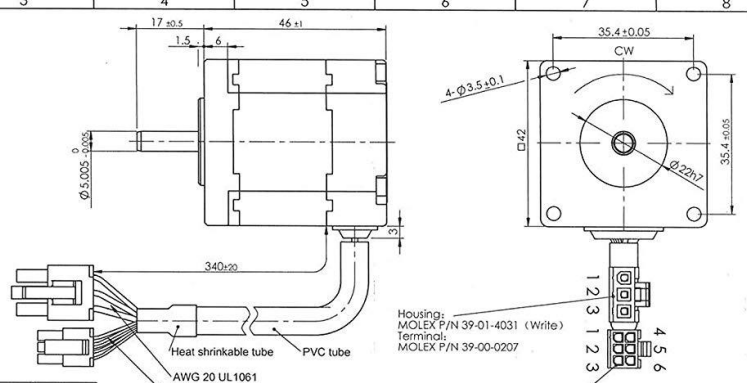
Appendices

Appendix A

A.1 Datasheet of Moons' 42BL30L2-5 BLDC motor.



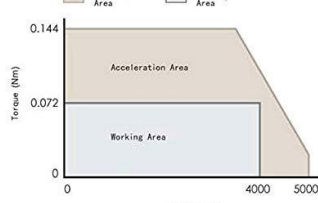
461113000067



Housing: MOLEX P/N 39-01-4031 (White)
Terminal: MOLEX P/N 39-00-0207

Housing: MOLEX P/N 43025-0600 (Black)
Terminal: MOLEX P/N 43030-0005

规格	Specifications		
1	额定电压/Rated power supply	24	VDC
2	额定输出功率/Rated output power	30	W
3	极数/Poles number	6	
4	相数/Phase number	3	
5	额定转速/Rated rotational speed	4000	RPM
6	最大转速/Maximum speed	5000	RPM
7	额定转矩/Rated torque	0.072	N·m
8	额定电流/Rated winding current	1.67	A
9	线反电势系数/Voltage constant±10%	2.95	Vrms/krpm
10	惯量/Moment of inertia	0.0388x10 ⁻⁴	kg·m ²
11	线电阻Ph/ph resistance±10% at 20°C	1.31	Ω
12	线电感Ph/ph inductance±30% at 1KHz 20°C	1.10	mH
13	电机质量/Motor mass	0.32	kg
14	绝缘等级/Insulation class	E	120°C
15	防护等级/Protective enclosure rating	IP40	



编号	线色	定义
1	红/red	VCC(+5V)
3	黄/yellow	Hv
4	蓝/blue	Hw
5	黑/black	GND
6	橙/orange	Hu

编号	线色	定义
1	蓝/blue	W
2	粉/orange	U
3	黄/yellow	V

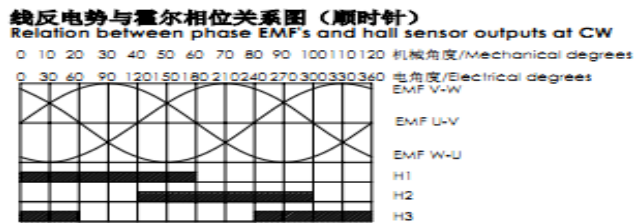
Unit: mm 第一角画法 FIRST ANGLE METHOD	批准 Approve 标准化 Standard	工艺 Technology	审核 Check	设计 Design
未注线性与角度尺寸公差 Tolerances for linear and angular dimensions without individual tolerance indicators GB/T 1804-m eqv ISO 2768-1:m	UNLESS OTHERWISE SPECIFIED	NAME	SIGNATURE	DATE

MOONS' Shanghai Moons' Electric Co., Ltd.

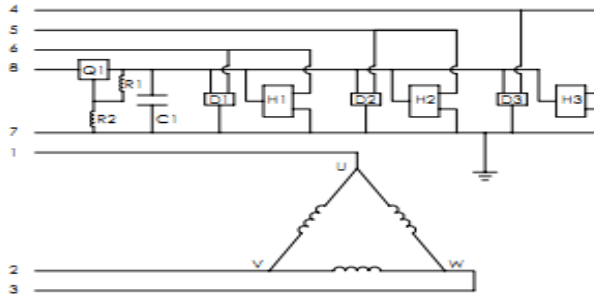
BRUSHLESS DC MOTOR

This document is released for information only. Its reproduction in whole or in part or its dissemination to third parties is prohibited. All trademarks and registered trademarks appearing on this document are the property of their respective owners.

Type: 42BL30L2-5 Rev. A SCALE 1:1 Sheet 1 of 1



电路图
Electric diagram



A.2 Datasheet of YA-070/ YA-071 PMDC motor.

Specifications:

- DC SERVO MOTOR YA-070/YA-071
- Maximum Supply Voltage : 24 VDC
- Current: 200 mA
- Speed : 3000 RPM
- 200 PULSES / REVOLUTION
- Encoder Model= H9700
- WIREES : red + black= motor power
- Green= SIG A
- white= SIG B
- diameter of the output shaft: 6mm
- length of Output shaft: 28.5mm
- Body diameter: 61mm
- Total length: 103mm



Figure (A.1): YA-070/071 PMDC motor.

A.3 Motor parameters conversion table.

Motor parameters conversion table

COLUMN A	COLUMN B	COLUMN C	COLUMN D	COLUMN E	COLUMN F	COLUMN G
Motor: X (Wye-wound armature)	Units: X Commutation six-step:	Column B to Column D conversions:	Units: X Commutation six-step:	Column D to Column F conversions:	Units: X Commutation sine wave:	Drive units required for sine commutation with wye-wound armature:
Tc	12.7_Nm	=	12.7_Nm	=	12.7_Nm	Tc (Nm)
Tp	41.2_Nm	=	41.2_Nm	=	41.2_Nm	Tp (Nm)
Ic (current/ Φ_1)	13_A/ Φ_1	=	13_A ₁	$\div \sqrt{1.5} =$	10.61_a-rms ₁	Ic (a-rms)/ phase(Φ) _{1,67}
Ip (current/ Φ_1)	53.3_A/ Φ_1	=	53.3_A ₁	$\div \sqrt{1.5} =$	43.52_a-rms	Ip (a-rms)/ phase(Φ) _{1,67}
Kt	1.00 Nm/A	=	1.00 Nm/A	$\cdot \sqrt{1.5} =$	1.224745 Nm/a-rms	Kt (Nm/a-rms)
Ke or Kb	0.57735 V(Φ)/rad/sec	$\cdot \sqrt{3} =$	1.00 V(L-L)/ rad/sec	$\cdot 1,000 \div 9.55$ $\div \sqrt{2} =$	74.05 Vrms(L-L)/ Krpm	Kb (Vrms/Krpm)
OR	0.57735 V(Φ)/rad/sec	$\cdot \sqrt{3} \cdot 1,000$ $\div 9.55 =$	104.72 V(L-L)/ Krpm	$\div \sqrt{2} =$	74.05 Vrms(L-L)/ Krpm	Kb (Vrms/Krpm)
Rm (ohms) ₁ (line-to-line) at 25°C	0.540_Ω/ Φ_3	$\cdot 2 =$	1.08_Ω (line to line)	=	1.08_Ω (line to line)	Rm (Ω: line to line) at 25°C
Lm ₂ (mH)	4.25_mH/ Φ_2	$\cdot 2 =$	8.5_mH(L-L)/	=	8.5_mH(L-L)/	L or Lm (mH: line to line)
Jm (inertia)	0.00152 kg-m ²	$\cdot 100^2 =$	15.2 kg-cm ²	=	15.2 kg-cm ²	Jm (kg-cm ²)
Motor poles	10_Poles	=	10_Poles	$\div 2 =$	5_Pole-pair	Pole-pair (PP)
Thermal resistance	0.467°C/W	=	0.467°C/W	=	0.467°C/W	
Winding temperature (maximum)	155°C	=	155°C	=	155°C	
Watts (loss) at 25°C ambient	$\{(155-25) \div 0.467\}w =$	278 =	$\{(155-25) \div 0.467\}w =$	278 =	$\{(130^\circ\text{C}_\text{rise}) \div 0.467\}w$	Based on data- sheet information.
Equality: Watts (loss) by power calculations						
Power (six-step) _{5,6,6} $= VI \cos\theta$ $= 2 \cdot I_\Phi^2 \cdot Rm_\Phi(\text{hot})_4$ $= I_\text{line}^2 \cdot Rm_L_L(\text{hot})_4$	2-132(0.54-1.525) =	278 =	$13^2 \cdot (1.08 \cdot 1.525) =$	278 =	Not applicable	Checks out based on given or converted data.
Power (sine wave) _{5,6,6} $= 3 \cdot V_\Phi^2 \cdot I_\Phi \cdot \cos\theta$ $= 3 \cdot I_\Phi^2 \cdot Rm_\Phi(\text{hot})_4$ $= 3 \cdot I_\Phi^2 \cdot Rm_L_L(\text{hot})_4 \div 2$	Not applicable	Not applicable	Not applicable	278 =	$3 \cdot 10.61^2 \cdot (1.08 \cdot 1.525) \div 2$	Checks out based on given or converted data.

Provided by Kollmorgen — 1.1.2013

This table converts common wye-wound motor parameters into units that can be entered into drives for sinusoidal operation.

1. Locate the required drive units in Column G.
2. For any motor parameter presented in the units of Column B, use the conversion factors of Column C to

get the units of Column D. Then use conversion factors of Column E to get the units of Column F, to match the specific units for the drive in Column G.

3. For any motor parameter presented in the units of Column D, use the conversion factors of Column E to get the units of Column F and match the specific units for the drive in Column G.

Appendix B

MALTLAB Codes

B.1 MATLAB M.file codes for BLDC motor.

```
% Clear MATLAB memory and close all figures
clear all; close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MOONS' BLDC motor parameters used in the modeling
Ra = 1.34;           % Ohms, Terminal Resistance phase
La = 0.00115;      % Henrys, Terminal Inductance phase
Kt = 0.043;       % Nm/Arms, Torque constant
Kb = 0.0281;      % Vrms/rad/sec, Voltage constant ke
tm = 0.0129;      % seconds, s, Mechanical Time constant
J = 0.0388e-4;    % kg.m^2, Rotor inertia,
ph= 3;            % Number of phases
te =2.8606e-4;    % seconds, s, Electrical Time constant
b=1.718e-4;       % Nm/rad/sec %viscous friction coefficient
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
G=tf([1/Kb],[tm*te tm 1])

% %%%%%%%%%Plots the Step Response diagram%%%%%%%%
figure;
step(G, 10);
title('Open Loop Step Response diagram');
xlabel('Time, secs')
ylabel('Voltage, volts')
grid on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%State space model of the Moons' BLDC motor%%%%%%%%
%%per phase modeling %%%%%%%%%
A=[-b/J Kt/J; -Kb/La -Ra/La];
B=[0; 1/La];
C=[1 0];
D=0;
motor_ss=ss(A,B,C,D)
tfff=tf(motor_ss)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%check controllability and Observability%%%%%%%%
%% controllability%%%%%%%%

p=ctrb(A,B);
if rank(A)==rank(p)
    disp('system is controllable')
else
    disp('system is not controllable')
end
%% observability%%%%%%%%
q=obsv(A,C);
if rank(A)==rank(q)
    disp('system is Observable')
else
    disp('system is not Observable')
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%observer Design %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%overshoot<0.5%,ts=0.0126sec,z=0.7

pos=0.5;
ts=0.0126;
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2))
wn=4/(z*ts)
r=roots([1,2*z*wn,wn^2]) %desired poles
% poles=10*r
% L_T=acker(A.',C.',poles)% state feedback observer ain matrix
%L=L_T'

k=place(A,B,r)
%%%check for closed loop eigenvalues%%
Acl=A-B*k;
Ecl=eig(Acl)
%%%creat closed loop system
syscl=ss(Acl,B,C,0);
step(syscl)
kdc=dcgain(syscl);
kr=1/kdc;
syscl_scaled=ss(Acl,B*kr,C,D);
step(syscl_scaled);
% %%% check poles of estimtor -error dynamic
obAcl=A-L.*C;
est_poles=eig(A-L.*C)

```

B.2: MATLAB M.file codes for PMDC motor.

```

% clear matlab memory and close all figures
clear all; close all;

% Define YA-070/ YA-071 PMDC motor parameters%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=0:0.01:20;
Ra=7; %Resistance
La=0.008436; % Inductance H
Kt=0.094; % Nm/Arms, Torque constant
Kb=0.094; % Vrms/rad/sec, Voltage constant ke
J=2.2097e-04; kg.m^2, Rotor inertia,
b=1.65e-04; %Nm/rad/sec %%%viscous friction coefficient
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s

% Define motor state variable model%%
A=[-b/J Kt/J; -Kb/La -Ra/La];
B=[0; 1/La];
C=[1 0];
D=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
motor_ss=ss(A,B,C,D)
tfff=tf(motor_ss)

% Plots the Step Response of state space diagram
% figure;
step(tfff, 10)
title('Open Loop Step Response state space diagram');

```

```

xlabel('Time, secs')
ylabel('Voltage, volts')
grid on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s
num =[Kt];
den = [(La*J)*s^2 + ((Ra*J)+(La*b))*s + ((Ra*b)+(Kt*Kb))] ;
plant = tf(num,sym2poly(den))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%stability%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
poles=eig(A)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%check controllability and observability
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%controllability%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p=ctrb(A,B);
if rank(A)==rank(p)
    disp('system is controllable')
else
    disp('system is not controllable')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%observability%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
q=obsv(A,C);

if rank(A)==rank(q)
    disp('system is Observable')
else
    disp('system is not Observable')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%observer Design %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pos=0.5;
ts=0.1;
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2))

wn=4/(z*ts)
r=roots([1,2*z*wn,wn^2])
% poless=10*r
% L_T=acker(A.',C.',poless)% state feedback observer ain matrix
% L=L_T'
k=place(A,B,r)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%check for closed loop eigenvalues%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Acl=A-B*k;
Ecl=eig(Acl)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%creat closed loop system
syscl=ss(Acl,B,C,0);

step(syscl)

kdc=dcgain(syscl);
kr=1/kdc;

syscl_scaled=ss(Acl,B*kr,C,D);
step(syscl_scaled)
% %%%% check poles of estimtor -error dynamc
obAcl=A-L.*C;
est_poles=eig(A-L.*C)

```

Appendix C

Schematics

C.1 Inverter Design.

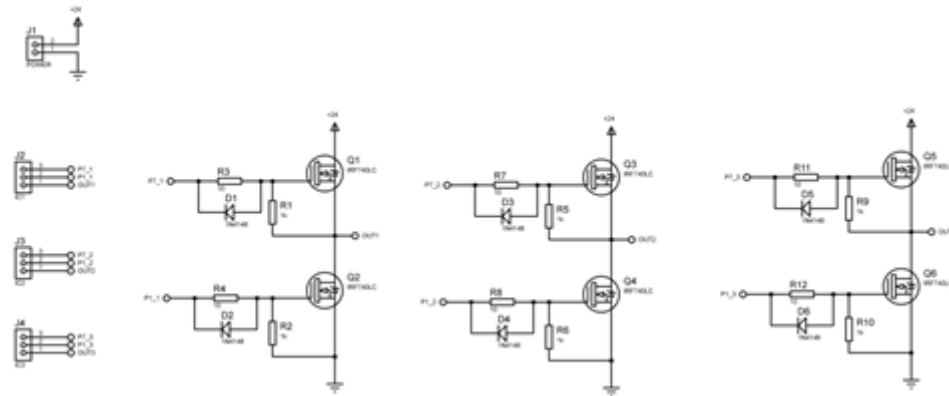


Figure (C.1): Schematic of the inverter circuit.

C.2 IR2110 Driver circuit.

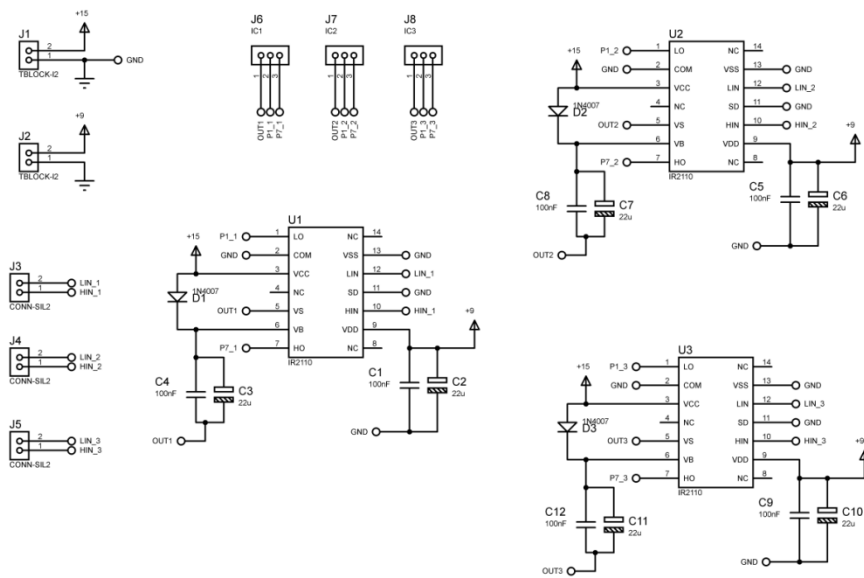


Figure (C.2): Schematic of the IR2110 Driver circuit.

C.3 Hall sensor circuit.

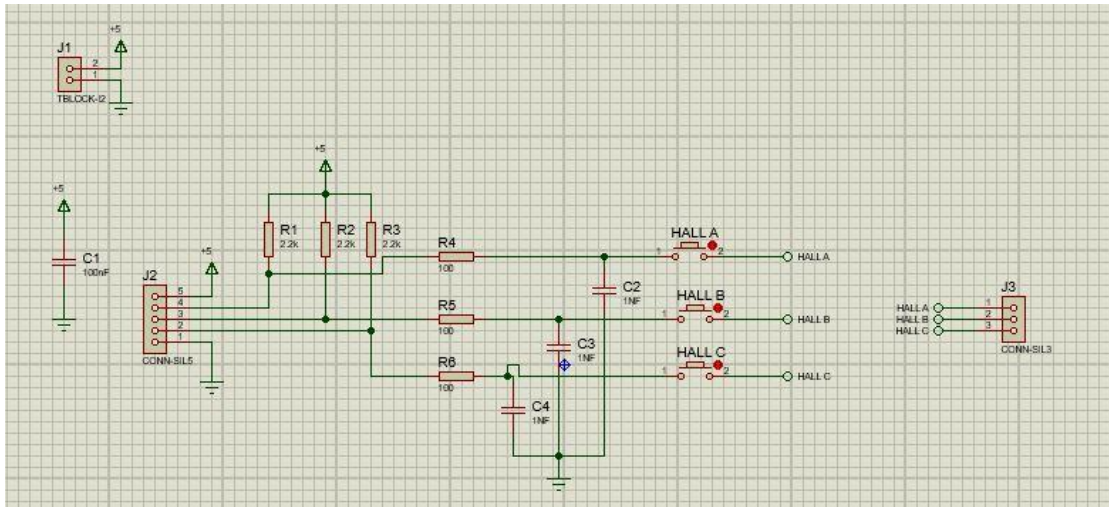


Figure (C.3): Schematic of the hall sensor circuit.

C.4 Motor driver (L298N H-bridge).circuit.

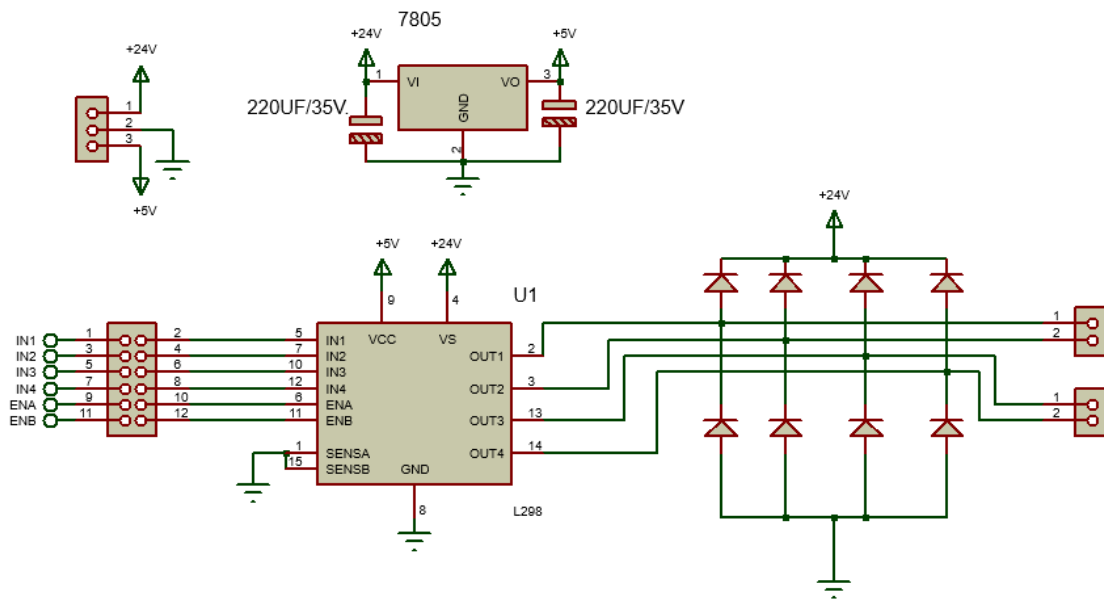


Figure (C.4): Schematic of the L298N H-bridge circuit (motor driver).

Appendix D

Data sheet

D.1 L298N data sheet.



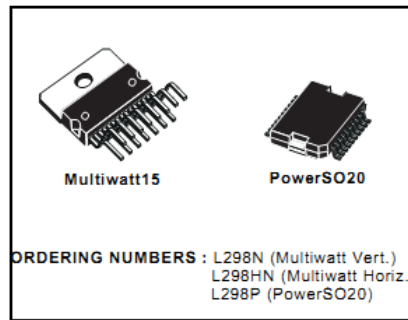
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

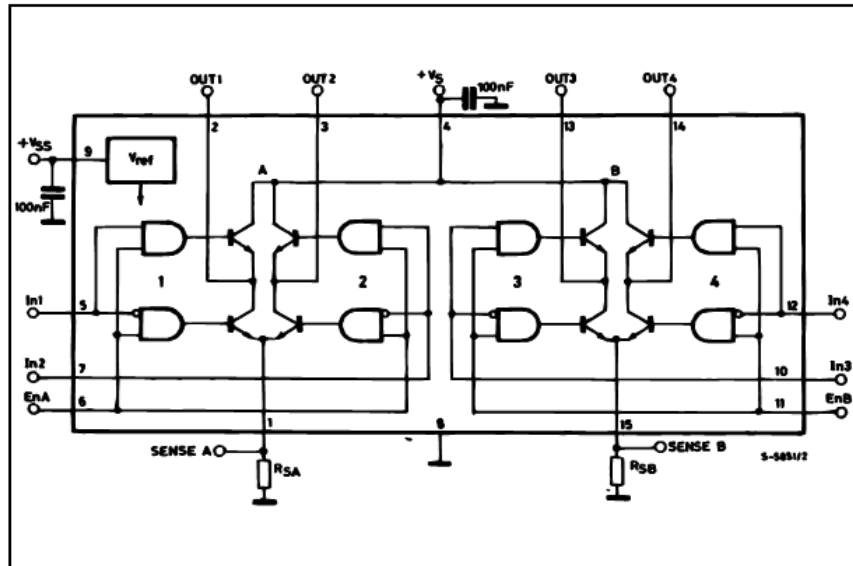
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



D.2 IRF740 data sheet.



IRF740, SiHF740

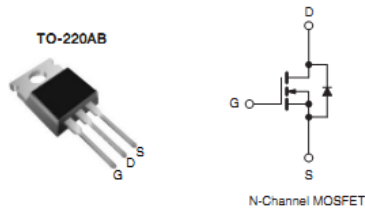
Vishay Siliconix

Power MOSFET

PRODUCT SUMMARY		
V_{DS} (V)	400	
$R_{DS(on)}$ (Ω)	$V_{GS} = 10$ V	0.55
Q_g (Max.) (nC)	63	
Q_{gs} (nC)	9.0	
Q_{gd} (nC)	32	
Configuration	Single	

FEATURES

- Dynamic dV/dt Rating
- Repetitive Avalanche Rated
- Fast Switching
- Ease of Paralleling
- Simple Drive Requirements
- Compliant to RoHS Directive 2002/95/EC



DESCRIPTION

Third generation Power MOSFETs from Vishay provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance and cost-effectiveness. The TO-220AB package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 W. The low thermal resistance and low package cost of the TO-220AB contribute to its wide acceptance throughout the industry.

ORDERING INFORMATION	
Package	TO-220AB
Lead (Pb)-free	IRF740PbF SiHF740-E3
SnPb	IRF740 SiHF740

ABSOLUTE MAXIMUM RATINGS ($T_C = 25$ °C, unless otherwise noted)			
PARAMETER	SYMBOL	LIMIT	UNIT
Drain-Source Voltage	V_{DS}	400	V
Gate-Source Voltage	V_{GS}	± 20	
Continuous Drain Current	I_D	V_{GS} at 10 V $T_C = 25$ °C	10
		$T_C = 100$ °C	6.3
Pulsed Drain Current ^a	I_{DM}	40	A
Linear Derating Factor		1.0	W/°C
Single Pulse Avalanche Energy ^b	E_{AS}	520	mJ
Repetitive Avalanche Current ^a	I_{AR}	10	A
Repetitive Avalanche Energy ^a	E_{AR}	13	mJ
Maximum Power Dissipation	P_D	125	W
Peak Diode Recovery dV/dt ^c	dV/dt	4.0	V/ns
Operating Junction and Storage Temperature Range	T_J, T_{stg}	- 55 to + 150	°C
Soldering Recommendations (Peak Temperature)	for 10 s	300 ^d	
Mounting Torque	6-32 or M3 screw		10
			1.1
			lbf · in
			N · m

Notes

- Repetitive rating; pulse width limited by maximum junction temperature (see fig. 11).
- $V_{DS} = 50$ V, starting $T_J = 25$ °C, $L = 9.1$ mH, $R_g = 25$ Ω , $I_{AS} = 10$ A (see fig. 12).
- $I_{SD} \leq 10$ A, $dI/dt \leq 120$ A/ μ s, $V_{DD} \leq V_{DS}$, $T_J \leq 150$ °C.
- 1.6 mm from case.

* Pb containing terminations are not RoHS compliant, exemptions may apply

Document Number: 91054
S11-0507-Rev. C, 21-Mar-11

www.vishay.com
1

This datasheet is subject to change without notice.
THE PRODUCT DESCRIBED HEREIN AND THIS DATASHEET ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc291000

D.3 IR2110 data sheet.

International
IR Rectifier

Data Sheet No. PD60147 rev.V

IR2110(S)PbF/IR2113(S)PbF

HIGH AND LOW SIDE DRIVER

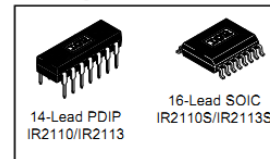
Features

- Floating channel designed for bootstrap operation
Fully operational to +500V or +600V
Tolerant to negative transient voltage
dV/dt immune
- Gate drive supply range from 10 to 20V
- Undervoltage lockout for both channels
- 3.3V logic compatible
Separate logic supply range from 3.3V to 20V
Logic and power ground $\pm 5V$ offset
- CMOS Schmitt-triggered inputs with pull-down
- Cycle by cycle edge-triggered shutdown logic
- Matched propagation delay for both channels
- Outputs in phase with inputs

Product Summary

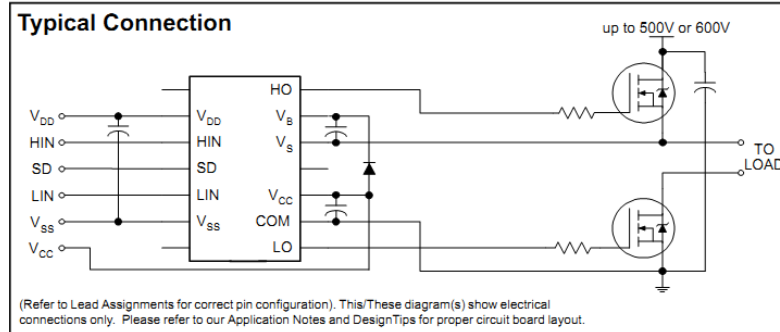
V_{OFFSET} (IR2110)	500V max.
(IR2113)	600V max.
$I_{\text{O+/-}}$	2A / 2A
V_{OUT}	10 - 20V
$t_{\text{on/off}}$ (typ.)	120 & 94 ns
Delay Matching (IR2110)	10 ns max.
(IR2113)	20ns max.

Packages



Description

The IR2110/IR2113 are high voltage, high speed power MOSFET and IGBT drivers with independent high and low side referenced output channels. Proprietary HVIC and latch immune CMOS technologies enable ruggedized monolithic construction. Logic inputs are compatible with standard CMOS or LSTTL output, down to 3.3V logic. The output drivers feature a high pulse current buffer stage designed for minimum driver cross-conduction. Propagation delays are matched to simplify use in high frequency applications. The floating channel can be used to drive an N-channel power MOSFET or IGBT in the high side configuration which operates up to 500 or 600 volts.



www.infineon.com/gatedriver

1

D.4 H9700 optical sensor data sheet.

HEDS-970x, HEDS-972x Series Digital Output Small Optical Encoder Modules



Data Sheet



Description

The HEDS-9700 series is a high performance, low cost, optical incremental encoder module. When operated in conjunction with either a codewheel or codestrip, this module detects rotary or linear position. The module consists of a lensed LED source and a detector IC enclosed in a small C-shaped plastic package. Due to a highly collimated light source and a unique photo-detector array, the module is extremely tolerant to mounting misalignment.

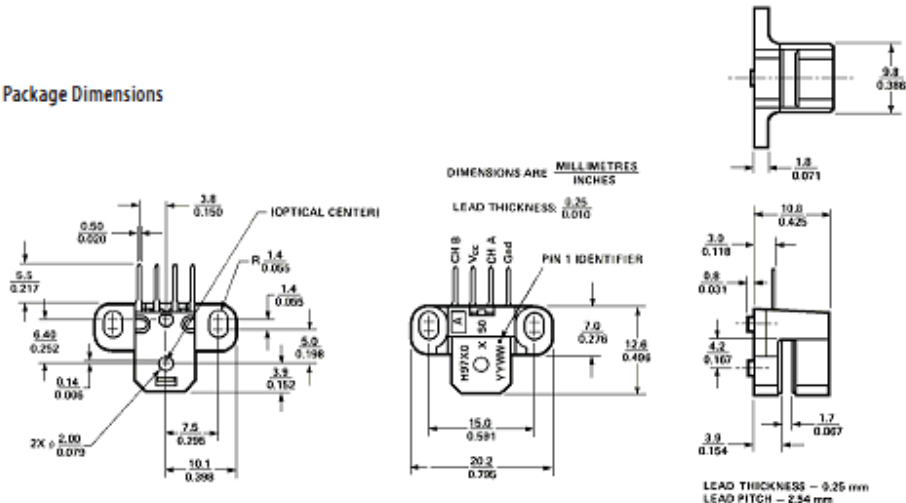
The two channel digital outputs and 5V supply input are accessed through four solder-plated leads located on 2.54 mm (0.1 inch) centers.

The standard HEDS-9700 is designed for use with an 11 mm optical radius codewheel, or linear codestrip. Other options are available. Please contact factory for more information.

Features

- Small Size
- Low Cost
- Multiple Mounting Options
- Wide Resolution Range
- Linear and Rotary Options Available
- No Signal Adjustment Required
- Insensitive to Radial and Axial Play
- -40°C to +85°C Operating Temperature
- Two Channel Quadrature Output
- TTL Compatible
- Single 5V Supply
- Wave Solderable

Package Dimensions



Mounting Option #50 - Standard (Baseplane Mounting) Contact Factory for Detailed Package Dimensions

ESD WARNING; NORMAL HANDLING PRECAUTIONS SHOULD BE TAKEN TO AVOID STATIC DISCHARGE.