ZİRVE UNIVERSITY

WIDE RANGE DIGITAL MOTOR SPEED MEASUREMENT USING MICROCONTROLLERS

by

KABIRU ALHAJI HASSAN

Submitted to the Institute for Graduate Studies in
Natural and Applied Sciences in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Thesis Supervisor: Asst. Prof. Dr. Hüseyin Emre KANKAYA

June,2014.

WIDE RANGE DIGITAL MOTOR SPEED MEASUREMENT USING MICROCONTROLLERS

APPROVED BY:

Asst. Prof. Dr. Hüseyin Emre KANKAYA ....
(Thesis Supervisor)

Asst. Prof. Dr. Mamatjan YASIN ....

Asst. Prof. Dr. Suat MERCAN ....

Prof. Dr. Sinan HINISLIOĞLU ....
Director of Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

All praise be to almighty Allah who made it possible for me to complete this research in His infinite mercy and guidance.

Certainly, it is my absolute pleasure to express my appreciation to my understanding and supportive thesis supervisor Asst. Prof. Dr. Hüseyin Emre Kankaya, whom without his help this research work, would not have come true. Besides my supervisor, I would like to extend my gratitude to Asst. Prof. Dr. Mamatjan Yasin and Asst. Prof. Dr. Suat Mercan for their advices, encouragements, insightful comments and corrections.

My profound appreciation goes to my parents and the entire family members for their enormous contributions to my career. I am also indebted to my friends and well wishers for their support in one way or the other towards my academic career.

My appreciation goes to Kano State government of Nigeria for its financial and moral support towards achieving an academic excellence in my studies.

To all those aforementioned persons and others whom I ought to have mentioned, extending my unreserved thanks to you all.

ABSTRACT

WIDE RANGE DIGITAL MOTOR SPEED MEASUREMENT USING MICROCONTROLLERS

KABIRU ALHAJI HASSAN

ZİRVE UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

Thesis Supervisor: Asst. Prof. Dr. Hüseyin Emre KANKAYA

In industrial applications, it is sometimes important to be able to control motor speed over a wide range with good speed resolution and efficiency. However, using the conventional analog control methods has some drawbacks such as nonlinearity in the analog speed transducers and inaccuracy in speed measurement. However, a digital speed control system eliminates nonlinearity in the speed transducers, thus giving better accuracy in speed measurements. A digital motor speed device uses digital techniques to measure, reverse, control and correct motor speed operations. However, this research focuses on the design, construction and development of a wide range digital motor speed measurement harnessing the functionality and flexibility of microcontrollers as the overall brain of the system. The proposed system is suitable for wide range motor speed measurement applications in providing fast dynamic response and high accuracy across wide range by using Constant Elapsed Time (CET) method and improving parameters of the frequency divider, and finally using the Simple Prediction Algorithms to obtain the next predicted output performance of the system.

Keywords: CET, Simple Prediction Algorithms, Microcontrollers, Frequency Divider and Motor Speed.

ÖZET

BAŞLIK

KABIRU ALHAJI HASSAN

ZİRVE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Tez Danışmanı: Tez Danışmanı

Endüstriyel uygulamalarda geniş aralıkta iyi hız çözünürlüğü ve verimliliği ile motor hızını kontrol etmek önemli olabilmektedir. Ancak, geleneksel analog kontrol yöntemleri kullanılması analog hız dönüştürücüde doğrusal olmama ve hız ölçümünde tutarsızlık gibi bazı sakıncaları içermektedir. Ancak, dijital hız kontrol sistemi, hız dönüştürücülerdeki doğrusal olmamayı ortadan kaldırmaktadır. Böylece hız ölçümlerine daha iyi doğru sonuca ulaşılabilmektedir. Dijital motor hızı aygıtı, hızı ölçmek, tersine çevirmek, kontrol ve düzeltmek için dijital teknikler kullanmaktadır. Ancak, bu araştırmada sisteminin genel olarak beyni olan mikroişlemcinin işlevselliğinden ve esnekliğinden yararlanarak geniş aralıkta dijital motor hızı ölçümünün tasarımı, yapımı ve geliştirilmesi üzerine odaklanılmıştır. Önerilen sistem Sabit Geçen Zaman (CET) yöntemini kullanarak geniş aralık boyunca hızlı dinamik tepki ve yüksek doğruluk sağlayan geniş motor hızı ölçüm uygulamaları için ve sisteminin bir sonraki tahmin edilen çıktı performansını elde etmek için frekans bölücü parametrelerin geliştirilmesine, ve son olarak Basit Tahmin Algoritmalarının kullanımına uygundur.

Anahtar Kelimeler: CET, Basit Tahmin Algoritmalar, Mikrodenetleyiciler, Frekans Bölücü ve Motor Hızı.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background

It is obvious that in industrial control systems, the regulation and measurement of digital motor speed over a wide range is of paramount importance so as to obtain a desired speed control, accuracy and reliable output performance of the system. Although, the conventional analog speed measuring devices has some limitations such as nonlinearity associated with analog speed control transducer and inaccuracy when transmitting analog signal across the system [1]. Some errors are also encountered when manipulating the signal to effect control action on the motor system as a result of temperature, component aging and extraneous disturbances [2].

However, digital speed control approach is superior in that it eliminates measurement nonlinearity in the speed transducer and makes the speed setting exactly reproducible. A digital speed motor measuring device has a shaft encoder, electronic circuits and the software part. Then the electric pulses (rising and falling edges) are obtained from the output of the encoder, and the frequency of these digital pulses is proportional to the rotational speed. Hence, digital speed can be calculated to obtain the measurement of the system.

The target of this research thesis is to explore the method of the Constant Elapsed Time (CET) in trying to increase the range of the motor speed measurement. The simulations and standard realization with reference to this method will be shown and analyzed, with indications of experimental results obtained from the developed wide range digital motor speed control system with expected minimum amount of errors.

1.2. Statement of Problem

The significance of motor speed measurement is a key factor in industrial applications. Digital motor speed control systems can be measured using various methods, although in this research, an overview of various methods are presented which can be used to increase the range of the motor speed measurement [2]. Hence, there is a limitation to the range of speed that can be measured, so the focus of this research is to increase measurement of the motor speed between the minimum and the maximum limits using microcontrollers.

1.3. Motivation

An increase in motor speed control system accuracy often leads to increased time measurement and narrows measurement range [2]. Hence, there is need to reach a compromise between resolution and time measurement. As such the CET method is proposed in this thesis to increase the speed measurement between upper and lower limits and yet providing an optimal compromise between resolution and measurement time with high accuracy, excellent dynamic performance and minimum error [1], [2]. Indeed, with this approach been realized, an intelligent measurement prototype can be modeled as part of some complex multifunctioning virtual speed control instruments, which is often requested in today's industrial applications.

1.4. Purpose of the Thesis

The purpose of this thesis is to increase the range of the digital motor speed measurement using microcontrollers.

1.5. Importance of the Thesis

The use of digital approach in motor speed measurement is usually more economical for high accuracy, low power consumption, flexibility [3]. Although, the beauty of this thesis is that with the incorporation of crystal oscillators, there is no need to use high-speed devices to increase the motor speed measurement. Applications of digital motor speed systems include speed of rotating shaft in an engine, fans speed measurement, optimum speed for drilling a hole on a metal piece, robotic arms, washing machines, electric trains, rolling mills, pulley system and elevator control systems and so on [4], [5].

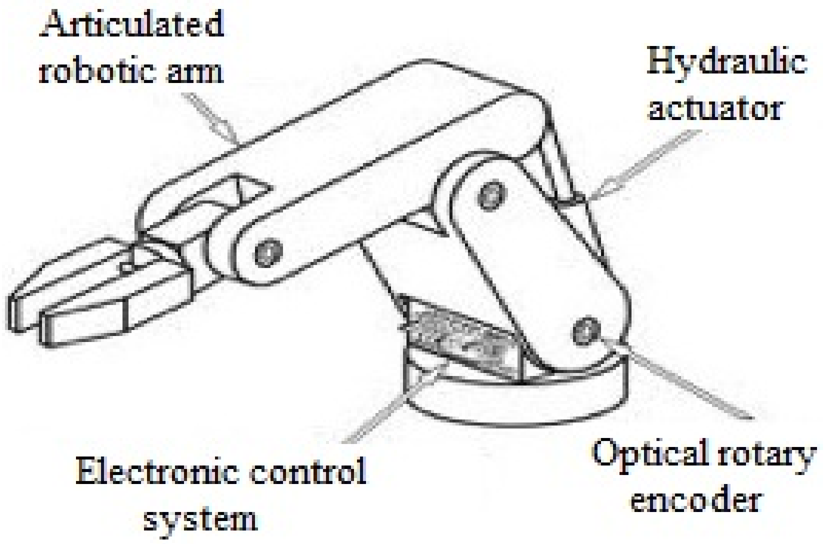


Figure 1.1. Digitally Controlled Robotic Arm [6]

1.6. Method and Procedures

The method of CET will be used in this thesis, which is a combination of encoder pulse counting and time measurement. The time is measured by counting clock pulses been generated by an interrupt service routine [1], [2]. The algorithm used is based on the Simple Prediction algorithm to obtain the system performance and changing the parameter of the frequency divider, which is integrated with microcontrollers.

1.7. Scope of the Thesis

With regards to achieving improved performance in this research, some designated scopes have been outlined such as the use of PIC16F887 microcontrollers which serve as the overall brain of the project, applying the CET method to reach a compromise between resolution and time measurement and also using the simple prediction algorithm so as to enhance higher performance of the motor speed measurement in RPM (Revolution per Minute) across a wider range over the previously obtained limits, and thus minimizing the error at a lower cost and lower power consumption. Hence, the measured speed will be displayed on a 16x2 LCD (Liquid Crystal Display) in line with the total time elapsed during the measurement.

1.8. Outline of the Thesis

This thesis is dealing with wide range digital motor speed control using microcontrollers by applying the CET method to optimize the performance of the system across a wider range. So, all the research, simulations and analysis of the thesis are presented in five chapters.

Chapter 1 discusses about the background studies, motivation, statement of problem, purpose of the thesis, importance, methodology and scope of the project research. While chapter 2 discusses more on theoretical background and literature reviews of related previous works that have been done. It also discusses about classical speed measurement methods, microcontrollers overview, technical data and some examples. Then chapter 3 is focused on design, construction of hardware and software implementation of the thesis. However, it involves block diagrams, flow charts and circuit diagrams of the thesis. In chapter 4, the results and discussion will be presented, this is important to determine whether the purpose of the research is achieved or not. Lastly, chapter 5 discusses the conclusion of the thesis and recommendations based on future works that can be done in relation to the thesis.

2. THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1. Introduction

This chapter discusses on the theoretical background studies of related projects and literature reviews of previous works carried out as related to digital motor speed measurement using microcontrollers across wide range. However, this includes the study of different types of electric motors drives, classical speed measurement methods, speed measurement using encoder marks, method of CET and types of digital motor speed controllers. It also gives an overview of microcontrollers, interrupts service routines, C programming language, 16x2 LCD, frequency counters, and yet giving some technical data, performance analysis, diagrammatic illustrations, formulae and related examples obtained from previous works.

2.2. Background Study of Related Works

With regards to this thesis, it is necessary to make some background study on various works or projects been carried out in relation to this research. Indeed, this is to improve or even make further contributions with respect to this research towards achieving better results. A digital tachometer was designed in which the advantages of CET method were analyzed, although some errors were associated with the experimental results obtained [2]. Also, a wind speed meter was designed to determine wind speed and direction with the aid of microcontrollers, and the current speed with its direction were both displayed on LCD [7]. R. Bonert also designed a microcontroller-based tachometer, which provides a digital measurement of speed with better accuracy and higher performance using CET method [1]. In 2004, a paper was presented which discusses the control of motor speed measurement using DC (Direct Current) chopper and PWM (Pulse Width Modulation) [8]. Another paper was published which illustrates a digital motor speed controller by employing digital approaches to determine motor speed [3]. S.M. Dawood also presented a paper, which focuses on the study of two DC motors speed measurement, direction and operating period using microcontrollers. The motor speeds are maintained by the controller upon load variation [9].

A related projected was carried out which describes the control of a DC motor speed using digital approach that uses Intel 8080 microcomputer and CMOS (Complementary Metal-Oxide Semiconductor) random logic which is fed by an SCR (Silicon Controlled Rectifier) chopper [4]. In November 2010, a project was conducted with the objective of obtaining the RPM of a motor speed with the aid of non-contact tachometer, the prototype was constructed using microcontrollers, an LCD module, battery and sensor [9]. Another project was designed which focuses on the implementation of a digital motor speed measurement system with the incorporation of PIC16F877A microcontroller. However, it involves a closed-loop control system, which is centered at maintaining the DC motor speed upon variation of load [5]. A stepper motor project was designed which describes the use of extended Kalman filter to implement the design of a controller which is flexible to load torques and yet accomplishing certain limits by estimating motor speed position mark of the motor via Kalman filter and the incorporation of microcontrollers [10]. Also, in March, 2012 a paper was published which discusses on the techniques introduced to model power components. The microcontroller based variable closed-loop digital motor speed controller was implemented using computer simulations as the basic tool for determining control and transient studies [11].

M.A. Fikri designed a project, which describes the expectation of obtaining a desired speed of DC motor and to energize the motor at the preset speed, however this requires the motor speed feedback constantly so as to maintain the speed [12]. Another project was focused on closed-loop real time system for digital motor speed measurement control system via the incorporation of PIC16F84 microcontroller in which the tachometer is synchronized with the motor shaft and a feedback speed signal is provided to the controller [13]. B. Upchurch designed a project to build a circuit that is capable of serving as a speedometer or an accelerometer for a digital motor, in which the obtained data can be associated to the speed, position and acceleration of the DC motor which can be employed in robotics design [14]. Another paper was published by J.P. Hui Wang which describes the variable M/T (Metal Tube) method for determining speed measurements using optical incremental encoders as related to motor servo systems, however in this project encoder and high frequency clock pulses are determined in a variable interval, thus enhancing high performance measurement across high and low speed [15]. A microcontroller based project was designed which focuses on PIC18F4431 microcontroller as the controller for the proposed implementation of a three phase induction motor via speed sensor which is extremely important in industrial and electrical applications [16].

Hence, based on the reviews of related works, it is obvious that none of the projects discussed about increasing the speed measurement across a wide range. As such, the contribution of this research is to increase the range of speed measurement between minimum and maximum limits.

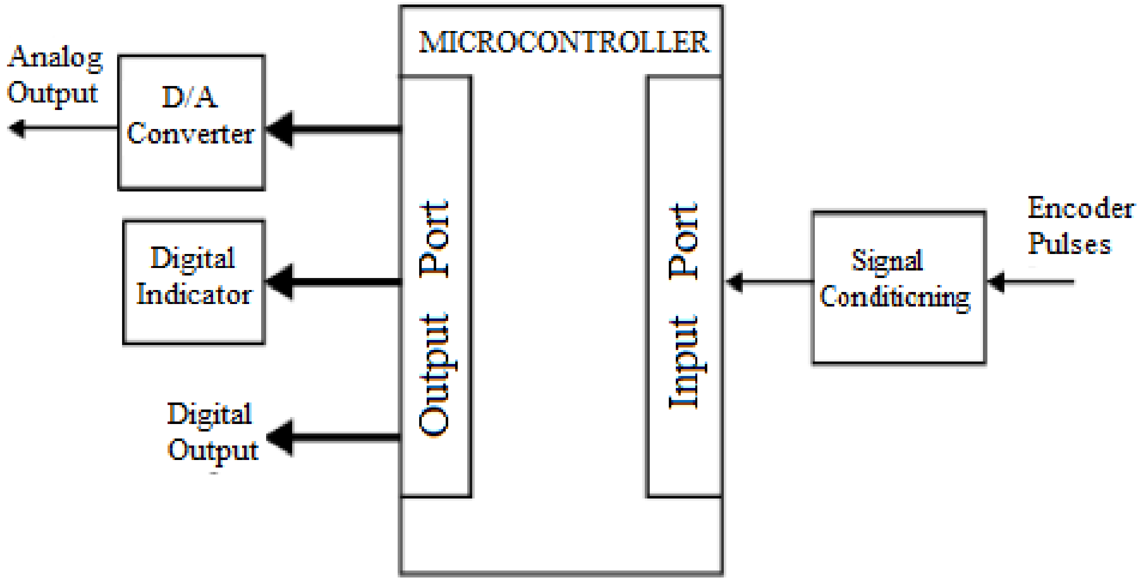


Figure 2.1. Block Diagram of Digital Tachometer [2]

2.3. Background Study of Hardwares Used

Based on the reviews of previous works as related to this research, there is need to make some studies on some of the hardwares used in the related works. Meanwhile, this will facilitate a better research to be conducted and indeed some remarkable contributions can be made in this research. Some of the hardwares include microcontrollers, encoders, frequency divider and electric motors etc.

2.3.1. Microcontrollers

A microcontroller is termed as a small computer designed on a single integrated circuit, which contains memory, processor and configurable input/output peripherals. Microcontrollers basically contain various general-purpose input/output pins (GPIO). However, GPIO pins can be configured to control devices such as LEDs (Light Emmitting Diodes) or motors [17]. A microcontroller is also referred to as an embedded controller because the microcontroller and its auxiliary circuits are largely built into, or embedded in the devices or systems they control. Microcontrollers are available in various word lengths, which include 4, 8,16, 32, 64 and 128 bits microcontrollers [17]. Indeed, microcontrollers are found in most kinds of digital or electronic devices nowadays, most devices or systems that are capable of controlling, processing measuring, calculating, storing or even displaying data or information have an embedded microcontroller chip inside that serves as the overall brain of the device. Hence, the largest single use of microcontrollers' lies in automobile industry in which microcontrollers are widely employed in controlling and regulating engines and power controls [17].

Microcontrollers have the capability of providing real time response to instructions in the embedded device or system under control. Meanwhile, when a particular event occurs, an interrupt system can alert the processor to hold on with processing or executing the current instruction sequence and to start an interrupt service routine. As such, the ISR (Interrupt Service Routine) will carry out any processing or execution required based on the source of the interrupt (such as internal timer overflow), then go-

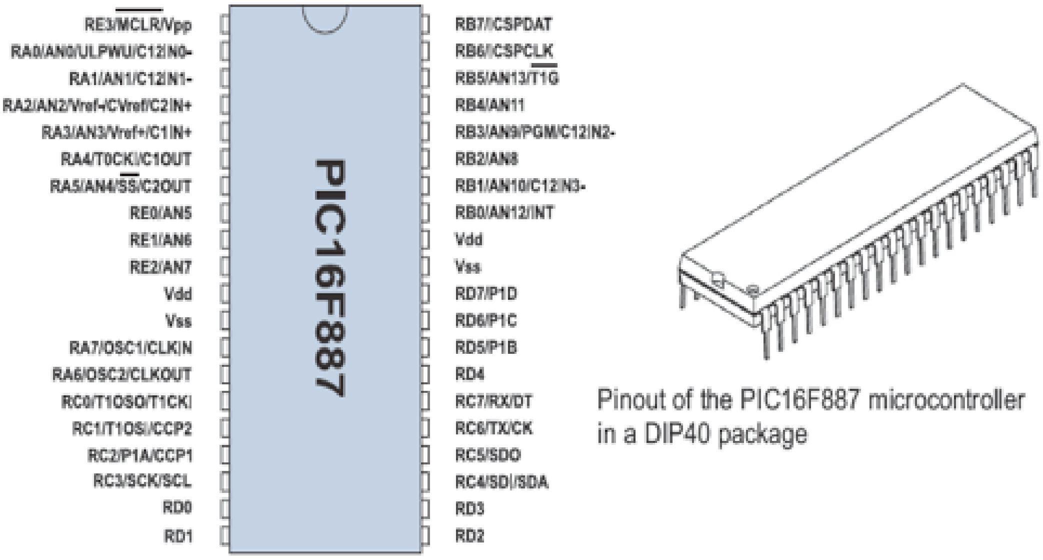


Figure 2.2. PIC16F887 Microcontroller [17]

ing back to the actual instruction sequence. The importance of microcontrollers cannot be over emphasized, microcontrollers are largely used in modern digital or electronics systems. Thus, some major applications of microcontrollers are stated below.

a) Used in biomedical instruments.

b) Widely used in communication systems.

c) Used as peripheral controller in personal computers.

d) Used in robotics.

e) Used in automobile fields, etc.

2.3.2. Encoders

An encoder can be termed as a device which can convert a mechanical information or data of a shaft or position into its equivalent electrical signal [18]. In otherwords, encoders are sensors or transducers that are capable of generating digital signals in relation or response to a specified motion or movement. As an electro-mechanical system, an encoder has the ability of providing motion or movement control system with related data corresponding to direction, velocity and position [19], [20].

Basically, we have two types of encoders, which include linear and rotary encoders. For a rotary encoder, it is designated to respond to rotational movement or motion, while a linear encoder has the ability to respond to movement along its way of motion. Incremental encoders are designated to produce series of pulses as they are set in motion. Therefore these pulses can be used to determine speed, or it can be fed to a counter to keep track of direction or position. While absolute encoders are capable of generating multi-bit digital words that show actual position directly [18], [20].

Encoders can be used as feedback devices for motor speed control systems, measurement of length, cutting and position and can also serve as input for speed and rate measurement systems. Motor Speed is one of the well known encoders application, where the encoder is directly attached to the end of a motor through a shaft. As such the basic parameter of determining measurement is motor speed in which a feedback is sent to a drive via the encoder, so as to authenticate the correctness of the speed and direction [19]. Some of the encoders' applications include: robotics, drilling machines, length measurement, conveyors, rolling mills, elevators, bottling machines.

Some of the advantages of encoders are high resolution, high accuracy, affordable feedback, portable size and synchronization with existing applications. While some of the disadvantages include interference with direct light source, easily contaminated by dust, dirt and oil. One of the limitations of an incremental encoder is the count loss, which happens during power failure. So when restarting, the equipment must be referred back to a home position so as to reinitialize the counter. For a linear encoder, the system makes use of magnetic sensor read head and a magnetic scale to detect the changes in magnetic field and outputs a signal. Since a linear encoder detects change in the magnetic field, then the interference of light, dust, oil and dirt have no effect on the system, as such they offer high reliability in harsh environments. Also, magnetic encoders do not make errors as a result of contamination because their sensors detect variations in magnetic fields embedded in the rotor, also oil, dirt and water have no effect on the magnetic fields.

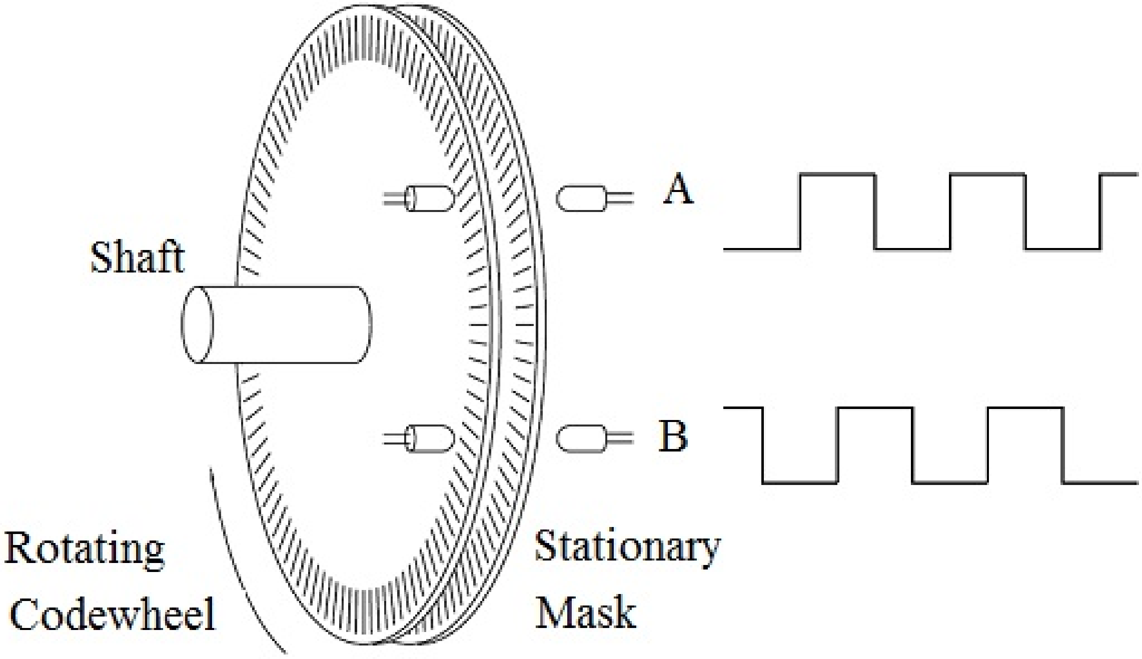


Figure 2.3. Rotary Encoder [21]

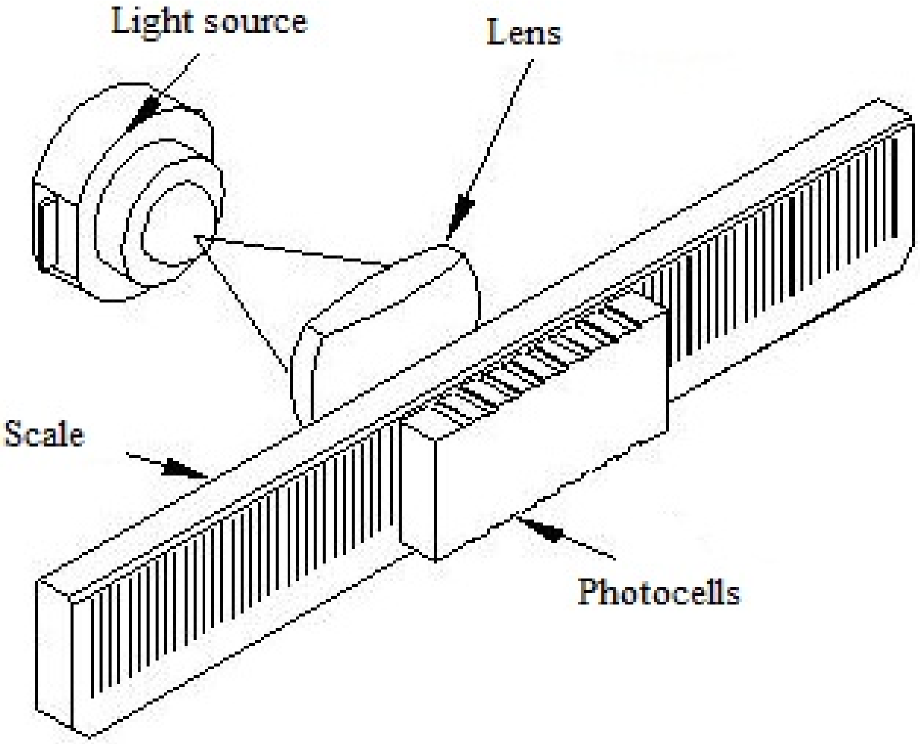


Figure 2.4. Linear Encoder [22]

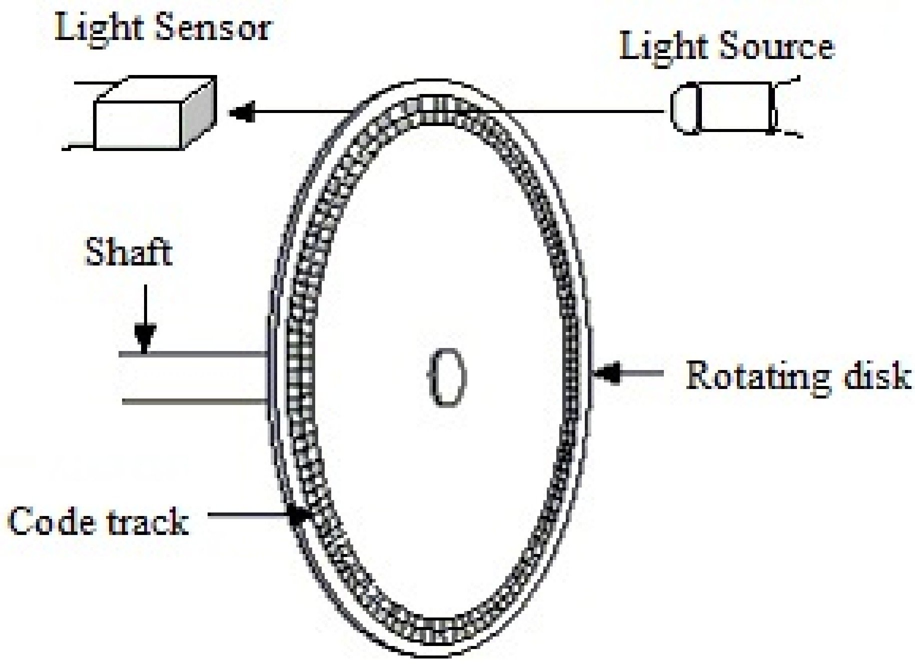


Figure 2.5. Incremental Encoder [23]

2.3.3. Frequency Divider

A frequency divider, which can also be called a clock divider or scalar or prescalar, can be described as a circuit which takes a signal frequency as an input, f_{in}, and then generates its corresponding output signal of the frequency. In otherwords modulo n frequency divider can be termed as a sequential logic circuit that gives a signal output with period n times greater than the periodic clock pulse [24], [25]. However, it is often said that a frequency divider is similar to a counter just that it is not so much important how data or information is encoded at the outputs, so the key point is the division ratio or proportion and the output signal filling factor. For instance, a phase-locked loop frequency synthesizer employs the use of frequency dividers to produce frequency that is a multiple of an input reference frequency. Frequency dividers can be developed and incorporated for both digital and analogue applications. For a frequency divider with power-of-2 integer division, a clocked input signal of a simple binary counter can be used to obtain the output, thus producing a toggle of the input clock [26]. Meanwhile, the least-significant output bit toggles at 1/2 the rate of the clock input, while the next bit alternates at 1/4 the rate, and the third bit also toggles at 1/8 the rate of the clock input, etc. However, the sequential patterns of flip-flops are a fundamental way for integer-n frequency division. Although, such division is frequency and phase coherent with respect to the source over environmental effects including temperature, thus the simplest arrangement is a series in which each flip-flop is a divide-by-2, as such for an arrangement of three of these; such output system would be a divide-by-8 clock frequency. Hence, by combining additional logic gates to the sequence of flip-flops, subsequent frequency division ratios can be produced. Indeed, an integrated logic circuit family is capable of providing a single chip solution for most common frequency division ratios [25], [26].

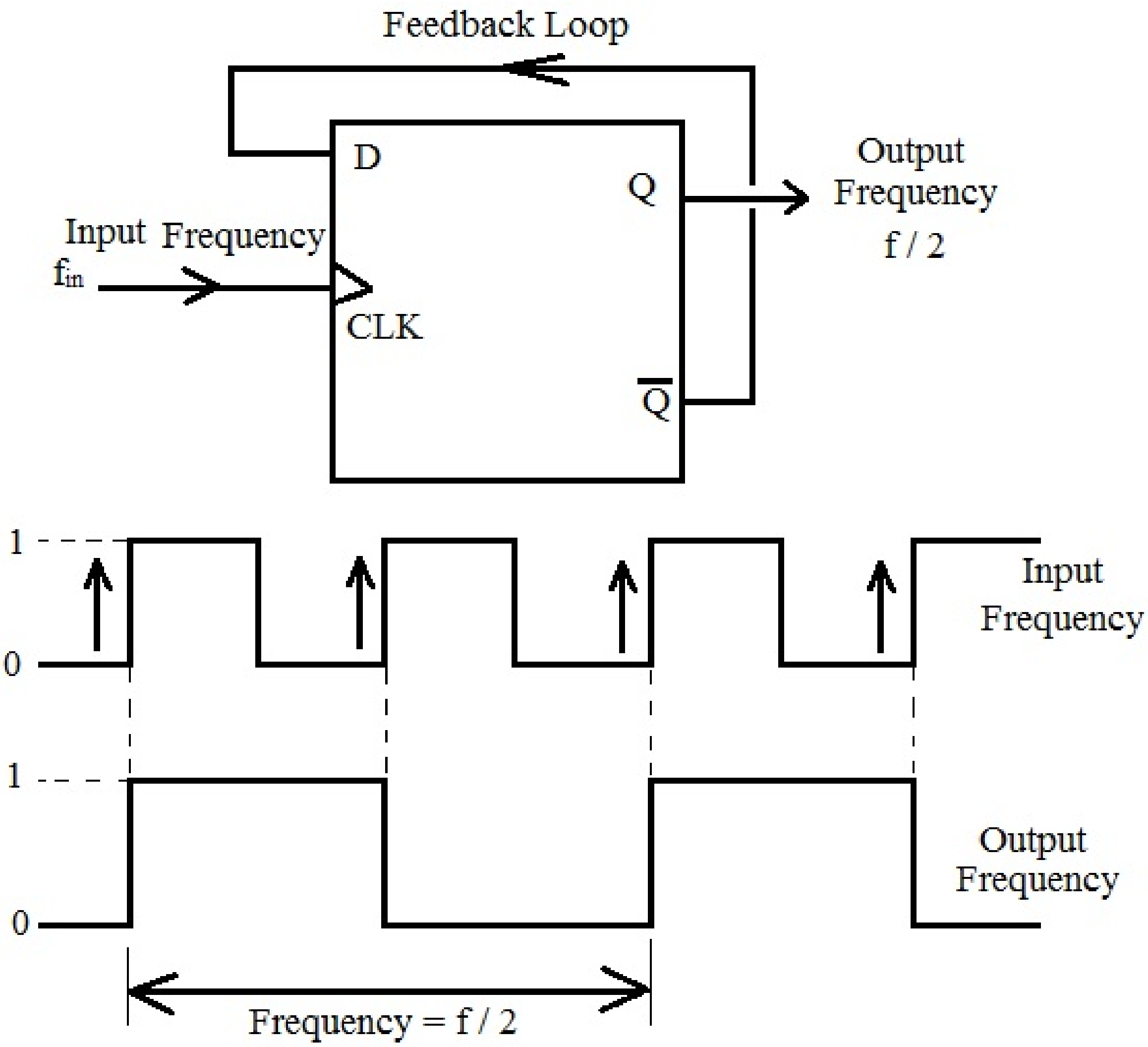


Figure 2.6. Frequency Divider

2.3.4. Electrical Motors

An electric motor is basically an electrical device, which is capable of converting electrical energy into its corresponding mechanical energy. However, in typical motoring perspective, interrelation process between electric motor's magnetic field and its winding currents to produce corresponding force within the electric motor give rise to electrical motor's operation [27]. Basically, electrical motors can be energized by either direct current sources or by alternating current sources. Hence, there are basically two types of electrical motors, which are direct current (DC) electrical motor and alternating current (AC) electrical motor. However, the reference of either DC or AC has to do with how electrical current is transferred to and fro the electrical motor, although both types of electrical motors have their corresponding functions and applications [27], [28].

2.4. Speed Control of DC Motor

DC motor speed control has to do with the purpose of varying motor speed, direction or perhaps controlled to act as dynamic brake system to a desired pattern to perform a specific task. Indeed, speed control is quite a different aspect from speed regulation in which there is a varying change in speed as a result of change in load across the shaft. However, speed control can either be obtained manually or by process of some automatic control systems. One of the fundamental characteristics of dc motor is that its speed can easily be controlled [29]. As such, DC motor speed can be controlled by varying:

i) Terminal voltage of the armature, V (Voltage control)

ii) External resistance in armature circuit, R (Rheostat control)

iii) Flux per pole (Flux control)

Hence, the mathematical equation of speed control of DC motor can be expressed as shown below [29].

$$\mathbf{N} = \mathbf{KV} - \frac{\mathbf{I_a}(\mathbf{R_a} + \mathbf{R})}{\phi} \tag{2.1}$$

Therefore having K as a constant, V as armature voltage, I_a as armature current, R_a as armature resistance, R as external resistance and ϕ as the flux. Then the speed (N) of DC motor are of 3 types which include; Series, Shunt and Compound which can all be controlled by varying the parameters on the RHS (Right Hand Side) of the given expression above. However, a robust way of speed control is by the use of microcontrollers, which enhances simplicity, reliability, minimised error and affordability.

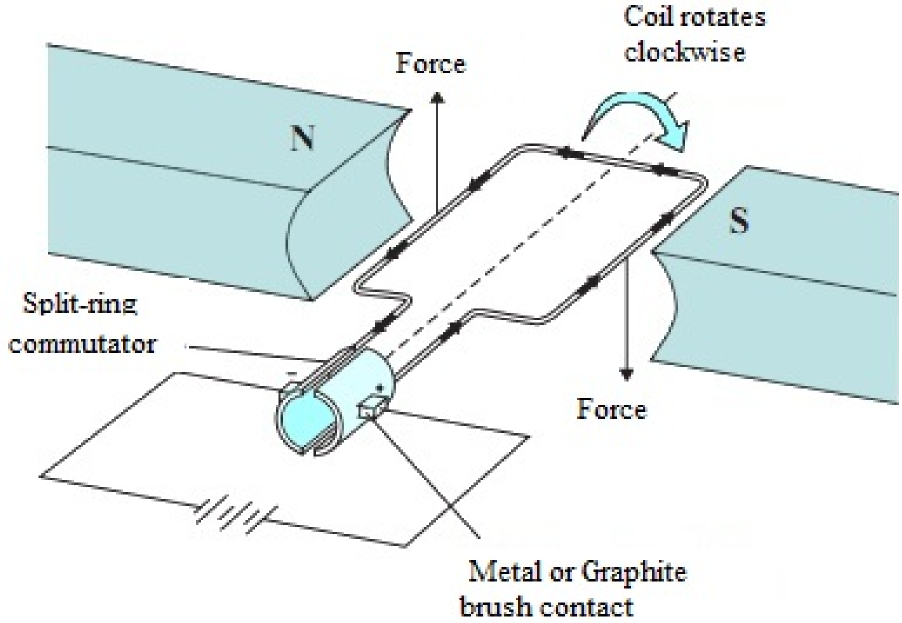


Figure 2.7. Electric Motor Diagram [29]

2.5. Speed Control DC Motor Using Microcontrollers

The development of fast response and high performance motor drives is very important in industrial and domestic applications. Consequently, the increasingly requirement for robust dynamic speed control tracking and load regulating response of motor control is of paramount importance to produce high performance response. The DC motors are used in various applications such as industries, robotics arm, washing machines etc. The use of microcontrollers to control the speed of DC motors provides simplicity, ease of controllability, reliability, error reduction and favorable cost, which are key factors in industrial applications. Infact, with the incorporation of microcontroller the DC motor can be controlled at a desired variable speed with minimum amount of error [12], [30]. However, the speed can be controlled by generating some rising or falling edges from the microcontrollers so as to facilitate pulse counting and time measurement by initialising timer0 and the interrupt service routines.

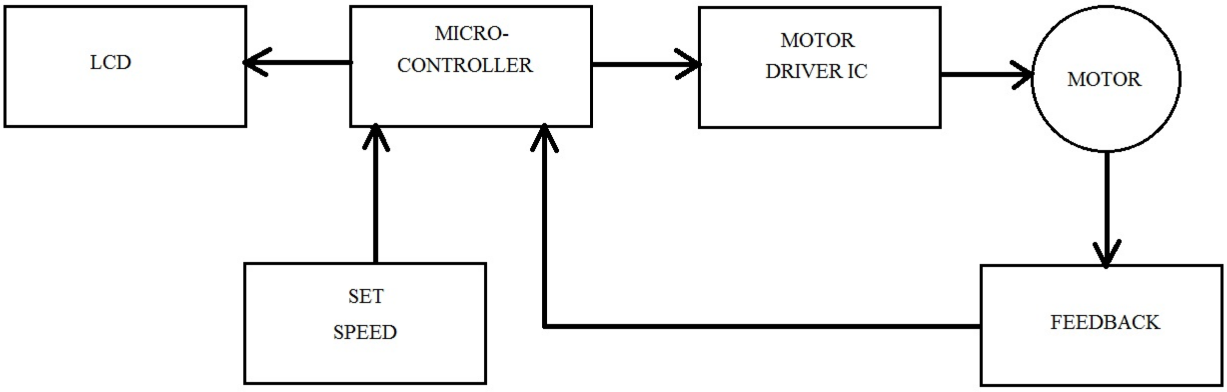


Figure 2.8. DC Motor Speed Control using Microcontrollers

The Fig 2.8 above describes the efficient and simple approach to control speed of digital motors using microcontrollers and incorporation of motor driver IC (Integrated Circuit). Firstly, using a variable resistance, which is connected to the microcontroller, the speed of the dc motor connected to the system is set. However, as the controller gets the speed set, it then starts counting the pulses that it is getting from a sensor and on the basis of that it measures the current speed of motor, which can be displayed on the LCD. And a feedback loop from the motor is sent to the microcontroller, which controls the driver IC again, and the whole process is continuous in a loop.

2.6. Classical Speed Measurement Methods

There are several methods that are available for motor speed measurement through various sources. However, details of these methods will be discussed so as to illustrate their performance, efficiency and limitations. Here are some known classical methods of speed measurement [1].

i) Direct Pulse Counting.

ii) Single Pulse Time Measurement.

iii) Constant Elapsed Time (CET).

iv) Pulse Time Measurement using variable number of counted pulses.

The direct pulse counting and single pulse time measurement methods are virtually similar as applicable in any timer/counter instrumentation. While the constant elapsed time and pulse time measurement methods has to do with combined form of time measurement and encoder pulse counting. Although, the CET method is associated with some advantages over the other methods. This method only needs a short measurement time, which slightly changes across the entire speed range of the motor. Furthermore, previous information with regards to the actual speed range is not needed, thus making the CET method a better choice over the other method of pulse time measurement using variable number of counted pulses [1]. Another advantage is that the CET method gives an optimal compromise between time measurement and resolution, which is not obtainable via the direct pulse counting and single pulse time methods.

2.7. Method of Constant Elapsed Time (CET)

The method of CET, is a combination of two operations, which include counting the encoder pulses and time measurement. However, the time can be determined by counting clock pulses of the clock pulse period. This method provides the satisfaction requirements for both short pulses and short time measurement time. Infact, the CET method ensures an optimal compromise between time measurement and resolution, which is not obtainable in other methods [1], [2]. An analytical comparison of the CET method with the direct pulse counting and single pulse time measurement methods is shown in Table 2.1.

Table 2.1. Comparison of CET Method with other methods [1]

Error (Δn/n)	***Pulse Counting***	***Pulse Time Measuring***	***CET***
At 30 rpm, At 3000 rpm	85.0%, 0.85%	0.025%, 2.5%	0.025%, 0.05%
Response Time (Δt)	***Pulse Counting***	***Pulse Time Measuring***	***CET***
At 30 rpm, At 3000 rpm	2.5ms, 2.3ms	1.95 ms, 0.02 ms	1.95 ms, 1.02 ms

The time interval can be measured with the use of the clock pulses from the internal oscillator of the microcontroller's timer. It should be noted that the selection of the measured time interval should be larger or equal to the desired CET interval T_{el}, or it should be longer than T_{el}, which contains an integer numbers of the encoder pulse period. However, the working principle of the CET measurement comprises of two counters as shown in Fig 2.9, in which one is basically for counting encoder pulses while the other is focused on measuring the time interval, which are simultaneously started at the same time by the positive going rising edge at the encoder pulse. The counter of the encoder pulses is stopped by the first positive going rising edge of the encoder pulse which occurs after the CET interval T_{el} has elapsed [1], [2]. As such the content of the encoder pulses counter is then C_p which is the number of the measured encoder pulses. While the content of the timer counter is denoted as C_t, which signifies the number of the measured clock pulses. Where P signifies the number of the encoder disc markers [2]. Consequently, from the correlation of the equations above, the rotational speed can be obtained as:

$$\omega = \frac{\mathbf{60C_p}}{\mathbf{PT_0C_t}} = \frac{\mathbf{60f_0C_p}}{\mathbf{PC_t}} \tag{2.2}$$

However, by analyzing the measurement process, it is obvious that the measurement

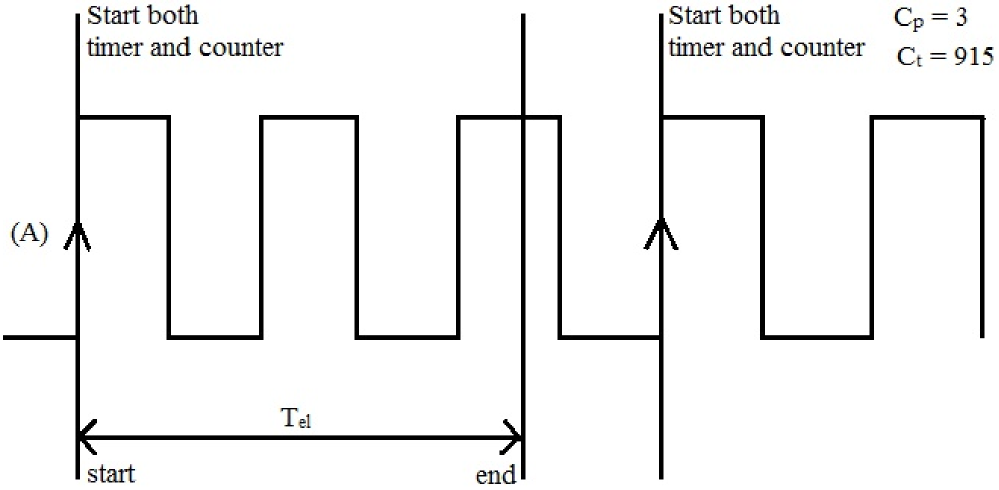


Figure 2.9. A: Method of CET Rotational Speed Measurement

or response time, Δ_t depends on the changes of the measurement speed, with variations in response time less than 1:2, in as much the duration of an encoder pulse period is less than the time T_{el}. It should be noted that the duration of the encoder pulses period increases relatively at very low rotational speeds and eventually exceeds the normal variations in response time of $2T_{el}$. Hence, at low rotational speeds the CET method is similar to the method of pulse time measurement at the encoder pulses period since the measurement of the time interval between the adjacent pulses is determined [1], [2]. Whenever the speed attains zero value then the encoder pulse duration and the obtainable response time go to infinity. Since, most industrial applications and practical control system require a response within a certain time frame, then a maximum response time, T_{max} of the control system should be defined, along which a rising pulse edge should occur as long as the counters have been initialized. If the encoder pulse

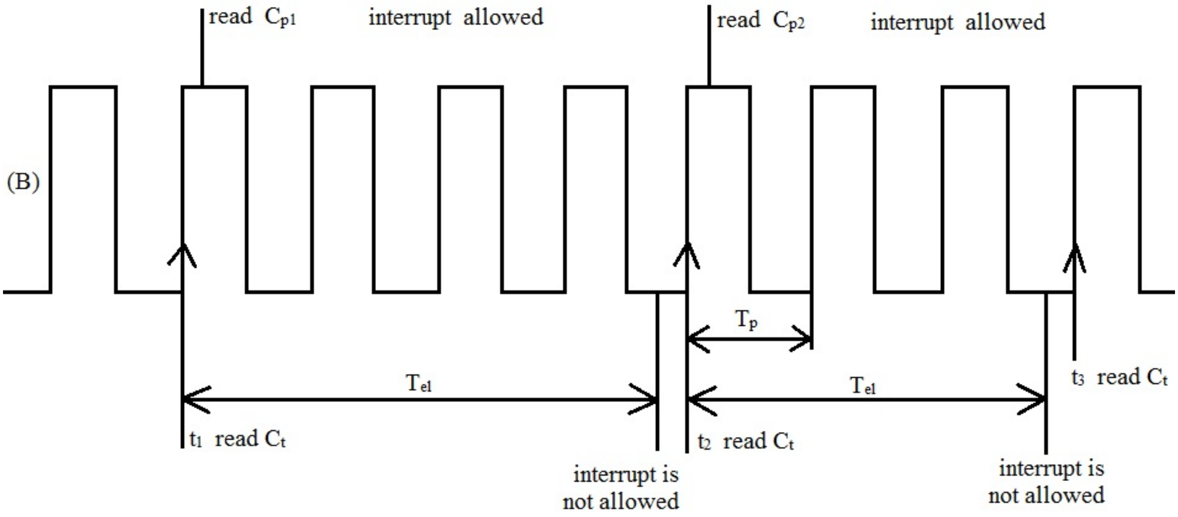


Figure 2.10. B: Method of CET Rotational Speed Measurement

duration goes beyond the maximum response time, T_{max} then speed signal will be set to be zero. Therefore the speed at which the signal measured is set zero is termed as the "minimum measurable speed", n_{min} along the predefined maximum response time. Hence, the property of minimum measurable speed is associated with digital speed measurement control systems [2].

3. HARDWARE IMPLEMENTATION AND SOFTWARE DEVELOPMENT

3.1. Introduction

The methodology involved in this research is the CET method. However, this method is based on frequency measurement method. A frequency measurement can be achieved by both pulse counting and time measurement during a predefined constant sampling time. The content of the free-running timer is updated to a program memory location upon each successive rising edge of an input pulse. The CET method is suitable for real-time speed measurement with the incorporation of microcontrollers and frequency counters. The overall project comprise of hardware design and software implementation part that involve MATLAB (Matrix Laboratory) simulations, C computer program using MikroC compiler.

3.2. Hardware Implementation

This section describes the hardware components and how to achieve a specific hardware design to be implemented. Thus, the hardware components include PIC16F887 Microcontroller, Signal generator, 8-Channel Multiplexer, 8-Bit Binary Counter, Easy PIC development board and an LCD. However, PC (Personal Computer) and Oscilloscope are used to monitor the performances of the system, while the power supply voltage is fixed at 5V DC. The complete circuit diagram of the overall system is shown in Fig 3.1. While the functional block diagram of the overall system describes the diagrammatic view of the wide range digital motor speed measurement as shown in Fig 3.2.

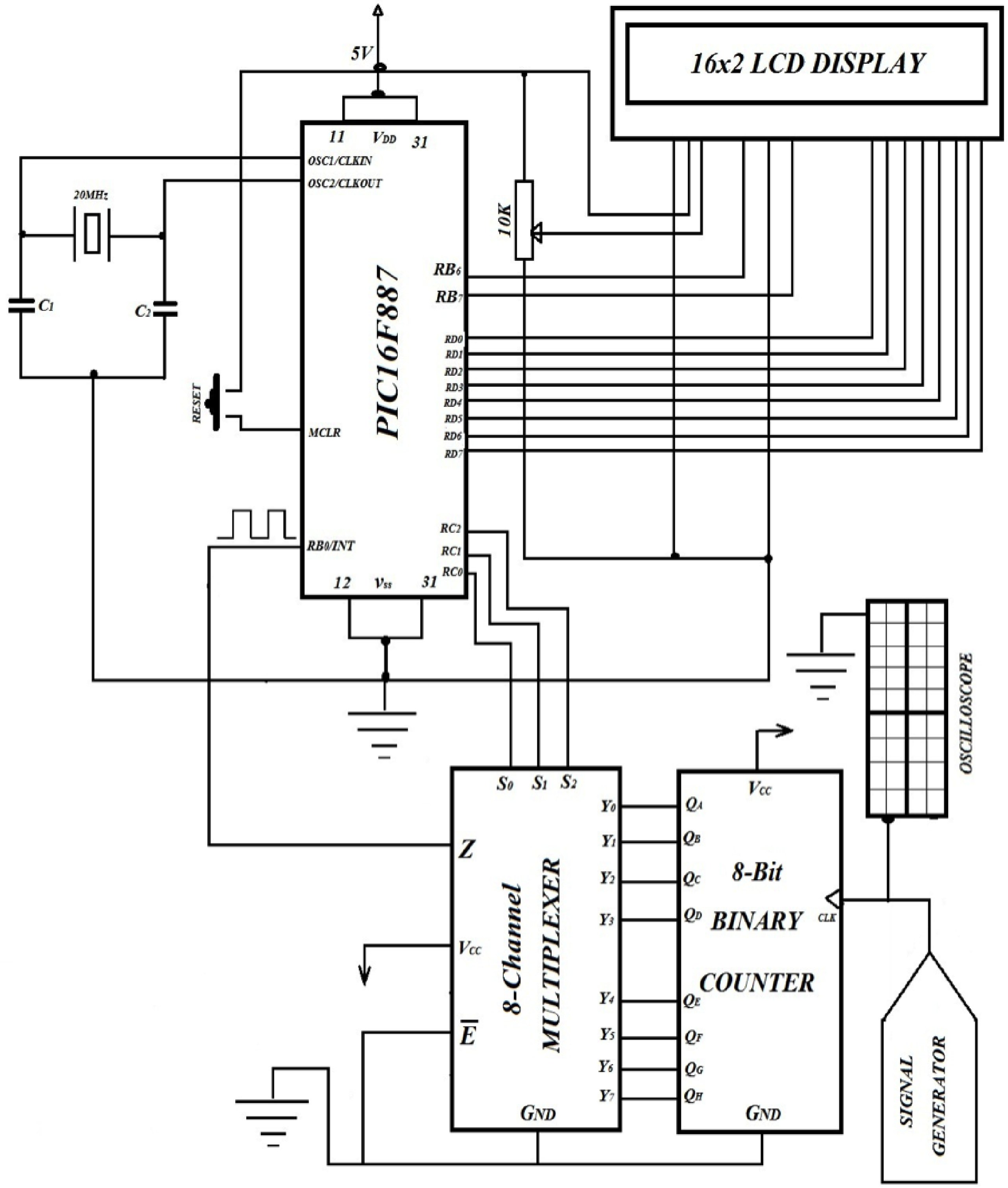


Figure 3.1. Complete Circuit Diagram

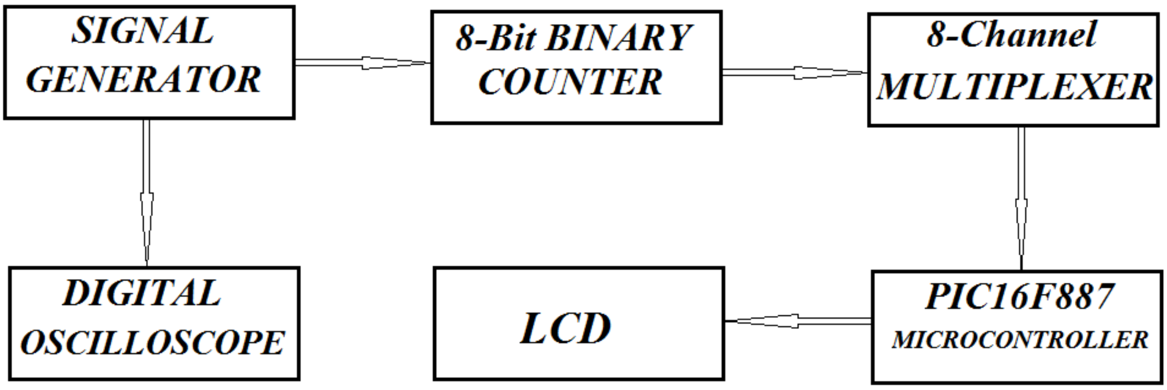


Figure 3.2. Functional Block Diagram

3.2.1. Signal Generator

A Signal generator is a key part of this research. This is an electronic device that is capable of producing repeating or non-repeating electronic signals. As related to industrial and research applications, a signal generator is required to test and analyze experimental set-ups using basic waveforms that include sine, square and triangular waves [31]. In this research work, when the frequency of the signal generator is set at a particular speed, its corresponding digital pulses are been generated simultaneously. Indeed, the digital output can be observed through the digital oscilloscope which displays the corresponding output pulses waveforms. Furthermore, the output pulses are transmitted to the 8-bit binary counter via the clock, while the output of the counter is sent to the input of the 8-Channel multiplexer, and the three digital select inputs controls the ranges of the speed measured which are connected to PortC as output of the microcontroller. Then the output of the multiplexer is sent to input port RB0 of the microcontroller and thereby calculating the pulse counts via the interrupt service routine of the software programming platform and increasing the counts upon receiving a positive going rising edge trigger. Hence, the computed number of speed measured can be displayed on the LCD. With the use of the signal generator, the speed or frequency can easily be varied, making the system to be more flexible. Also, using the function generator gives an excellent level of accuracy with lowest amount of error when compared to using an encoder incorporated to a motor, also the response time is faster when using the signal generator. The digital clock pulses from the signal generator is shown in Fig 3.3. However, the output pulses from the signal generator will be channeled to the 8-bit binary counter where multiple digital output will be generated from the counter.

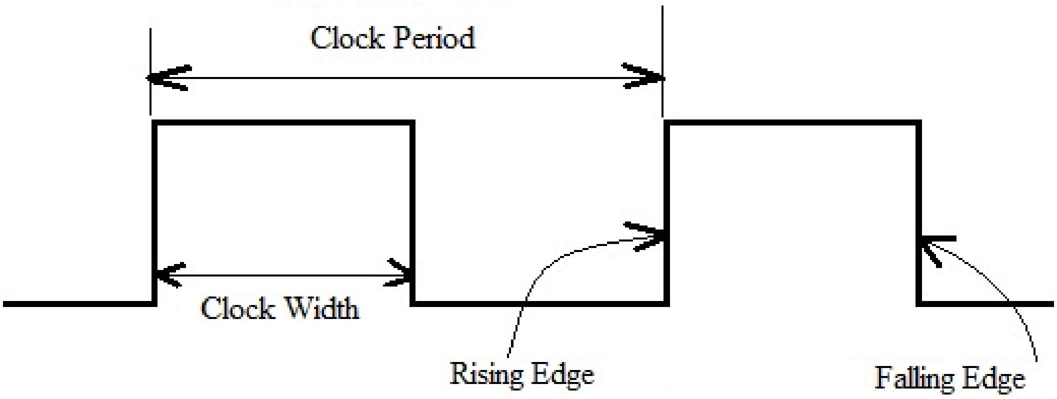


Figure 3.3. Digital Clock Pulses

3.2.2. Binary Counter

The 8-bit binary counter comprises of dual 4-bit ripple carry binary counters, which can easily be interconnected to operate as a divide by 256 counter. Also, each of the two 4-bit binary counters is increased via the clock input on a HIGH to LOW transition i.e. negative edge, with each having a separate clear input. However, when clear input pin is set HIGH, then all four bits of each binary counter are consequently set to a low status. Thus, this enhances count truncation and supports the implementation of divide by N counter configurations [24].

Meanwhile, based on this research, the binary counter is initialized through pulses generated from the signal generator, while the output of the counter is channeled to the input of the 8-channel multiplexer. With reference to the clock input from the signal generator, the binary counter will generate multiple outputs by producing toggles of the input clock. Therefore, the least significant output bit will toggle at 1/2 the rate of the clock input from the signal generator. Consequently, the next output bit will also toggle at 1/4 the rate of the clock input, while the third bit will also toggle at 1/8 the rate of the clock input, and so on. Since it is an 8-bit binary counter (having 8 outputs), the division will continue until it reaches a divide by 256 of the clock input. The logic timing waveforms is shown in Fig 3.4 which will be interfaced to the input of the multiplexer.

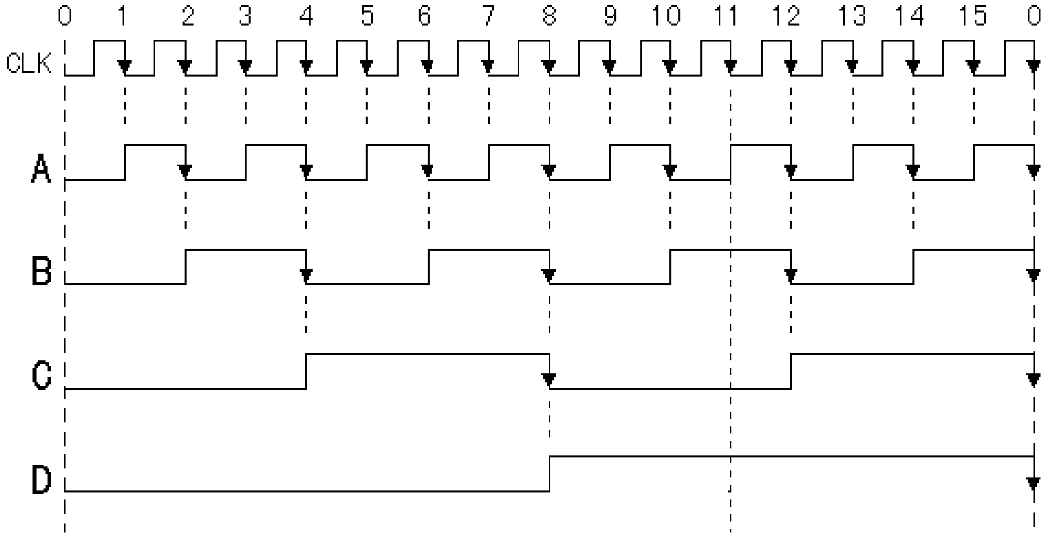


Figure 3.4. Logic Timing Waveforms

3.2.3. Multiplexer

A multiplexer is a device that selects one of many analog or digital input signals and transfers the selected input signal into a single line. For a multiplexer of degree 2^n inputs have n select lines, which can be used to control or select the input line to be channeled to the output. The 8-channel analog multiplexer comprises of three digital select inputs from S_0 to S_2, eight independent outputs/inputs from Y_0 to Y_7, an active LOW enable input, E and a common input/output (Z). However, setting E to LOW, then one of the eight switches will be selected by S_0 to S_2. While setting E to HIGH, renders all switches to high-impedance OFF-state, regardless of S_0 toS_2 [32]. The multiplexer gets input from the output of the binary counter while the three digital select pins selects the corresponding value based on the microcontroller and then sends it to the output of the multiplexer. The output of the multiplexer is then interfaced with the microcontroller.

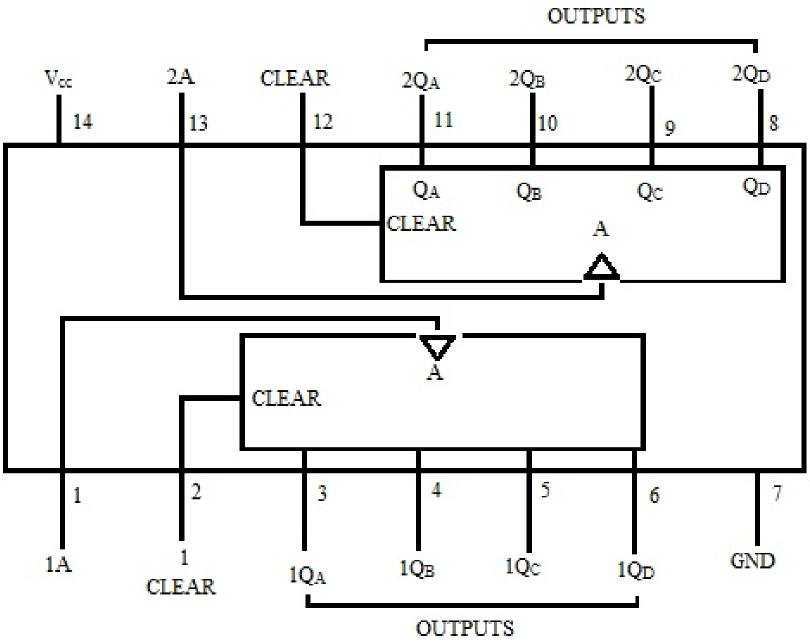


Figure 3.5. Binary Counter Pin Connections

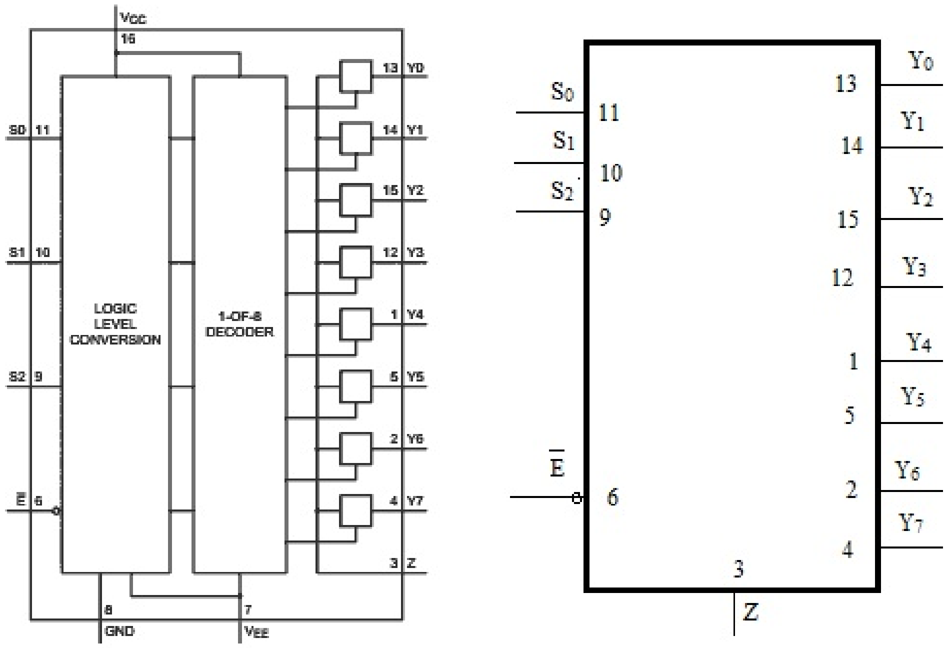
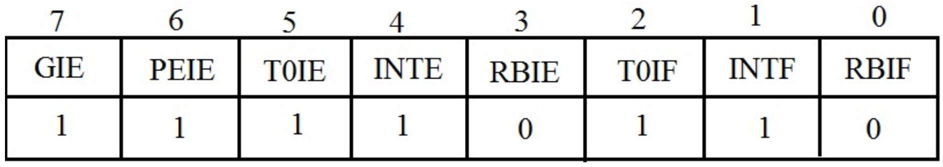


Figure 3.6. Functional Diagram and Logic Symbol of a Multiplexer [32]

3.2.4. PIC16F887 Microcontrollers

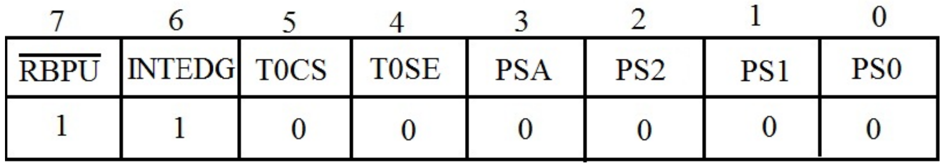
In order to run the real-time control algorithm and generate pulses from interrupt service routine, a PIC16F887 microcontroller is used to serve as the overall brain of the system. However, based on this research the microcontroller is interfaced with a 16x2 LCD module using Mikro C compiler, which displays the measured speed output. Also, the PIC16F887 microcontroller has a RAM (Random Access Memory) capacity of 368 bytes which is sufficient for the system, while the CPU (Central Processing Unit) speed is at 5 MIPS (Million Instructions per Second) [17]. The interface between the 16x2 LCD module and PIC16F887 microcontroller is in 4-bit mode, in which data and commands are sent through the data lines. Based on the configuration of the microcontrollers, they are two options for the pulses which can either be rising or falling edges. However in this research, the rising edge is used via the configuration of the registers and control bits.

Based on the requirements, some registers and control bits configurations were used such as INTCON (Interrupt Control) registers, Option registers and ANSEL/ANSELH (Analogue Select) registers . Based on the configurations of the registers, the INTCON register basically controls the enable and flag bits for the timer0 register overflow and external INT pin interrupts. The bits configurations as used in this research is shown in Fig 3.7. While the Option register is a readable/writable register which basically comprises of some control bits used for the configuration of the timer0 and external interrupts. The bits configurations as used in this research is shown in Fig 3.8. The ANSEL register is basically used to configure the Input mode of an input/output pin to analog. However, configuring the appropriate ANSEL bit high will render all digital reads on the pin to be read as ‘0’. Then this allows analog functions on the pin to operate properly, rendering the analog pins to be configured as digital inputs/outputs. While the ANSELH register is also used to configure the input mode of an input/output pin to analog. Meanwhile, configuring the proper ANSELH bit high will make all digital reads on the pin to be read as ‘0’ and thus making analog functions on the pin to function correctly as digital inputs/outputs.



7	6	5	4	3	2	1	0
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
1	1	1	1	0	1	1	0

Figure 3.7. Interrupt Control Register



7	6	5	4	3	2	1	0
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
1	1	0	0	0	0	0	0

Figure 3.8. Option Register

3.2.5. Frequency Counter

Basically, a frequency counter determines or measures frequency by counting the number of rising or falling edges of an input signal across an allocated period of time. With regards to this research work, the frequency counter is based on PIC (Peripheral Interface Controller) microcontrollers, which uses a 16x2 LCD to display the number pulse counts generated by an external interrupt service routine using a digital signal generator and PIC timer0 (TMR0) as an internal interrupt service routine to determine the total elapsed time. However, it uses TMR0 in 8-bit counter mode to count the input signal upon seeing rising edges and timer overflows of the counter are cumulated to obtain the total counts in multiples of 256, since timer0 is in 8-bit mode, the sampling period has to be short enough so as to accommodate less than 256 cycles per period across the maximum input frequency. The crystal oscillator used is basically a typical crystal and two capacitors that are connected to the PIC microcontroller oscillator pin at OSC_1 and OSC_2 (Oscillator). Also the code uses the built in LCD driver modules that are automatically included by the mikroC compiler, although automatic include file is unusual but it is very efficient in mikroC interface. The number of rising edges across a measurement period from an input signal is shown in Fig 3.9. The performances and the output waveforms can be monitored using the digital oscilloscope and an LCD.

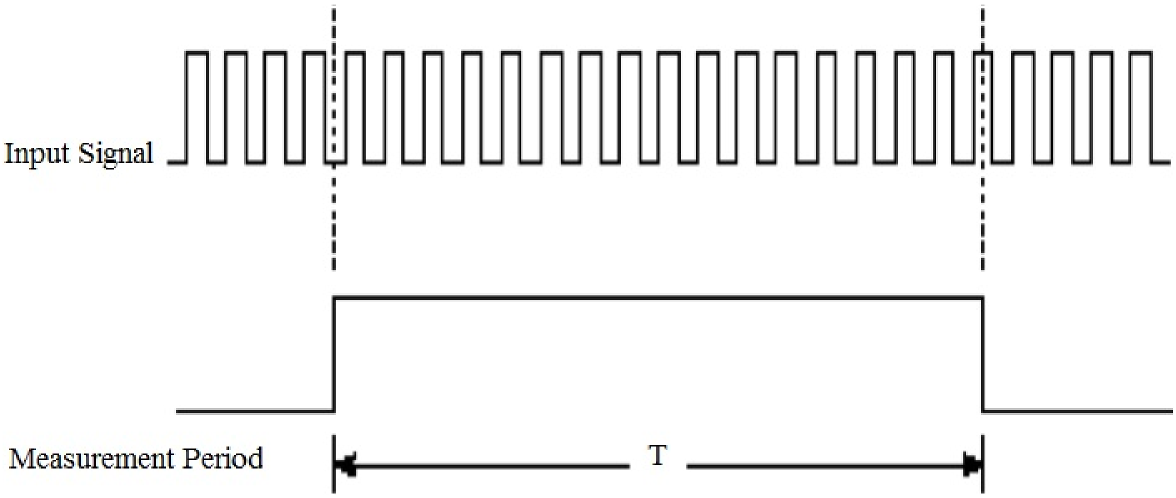


Figure 3.9. Pulse Generation

3.2.6. Digital Oscilloscope

An oscilloscope is specifically a graph-displaying equipment of a given electrical signal, the displayed graph illustrates the variation of electrical signals over time. However, digital oscilloscope functions with discrete binary numbers that may represent voltage samples. Hence, a digital oscilloscope samples the waveform and uses an analog-to-digital converter to convert the voltage being measured into its corresponding digital data and then displaying the information for the waveform unto the screen. [33] However, based on this research work, the digital oscilloscope is used to display the output of digital pulses from the signal generator. Meanwhile, with the incorporation of the digital oscilloscope it will be easier to visualize and analyze the generated running pulses. The block diagram of a digital oscilloscope is shown in Fig 3.10.

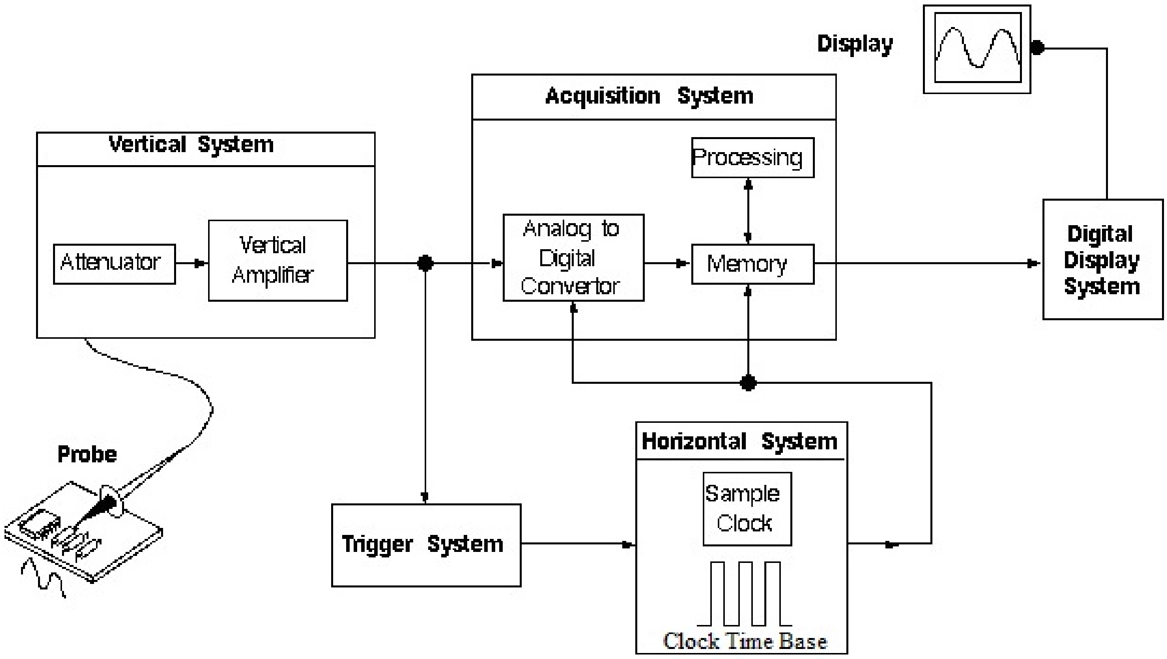


Figure 3.10. Block Diagram of a Digital Oscilloscope [33]

3.3. Software Development

Before implementing the system into its actual hardware, the software part has to be fully implemented. However, based on this research work, it uses a C program using the MikroC compiler with its debugger. Meanwhile, the entire development cycle of the software design is classified into two main parts: initialization module and the interrupt service routine module in which the total time elapsed can be determined. However, the initialization module describes and declares the software variables, constants and specific registers used in the program. The software constants and uninitialized variables can also be declared in this part or perhaps at the initial stage of each module. Although, it is a good programming practice to test each module before building up the main program. With regards to developing or testing a software design, a specific development cycle is followed comprising of some stages to be implemented. Incase there is an error or bug, the program is re-edited and the entire cycle is repeated. A typical development cycle is shown in Fig 3.11.

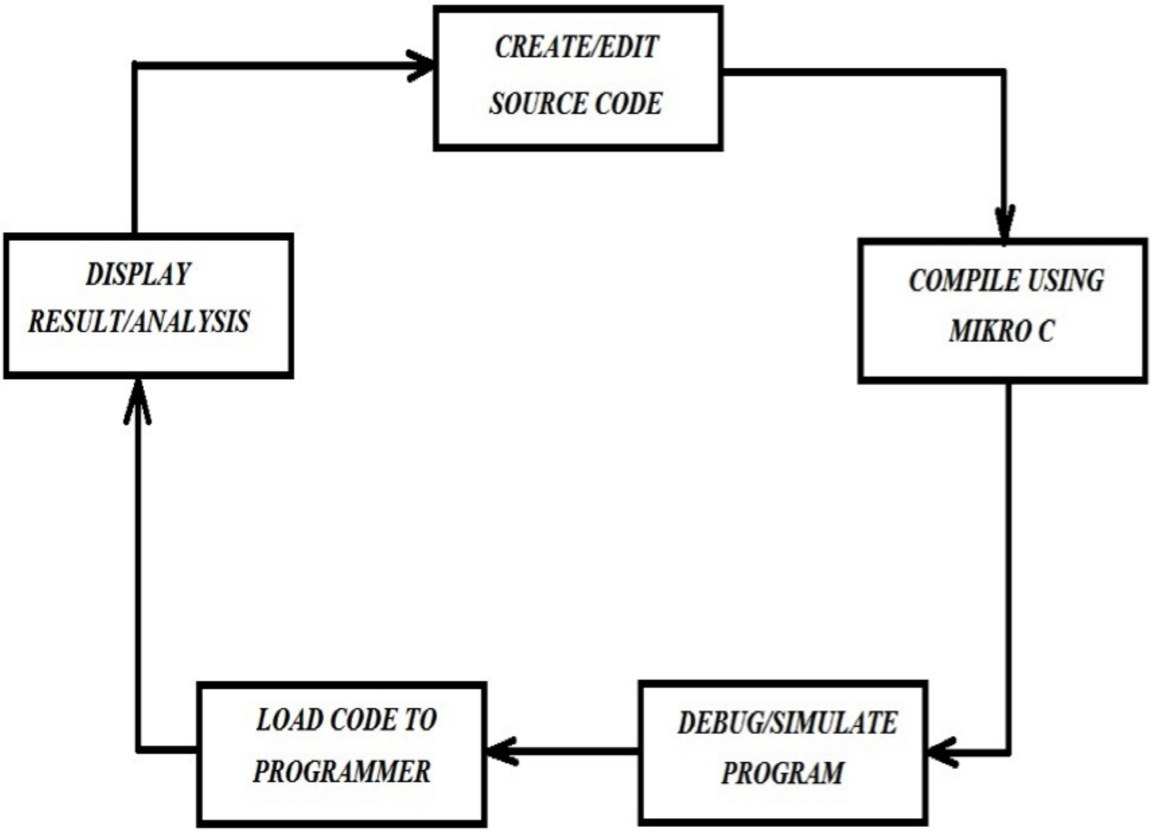


Figure 3.11. Software Development Cycle

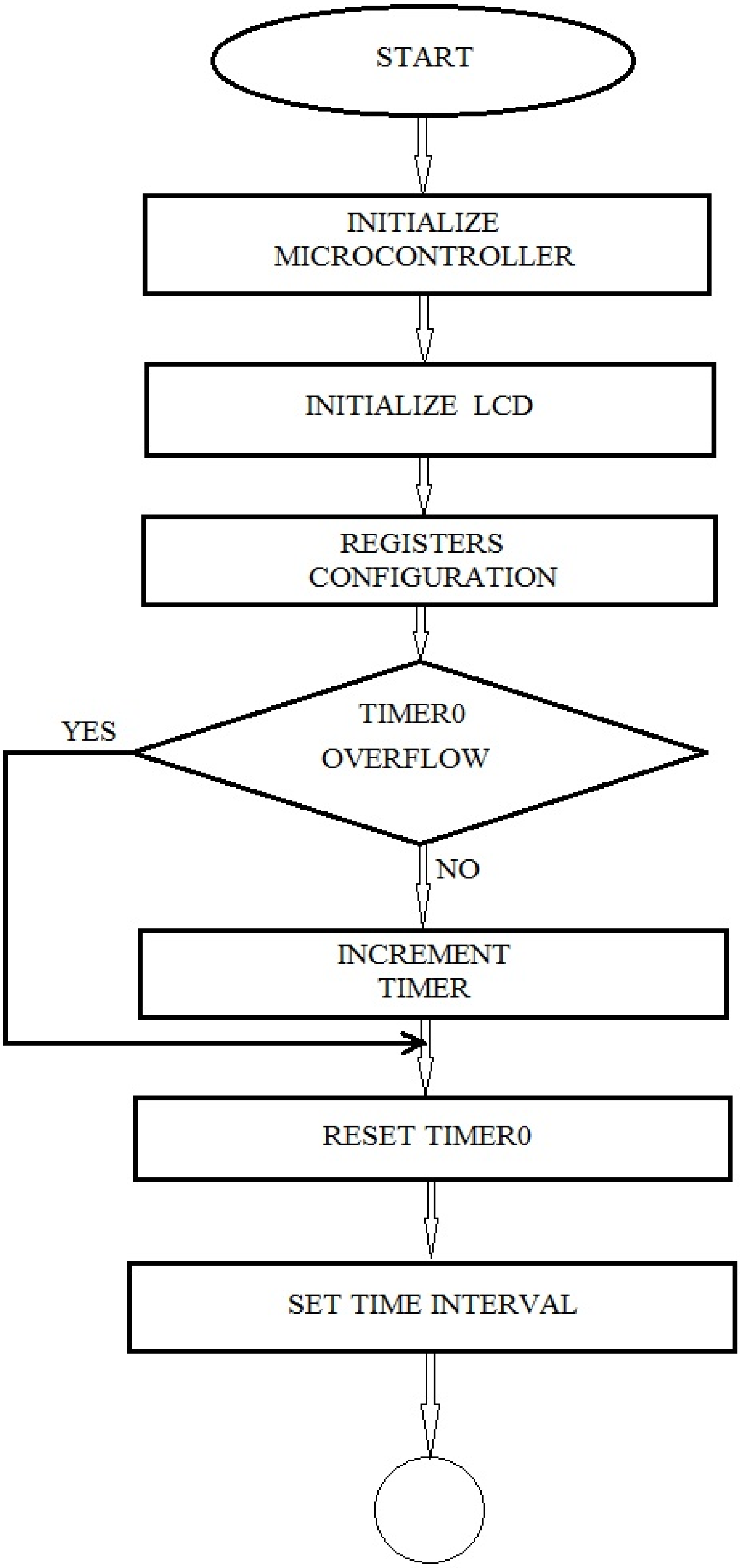


Figure 3.12. Program Flowchart

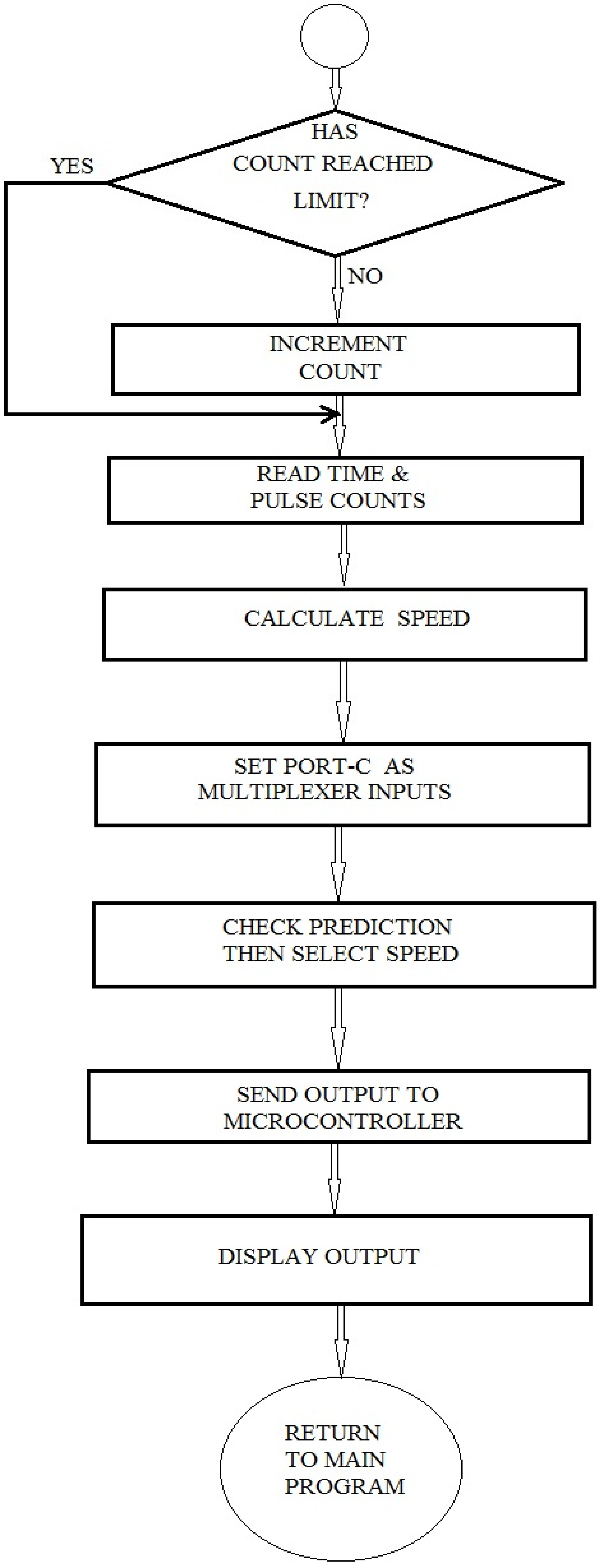


Figure 3.13. Program Flowchart

The initialization of the PIC16F887 microcontroller is a key factor in this program. Meanwhile, the initialization processes involves registers and control bits configurations such as INTCON registers, option registers, ANSEL/ANSELH registers and the inputs/outputs ports configurations registers. The INTCON registers basically control the enable and flag bits for the timer0 register overflow and external INT pin interrupts. While the option register is a readable/writable register, which basically comprises of some control bits used for the configuration of the timer0 and external interrupts. The ANSEL/ANSELH registers are basically used to configure the Input mode of an input/output pin to analog. However, configuring the appropriate ANSEL/ANSELH bit high will render all digital reads on the pin to be read as '0'. Then this allows analog functions on the pin to operate properly, rendering the analog pins to be configured as digital inputs/outputs.

The numbers of input/output pins that enables connections to the peripheral modules is of paramount importance. However, there are 35 general-purpose input/output pins associated with the PIC16F887 microcontrollers, which is adequate for this program. Also, in order to synchronize the operation of input/output ports with the internal organization of the microcontrollers, they are similar registers which are grouped into five ports denoted by letters A, B, C, D and E. Furthermore, every input/output port is basically accompanied by the corresponding TRIS (Tri Sate) register, these include TRISA, TRISB, TRISC etc. which actually determines the performance, but not the contents of the input/output port bits. Meanwhile, by clearing any bit of the TRIS register (bit=0), then the corresponding port pin is configured as an output port. In a similar fashion, by setting any bit of the TRIS register (bit=1), the corresponding port pin is configured as an input.

LCD initialization is also an important stage in this program. Before using the LCD for display purpose, LCD has to be initialized by sending some set of commands to initialize the LCD. In mikroC for PIC, it is sufficient to write only one function to perform all operations for LCD initialization. The LCD is automatically cleared when powered up, before it gets ready for operation. However, the LCD used in this program is the 16x2 LCD, the LCD module connections is synchronized with portD. Since

mikroC for PIC is being used in this program, then the LCD initialization command will be;
$Lcd\_Init(); //InitializeLCD$

The registers and control bits configurations are very important in this program. With respect to the system, some registers and control bits configurations were used such as INTCON registers, Option registers and ANSEL/ANSELH registers. Based on the configurations of the registers, the INTCON register basically controls the enable and flag bits for the timer0 register overflow and external INT pin interrupts. While the Option register is a readable/writable register that basically comprises of some control bits used for the configuration of the timer0 and external interrupts. The ANSEL register is basically used to configure the input mode of an input/output pin to analog. However, configuring the appropriate ANSEL bit high will render all digital reads on the pin to be read as '0'. Then this allows analog functions on the pin to operate properly, rendering the analog pins to be configured as digital inputs/outputs. While the ANSELH register is also used to configure the input mode of an input/output pin to analog. Meanwhile, configuring the proper ANSELH bit high will make all digital reads on the pin to be read as '0' and thus making analog functions on the pin to function correctly as digital inputs/outputs. However, improper configurations of registers and control bits can lead to malfunction of the program.

Timer0 has a register called TMR0 register, which has 8 bits of size. The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h, maximum value that can be assigned to this register is 255. We can read the value of the register TMR0 and write into. Based on this research, the Timer0 register increments on every rising edge of the T0CKI (Timer0 Clock) pin. Indeed, the incremental edge is controlled by the T0SE (Timer0 Source Edge) bit configuration of the OPTION register. Timer0 generates an interrupt when the TMR0 register overflows. The T0IF (Timer0 Interrupt Flag) interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. To avoid malfunctioning of the system, the T0IF bit must be cleared in the program. The Timer0 interrupt enable bit is the T0IE (Timer0 Interrupt Enable) bit of the

INTCON register that is set to ‘1‘.

However, from the initialized counters, the pulse counts and time counts can be obtained within the time interval. Based on the counts, the speed can be calculated. Also, the microcontroller is interfaced with the digital multiplexer, where portC is configured as input to the multiplexer, which serves as the select pins that controls the selection processes of the calculated speed based on the prediction algorithms. Then the output of the multiplexer is sent to the microcontroller where the output will be sent to the LCD for display. Meanwhile, the whole processes continue in a loop.

3.3.1. Interrupts Processes Overview

The microcontroller will execute the interrupt routine process instead of the void main program that is running whenever it receives an interrupt flag. The interrupt service routine is associated with both internal and external interrupts in the program. Indeed, the setting of pin RB0/INT to execute an external interrupt routine is a key point, the external interrupt is set high by an input on RB0 as the pulses go from 0 to 1 (due to register configuration), which is called the rising edge. This sets the interrupt flag to go high, hence if the interrupt is enabled, the microcontroller automatically goes to the interrupt subroutine. Timer0 is perhaps mostly used timer in various microcontrollers. However, it has the ability to function as a simple counter and overflow upon reaching its maximum count, since it is an 8-bit register (counts from 0 to 255), generating an internal interrupt and then repeat the count again. Based on this research work, the timer0 works simultaneously with the RB0/INT pin, meanwhile as the pulses are been generated, the timer is also started and it counts up to its maximum value of 255, then it overflows and increments the counts, as such the total time elapsed can be determined.

3.3.2. Timer0 Interrupt

Based on this research, the Timer0 register increments on every rising edge of the T0CKI pin. Indeed, the incremental edge is controlled by the T0SE bit configuration of the OPTION register. Timer0 generates an interrupt when the TMR0 register overflows from 0 to 255. The T0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. To avoid malfunctioning of the system, the T0IF bit must be cleared in the program. The Timer0 interrupt enable bit is the T0IE bit of the INTCON register which is set to '1'. The digital pulse counts from timer0 overflow is shown in Fig 3.14.

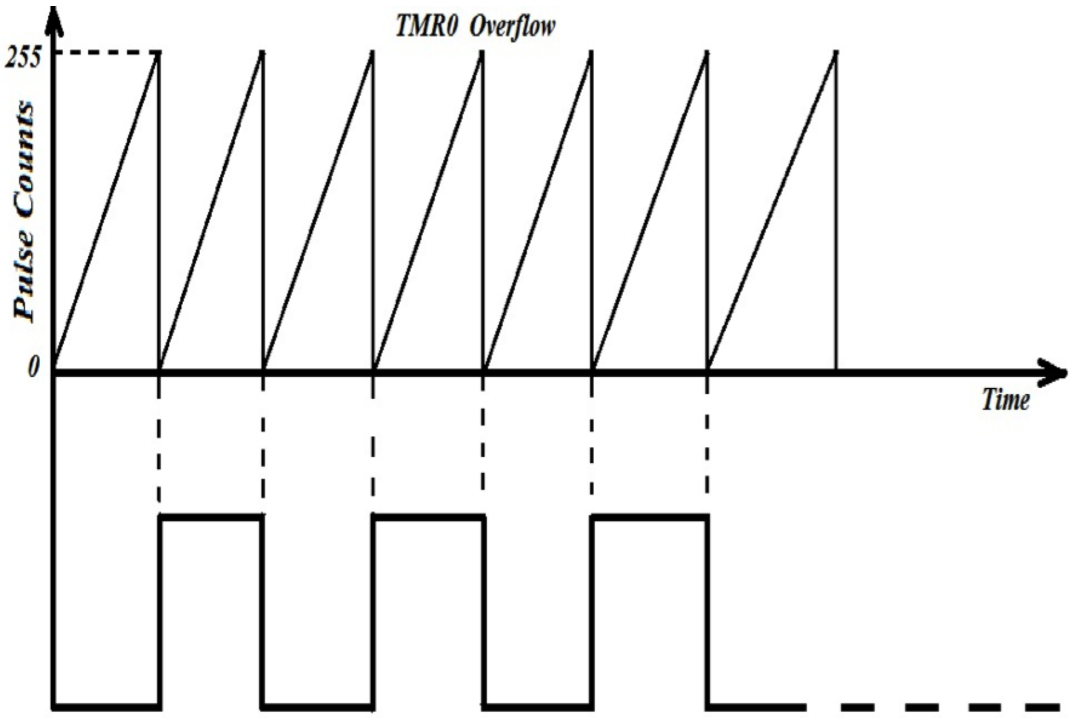


Figure 3.14. Pulse Counts from Timer0 Overflow

3.3.3. Speed Calculation in RPM

With reference to pulse counts (rising edges) and the total time elapsed (timer0 overflow counts), the desired speed can be determined in RPM. However, some variables are also declared as follow:

I_2=Timer0 $Overflow\ Counts$
I_5=$Time\ Elapsed$
I_6=$Pulse\ Counts$
I_8=$Speed(RevolutionPerSecond)$
I_9=$Speed(RevolutionPerMinute)$
m_1=$Encoder\ Marks$
k_1=$Frequency\ Ratio\ Factor$
f_1=$Crystal\ Frequency$

According to TMR0 interrupt flag, which increments by 1 up to maximum of 256 and then overflows,

I_2=I_2+1

But if:

(I_2=10000)

Then;

I_3=I_3+1

I_4=($I_3 \times$ 10000)+I_2

Hence, the Total Time elapsed can be calculated as:

I_5=($I_4 \times$ 256)+Tmr_0

But if:

($I_5 \leq$ 2000000)

Then

I_6=I_6+1 ($Number\ of\ Rising\ Edges$)

Else if:

(2000000 $\leq I_5 \leq$ 4000000)

Therefore, the Speed can be calculated as:

$I_8 = (I_6 \times f_1) \,/\, (I_5 \times m_1 \times 4)$

Hence, the RPM Speed can be calculated as:

$I_9 = I_8 \times 60.$

$I_{10} = I_9 \times k_1$

Otherwise if:

$(I_5 > 4000000)$

Then the Speed will be zero, i.e

$I_9 = 0.$

3.4. Linear Prediction Algorithms

Predicting or estimating some samples given some other samples is one of the most frequently occurring problems in real life or time series analysis. However, this algorithm is a MATLAB based implementation using the Weighted Least Squares Error Algorithm (WLSE) for the Simulation and result Analysis [34]. Obviously, the target is to obtain predicted values based on given actual variables, and yet minimizing the error between the predicted values and the actual values. In the WLSE Algorithm, the weighted sum of the error is defined and minimized for a specific set of weights and having coefficients at each instant of time, $t = k$, and then applying those set of coefficients to predict or estimate the value at the next instant of time $t = k + 1$. Meanwhile, the coefficients will be subsequently changed so as to achieve the minimum WLSE criterion. i.e.

$$\beta(\mathbf{a}) = \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{i=1}}^{\mathbf{k}} \alpha_{\mathbf{i}} \left[\mathbf{a_M^T u(i) - x(i)}\right]^{\mathbf{2}} \tag{3.1}$$

$$\mathbf{u(k) = [\; x(k-1)\; x(k-2)\; x(k-3)\; ...\; x(k-M)\;]^T} \tag{3.2}$$

$$\mathbf{\hat{x}(k) = a_M^T(k-1)u(k)}. \tag{3.3}$$

Where α_i are the weights and $u(k)$ is the input to the filter at time, $t = k$

$\beta(a)$ is the minimum WLSE criterion.

a_M^T is the transpose of coefficient vector.

u(i) is the new input to the filter.

x(i) is the current output.

$\hat{x}(k)$ is the predicted output.

P is a matrix of size M x M.

The weighting should de-emphasize the previous data points because in the implementation of the algorithm, memory space will be limited if all previous data samples are stored, and perhaps the memory limit might be exceeded. The algorithms are outlined below: [34]

$For\ k = 2\ to\ \infty\ and\ \alpha = 0.99$

1. Calculate the current predicted output

$$\mathbf{\hat{x}(k) = a_M^T(k-1)u(k)} \tag{3.4}$$

2. Update the coefficient vector

$$\mathbf{a_M(k) = a_M(k-1)} + \frac{\boldsymbol{P}\mathbf{(k-1)u(k)}}{\alpha + \mathbf{u^T(k)P(k-1)u(k)}}\,[\mathbf{x(k) - \hat{x}(k)}] \tag{3.5}$$

3. Update the P matrix

$$\mathbf{P(k)} = \frac{\mathbf{1}}{\alpha}\left\{\mathbf{P(k-1)} - \frac{\mathbf{P(k-1)u(k)u^T(k)P(k-1)}}{\alpha + \mathbf{u^T(k)P(k-1)u(k)}}\right\} \tag{3.6}$$

Note that the algorithm begins with $a_M = [1, 0, 0, ..., 0]^T$ and P (1) = I, which signifies the M × M Identity Matrix [34]. As such, the subsequent samples or variables of the actual samples can be iteratively predicted or estimated. Hence, this procedure is employed in the implementation of the linear prediction algorithm and thus, this algorithm does not require the calculation of the autocorrelation function of the actual samples [34]. During this research, some limitations were encountered while using the linear prediction algorithm, which is related to the programming aspect. Meanwhile, the linear prediction algorithm is not suitable for higher mathematical calculations involving computer programming, as it leads to malfunction of the program. Hence, the algorithm was modified by implementing the Simple Prediction algorithm to solve these limitations.

3.4.1. Simple Prediction Algorithms

The simple prediction algorithms is introduced to overcome the limitations of the linear prediction algorithms. However, based on the minimum and maximum values of the motor speed obtained, the threshold and optimum values of motor RPM speed using frequency divider parameters can be calculated using MATLAB simulations. Although, this estimation has less precision when compared to the linear prediction algorithms. However, the speed prediction is initialised at 26 rpm because the minimum measurable speed is 25.5 rpm, while the maximum value is 625000 rpm.
The basic steps involved in the simple prediction algorithms are outlined below:

For frequency values, f = 1, 2, 4, 8, 16, 32, 64 and 128.

1. Obtain minimum, $X_{min}(f)$ and maximum, $X_{max}(f)$ values of speed with corresponding frequency values.

2. Set an arbitrary value between $X_{min}(f)$ and $X_{max}(f)$ as X

3. Determine the difference between X and $X_{min}(f)$ as $S_1(f)$ = X - $X_{min}(f)$

4. Determine the difference between $X_{max}(f)$ and X as $S_2(f) = X_{max}(f)$ - X

5. If S_1 or S_2 is less than or equal to zero, skip.

6. Obtain the minimum value between $S_1(f)$ and $S_2(f)$ as $S_{min}(f)$.

7. Repeat procedures for other values of $X_{min}(f)$ and $X_{max}(f)$ with corresponding frequencies, f.

8. Determine the maximum value of $S_{min}(f)$ as $S_{max}(f)$.

4. SIMULATIONS AND DISCUSSION OF RESULTS

4.1. Introduction

In this chapter, some simulations were carried out on the system by applying prediction algorithms using MATLAB tool to obtain simulations results. Also, the system's estimated algorithms were incorporated into real hardware via EasyPIC development board using MikroC compiler to obtain experimental results. Also, this chapter discusses about the results obtained from the simple prediction algorithms by calculating the threshold and optimum values of the speed. Furhtermore, it also outlines the comparisons between the theoretical and experimental results, it also discusses about the degree of errors associated with the results obtained. However, some of the limitations and sources of errors will also be discussed as related to this research.

4.2. Results of Simple Prediction Algorithms

In this part the simulations results obtained from the simple prediction algorithms using MATLAB is presented. The simple prediction algorithms implemented makes subsequent prediction computations and updates the next values from the inputs with respect to the parameters of the frequency divider. Meanwhile, the simple prediction algorithm is basically used to obtain the threshold and optimum values within a speed range, which is corresponding to the frequency division parameters. However, the threshold and optimum values simply relates to the minimum and maximum speed range within the frequency divisions.

Based on the prediction parameters, the simple prediction algorithm is initialised at the value of 26 rpm because the minimum measurable speed value is 25.5 rpm, while the maximum value is 625000 rpm. Based on the modified simple prediction algorithms, the threshold and optimum speed values were obtained with their corresponding frequency divider parameters. However, the simulations show a stairs-like plot indicating the frequency divider effects on the prediction algorithms for speed measurement across wide range. Thus, it can be observed from the plot that the speed threshold and optimum values intervals increases as the frequency division ratio increases. Since the frequency divider has a power-of-2 integer division, then the output will be a toggle of the input value. In otherwords, the least significant output bit toggles at 1/2 the rate of the input value, while the next toggles at 1/4 the rate of the input and the third bit toggles at 1/8 of its input value. In a similar fashion, the subsequent bits toggle at 1/16, 1/32, 1/64 and 1/128 the rates of their respectives input values. Therefore, the increment in ranges of speed is due to the frequency division parameters. Also, it can be observed that, for the speed range of 26-2466, the corresponding frequency is 1. While for the 2467-4933, the frequency is 2 and for 4934-9867, the corresponding frequency is 4 and so on.

Based on the simple prediction algorithms, the results for the threshold and optimum values for the RPM speed were obtained against their corresponding frequency divider parameters. Table 4.1 shows the threshold and optimum values of the speed obtained.

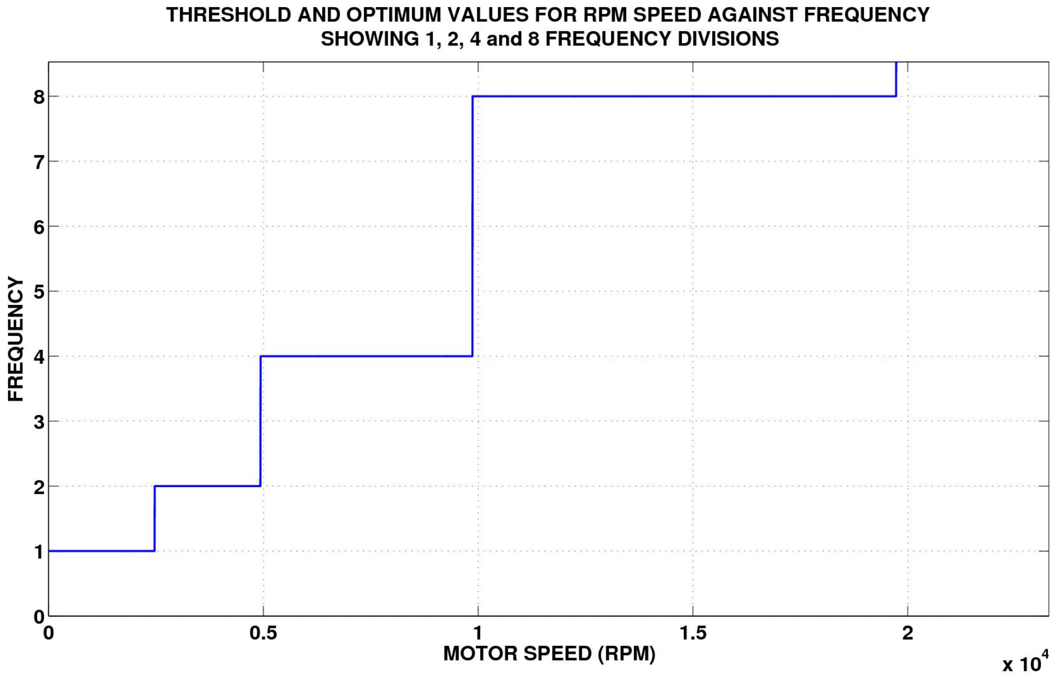


Figure 4.1. Values for Speed showing 1, 2, 4 and 8 Frequency Divisions

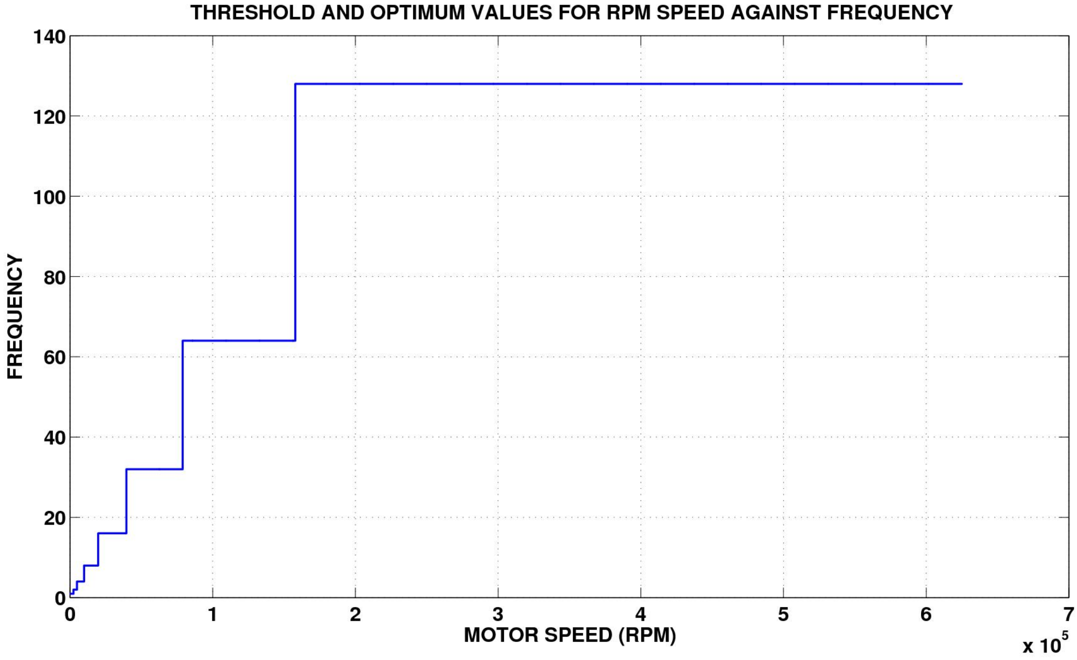


Figure 4.2. Threshold and Optimum Values for RPM Speed against Frequency

Table 4.1. Threshold and Optimum Values of RPM Speed

FREQUENCY DIVIDER(Hz)	**THRESHOLD/OPTIMUM VALUES(RPM)**
1	26– 2466
2	2467– 4933
4	4934– 9867
8	9868– 19735
16	19736– 39470
32	39471– 78940
64	78941– 157880
128	157881-625000

4.3. Experimental Results

Based on the simple prediction algorithms, some experimental results were obtained from the system. However, the speed values were obtained with their corresponding frequency parameters. As the frequency of the signal generator is increased the speed also increases correspondingly, meanwhile this shows that frequency is proportional to the speed. However, the highest speed was measured at 30960 rpm while the minimum measurable speed was measured at 240 rpm. However, a comparison between theoretical and experimental results was carried out, having the maximum error of 4 rpm. Also, a graph of both experimental and theoretical speed results is plotted against their frequency values as shown in Fig 4.3. Also, the errors obtained between the theoretical and experimental values are shown in Fig 4.5. Meanwhile, the errors are due to some sources of errors from crystal stability, mikroC compiler errors, computational errors and triggering circuits errors which affects the speed results obtained. Also, in the experimental analysis it can be observed that some parts of the mikroC code (which has to do with the simple prediction algorithms with respect to the multiplexer) were commented due to some bugs in the mikroC compiler, otherwise it will constitute some errors to the system. Although, this occurs at higher frequency range in the system.

Moreover, it can be observed from the graph of Fig 4.3 which shows the correlation between RPM speed and frequency. Although, the theoretical and experimental plots seem to overlap each other, this is because lots of data were taken during the experiemental results. But in reality there are still some errors between the plots as shown in Fig 4.5 in which the maximum error is 4 rpm.

However, Fig 4.4 clearly illustrates the variations between the theoretical and experimental results obtained. From the plot of the error graph between theoretical and experimental results, it can also be observed that the graph has an irregular pattern, this is because the intervals between the theoretical and experimental results is also irregualr as the frequency changes. Furthermore, it can also be observed in Fig 4.6 that, the variations between the theoretical and experimental results only occur along the rpm speed axis of the graph, which shows the errors between theoretical and experimental results at a particular frequency.

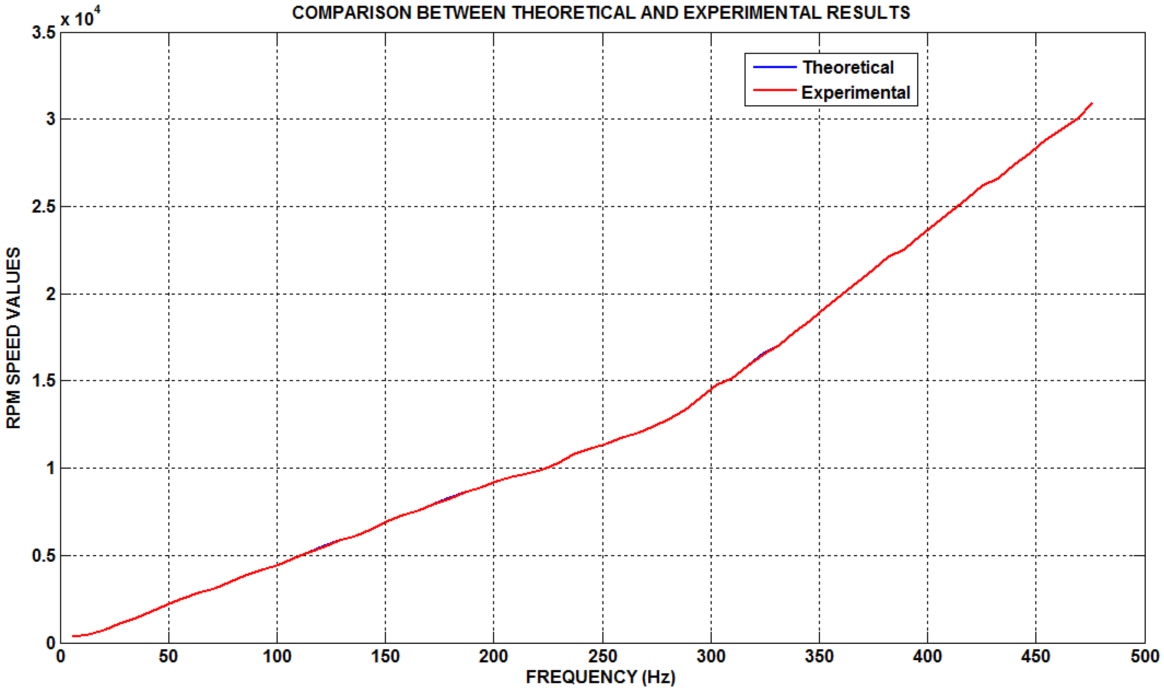


Figure 4.3. Theoretical and Experimental Results

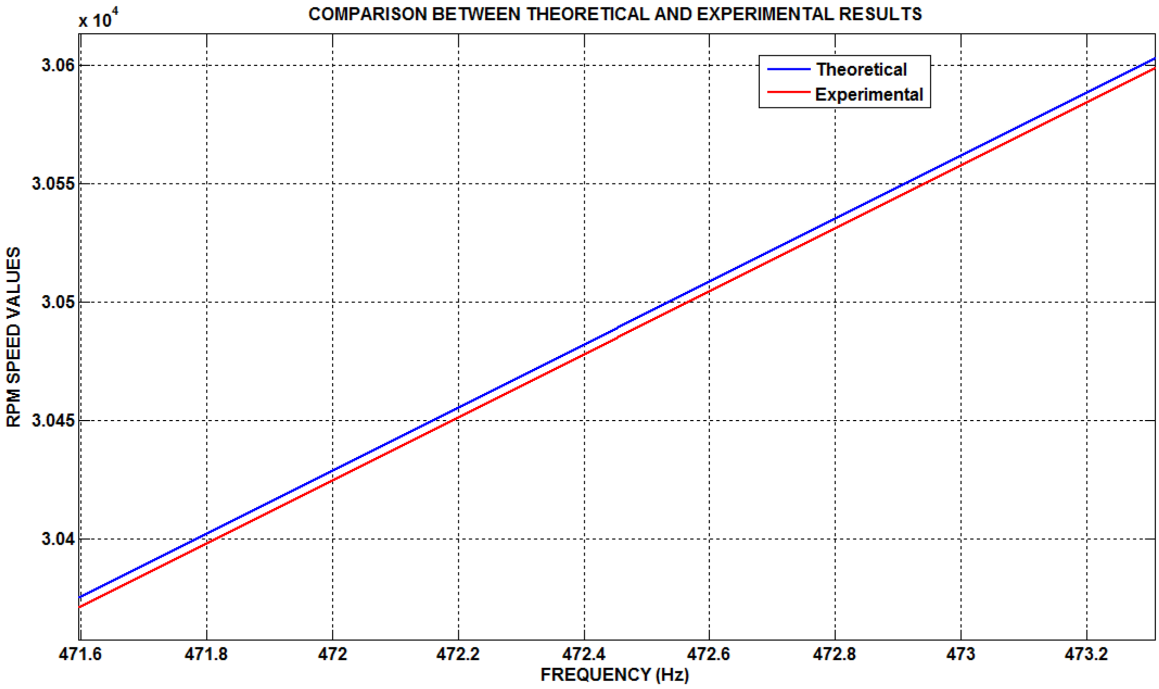


Figure 4.4. Theoretical and Experimental Results

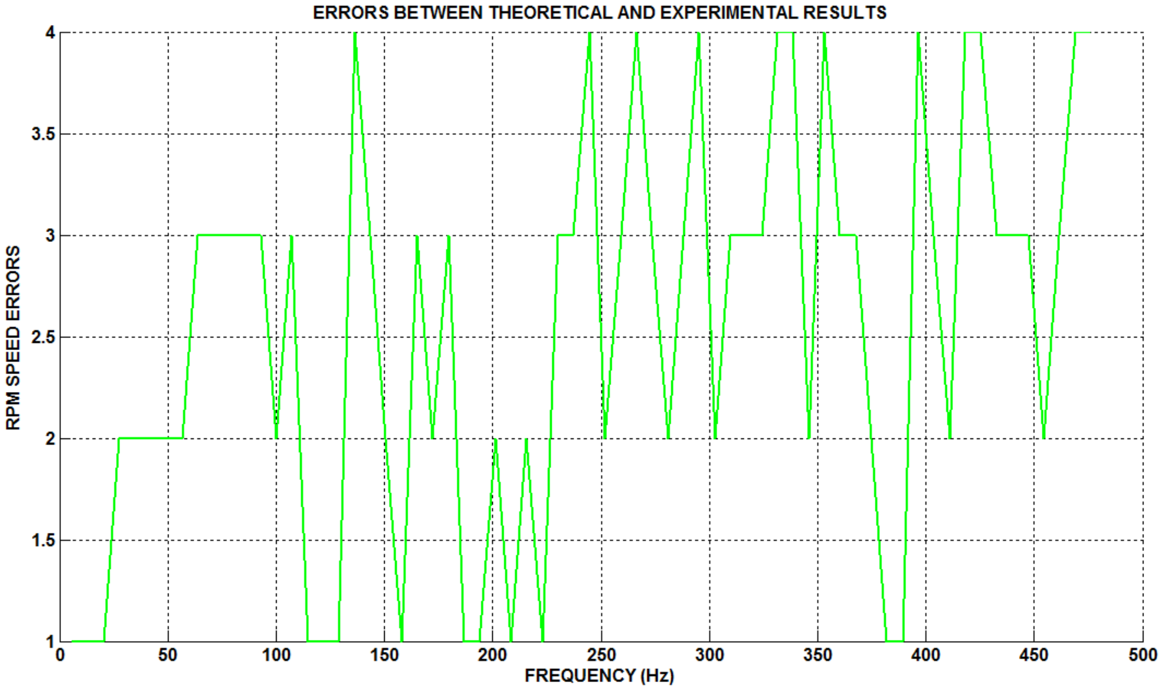


Figure 4.5. Error Results

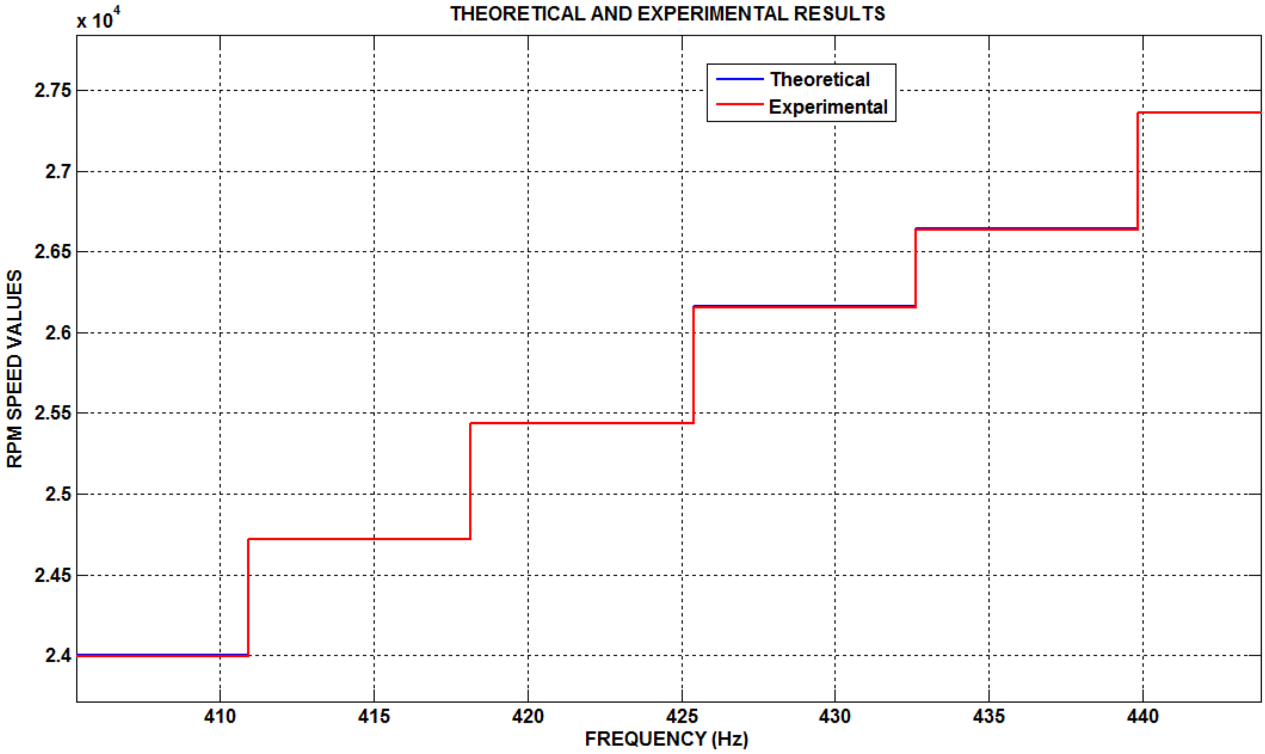


Figure 4.6. Theoretical and Experimental Results

4.4. Results of Linear Prediction Algorithms

The MATLAB plot results below are simulated by assigning some conventional sinusoidal test inputs and linearly estimating the subsequent values based on previous predictions. However, we can observe the correlations between the actual and predicted plots as shown in the figures below, although there are some errors between the actual and predicted values.

Meanwhile, the linear prediction algorithms involve higher mathematical computations, as such it requires large amount of time for calculations in the microprocessor, also since the computations involve matrices and floating point numbers, then the microprocessor can easily run out of memory. Indeed, these limitations are not suitable for this research. Due to these limitations, the simple prediction algorithm has to be implemented in this research. By and large, the simulation results obtained from the prediction algorithms is shown in Fig 4.7 while the errors associated with the results is also shown in Fig 4.8.

4.5. Discussion of Results

It is pertinent to note that MikroC-compiler floating points computations take about 150 microseconds to execute one multiplication instruction [35]. Indeed while updating subsequent matrix values in the prediction algorithms, lots of floating points multiplications were involved. As such, this often leads to a mismatch between hardware and software implementations thereby making the prediction results inconsistent. Due to the PIC limitations of dealing with matrix computations, the prediction algorithms has to be modified into the simple prediction algorithms where the threshold and optimum values of the speed can be obtained via Matlab simulations with corresponding frequency divisions, although this algorithm has less precision when compared to the previous algorithms.

The simple prediction algorithm is basically used to obtain the threshold and optimum values of speed with respect to the corresponding frequency divider parameters. How-

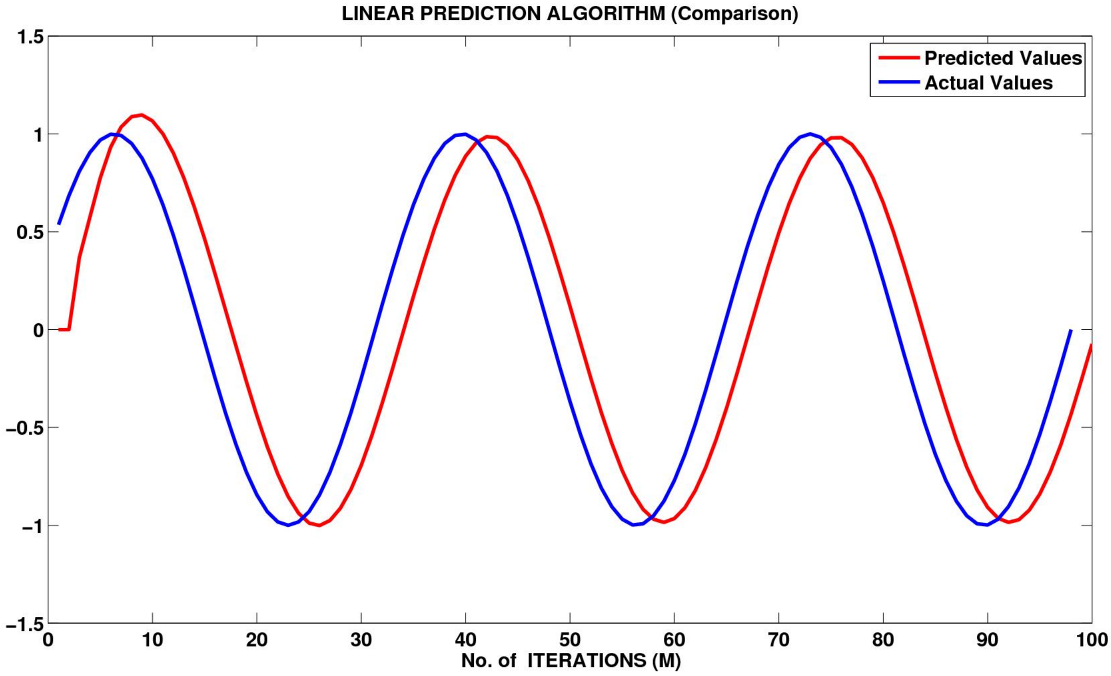


Figure 4.7. Comparison between Actual and Predicted Values

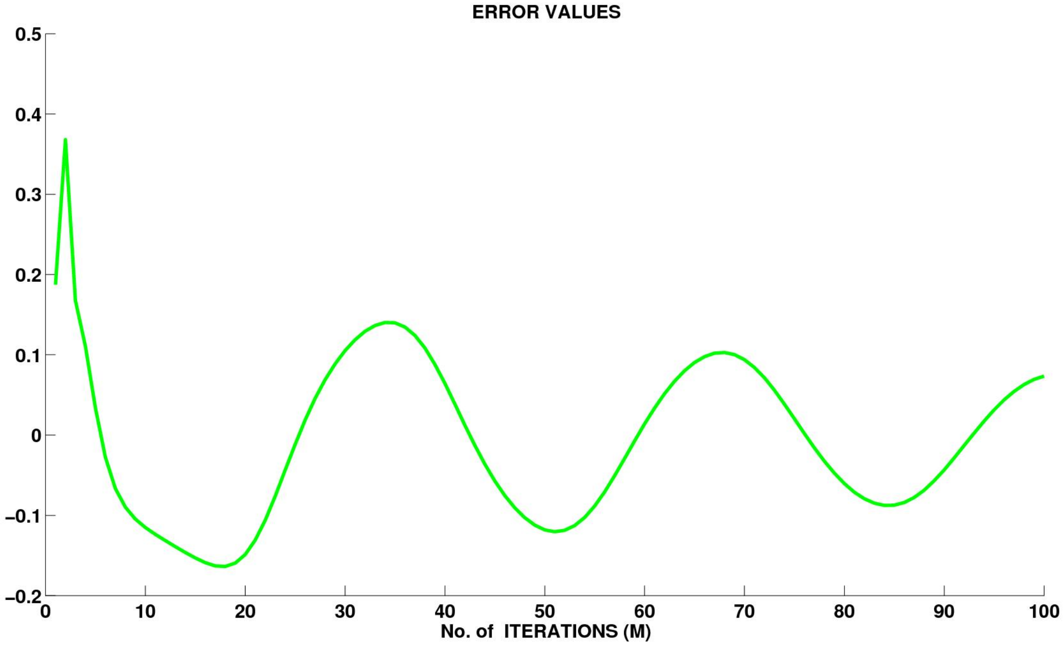


Figure 4.8. Errors between Actual and Predicted Values

ever, it can be observed from the plots that the minimum and maximum measurable speed was obtained, with the speed interval increasing as it goes across the speed axis. Hence, this is a result of the toggling of the frequency divider parameters associated. Also, some comparisons between the actual and predicted values was also reviewed, in relation to the errors associated. Although, the simple prediction algorithms has less accuracy when compared to the linear prediction algorithms as seen from the Matlab simulations. But due to its limitations, it is not suitable in this research.

Based on the theoretical and experimental results obtained from the system, it can be observed that the maximum speed was measured at 30960 rpm while the minimum speed was obtained at 240 rpm. Also, there are some errors associated with the experimental results, in which maximum error obtained is 4 rpm. Although, these errors are due to some sources of errors associated with the system such as crystal stability, computational errors, mikroC compiler errors and triggering circuits errors.

4.6. Sources of Errors

It is obvious that in practical applications involving measurement and computations, the issue of errors is often associated with the systems. Indeed, in this research some instrumental errors are associated with the speed measurements and the frequency values. Meanwhile, when the measurements has some degree of errors, then the speed measurement results are distorted with less accuracy. Although, the target is always meant to minimise such errors from their sources. With respect to this research some of the sources of errors are outlined below:

1. Crystal stability.
2. MikroC compiler errors.
3. Computational errors.
4. Triggering circuits errors.

4.6.1. Crystal Stability

Modern microcontrollers need clock generation, which enhance processors to execute instructions. However, a crystal oscillator is used in this research, which can be a source of error according to its stability. The main source of crystal stability errors is largely related to the momentary variation in frequency of the oscillator. However, it may occur as a result of voltage transients, shocks and vibrations. Also, the crystal stability is also dependent on the measurement interval, as such longer measurement time has more error magnitude [36].

4.6.2. MikroC Compiler

Since the mikroC compiler is used in this research, some errors can also be associated with the compiler. Meanwhile, the compiler gives some error messages when computations involve floating point numbers are included. Hence, this can also constitute a source of error in the system.

4.6.3. Computational Errors

This research work basically involves estimated values from the simple prediction algorithms used. Therefore, there are some distortions between the actual and predicted values during computations. As such, computational values can be another source of error in the system.

4.6.4. Triggering Circuits

This research work is a microcontroller based project. However, microcontrollers require triggering circuits for their operations. Although, these triggering circuits can also be a source of error to the system as a result of manufacturer's defect or systematic errors associated with the components.

5. CONCLUSION AND RECOMMENDATIONS

Based on the results obtained from Matlab simulations and experimental analysis inline with the computer program, it can be concluded that the aim of the thesis has been achieved. However, the speed measurement was increased across a wide range using microcontrollers with reference to the improved prediction algorithms.

Although, some limitations were also encountered, as such to improve on the system for future works, a more efficient and faster code compiler can be used. Perhaps, we can also swap the microcontroller with a DSP (Digital Signal Processor), which has larger memory space and has better response time for computations involving floating point and fixed-point numbers. Furthermore, a more robust crystal oscillator with higher stability can be used, although such higher stability oscillators are very expensive.

.1.
APPENDIX - MikroC Code

```
/*
Project Name: Wide Range Digital Motor Speed Measurement using
              Microcontrollers.
Description: This program measures a motor speed in RPM using the
             Constant Elapsed Time (CET) method.

MCU: PIC18F4620
Development Board: EasyPIC7
Crystal Oscillator: 20MHz
Compiler: MikroC for PIC
Display: 16x2 LCD Module
Author: Kabiru Alhaji Hassan, June,2014.
*/

long i2=0;
long i3=0;
long i4=0;
long i5=0;
long i6=0;
long i8=0;
long i9=0;
long i10=0;
int j1=0;
int m1=2;
int k1=0;
int tmrcopy=0;


// LCD module connections
```

```
sbit LCD_RS at LATD4_bit;        //for output use latch (PIC18 family)
sbit LCD_EN at LATD5_bit;        //for output use latch (PIC18 family)
sbit LCD_D4 at LATD0_bit;        //for output use latch (PIC18 family)
sbit LCD_D5 at LATD1_bit;        //for output use latch (PIC18 family)
sbit LCD_D6 at LATD2_bit;        //for output use latch (PIC18 family)
sbit LCD_D7 at LATD3_bit;        //for output use latch (PIC18 family)
sbit LCD_RS_Direction at TRISD4_bit;
sbit LCD_EN_Direction at TRISD5_bit;
sbit LCD_D4_Direction at TRISD0_bit;
sbit LCD_D5_Direction at TRISD1_bit;
sbit LCD_D6_Direction at TRISD2_bit;
sbit LCD_D7_Direction at TRISD3_bit;
// End LCD module connections


char txt[12];
char txt1[12];
char txt2[12];


void interrupt()
{
    if (INTCON.TMR0IF)
    {
        i2=i2+1;
        INTCON.TMR0IF = 0x00;


        if  (i2==10000)
        {
            i3 = i3+1;
```

```
                i2=0;
            }
        }

        else if (INTCON.INT0IF)
        {
             tmrcopy=TMR0L;
             TMR0L=0;



             INTCON.INT0IF = 0;

             i4=(i3*10000)+i2  ;
             i5=(i4*256)+tmrcopy;



        if (i5<=2000000)
        {
            i6=i6+1;
        }

        else if ((2000000 <= i5) && (i5 <= 4000000))
        {
                  j1=1;
                  i8=(i6*20000000)/(i5*m1*4);
                  i9=i8*60;          // RPM Speed

                  i2=0;
                  i3=0;
                  i4=0;
                  i5=0;
```

```
i6=0;

if ((26 <= i9) && (i9 <= 2456))
{
     PORTC=0x01;
     k1=2;
 }

else if ((2457 <= i9) && (i9 >= 4933))
{
          PORTC=0x02;
          k1=4;
 }
/*else if ((4934 <= i9) && (i9 >= 9867))
{
        PORTC=0x03;
        k1=8;
}


else if ((9868 <= i9) && (i9 <= 19735))
{
        PORTC=0x04;
        k1=16;
}

else if ((19736 <= i9) && (i9 <= 39470))
{
        PORTC=0x05;
        k1=32;
}
```

```
                    else if ((39471 <= i9) && (i9 <= 78940))
                    {
                              PORTC=0x06;
                              k1=64;
                    }

                     else if ((78941 <= i9) && (i9 <= 157880))
                     {
                               PORTC=0x07;
                               k1=128;
                     }

                     else
                     {
                               PORTC=0x00;
                      }*/

                                i10=i9*k1;
        }

        else if (i5>4000000)
        {
                    j1=1;
                    i9=0;
                    i10=0;
                    i6=0;
        }

        }
}
```

```
void main()
{

    T0CON.TMR0ON=0;            //Timer0 OFF control bit
    RCON.IPEN=1;               //Enable priority levels on interrupts
    ADCON0=0x01;              //A/D Converter module is enabled
    ADCON1=0x0F;              //A/D Port Configuration Control bits



    CMCON=0x07;                //Turn off Comparators
    OSCCON=0x72;              //Oscillator Control Register
    OSCTUNE=0xCF;

    TMR0L=0x00;                //Timer0 Reg. for 8-bit mode

    T0CON.T08BIT=1;            //Timer0 8-bit/16-bit Control bit
    T0CON.T0CS=0;              //Timer0 Clock Source Select bit
    T0CON.T0SE=0;              //Timer0 Source Edge Select bit
    T0CON.PSA=0;               //Timer0 Prescaler Assignment bit
    T0CON.T0PS2=0;             //Timer0 Prescaler Select bits
    T0CON.T0PS1=0;             //Timer0 Prescaler Select bits
    T0CON.T0PS0=0;             //Timer0 Prescaler Select bits

    INTCON=0xB0;               //Interrupt Control Register

    T0CON.TMR0ON=1;            //Timer0 ON Control bit

    TRISB = 0xFF;             //Set PortB as input
    LATB = 0x00;               //PORTB Data Latch Register
    PORTB = 0x00;             //Set PortB low
    TRISD = 0x00;             //Set PortD as output
    LATD = 0x00;              //PORTD Data Latch Register
    PORTD = 0x00;             //Set PortD low

    TRISC = 0x00;             //Set PortD as output
    LATC = 0x00;              //PORTD Data Latch Register
    PORTC = 0x00;            //Set PortD low

    Lcd_Init();               // LCD display initialization
    Lcd_Cmd(_LCD_CURSOR_OFF);// LCD command (cursor off)
    Lcd_Cmd(_LCD_CLEAR);      // LCD command (clear LCD)

while(1)
{

      if (j1==1)
      {

         Lcd_Cmd(_LCD_RETURN_HOME);

         //LongToStr(i5, txt);
         //LongToStr(i6, txt1);
         LongToStr(i10, txt2);

         //Lcd_Out(1,1,txt);      // Write first message in first line
         //Lcd_Out (2,1,txt1);  // Write first message in first line
         Lcd_Out (2,1,txt2);    // Write first message in first line

         j1 = 0;
```

```
        }
    }
    }
```

REFERENCES

1. Bonert R., "Design of a High Performance Digital Tachometer with a Microcontroller", *Instrumentation and Measurement*, IEEE Transactions on, vol. 38, no. 6, pp. 1104 - 1108, 1989.

2. Denic D., G. Miljkovic and D. Zivanovic, "Microcomputer Based Wide Range Digital Tachometer", ISSN, vol. 1392, no. 1215, pp. 31 - 36.

3. Maloney T. J., F. L. Alvarado, "A Digital Method for DC Motor Speed Control", *Industrial Electronics and Control Instrumentation*, IEEE Transactions on, no. 1, pp. 44 - 46, 1976.

4. Damle P. D. and G. Dubley, "A Digital Computer Program for Chopper-fed DC Motors," *Industrial Electronics and Control Instrumentation*, IEEE Transactions on, no. 3, pp. 408–412, 1975.

5. Shahril M. and A. Bakar, "DC Motor Speed Control using Microcontroller PIC16F877A", 2009.

6. Grimm T. M., J. W. Hill, G. W. Rhodes and C. M. Smith, "Robotic Arm," *uS Patent*, 4, 806, 066, Feb. 21 1989.

7. Mokhtar M. I., *Development of Digital Wind Speed Meter with Wind Direction*, Ph.D. Dissertation, Universiti Malaysia Pahang, 2007.

8. Ettomi Y. S. A., *Microcontroller Based Adjustable Speed Closed-loop DC Motor Drive*, Ph.D. Dissertation, Universiti Putra Malaysia, 2004.

9. Hsiung S. C., "The use of PIC Microcontrollers in Multiple DC Motors Control Applications," *Journal of Industrial Technology*, vol. 23, no. 3, pp. 2–3, 2007.

10. Dhaouadi R., N. Mohan and L. Norum, "Design and Implementation of an Ex-

tended Kalman Filter for the State Estimation of a Permanent Magnet Synchronous Motor," *Power Electronics*, IEEE Transactions on, vol. 6, no. 3, pp. 491–497, 1991.

11. Obermeier C., H. Kellermann, G. Brandenburg, and J. Heinzl, "Sensorless Field Oriented Speed Control of a Hybrid Stepper Motor using an Extended Kalman Filter," in *EUROPEAN CONFERENCE ON POWER ELECTRONICS AND APPLICATIONS*, PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS, vol. 1, pp. 1–238, 1997.

12. Amir M. and F. Awang, *DC Motor Speed Controller*, Ph.D. dissertation, Universiti Malaysia Pahang, 2010.

13. Wu R. and G. R. Slemon, "A Permanent Magnet Motor Drive without a Shaft Sensor," *Industry Applications*, IEEE Transactions on, vol. 27, no. 5, pp. 1005–1011, 1991.

14. Ehsan A. M., "Pi Algorithm for a Speed Control of DC Motor with Tachometer Feedback," 2009.

15. Shanmugasundram R., K. M. Zakaraiah and N. Yadaiah, "Modeling, Simulation and Analysis of Controllers for Brushless Direct Current Motor Drives," *Journal of Vibration and Control*, vol. 19, no. 8, pp. 1250–1264, 2013.

16. Pu J. T. and H. Wang, "A Novel Variable M/T Method for Speed Measurement with High Precision in a Wide Speed Range," in *2nd International Conference on Electronic and Mechanical Engineering and Information Technology*, Atlantis Press, 2012.

17. Hintz K. J. and D. Tabak, "Microcontrollers: Architecture, Implementation and Programming", *McGraw-Hill Professional*, 1992.

18. Eckmiller R., "Adaptive Sensory-motor Encoder for Visual or Acoustic Prosthesis," uS Patent 6,400,989, Jun. 4, 2002.

19. Kieselbach J. and K. Mleinek, “Encoder having First and Second Housing Portions and Guide Means,” uS Patent 5,859,425, Jan. 12 1999.

20. Braunl T., ”Embedded robotics: mobile robot design and applications with embedded systems”, Springer, 2008.

21. Itonaga M., M. Kai, K. Kayanuma, N. Nakamura, and J. Shikunami, “Reflective Optical Rotary Encoder Disc,” uS Patent 5,006,703, Apr. 9 1991.

22. Arai N. and H. Shinada, “Optical Linear Encoder,” uS Patent 4,667,099, May 19 1987.

23. Avins J. Y., “Incremental Encoder for Measuring Positions of Objects such as Rotating Shafts,” uS Patent 4,308,500, Dec. 29 1981.

24. Kennedy R. A., “Bit binary counter,” uS Patent 3,916,484, Oct. 5, 1975.

25. Santonico R. and R. Cardarelli, “Development of Resistive Plate Counters,” *Nuclear Instruments and Methods in Physics Research*, vol. 187, no. 2, pp. 377–380, 1981.

26. Holst P. F. and L. R. Kirkwood, “Frequency Counter Circuit,” uS Patent 2,411,573, Nov. 26 1946.

27. Karnopp D. C., D. L. Margolis, and R. C. Rosenberg, ”System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems,” Wiley, 2012.

28. Mohanty N. K. and R. Muthu, “Microcontroller based PWM Controlled Four Switch Three Phase Inverter Fed Induction Motor Drive,” *Serbian Journal of Electrical Engineering*, vol. 7, no. 2, 2010.

29. Matsui N., “Sensorless PM Brushless DC Motor Drives,” *Industrial Electronics*, IEEE Transactions on, vol. 43, no. 2, pp. 300–308, 1996.

30. Kuo B. C. and M. F. Golnaraghi, ”Automatic Control Systems”, John Wiley and Sons New York, vol. 4, 2003.

31. Campbell K. J., “RF Signal Generator,” uS Patent 4,047,121, Sep. 6 1977.

32. Dragone C., “Optical Multiplexer / Demultiplexer,” uS Patent 5,002,350, Mar. 26 1991.

33. Conway, B. E., L. Bai, and D. F. Tessier, “Digital Oscilloscope Control Apparatus,” uS Patent 3,836,851, Sep. 17 1974.

34. Garg M., “Linear Prediction Algorithms,” *Indian Institute of technology*, Bombay, India, 2003.

35. Saltz J., H. Berryman, and J. Wu, “Multiprocessors and Run-time Compilation,” *Concurrency: Practice and Experience*, vol. 3, no. 6, pp. 573–592, 1991.

36. Bakshi V., ”Extreme Ultraviolet (EUV) Lithography”, *Spie Press Bellingham*, vol. 178, 2009.