Sudan University of Science and Technology

College of Engineering

Electrical Engineering Department

DC-Motor Control of Speed Using FUZZY-PID Controller

التحكم في سرعة محرك تيار مستمر باستخدام المتحكم التناسبى التفاضلى التكاملى الغامض

A Project Submitted in Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) In Electrical Engineering

Prepared By:

1. **Ahmed Alhussain Ali Ahmed.**
2. **Samar Mohammed Abdullah.**
3. **Mohamed Omer Babiker Mohamed Ali.**
4. **Abdulmajeed Abdulmonim Abdullatif Abdulrahman.**

Supervised By:

Ust. Galal Abdalrahman Mohammed

September 2017

الاية

بسم الله الرحمن الرحيم

قال تعالى :

قُلْ لَا أَقُولُ لَكُمْ عِنْدِي خَزَائِنُ اللَّهِ وَلَا أَعْلَمُ الْغَيْبَ وَلَا أَقُولُ لَكُمْ إِنِّي مَلَكٌ ۖ إِنْ أَتَّبِعُ إِلَّا مَا يُوحَىٰ إِلَيَّ ۚ قُلْ هَلْ يَسْتَوِي الْأَعْمَىٰ وَالْبَصِيرُ ۚ أَفَلَا تَتَفَكَّرُونَ

الأنعام ﴿ 50 ﴾

صدق الله العظيم

DEDICATIONS

This study is lovingly dedicated to our parents for their emotional and financial support, our brothers, our sisters and our friends whose has been constant source of inspiration for us.

They have given us the drive and discipline to tackle any task with enthusiasm and determination.

Without their love and support this project would not have been made possible.

ACKNOWLEDGEME

We wish to express our profound gratitude to our Supervisor assistant **Galal Abd-Alrahman Mohammed** for his valuable guidance , continues encouragement , worthwhile suggestions and constructive ideas throughout this project . His support, pragmatic analysis and understanding made this study a success and knowledgeable experience for us.

Abstract

There are many problems in obtaining very fast speed in some production processes. Many sensitive areas need a very precise and precise speed, for example in the medical fields (pharmaceutical industry, droplets, and intravenous fluids), food products. This dilemma was chosen for its importance in daily life and its sensitivity in terms of products.

A brushless DC motor with a magnet being the rotor and stationary windings forming the stator, this design provides many advantages over other motors, they are becoming widely used in various consumer and industrial systems.

One of the applications used here is the speed control of DC motor. DC motors are used in industry extensively due to their low cost, low energy consumption and their easy adaptation with digital systems. Controlling the speed of a DC motors is very important as any small change can lead to instability of the closed loop system.

The most commonly used controller for the position control of DC servomotor is conventional Proportional –Integral –Derivative (PID) controller. However, the PID controller has some disadvantages such as: the high starting overshoot, sensitivity to controller gains and sluggish response due to sudden disturbance. So, the relatively design PID controller with computational optimization approach method is proposed to overcome the disadvantages of the conventional PID controller. In position control system, PID controller sometimes cannot make this application accurate because of nonlinear properties. Therefore, in this thesis the fuzzy logic controller is proposed to overcome the problem of PID controller. Fuzzy logic controller has ability to control the nonlinear system because of the algorithm is implementing in language. Based on the simulation result the fuzzy logic controller designed is able to improve the performance of the position control system compared to the PID controller in term of rise time (Tr) and settling time (Ts).

This research presents a design and implementation of Fuzzy Logic PID Controller using 8-bit microcontroller U-Board speed control. The speed of system depends on the speed of considered DC motor. So, the main speed control is to control the speed of considered DC motor. The frequency response of the system and its specifications were calculated, and Comparison between the conventional output and the fuzzy self-tuning output was done on the basis of the simulation result obtained by MATLAB Simulink.

This study is a practical study aimed at solving the forms of speed accuracy, or at least contributing and adding in this subject in an effective manner, using the two methods of simulation using the PROTEUS program, and practical application of a real circuit.

مستخلص

كثير ما تواجهنا مشاكل في الحصول على سرعه دقيقه جدا في بعض العمليات الإنتاجية، فكثير من المجالات الحساسة تحتاج لسرعه مضبوطة وغايه في الدقة، على سبيل المثال في المجالات الطبية (صناعة الأدوية والقطْرات، والسوائل الوريدية)، وفي مجالات الأغذية، ومجالات صناعة الأوراق، ولهذا وقع الاختيار على هذا الإشكال لأهميته في الحياة اليومية، ولحساسيته من حيث المنتجات والمتطلبات.

محركات التيار المباشر تساهم بدور أساسي وفعال جدا في هذا المجال حيث أن محرك التيار المستمر ذو المغنطيسية الدائمة وعديم الفرش الكربونية يمتلك الكثير من الخصائص الايجابية والتي تساهم بشكل فعال في استقرار الانظمة وهي سهلة التحكم وتستخدم بصورة واسعة جدا في الصناعة.

واحدة من التطبيقات المستخدمة هنا هو سرعة التحكم في محرك التيار المستمر. وتستخدم محركات التيار المستمر في الصناعة على نطاق واسع نظرا لانخفاض تكلفتها، وانخفاض استهلاك الطاقة وسهولة التكيف مع النظم الرقمية. السيطرة على سرعة محركات التيار المستمر مهم جدا كما أي تغيير صغير يمكن أن يؤدي إلى عدم استقرار نظام حلقة المغلقة.

إن أغلب المتحكمات المستخدمة للتحكم في الوضع، هي المتحكمات التقليدية المعروفة بالمتحكم التناسبي التكاملي التفاضلي، لكن رغم ذلك نجد أن المتحكم التناسبي التكاملي التفاضلي، لديه بعض العيوب، مثل البدء بتجاوز عالي للهدف، حساسية لكسب المتحكمة والاستجابة البطيئة للتغيرات الفجائية. وعليه فان مقترح تصميم المتحكم التناسبي التكاملي التفاضلي بالطريقة الحسابية المثلي هو يتغلب على بعض عيوب المتحكم التناسبي التكاملي التفاضلي التقليدي. عند استخدامه للتحكم في أنظمة التحكم في الوضع، وجد أنه لا يعطي الدقة المطلوبة، نسبة لخواصها غير الخطية. ولذلك في هذا البحث تم تصميم المتحكم الغامض لتفادي هذه المشكلة. للمتحكم الغامض القدرة على التحكم في الأنظمة غير الخطية، لأن الخوارزمية المستخدمة تنفذ بتعليمات لغوية. واعتمادا على نتائج المحاكاة، نجد أن المتحكم الغامض له القدرة علي تحسين أداء النظام المتمثل في تقليل زمن الصعود وزمن السكون، مقارنة بالمتحكم التناسبي التكاملي التفاضلي.

يقدم هذا البحث تصميم وتنفيذ التناسبي التكاملي التفاضلي و الغامض باستخدام متحكم سرعة 8بت . سرعة النظام تعتمد على سرعة المحرك المستمر. لذا، فإن التحكم الرئيسي بالسرعة هو التحكم في سرعة المحرك المستمر المعتبر. تم حساب الاستجابة الترددية للنظام، ومواصفاته، وأجريت المقارنة بين التحكم التقليدي والمتحكم الضبابي عبر برنامج المحاكاة الماتلاب .

هذه الدراسة هي دراسة عملية تهدف الى حل اشكال دقة السرعة، او على الأقل الاسهام والإضافة في هذا الموضوع بصورة مؤثرة وفعالة، وتم فيها استخدام وسيلتي المحاكاة باستخدام برنامج المحاكاة البروتوس، والتطبيق العملي على دائرة حقيقية.

TABLE OF CONTENTS

LIST OF FIGIURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Alternating Current
CPU	Central Processing Unit
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read Only Memory
EMF	Electromotive Force
EPROM	Erasable Programmable Read Only Memory
Hz	Hertz
LED	Light Emitted Diode
MMF	Magnetic Motive Force
PD	Proportional Derivative controller
PI	Proportional Integral controller
PID	Proportional Integral Derivative
PROM	Programmable Read Only Memory
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RX	Receiver mode
TX	Transmitter mode
LCD	liquid-crystal display
MOSFET	The metal–oxide–semiconductor field-effect transistor

LIST OF SYMBOLS

Kp	Proportional gain
Ki	Integral gain
Kd	Derivative gain
Ti	Integral time
Td	Derivative time
T	Motor torque
i	Armature Current
Kt	Torque constant
e	Electromotive force
$\dot{\theta}$	Angular velocity of the shaft
K_b	Electromotive force constant
K	Motor torque and back EMF constant
J	The moment of inertia of the rotor
b	The motor viscous friction constant
La	Electric inductance
Ra	Electric resistance
V	Voltage source
Kg	Gears ratio
θ	gear angle

CHAPTER ONE

INTRODUCTION

1.1 Project Background

The development of high performance motor drives is very important in industrial as well as other purpose applications such as steel rolling mills, electric trains and robotics. Generally, a high-performance motor drive system must have good dynamic speed command tracking and load regulating response to perform task. DC drives, because of their simplicity, ease of application, high reliabilities, flexibilities and favorable cost have long been a backbone of industrial applications, robot manipulators and home appliances where speed and position control of motor are required.

There are two main types of electrical motors. There are direct current or DC and alternating current or AC motors. The reference of DC or AC refers to how the electrical current is transferred through and from the motor. Both types of motors have different functions and uses. DC motors come in two general types. They can have brushes or be brushless (synchronous motor). Then, AC motors come in two different types which are they can be single phase and three phases. This project will cover and design the controller for a DC motor only, the next subtopic will be discussed details on types on DC motor.

DC drives are less complex with a single power conversion from AC to DC. Again, the speed torque characteristics of DC motors are much more superior to That of AC motors. A DC motors provide excellent control of speed for acceleration and deceleration. DC drives are normally less expensive for most horsepower ratings. DC motors have a long tradition of use as adjustable speed machines and a wide range of options have evolved for this purpose. In these applications, the motor should be precisely controlled to give the desired performance.

1.2 Problem Statements

The nonlinear characteristics of a DC motor such as saturation and friction could degrade the performance of conventional controllers. Many advanced model based control methods such as variable structure control and model reference adaptive control have been developed to reduce these effects. However, the performance of these methods depends on the accuracy of the system models and parameters. Generally, an accurate non-linear model of an actual DC motor is difficult to find, and parameter values obtained from system identification may be only approximated values.

Control system that could be able to give a fast response in order to maintain the speed of the DC motor at the desired value with a minimum overshoot, minimum steady state error, minimum settling time and fast rising time are very important and crucial in industrial application.

Conventional control has proven for a long time to be good enough to handle control tasks on system control, however this implementation relies on an exact mathematical model of the plan to be control and not a simple mathematical operation.

1.3 Objectives

The objectives of this project are as follows:

- ✓ To design a FUZZY-PID logic controller as another type of controller that can be used on to control speed of the DC motor.
- ✓ To analyze the performance comparison between PID and Fuzzy Logic Controller in order to control speed of the DC motor by simulation and real circuit.

- ✓ To evaluate and validate performance of the design FUZZY-PID output by using microcontroller.
- ✓ Improve responses performances for DC motor by PID and fuzzy logic implementation.

1.4 Methodology

Study all the previous studies in the field and Analysis plot shows the transient and steady state response of the system using MATLAB Simulink the simulation for the practical was performed in PROTEUS simulator.

1.5 Project Layout

This thesis consists of five chapter: Chapter one presents an introduction to the principles of the study, general concepts, overview, the problem statement and also discusses the objectives and outline of methodologies of the study. Chapter two discusses a theoretical background of DC motor, PID controller and fuzzy system controller, and the Microcontroller. Chapter three presents DC motor modeling and the circuit component. Chapter four includes simulation results and the practical implementations and operation. Finally, Chapter five provides the conclusion and recommendations.

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Introduction to brushless dc motor controlling

Emerging intelligent techniques have been developed and extensively use to improve or to replace conventional control techniques because these techniques do not require a precise model [4]. One of the intelligent techniques, fuzzy logic developed by Lotfi A. Zadeh [5] is applied for controller design in many applications [4,6,7]. Fuzzy Logic has been successfully applied to a large number of control applications and platforms such as FPGA [8], personal computer [9] or off the shelf microcontroller [10].

At present, Proportional-Integral-Derivative "PID" controller, due to its simplicity, stability, and robustness, is a type of controllers that is most widely applied [11,12]. For brushed DC motors, factors such as unknown load characteristic and parameter variation influence seriously the controlling effect of speed controller [8].

Fuzzy logic controller provides an alternative to PID controller since it is a good tool for the control of systems that are difficult in modelling. The control action in fuzzy logic controllers can be expressed with simple "if-then" rules.

Fuzzy control gives robust performance for a linear or nonlinear plant with parameter variation. Hardware implementation of the controller can be achieved in a number of ways to create new products [8].

2.2 DC Motor

A dc motor converts dc electrical energy into rotational mechanical energy. A major part of the torque generated in the rotor (armature) of the motor is available to drive an external load. The dc motor is probably the earliest form of electric motor. Because of features such as high torque, speed controllability over a wide range, portability, well-behaved speed–torque characteristics, easier and accurate modeling, and adaptability to various types of control methods, dc motors are still widely used in numerous engineering applications including robotic manipulators, vehicles, transport mechanisms, disk drives, positioning tables, machine tools, biomedical devices, and servo valve actuators. In view of effective control techniques that have been developed for ac motors, they are rapidly becoming popular in applications where DC motors had dominated. Still, DC motor is the basis of the performance of an AC motor which is judged in such applications. [3]

2.3 Construction

DC motors consist of one set of coils, called armature winding, inside another set of coils or a set of permanent magnets, called the stator. Applying a voltage to the coils produces a torque in the armature, resulting in motion.

2.3.1 Stator

- ✓ The stator is the stationary outside part of a motor
- ✓ The stator of a permanent magnet dc motor is composed of two or more permanent magnet pole pieces.
- ✓ The magnetic field can alternatively be created by an *electromagnet*. In this case, a DC coil (field winding) is wound around a magnetic material
- ✓ that forms part of the stator.

2.3.2 Rotor

- ✓ The rotor is the inner part which rotates.
- ✓ The rotor is composed of windings (called armature windings) which are connected to the external circuit through a mechanical commutator.
- ✓ Both stator and rotor are made of ferromagnetic materials. The two are separated by air-gap.

2.3.3 Winding

- ✓ A winding is made up of series or parallel connection of coils.
- ✓ Armature winding - The winding through which the voltage is applied or induced.
- ✓ Field winding - The winding through which a current is passed to produce flux (for the electromagnet)
- ✓ Windings are usually made of copper.

2.4 Principle of Operation

If electrical energy is supplied to a conductor lying perpendicular to a magnetic field, the interaction of current flowing in the conductor and the magnetic field will produce mechanical force (and therefore, mechanical energy).

Consider a coil in a magnetic field, when the two ends of the coil are connected across a DC voltage source, a current flow through it. A force is exerted on the coil as a result of the interaction of magnetic field and electric current. The force on the two sides of the coil is such that the coil starts to move in the direction of force. In an actual DC motor, several such coils are wound on the rotor, all of which experience force, resulting in rotation. The greater the current in the wire, or the greater the magnetic field, the faster the wire moves because of the greater force created. At the same time, this torque is being produced, the conductors are moving in a magnetic field. At different positions, the flux linked with it changes, which causes an *emf* to be induced. This voltage is in opposition to the voltage that causes current flow through the conductor and is referred to as a *counter-voltage* or *back emf.* The value of current flowing through the armature is dependent upon the difference between the applied voltage and this counter-voltage. The current due to this counter-voltage tends to oppose the very cause for its production according to Lenz's law. It results in the rotor slowing down. Eventually, the rotor slows just Induced emf Flux enough so that the force created by the magnetic field equals the load force applied on the shaft. Then the system moves at constant velocity. [3]

2.5 DC Machine Classifications

DC Machines can be classified according to the electrical connections of the armature winding and the field windings. The different ways in which these windings are connected lead to machines operating with different characteristics. The field winding can be either self-excited or separately-excited, that is, the terminals of the winding can be connected across the input voltage terminals or fed from a separate voltage source. Further, in self-excited motors, the field winding can be connected either in series or in parallel with the armature winding. These different types of connections give rise to very different types of machines[3].

2.5.1 Separately excited machines

- ✓ The armature and field winding are electrically separate from each other.
- ✓ The field winding is excited by a separate DC source.

2.5.2 Self excited machines

In these machines, instead of a separate voltage source, the field winding is connected across the main voltage terminals.

2.5.2.1 Shunt machine

- ✓ The armature and field winding are connected in parallel.

- ✓ The armature voltage and field voltage are the same.

2.5.2.2 Series machine

- ✓ The field winding and armature winding are connected in series.
- ✓ The field winding carries the same current as the armature winding.

2.5.3 Compound DC machine

If both series and shunt field windings are used, the motor is said to be compounded. In a compound machine, the series field winding is connected in series with the armature, and the shunt field winding is connected in parallel. Two types of arrangements are possible in compound motors:

2.5.3.1 Cumulative compounding

If the magnetic fluxes produced by both series and shunt field windings are in the same direction (i.e., additive), the machine is called cumulative compound.

2.5.3.2 Differential compounding

If the two fluxes are in opposition, the machine is differential compound. In both these types, the connection can be either short shunt or long shunt.

2.5.4 Brushed DC Motor

The brushed DC motor generates torque directly from DC power supplied to the motor by using internal commutation, stationary permanent magnets, and rotating electrical magnets. Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Although that, the brushed DC motor need maintenance regularly by replacing the brushes and springs which carry the electric current, as well as cleaning or replacing the commutated. These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor. The Figure 2-1 below shows the brushed DC motor. [3]

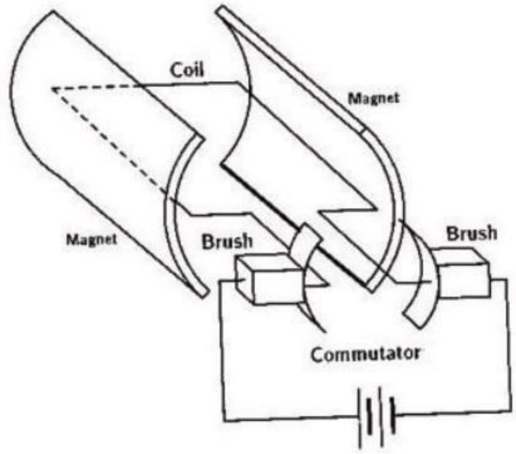


Figure 2.1: Brushed DC motor

2.5.5 Synchronous DC Motor

There are two types of synchronous DC motor which are the brushless DC motor and the stepper motor. Both require external commutation to generate torque. The motor is lock up if driven by DC power.

2.5.5.1 Brushless DC Motor

Brushless DC motor use a rotating permanent magnet in the rotor and stationary electrical magnets on the motor housing. Brushless motor consists a controller that used to converts from DC to AC. This design is simpler than brushed motor because it eliminates the complication of transferring power from outside the motor to the spinning rotor. This type of motor needs no maintenance and more efficient compared to the brushed motor that discussed. Figure 2-4 shows that the brushless DC motor using three poles to operates. [3]

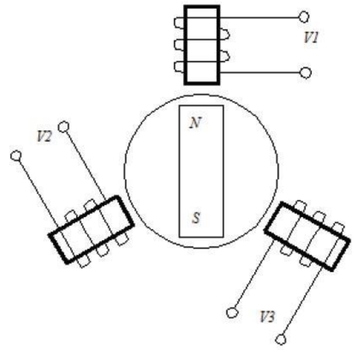


Figure 2.2: Brushless DC motor

2.5.5.2 Stepper Motor

A stepper motor is the electric motor that can divide a full rotation into a large number of steps. The motor’s position can be controlled precisely without any feedback mechanism. Stepper motors are similar to switched reluctance motors which are very large stepping motors with a reduced pole count, and generally are closed loop commutated. Figure 2-5 shows the operation of the stepper motor.[3]

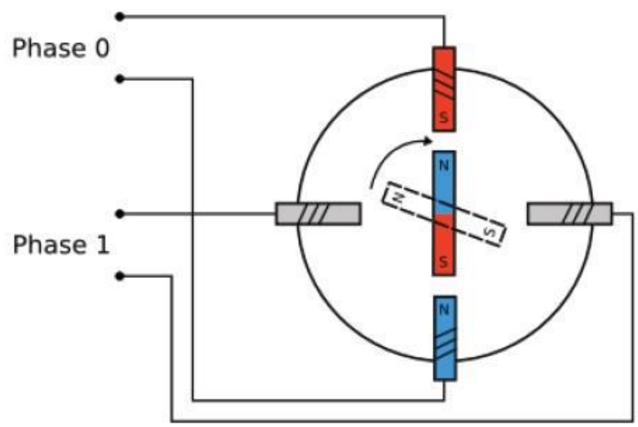


Figure 2.3: Stepper motor

2.6 Control Systems

Control systems are an integral part of modern society. Numerous applications are all around us like the rockets fire, the space shuttle lifts off, the earth orbit, in splashing cooling water, a metallic part is automatically machined and a selfguided vehicle delivering material to workstations in an aerospace assembly plant glides along the floor seeking its destination [1].

Automatic control has played a vital role in the advance of engineering and science. In addition to its extreme importance in space-vehicle systems, missile guidance systems, robotic systems, and the like, automatic control has become an important and integral part of modern manufacturing and industrial processes. For example, automatic control is essential in the numerical control of machine tools in the manufacturing industries, in the design of autopilot systems in the aerospace industries, and in the design of cars and trucks in the automobile industries. It is also essential in such industrial operations as controlling pressure, temperature, humidity, viscosity, and flow in the process industries [2].

Humans are not the only creators of automatically controlled systems; these systems also exist in nature. Within our own bodies are numerous control systems, such as the pancreas, which regulates our blood sugar. In time of "fight or flight," our adrenaline increases along with our heart rate, causing more oxygen to be delivered to our cells. Our eyes follow a moving object to keep it in view; our hands grasp the object and Place it precisely at a predetermined location [1].

2.6.1 Historical review

The first significant work in automatic control was James Watt's centrifugal governor for the speed control of a steam engine in the eighteenth century. Other significant works in the early stages of development of control theory were due to Minorsky, Hazen and Nyquist among many others. In 1922, Minorsky worked on automatic controllers for steering ships and showed how stability could be determined from the differential equations describing the system. In 1932, Nyquist developed a relatively simple procedure for determining the stability of closed-loop systems on the basis of open-loop response to steady-state sinusoidal inputs. In 1934, Hazen, who introduced the term servomechanisms for position control systems, discussed the design of relay servomechanisms capable of closely following a changing input. Since the late 1950s, the emphasis in control design problems has been shifted from the design of one of many systems that work to the design of one optimal system in some meaningful sense. As modern plants with many inputs and outputs become more and more complex, the description of a modern control system requires a large number of equations.

Classical control theory, which deals only with single input single output systems, becomes powerless for multiple input multiple output systems. Since about 1960, because the availability of digital computers made possible time-domain analysis of complex systems, modern control theory, based on time-domain analysis and synthesis using state variables, has been developed to cope with the increased complexity of modern plants and the stringent requirements on accuracy, weight, cost in military, space and industrial applications.

2.6.2 Open and closed loop control systems

Feedback control systems are often referred to as closed-loop control systems. In practice, the terms feedback control and closed-loop control are used interchange-ably. In a closed-loop control system the actuating error signal, which is the difference between the input signal and the feedback signal (which may be the output signal itself or a function of the output signal and its derivatives and/or integrals), is fed to the controller so as to reduce the error and bring the output of the system to a desired value. The term closed-loop control always implies the use of feedback control action in order to reduce system error.

Those systems in which the output has no effect on the control action are called open-loop control systems. In other words, in an open loop control system the output is neither measured nor feedback for comparison with the input. One practical example is a washing machine. Soaking, washing, and rinsing in the washer operate on a time basis. The machine does not measure the output signal, that is, the cleanliness of the clothes. In any open-loop control system the output is not compared with the reference input. Thus, to each reference input there corresponds a fixed operating condition; as a result, the accuracy of the system depends on calibration. In the presence of disturbances, an open-loop control system will not perform the desired task. Open-loop control can be used, in practice, only if the relationship between the input and output is known and if there are neither internal nor external disturbances. Clearly, such systems are not feedback control systems. Note that any control system that operates on a time basis is open loop. For instance, traffic control by means of signals operated on a time basis is another example of open-loop control.

An advantage of the closed loop control system is the fact that the use of feedback makes the system response relatively insensitive to external disturbances and internal variations in system parameters. It is thus possible to use relatively inaccurate and inexpensive components to obtain the accurate control of a given plant, whereas doing so is impossible in the open-loop case.

From the point of view of stability, the open-loop control system is easier to build because system stability is not a major problem. On the other hand, stability is a major problem in the closed-loop control system, which may tend to overcorrect errors and thereby can cause oscillations of constant or changing amplitude. It should be emphasized that for systems in which the inputs are known ahead of time and in which there are no disturbances it is advisable to use open-loop control. Closed loop control systems have advantages only when unpredictable disturbances and/or unpredictable variations in system components are present. Note that the output power rating partially determines the cost, weight, and size of a control system. The number of components used in a closed-loop control system is more than that for a corresponding open-loop control system. Thus, the closed-loop control system is generally higher in cost and power. To decrease the required power of a system, open-loop control may be used where applicable. A proper combination of open-loop and closed-loop controls is usually less expensive and will give satisfactory overall system performance [2].

2.6.3 Advantages of control systems

Control systems giving the ability to move large equipment with precision that would otherwise be impossible. For example pointing huge antennas toward the farthest reaches of the universe to pick up faint radio signals; controlling these antennas by hand would be impossible. Because of control systems, elevators carry us quickly to our destination, automatically stopping at the right floor. Humans alone could not provide the power required for the load and the speed; motors provide the power, and control systems regulate the position and speed.

Control systems build for primary reasons; Power amplification, remote control, convenience of input form and compensation for disturbances [1].

2.7 Nonlinear Systems

A system is nonlinear if the principle of superposition does not apply. Thus, for a nonlinear system the response to two inputs cannot be calculated by treating one input at a time and adding the results. Although many physical relationships are often represented by linear equations, in most cases actual relationships are not quite linear. In fact, a careful study of physical systems reveals that even so-called "linear systems" are really linear only in limited operating ranges. In practice, many electromechanical systems, hydraulic systems, pneumatic systems, and so on, involve nonlinear relationships among the variables. For example, the output of a component may saturate for large input signals. There may be a dead space that affects small signals. (The dead space of a component is a small range of input variations to which the component is insensitive). Square-law nonlinearity may occur in some components. For instance, dampers used in physical systems may be linear for low-velocity operations but may become nonlinear at high velocities, and the damping force may become proportional to the square of the operating velocity.

In control engineering a normal operation of the system may be around an equilibrium point, and the signals may be considered small signals around the equilibrium. (It should be pointed out that there are many exceptions to such a case). However, if the system operates around an equilibrium point and if the signals involved are small signals, then it is possible to approximate the non- linear system by a linear system. Such a linear system is equivalent to the nonlinear system considered within a limited operating range. Such a linearized model (linear, time-invariant model) is very important in control engineering.

The linearization procedure to be presented in the following is based on the expansion of nonlinear function into a Taylor series about the operating point and the retention of only the linear term. Because of neglecting higher-order terms of Taylor series expansion, these neglected terms must be small enough; that is, the variables deviate only slightly from the operating condition [2].

2.8 Speed Control Method of DC Motor

The speed of DC motor can be varied by controlling the field flux, the armature resistance or the terminal voltage that applied to the armature circuit (armature voltage). The three most common speed control methods are field resistance control, armature voltage control, and armature resistance control. [13]

2.8.1 Field Resistance Control Method

In the field resistance control method, a series resistance is inserted in the shunt-field circuit of the motor in order to change the flux by controlling the field current. It is theoretically expected that an increase in the field resistance will result in an increase in the load speed of the motor and in the slope of torque speed curve.

2.8.2 Armature Voltage Control Method

In the armature voltage control method, the voltage applied to the armature circuit, is varied without changing the voltage applied to the field circuit of the motor. Therefore, the motor must be separately excited to use armature voltage control. When the armature voltage is increased, the no-load speed of the motor increases while the slope of torque speed curve remains unchanged since the flux is kept constant.

2.8.3 Armature Resistance Control Method

The armature resistance control is the less commonly used method for speed control in which an external resistance is inserted in series with the armature circuit. An increase in the armature resistance results in a significant increase in the slope of the torque speed characteristic of the motor while the no-load speed remains constant.

2.9 Pulse Width Modulation (PWM)

Pulse-width modulation (PWM), is a modulation technique that conforms the width of the pulse, formally the pulse duration, based on modulator signal information. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors.

The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast pace. The longer the switch is on compared to the off periods, the higher the power supplied to the load is.

The PWM switching frequency has to be much faster than what would affect the load, which is to say the device that uses the power. Typically switching have to be done several times a minute in an electric stove, 120 Hz in a lamp dimmer, from few kilohertz (kHz) to tens of kHz for a motor drive and well into the tens or hundreds of kHz in audio amplifiers and computer power supplies.

The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time. A low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on.[13]

2.10 Proportional Integral Derivative Controller

A proportional integral derivative (PID) controller is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

The PID controller algorithm involves three separate constant parameters as shown is Figure (2.6) and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element , Table (2.1) show the effect of increasing a parameter independently.

or the power supplied to a heating element , Table (2.1) show the effect of increasing a parameter independently.

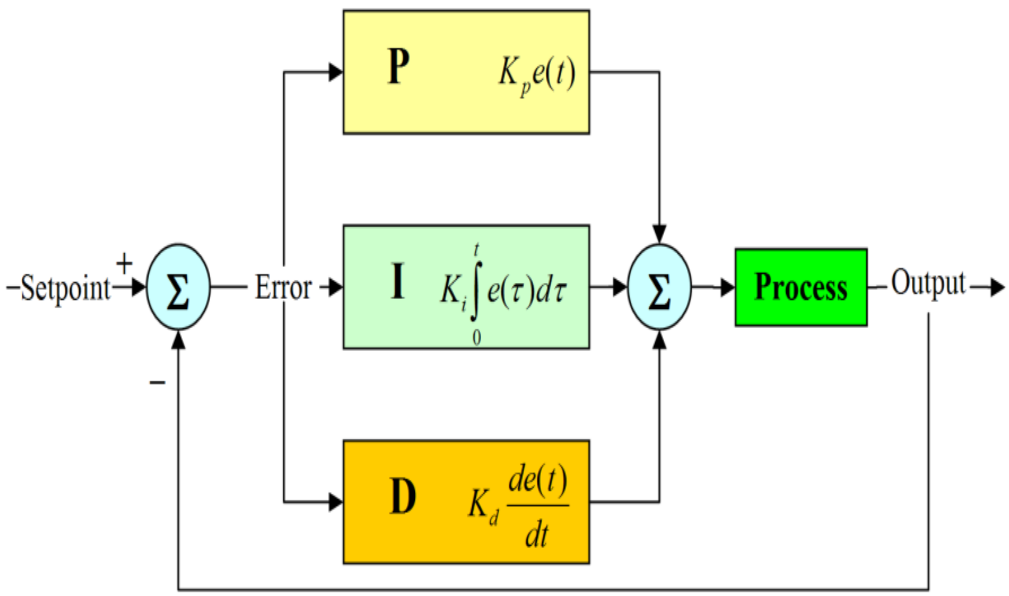


Figure 2.4: Proportional Integral Derivative (PID)

Table (2.1): Effects of increasing a parameter independently

Parameter	**Rise time**	**Overshoot**	**Settling time**	**Steady-state**
$\mathbf{K_P}$	Decrease	Increase	Small change	Decrease
$\mathbf{K_I}$	Decrease	Increase	Increase	Eliminate
$\mathbf{K_D}$	No change	Decrease	Decrease	Small effect

2.10.1 Proportional controller

The Proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p , called the proportional gain constant.

The proportional term is given by:

$$P = K_p (e(t)) \quad (2.1)$$

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances. Tuning theory and industrial practice indicate that the proportional term should contribute the bulk of the output change.

2.10.2 Proportional integral controller

The main function of the integral action is to make sure that the process output agrees with the set point in steady state. With Proportional control, there is normally a control error in steady state. With integral action, small positive error will always lead to an increasing signal , and a negative error will give a decreasing control signal no matter how small the error is.

$$PI = K_p (e(t)) + K_i \int \big(e(t)\big) dt \quad (2.2)$$

2.10.3 Proportional derivative controller

The purpose of the derivative action is to improve the close-loop stability. Because of the process dynamics, it will take some time before a change in the control variable is noticeable in the progress output. Thus, the control system will be late in correction for an error. The action of a controller with proportional and derivative may be interpreted as if the control is made proportional to the predicted process output, where the prediction is made by extrapolating the error by the tangent to the error curve.

$$PD = K_p (e(t)) + K_d \left(\frac{d(e(t))}{dt}\right) \quad (2.3)$$

2.10.4 Proportional integral derivative controller

The Proportional Integral Derivative (PID) controller has three terms. The proportional term (P) corresponds to proportional control. The integral term (I) give a control action that is proportional to the time integral of the zero. The derivative term (D) is proportional to the time derivative of the control error. This term allows prediction of the future error. There are many variations of the PID algorithm that will substantially improve its performance and operability. Those variations are discussed in the next section [14].

$$\text{PID}(t) = K_p(e(t)) + K_i\int\big(e(t)\big)dt + K_d\left(\frac{d(e(t))}{dt}\right) \tag{2.4}$$

By taking Laplace transform PID controller transfer function become:

$$C(s) = Kp + \frac{Ki}{s} + Kd * s \tag{2.5}$$

2.10.5 Tuning of proportional integral derivative controller

The process of selecting the controller parameters to meet given performance specifications is known as controller tuning.

There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, and then choosing P, I, and D based on the dynamic model parameters.

In particular, when the mathematical model of the plant is unknown and therefore analytical design methods cannot be used, PID controls prove to be most useful. In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control.

If a mathematical model of the plant can be derived, then it is possible to apply various design techniques for determining parameters of the controller that will meet the transient and steady-state specifications of the closed-loop system. However, if the plant is so complicated that its mathematical model cannot be easily obtained, then an analytical or computational approach to the design of a PID controller is not possible. Then we must resort to experimental approaches to the tuning of PID controllers[2].

There are many methods of PID tuning such as:

- ✓ Manual tuning.
- ✓ Ziegler-nichols.
- ✓ Tyreus luyben.
- ✓ Cohen-coon.

2.11 Fuzzy logic

One of the most popular new technologies is ‘’ intelligent control ‘’ which is defined as a combination of control theory, operations research, and artificial intelligence (AI).judging by the billions of dollars’ worth of sales and thousands of patents issued worldwide, led by Japan since the a noun cement of the first fuzzy chips in 1987, fuzzy logics still perhaps the most popular area in AI.

To understand fuzzy logic it is important to discuss fuzzy sets .in 1965 ,Zadeh wrote a seminal paper in which he introduced fuzzy sets ,that is ,sets with un sharp boundaries .these sets are generally in better agreement with the human mind and reasoning that works with shades of gray ,rather than with just black or white .fuzzy sets are typically able to represent linguistic terms ,for examples ,warm ,hot, high ,low ,close ,far ,etc. Nearly 190 years later (in 1974) ,Japan ,united states ,Europe ,Asia ,and many other part of the world ,fuzzy control is widely accepted and applied .

Conventional set theory distinguishes between those elements that are members of a set and those are not, there being very, clear or crisp boundaries. In a fuzzy set we name all the elements of the universe and supplement to them a number between 0and 1.This number demonstrates to what degree this generic element belongs to the defined fuzzy set. Actually, according to this definition we just add to every element a number which constitutes the membership degree of this element. Actually, a fuzzy set is given by its membership function. The value of this function determines if the element belongs to the fuzzy set and in what degree [15].

2.11.1 Linguistic variables

To specify rules for the rule-base, the expert will use a "linguistic description"; hence, linguistic expressions are needed for the inputs and outputs and the characteristics of the inputs and outputs. "Linguistic variables" is used (constant symbolic descriptions of what are in general time-varying quantities)to describe fuzzy system inputs and outputs. For our fuzzy system, linguistic variables denoted by ~ui are use to describe the inputs ui .Similarly, linguistic variables denoted by~yi are used to describe outputs yi .For instance, an input to the fuzzy system may be described as ~u1 ="position error "or ~u2 ="velocity error", and an output from the fuzzy system may be ~y1 ="voltage in" [16].

2.11.2 Membership Functions:

The simplest membership functions are formed using straight lines. Of these, the simplest is the triangular membership function, and it has the function name trimf. It is nothing more thana collection of three points forminga triangle. The trapezoidal membership function, trapmf, has aflat top and really is just a truncated triangle curve as showninfigure (2.5).

These straight line membership functions have the advantage of simplicity.

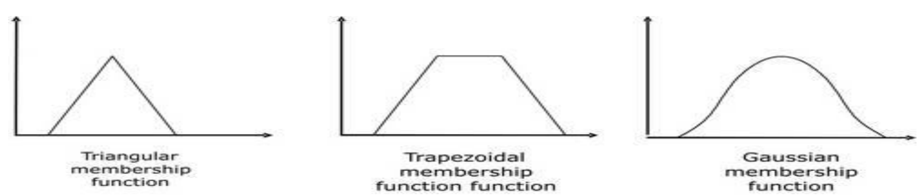


Figure (2.5): Membership function

2.11.3 Fuzzy Sets Operations

There are three types of fuzzy set operation:

- ✓ **Union**

The membership function of the Union of two fuzzy sets A and B with membership functions µ A and µ B respectively is defined as the maximum of the two individual membership functions .This is called the maximum criterion.The Union operation in Fuzzy set theory as shown is Figure (2.6)

µAB=max (µA, µB)

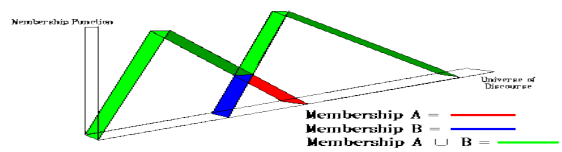


Figure (2.6):The Union operation in Fuzzy set theory

The union operation in Fuzzy set theory is the equivalent of the OR operation in Boolean algebra.

- ✓ **Intersection**

The membership function of the intersection of two fuzzy sets A and B with membership functions µ A and µ B respectively is defined as the minimum of the two individual membership functions. This is called the minimum criterion. The Intersection operation in Fuzzy set theory as shown is Figure (2.7)

µÂB=min (µA, µB)

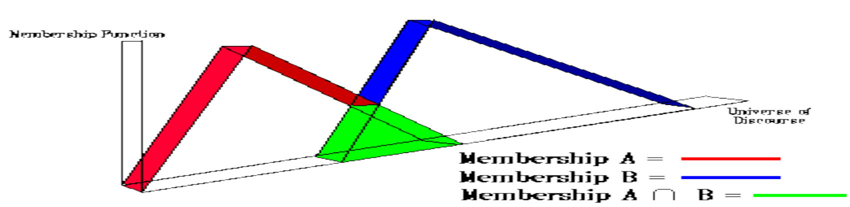


Figure (2.7): The Intersection operation in Fuzzy set theory

The intersection operation in Fuzzy set theory is the equivalent of the AND operation in Boolean algebra.

- ✓ **Complement**

The membership function of the complement of a fuzzy set with membership function μ A is defined as the negation of the specified membership function. The complement operation in Fuzzy set theory as shown is Figure (2.8)
This is called the negation criterion.

$\mu\bar{A}=1-\mu A$

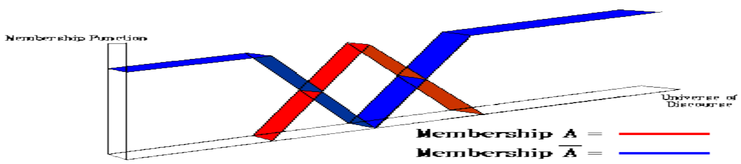


Figure (2.8):The complement operation in Fuzzy set theory

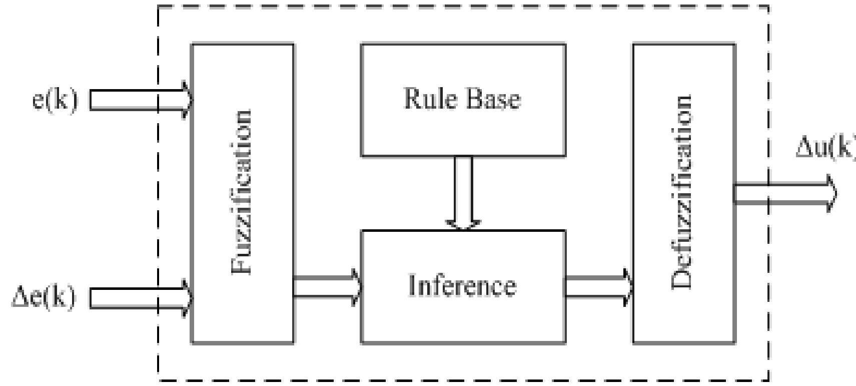
The complement operation in Fuzzy set theory is the equivalent of the NOT operation in Boolean algebra.

2.11.4 The Basics of Fuzzy logic

A fuzzy control is a controller that is intended to manage some vaguely known or vaguely described process. The controller can be used with the process in two modes:

- ✓ Feedback mode when the fuzzy controller acts as a control device.
- ✓ Feed forward mode where the controller can be used as a prediction device.

Illustrates the basic components of fuzzy logic controller, as shown is Figure (2.9)



Figure(2.9): Fuzzy Logic Control System

The plant output are denoted by Δu(k), the input are denoted by e(k), and the reference input to the fuzzy controller is denoted by Δe(k).

The fuzzy controller has for main components:

- ✓ **Fuzzification**

The first block of the fuzzy controller is fuzzification, which converts each piece of input data to degrees of membership by a look up in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable.

- ✓ **Inference mechanism**

Inference mechanism or engine is the processing program in a fuzzy control System. It derives a conclusion from the facts and rules contained in the knowledge base using various human expert techniques.

- ✓ **Rule-base**

A group of rules may use several variables both in the condition and the conclusion of the rules. They are based on a set of rules that a human expert would follow in diagnosing a problem. Rule-base also where the knowledge is stored.

- ✓ **Defuzzification**

Defuzzification is a process that maps a fuzzy set to a crisp set and has attracted far less attention than other processes involved in fuzzy systems and technologies. Four most common defuzzification methods are:

- ✓ Maximum membership method.
- ✓ Center of gravity method.
- ✓ Weight average method.

2.12 Microcontrollers

It is a highly integrated chip that contains all the components comprising a controller. Typically, this includes a CPU, RAM, ROM and I/O ports. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production cost.

2.12.1 History of microcontroller

The first computer system on a chip optimized for control applications was the Intel 8048 microcontroller with both RAM and ROM on the same chip. Most microcontrollers at that time had two variants; one had an erasable EEPROM program memory, which was significantly more expensive than the PROM variant which was only programmable once.

The introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16x84) to be electrically erased quickly without an expensive package as required for EPROM. The same year, Atmel introduced the first microcontroller using Flash memory. Other companies rapidly followed suit, with both memory types. Nowadays microcontrollers are low cost and readily available for hobbyists, with large online communities around certain processors.

Microcontrollers have traditionally been programmed using the assembly language of the target device. Although the assembly language is fast.The microcontrollers manufactured by different firms have different assembly languages, so the user must learn a new language with every new microcontroller he or she uses.

Microcontrollers can also be programmed using a high-level language, such as BASIC, PASCAL, or C. High-level languages are much easier to learn than assembly languages and also facilitate the development of large and complex programs [17].

2.12.2 Microcontroller application

Microcontroller applications found in many lives filed, for example in Cell phone, watch, recorder, calculators, mouse, keyboard, modem, fax card, sound card, battery charger, door lock, alarm clock, thermostat, air conditioner, TV Remotes, industrial equipment like Temperature and pressure controllers, counters and timers.

2.12.3 Arduino microcontroller

Arduino is a small microcontroller board with a USB plug to connect to your computer and a number of connection sockets that can be wired up to external electronics, such as motors, relays, light sensors, laser diodes, loudspeakers, microphones, etc. Arduino can either be powered through the USB connection from the computer or from a 9V battery. Arduino can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently Arduino microcontroller board is shown in figure (2.10) [18].

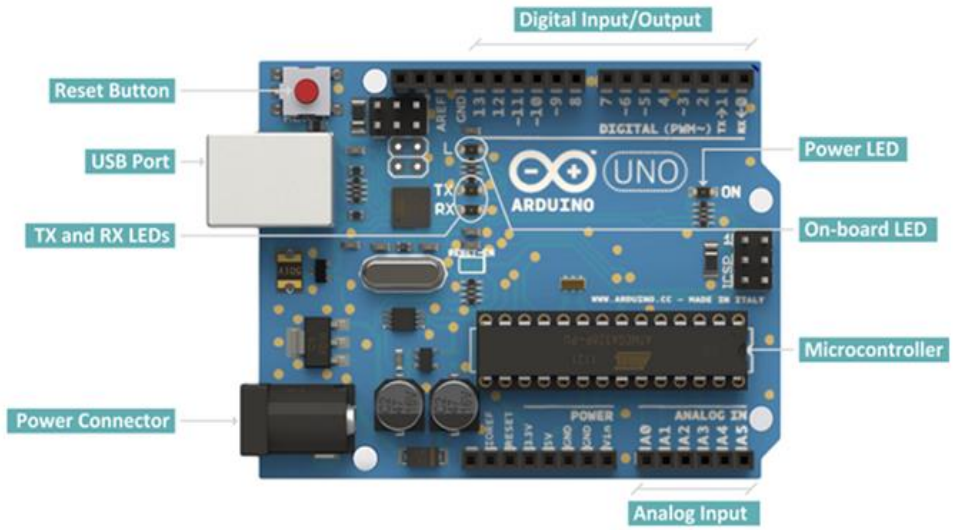


Figure (2.10): Arduino microcontroller board

2.12.4 The Arduino board

It is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating inter-active objects or environments in simple terms, the Arduino is a tiny computer system that can be programmed with your instructions to interact with various forms of input and output. The current Arduino board model, the Uno, is quite small in size compared to the average human hand, as shown in Figure (2.10).

Although it might not look like much to the new observer, the Arduino system allows creating devices that can interact with the world. By using an almost unlimited range of input and output devices, sensors, indicators, displays, motors, and more, the exact interactions required to create a functional device can be programmed. For example, artists have created installations with patterns of blinking lights that respond to the movements of passers-by, high school students have built autonomous robots that can detect an open flame and extinguish it, and geographers have designed systems that monitor temperature and humidity and transmit this data back to their offices via text message. In fact, there are infinite numbers of examples with a quick search on the Internet.

By taking a quick tour of the Uno Starting at the left side of the board there are two connectors, as shown in Figure (2.11)

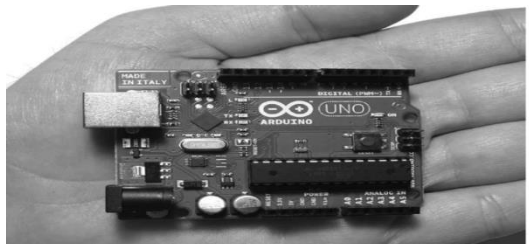


Figure (2.11): The USB and power connectors

On the far left is the Universal Serial Bus (USB) connector. This connects the board to your computer for three reasons; to supply power to the board, to upload the instructions to the Arduino, and to send data to and receive it from a computer. On the right is the power connector, this connector can power the Arduino with a standard mains power adapter.

At the lower middle is the heart of the board: the microcontroller, as Shown in Figure (2.12).

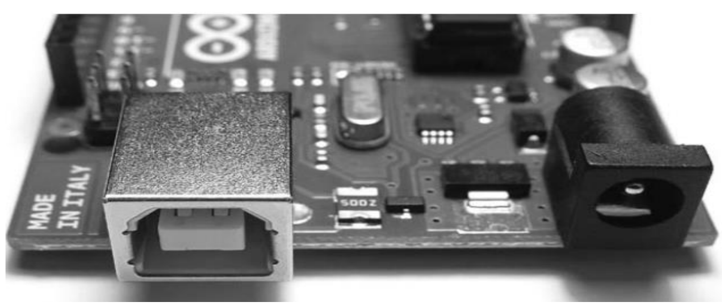


Figure (2.12): The microcontroller

The microcontrollers represent the “brains” of the Arduino. It is a tiny computer that contains a processor to execute instructions, includes various types of memory to hold data and instructions from our sketches, and provides various avenues of sending and receiving data. Just below the microcontroller are two rows of small sockets, as shown in Figure (2.13).

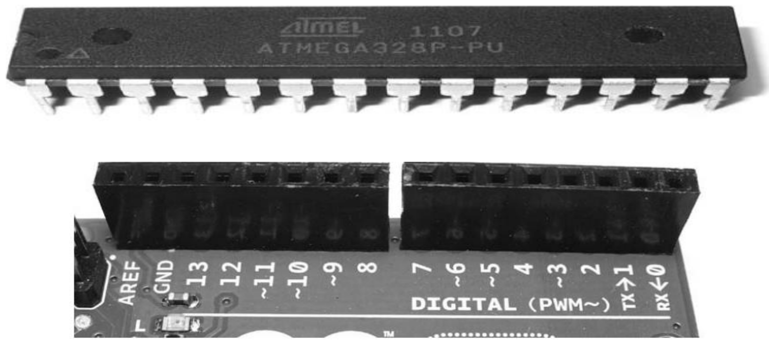


Figure (2.13): The power and analog sockets

The first row offers power connections and the ability to use an external RESET button. The second row offers six analog inputs that are used to measure electrical signals that vary in voltage. Furthermore, pins A4 and A5 can also be used for sending data to and receiving it from other devices. Along the top of the board are two more rows of sockets, as shown in Figure (2.14).

Figure (2.14): The digital input/output pins

Sockets (or pins) numbered 0 to 13 are digital input/output (I/O) pins. They can either detect whether or not an electrical signal is present or generate a signal on command. Pins 0 and 1 are also known as the serial port, which is used to send and receive data to other devices, such as a computer via the USB connector circuitry. The pins labeled with a tilde (~) can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors.

Next are some very useful devices called light-emitting diodes (LEDs); these very tiny devices light up when a current pass through them. The Arduino board has four LEDs: one on the far right labeled ON, which indicates when the board has power, and three in another group, as shown in Figure (2.15).

The LEDs labeled TX and RX light up when data is being transmitted or received between the Arduino and attached devices via the serial port and USB. The L-LED connected to the digital I/O pin number 13. The little black square part to the left of the LEDs is a tiny microcontroller that controls the USB interface that allows Arduino to send data to and receive it from a computer [19].

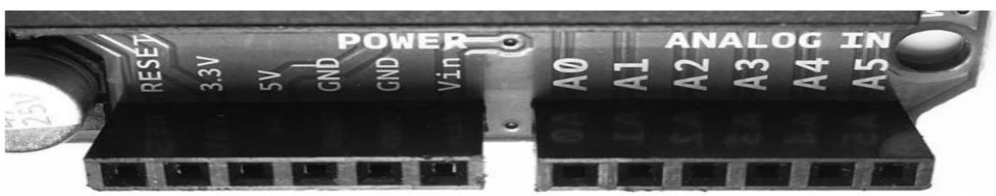


Figure (2.15): The onboard LEDs

And, finally, the RESET button is shown in Figure (2.16).

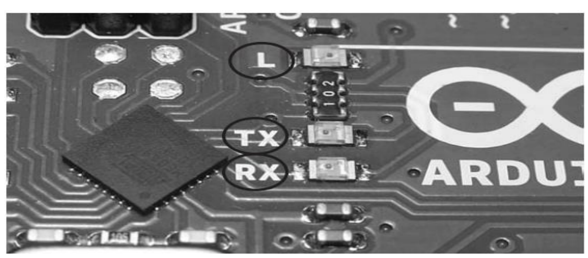


Figure (2.16): The RESET button.

CHAPTER THREE

MODEL DESIGN AND METHODOLOGY

3.1 Introduction

using microcontroller to implement a FL-PID controller is inexpensive and the physical size of the system is small; however, it is suitable for movable equipment, but the FL-PID controller requires longer processing time to complete the performance [8].

Based on advantages of higher level microcontroller device, microchip PIC18F family can be used to integrate large amounts of code in a single IC. PIC microcontrollers are one of the fastest growing parts of the embedded integrated circuit market in recent times [20]. PIC18F2550 is suitable for fast implementation controller and can be programmed to process any type of digital functions and for motor control applications.

3.2 Components

The design of the following model is required:

- ✓ DC motor.
- ✓ Arduino.
- ✓ Optocoupler.
- ✓ MOSFET transistor.
- ✓ LCD screen.
- ✓ Diode.
- ✓ Push buttons.
- ✓ Resistors.
- ✓ U-board.
- ✓ Potentiometer.

3.3 components specifications

define components separately.

3.3.1 Motor Dc

A permanent magnet (pm) brushless can produce magnetic field in air gap with no excitation winding and no dissipation of electric power.

permanent magnet is no electrical energy is absorbed by the field excitation system and thus there are no excitation losses which means substantial increase in the efficiency. [3]

3.3.1.1 Physical Description

A permanent magnet DC motor is a mechanism which converts electrical power to mechanical power via magnetic coupling. The electrical power is provided by a voltage source, while the mechanical power is provided by a spinning rotor. A very basic DC motor is constructed of two main components: the rotor or armature and the stator. The armature rotates within the framework of the stationary stator. A simple illustration of a dc motor is given in Fig. (3.1): [3]

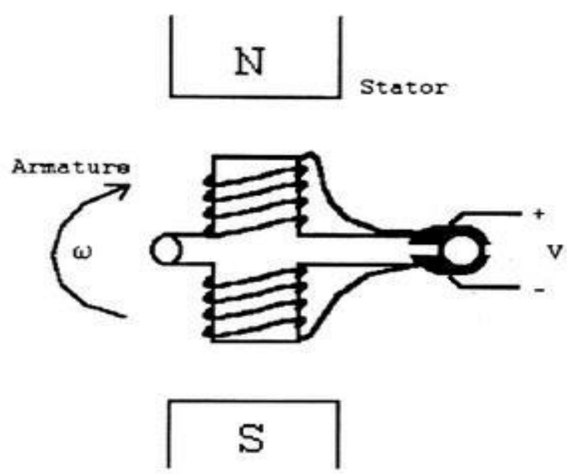


Figure (3.1): Simple DC motor

The stator consists of permanent magnets which create a magnetic field. The armature consists of an electromagnet created by a coil wound around an iron core. The armature rotates due to the phenomenon of attracting and opposing forces of the two magnetic fields. A magnetic field is generated by the armature by sending an electrical current through the coil and the polarity is constantly changed by alternating the current through the coil (also known as commutation) causing the armature to rotate (done electronically).

3.3.1.2 DC Motor Modelling

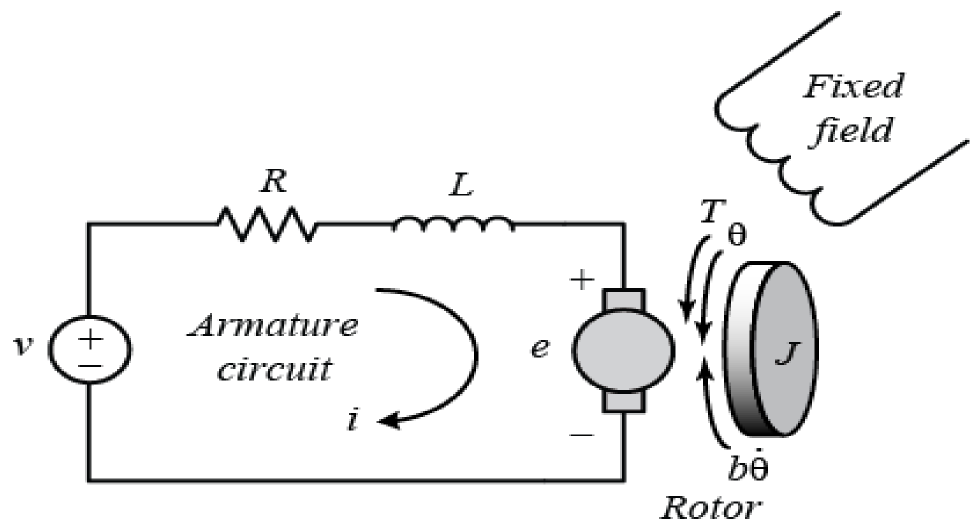


Figure (3.2): Equivalent circuit of DC motor

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. Assume that the magnetic field is constant and, therefore, that the motor torque (T) is proportional to only the armature current (i) by a constant factor (K_t) called torque constant as shown in the Equation (3.2). This is referred to as an armature-controlled motor.

$$T = K_t\, i \tag{3.1}$$

The back electromotive forceemf (e) is proportional to the angular velocity of the shaft ($\dot{\theta}$) by a constant factor (K_b) called electromotive force constant as shown in the Equation (3.2).

$$e = K_b\dot{\theta} \quad (3.2)$$

In system international units, the motor torque and back emf constants are equal, that is, $K_t = K_b$; therefore, (K) will be used to represent both the motor torque constant and the back emf constant.

By using Figure (3.2) to derive the following governing equations based on Newton's 2nd law in Equation(3.3)and Kirchhoff's voltage law in Equation (3.4).

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (3.3)$$

$$La\frac{di}{dt} + Ra * i = V - K\dot{\theta} * b \quad (3.4)$$

Where (J) is the moment of inertia of the rotor, (b) is the motor viscous friction constant, (La) is the electric inductance, (Ra) is the electric resistance, and (V) is the voltage source.

Applying the Laplace transform, the modeling equations can be expressed in terms of the Laplace variables as shown inEquation (3.5) and Equation (3.6).

$$s(Js + b)\theta(s) = KI(s) \quad (3.5)$$

$$(Las + Ra)I(s) = V(s) - Ks\theta(s) \quad (3.6)$$

Following open-loop transfer function had arrived by eliminating I(s) in Equations (3.7), where the rotational speed is considered the output and the armature voltage is considered the input.

$$G(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Las + Ra) + K^2} \ \left[\frac{rad/sec}{V}\right] \quad (3.7)$$

Table (3.1): DC motor parameter

Kt	0.01 N.m/A
Kb	0.01V/rad/s
Ra	1Ω
B	0.1N.m/rad/s

However, during this model continue looking at the position as the output. The position has been obtained by integrating the speed; therefore, it needed to divide the transfer function in Equation (3.8).

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((Js+b)(Las+Ra)+K^2)} \quad \left[\frac{rad}{V}\right] \tag{3.8}$$

The position $\theta(s)$ in Equation (3.8) is the rotational displacement which produced on the motor's shaft but there are gears between shaft and load with ratio known as gears ratio (K_g) , so the transfer function becomes as shown in Equation (3.9).

$$\frac{\theta(s)}{V(s)} = \frac{K*Kg}{s((Js+b)(Las+Ra)+K^2)} \left[\frac{rad}{V}\right] \tag{3.9}$$

The physical parameters of the DC motor used for simulation testing are given in table (3.1) . Substitute there parameters in equation (3.9) , the transfer function becomes as follow:

$$\frac{\theta m(s)}{Ea(s)} = \frac{3.839}{0.004s2 + 0.34s + 1}$$

3.3.2Arduino Uno

Arduino Uno as shown in Figure (3.3) is a microcontroller board based on ATmega328P microcontroller. It has 14 digital input/output pins (of which 6 can be used as pulse width modulation (PWM) outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, header and a reset button. Using PID library in Arduino Uno code (See Appendix A) as a PID controller has many benefits such as:

- ✓ There are many ways to write the PID algorithm. A lot of time was spent making the algorithm in this library as solid as any found in industry.
- ✓ When using the library all the PID code is self-contained. This makes your code easier to understand. It also lets you do more complex stuff, like having 8 PIDs in the same program.

Figure (3.3): Arduino Uno

3.3.3 Optocoupler

In electronics, an opto-isolator, also called an optocoupler, photocoupler, or optical isolator, is a component that transfers electrical signals between two isolated circuits by using light. Opto-isolators prevent high voltages from affecting the system receiving the signal. Commercially available opto-isolators withstand input-to-output voltages up to 10 kV and voltage transients with speeds up to 10 kV/μs.

A common type of opto-isolator consists of an LED and a phototransistor in the same opaque package. Other types of source-sensor combinations include LED-photodiode, LED-LASCR, and lamp-photoresistor pairs. Usually opto-isolators transfer digital (on-off) signals, but some techniques allow them to be used with analog signals.[21]

3.3.3.1 Operation

An opto-isolator contains a source (emitter) of light, almost always a near infrared light-emitting diode (LED), that converts electrical input signal into light, a closed optical channel (also called dielectrical channel), and a photosensor, which detects incoming light and either generates electric energy directly, or modulates electric current flowing from an external power supply. The sensor can be a photoresistor, a photodiode, a phototransistor, a silicon-controlled rectifier (SCR) or a triac.

Because LEDs can sense light in addition to emitting it, construction of symmetrical, bidirectional opto-isolators is possible. An optocoupled solid-state relay contains a photodiode opto-isolator which drives a power switch, usually a complementary pair of MOSFETs. A slotted optical switch contains a source of light and a sensor, but its optical channel is open, allowing modulation of light by external objects obstructing the path of light or reflecting light into the sensor.[21]

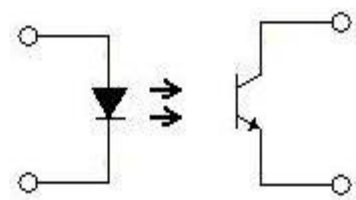


Figure (3.4): Optocoupler

3.3.4 MOSFET Transistor

The metal–oxide–semiconductor field-effect transistor (MOSFET, MOS-FET, or MOS FET) is a type of field-effect transistor (FET). It has an insulated gate, whose voltage determines the conductivity of the device. This ability to change conductivity with the amount of applied voltage can be used for amplifying or switching electronic signals. A metal–insulator–semiconductor field-effect transistor or MISFET is a term almost synonymous with MOSFET. Another synonym is IGFET for insulated-gate field-effect transistor.

The basic principle of the field-effect transistor was first patented by Julius Edgar Lilienfeld in 1925.

The main advantage of a MOSFET is that it requires almost no input current to control the load current, when compared with bipolar transistors. In an "enhancement mode" MOSFET, voltage applied to the gate terminal increases the conductivity of the device. In "depletion mode" transistors, voltage applied at the gate reduces the conductivity.[21]

The "metal" in the name MOSFET is now often a misnomer because the gate material is often a layer of polysilicon (polycrystalline silicon). "Oxide" in the name can also be a misnomer, as different dielectric materials are used with the aim of obtaining strong channels with smaller applied voltages. The MOSFET is by far the most common transistor in digital circuits, as hundreds of thousands or millions of them may be included in a memory chip or microprocessor. Since MOSFETs can be made with either p-type or n-type semiconductors, complementary pairs of MOS transistors can be used to make switching circuits with very low power consumption, in the form of CMOS logic.

3.3.4.1 Operation

The traditional metal–oxide–semiconductor (MOS) structure is obtained by growing a layer of silicon dioxide (SiO2) on top of a silicon substrate and depositing a layer of metal or polycrystalline silicon (the latter is commonly used). As the silicon dioxide is a dielectric material, its structure is equivalent to a planar capacitor, with one of the electrodes replaced by a semiconductor.

When a voltage is applied across a MOS structure, it modifies the distribution of charges in the semiconductor. If we consider a p-type semiconductor (with {\displaystyle N_{A}} N_{A} the density of acceptors, p the density of holes; p = NA in neutral bulk), a positive voltage, {\displaystyle V_{GB}} V_{GB}, from gate to body (see figure) creates a depletion layer by forcing the positively charged holes away from the gate–insulator/semiconductor interface, leaving exposed a carrier-free region of immobile, negatively charged acceptor ions (see doping (semiconductor)). If {\displaystyle V_{GB}} V_{GB} is high enough, a high concentration of negative charge carriers forms in an inversion layer located in a thin layer next to the interface between the semiconductor and the insulator. Conventionally, the gate voltage at which the volume density of electrons in the inversion layer is the same as the volume density of holes in the body is called the

threshold voltage. When the voltage between transistor gate and source (VGS) exceeds the threshold voltage (Vth), it is known as overdrive voltage.
This structure with p-type body is the basis of the n-type MOSFET, which requires the addition of n-type source and drain regions.

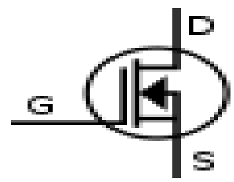


Figure (3.5): Mosfet Transistor

3.3.5 LCD Screen

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.[21] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.
LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma

displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big-screen television sets.

Since LCD screens do not use phosphors, they do not suffer image burn-in when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft schedule on an indoor sign). LCDs are, however, susceptible to image persistence.[21] The LCD screen is more energy-efficient and can be disposed of more safely than a CRT can. Its low electrical power consumption enables it to be used in battery-powered electronic equipment more efficiently than CRTs can be. By 2008, annual sales of televisions with LCD screens exceeded sales of CRT units worldwide, and the CRT became obsolete for most purposes.

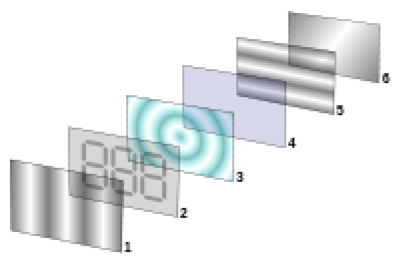


Figure (3.6): LCD Screen

3.3.6 Diode

In electronics, a diode is a two-terminal electronic component that conducts primarily in one direction (asymmetric conductance); it has low (ideally zero) resistance to the current in one direction, and high (ideally infinite) resistance in the other. A semiconductor diode, the most common type today, is a crystalline piece of semiconductor material with a p–n junction connected to two electrical terminals. A vacuum tube diode has two electrodes, a plate (anode) and a heated cathode. Semiconductor diodes were the first semiconductor electronic devices. The discovery of crystals' rectifying abilities was made by German physicist Ferdinand Braun in 1874. The first semiconductor diodes, called cat's whisker diodes, developed around 1906, were made of mineral crystals such as galena. Today, most diodes are made of silicon, but other semiconductors such as selenium and germanium are sometimes used.

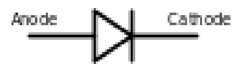


Figure (3.7): Diode

3.3.7 U-Board

A single-board microcontroller is a microcontroller built onto a single printed circuit board. This board provides all of the circuitry necessary for a useful control task: a microprocessor, I/O circuits, a clock generator, RAM, stored program memory and any necessary support ICs. The intention is that the board is immediately useful to an application developer, without requiring them to spend time and effort to develop controller hardware.

As they are usually low-cost, and have an especially low capital cost for development, single-board microcontrollers have long been popular in education. They are also a popular means for developers to gain hands-on experience with a new processor family. [21]

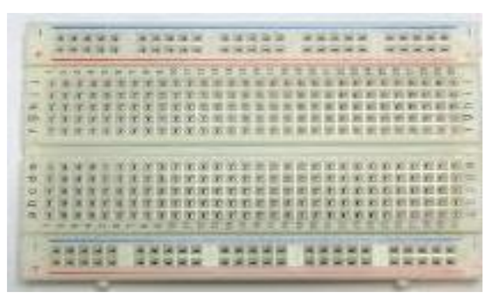


Figure (3.8) : U-Board

3.3.8 Potentiometer

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider.[21] If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

Figure (3.9) : potentiometer

3.3.9 Resistors

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in electronic equipment. Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within integrated circuits.

The electrical function of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than nine orders of magnitude. The nominal value of the resistance falls within the manufacturing tolerance, indicated on the component.[21]

Figure (3.10) : Resistor

3.3.10 Push Button

A push-button (also spelled pushbutton) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal.[21] The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, although many un-biased buttons (due to their physical nature) still require a spring to return to their un-pushed state. Different people use different terms for the "pushing" of the button, such as press, depress, mash, hit, and punch .

Figure (3.11) : push button

3.4 Block diagram

Components can be represented in the following block diagram:

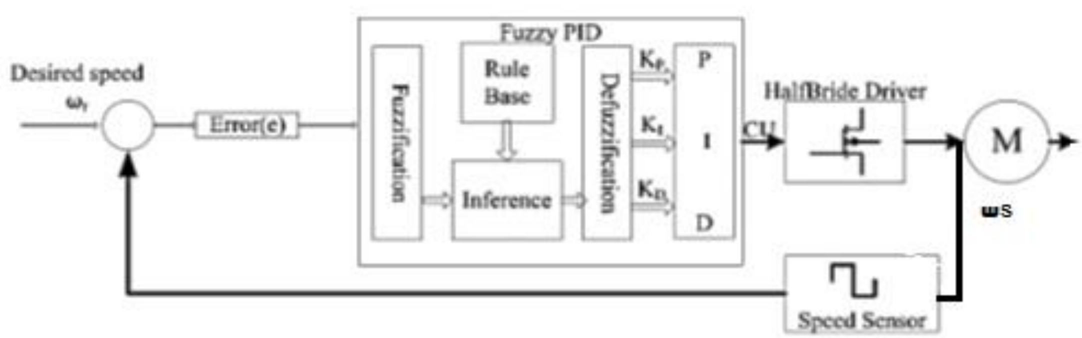


Fig (3.12) Block diagram of Fuzzy PID controller

CHAPTER FOUR
SIMULATION & PRACTICAL

4.1 Introduction

Computer simulations have become a useful part of mathematical modeling of many natural systems to observe their behavior. It allows the engineer to test the design before it is built for real. The simulation for the project were performed in PROTEUS program SIMULINK. These software applications are widely used in control engineering, for both simulation and design.

Beside simulation we had implemented a real circuit with real components, to help us in real life industrial applications, and to fix the errors those faced us in simulation.

4.2 Simulation of DC Motor Using MATLAB

After entering the transfer function in the MATLAB program in the previous chapter, the following responses are presented:

4.2.1 Response of Dc motor without controller

shows the SIMULINK block diagram of DC motor without controller.

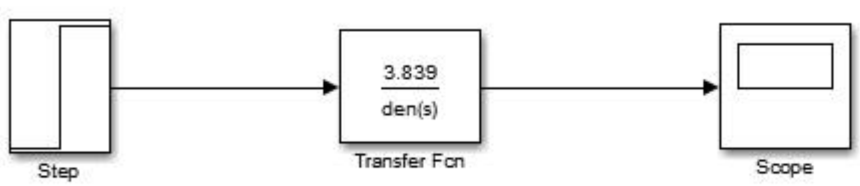


Figure (4.1) : the SIMULINK block diagram of DC motor without Controller

The step response of DC motor without controller as show in Fig (4.2).

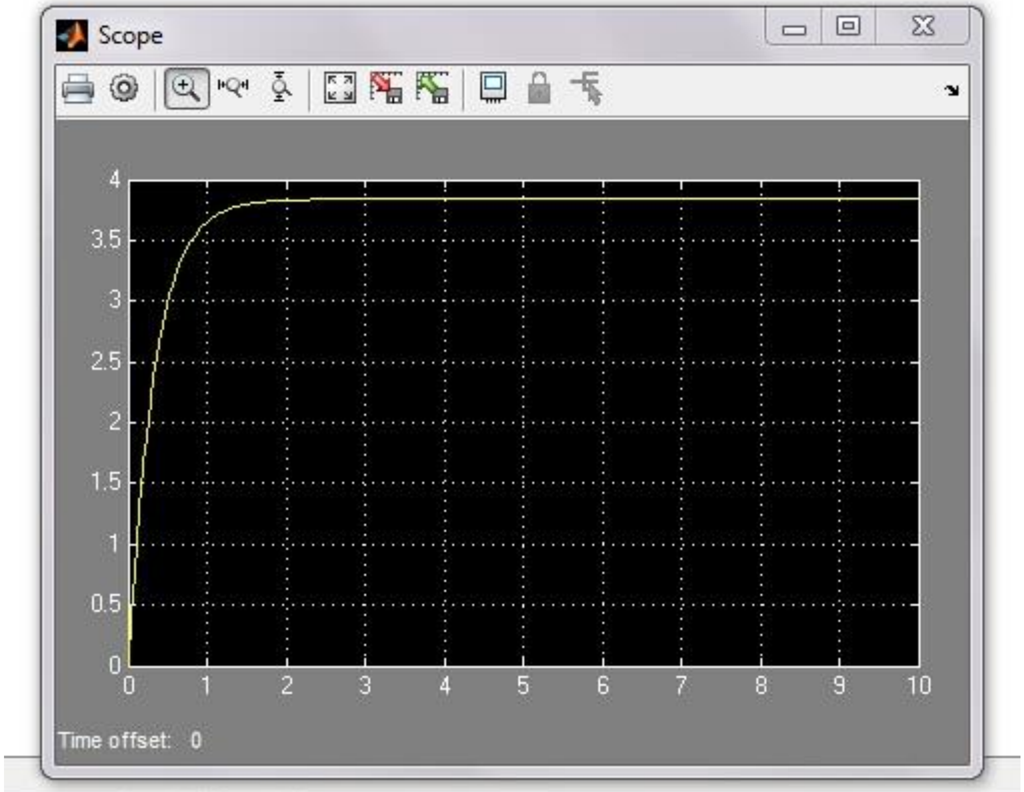


Figure (4.2): the step response of DC motor without controller

4.2.2 Simulation result is presented to evaluate the effectiveness of the PID controller.

Figure (4.3) shows the SIMULINK block diagram of position control of DC motor using PID controller .the PID controller gain selected as Kp=4.7198 ,Ki=54.2762 ,Kd=0.06912 . Figure (4.4) shows the step response of PID controller.

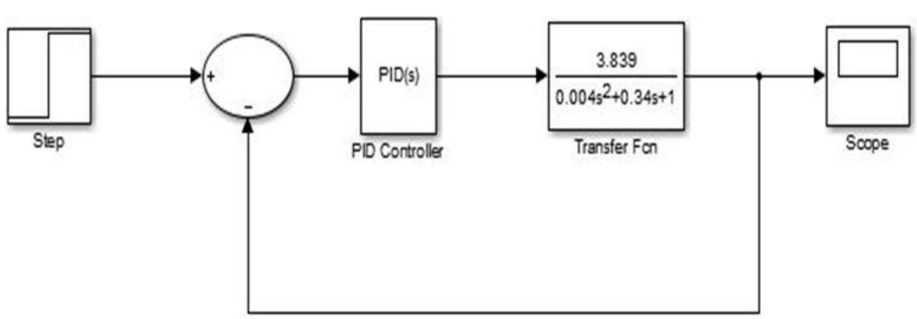


Figure (4.3): the Simulink block diagram of PID controller

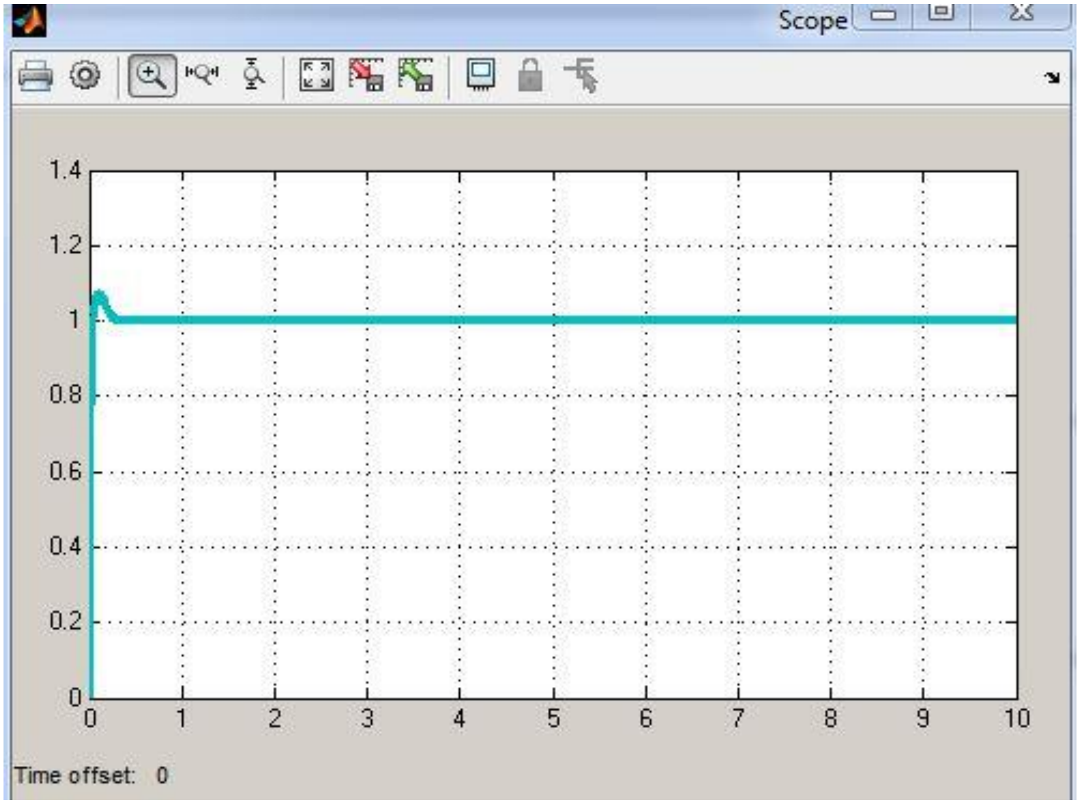


Figure (4.4): the step response of PID controller

4.2.3Simulation of (5*5) Fuzzy Logic Controller

Figure (4.5) shows the Simulink block diagram of (5*5) fuzzy logic controller. Figure (4.6) shows the step response of (5*5) fuzzy logic controller

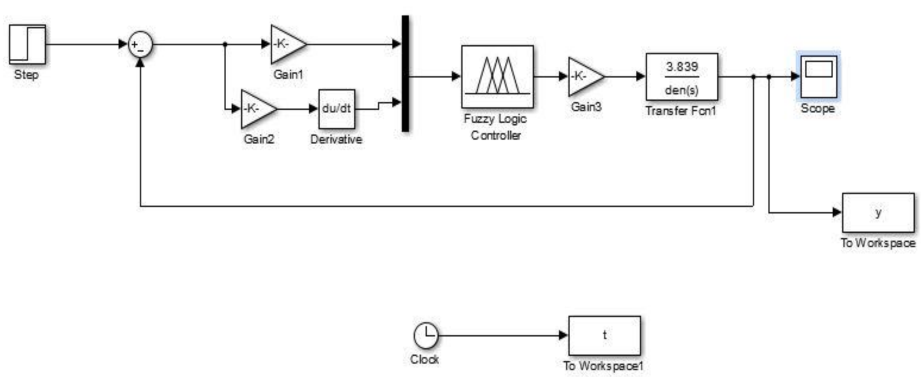


Figure (4.5): the block diagram of (5*5) fuzzy logic controller

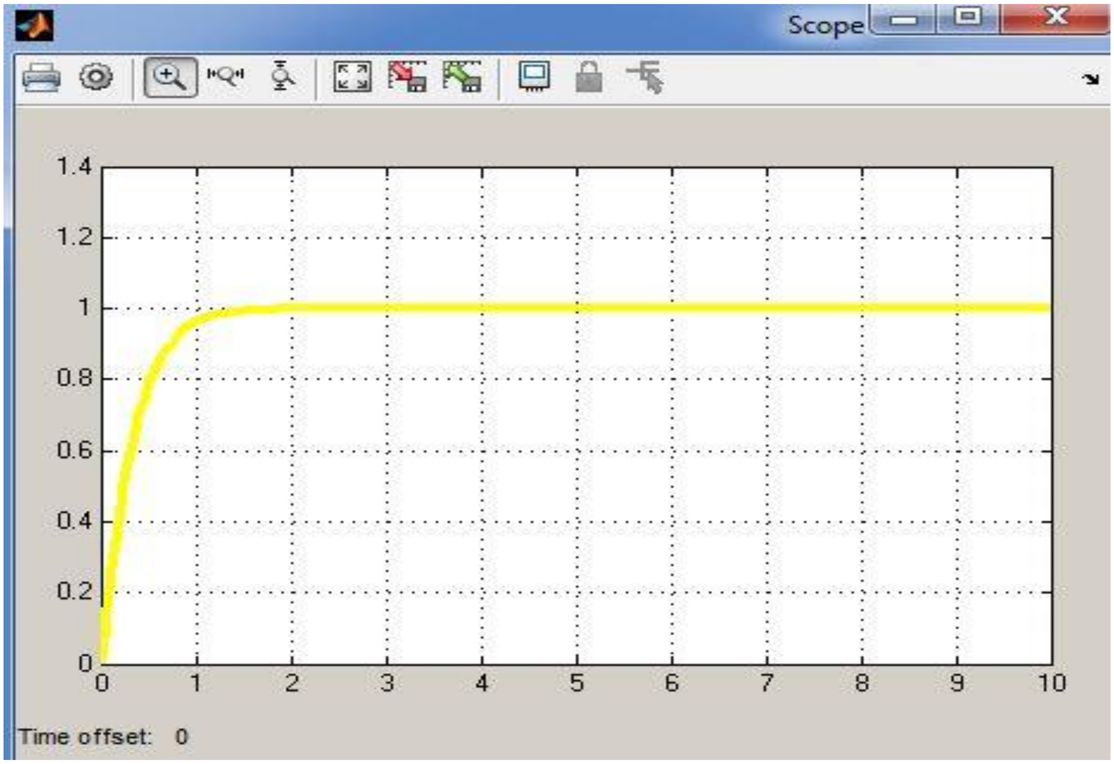


Figure (4.6): the step response of (5*5) fuzzy logic controller

4.2.4 Simulation of DC motor by using two controllers (PID and FLC)

Figure (4.7) shows the Simulink block diagram of position control system by using two controllers. Figure (4.8) show the step response of position control system using two controllers

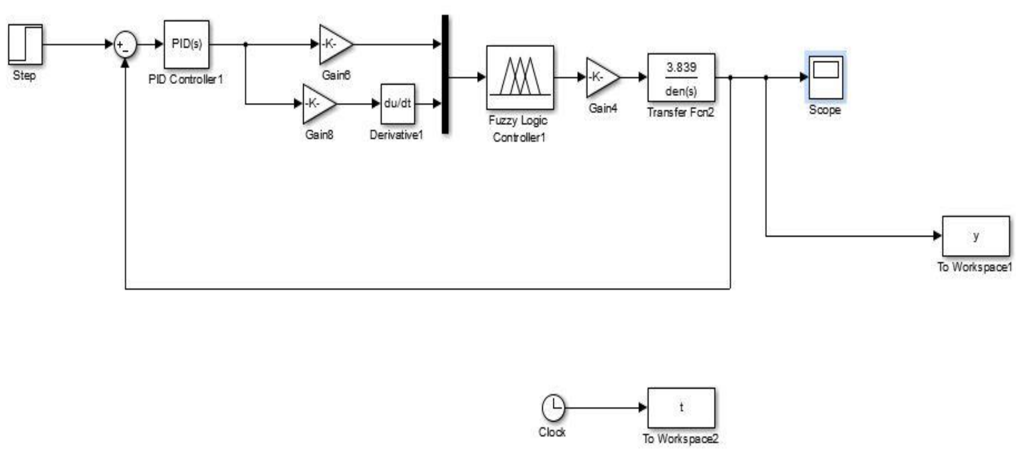


Figure (4.7): the Simulink block diagram of PID with fuzzy controller

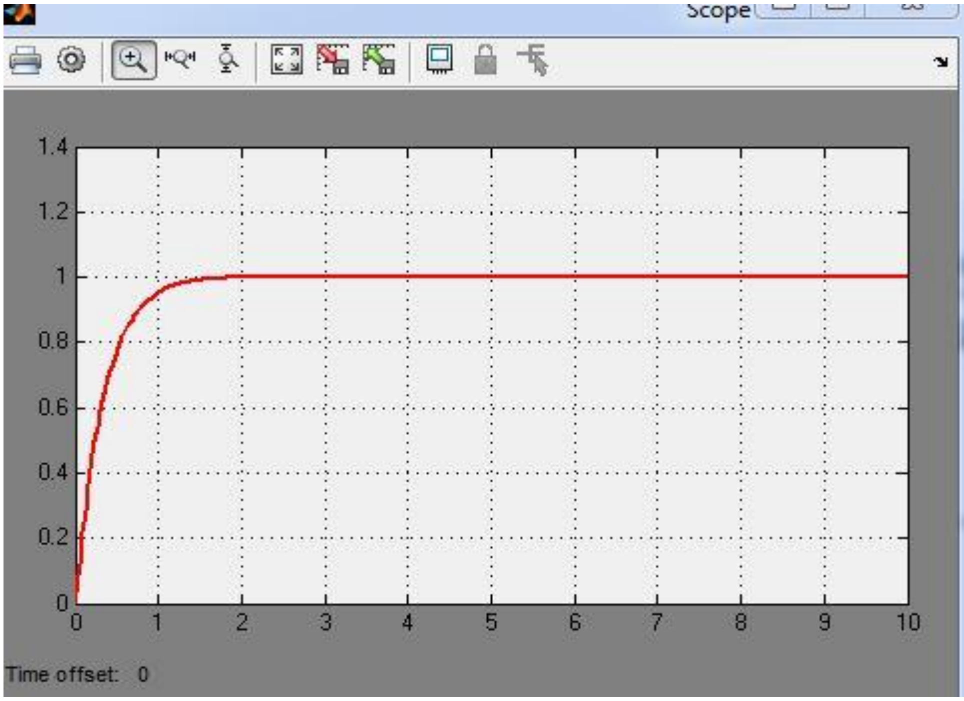


Figure (4.8): the step response of position control by using PID with fuzzy controller

4.2.5 Comparison and Discussion

In order to validate the control strategies as described above, digital simulation was carried out on a converter DC motor drive system whose parameters are given in pervious chapter. The MATLAB/SIMULINK models of system under study with the two controllers are shown in figure (4.4), and (4.6). First a comparison has been made between the maximums overshoot, rise times and settling time illustrated in table (4.1)

Table (4.1) comparison between, maximums overshoot, rise times and settling time

Controller	Ts(sec)	Tr(sec)	Overshoot MP
PID	0.218	0.0344	8.87%
Fuzzy	1.34	0.721	0%

4.3 Software Program

Proteus combines ease of use with powerful features to help us design, test and layout professional PCBs like never before. With nearly 800 microcontroller variants ready for simulation straight from the schematic, one of the most intuitive professional PCB layout packages on the market and a world class shape based auto router included as standard, Proteus Design Suite 8 delivers the complete software package for today and tomorrow's engineers.

Proteus VSM Simulation was the world's first schematic based micro-controller simulation tool and quickly became a de-facto standard for teaching embedded systems within education. Today, we support more processor families along with more embedded peripherals and more technologies than any other tool on the market and we remain world leaders in the field.

The Proteus PCB Design and Layout tools have successfully served both commercial and educational needs for over twenty-five years. Students benefit from exposure to professional grade tools with an intuitive user interface and a quick learning curve.

From China and India, through South America and the USA, and across the UK and Europe, the Proteus Design Suite is trusted as the tool of choice for embedded engineering and electronics learning.

The Proteus schematic capture program is an experimental canvas for students. Placing and wiring is very intuitive and with tens of thousands of components to simulate, curiosity and creativity can be encouraged in equal measure. Together with our world class mixed-mode SPICE simulation engine Proteus provides a safe, fast and immersive learning environment for students.

The ability to interact with a running simulation in Proteus by pressing buttons, ramping POTs or flicking switches makes it ideally suited for engaging students in learning electronic theory.

At introductory levels, simple animations for voltage levels on pins and current flow can be turned on to help students visualize what is happening. As students advance they can use basic meters to take measurements and then be introduced to instrumentation such as an oscilloscope or logic analyzer for analysis. Advanced students can then work with more complex circuitry and use graphs to perform a host of more detailed analyses such as frequency, Fourier or distortion. Microcontroller simulation is where Proteus truly leads the way. The whole learning process takes place in software with the schematic capture module serving as the 'virtual hardware' and the VSM Studio IDE module enabling firmware development and compilation. Basic concepts such as using interrupts, reading from an ADC or setting up a UART can be shown in the context of a simulated embedded system. Educators or students can set breakpoints and pause at any time, examining source code or voltage levels on the schematic and then single stepping through the code. A host of register, variable and watch windows can be used to display relevant information and there is even diagnostics display that provides command and data information from the entire simulation in plain text form. The detail and accuracy of our processor models mean that they will run third party libraries and code examples. This helps more advanced students experiment with advanced on-board peripherals such as USB. Meanwhile, our support for multiple 8-bit, 16-bit and 32-bit processor families enables educators to cover a broad range of embedded architectures and discuss the benefits, drawbacks and typical application areas of each.

4.4 Procedure of Simulation

Before simulation can be run, the procedure is illustrated step by step: First step: the PROTEUS program is chosen from program menu and PROTEUS program is chosen from ISIS7 professional.

After clicking the ISIS7 professional a work area appears as in figure (4.9).

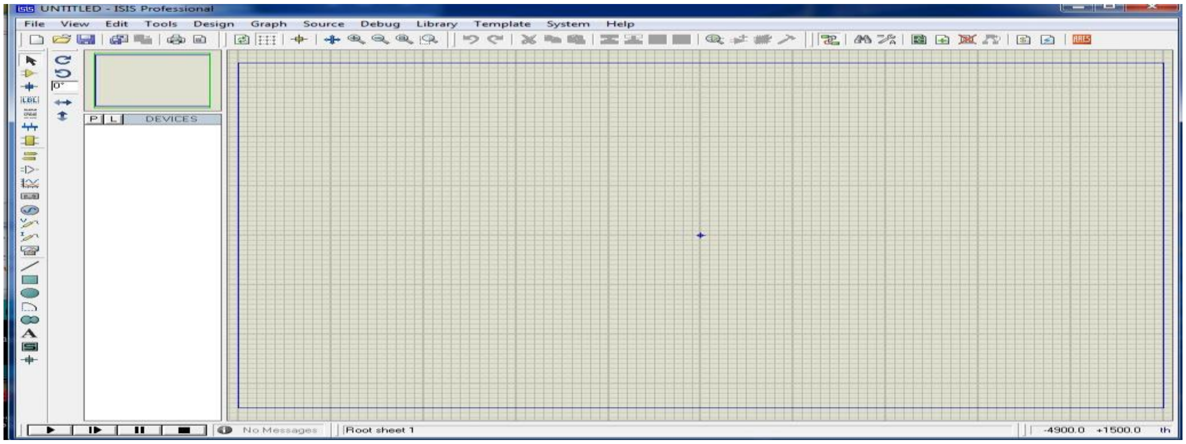


Figure (4.9) the work area

Second step: the tools needed for the design is chosen as in figure (4.10).

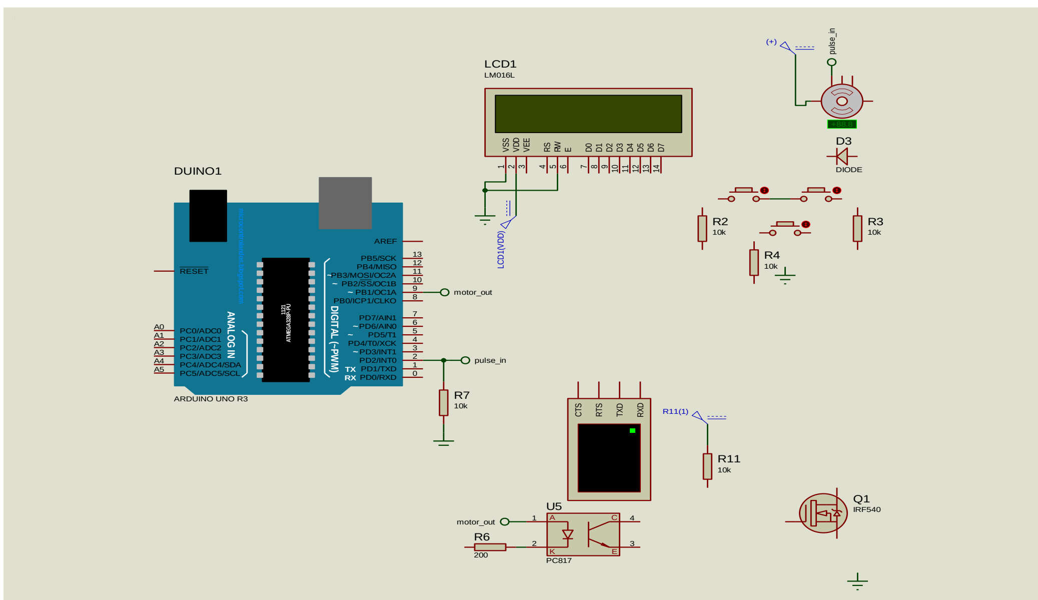


Figure (4.10) all tools DC Motor

Third step: after choosing all tools needed in the design, the tools are assembled as shown in figure (4.11).

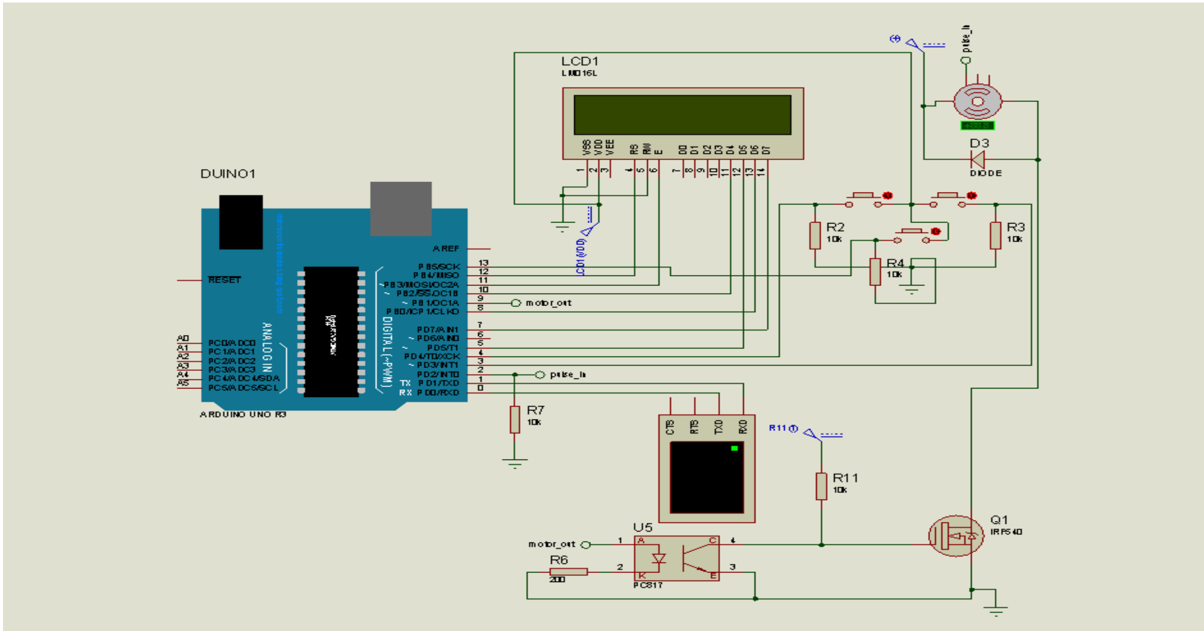


Figure (4.11): the main circuit design

4.4 Operation

When the power supply is connected, the DC motor starts rotating. When the power push button is pressed, the microcontroller starts to operate. We adjust the required speed through push buttons and displays it on a LCD screen. At the same time, the tachometer calculates the motor cycles per minute(r.p.m) and sends a signal to the microcontroller to turn it into a pulse wave. The microcontroller compare it with the required speed signal to change between the two waveforms at a given speed to obtain a new wave with a particular width, generating a new output voltage being fed to the optocoupler, which separates the motor circuit and the microcontroller's to ensure that (eddy current) of the motor's coil will not cause a disturbance in the microcontroller circuit.

When the microcontroller gives a signal to the optocoupler, it sends a light to the transistor to connect the circuit and pass the current to the drive (MOSFET transistor) and controls the motor.

The motor is connected in parallel to a diode and its function is to eliminate the remaining magnetism.

The current speed changes near the desired value of the speed and displays changes in the LCD screen shown in figure (4.12).

The controller replicates the process by taking the tachometer measurements and comparing them with the speed required until we obtain the required speed value.

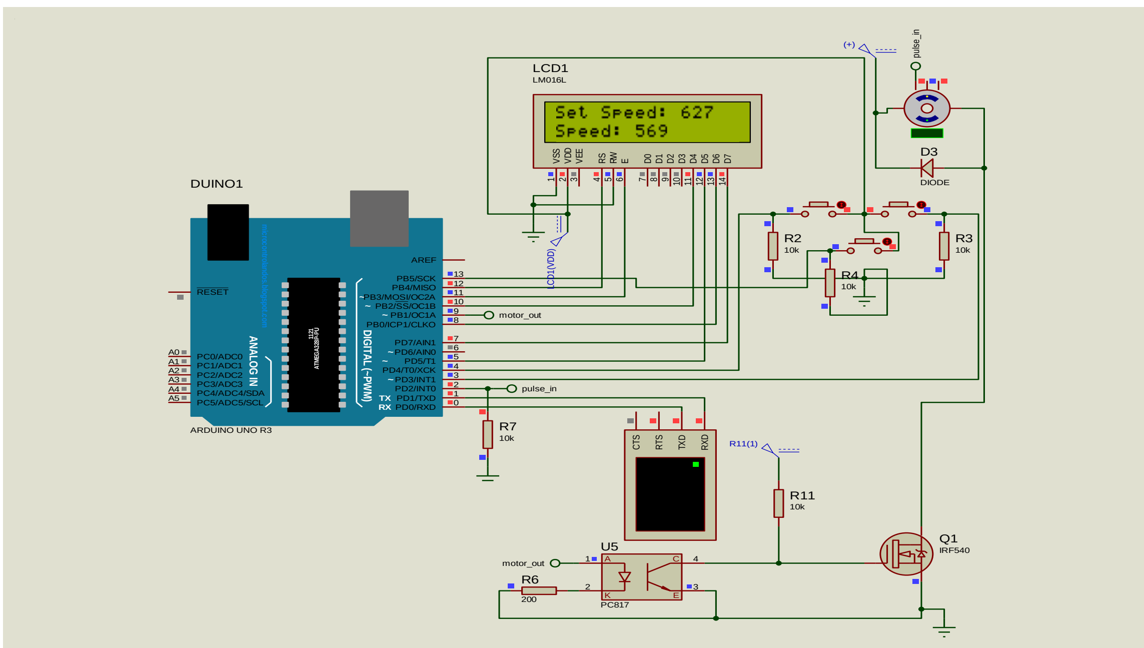


figure (4.12): the operation of simulation

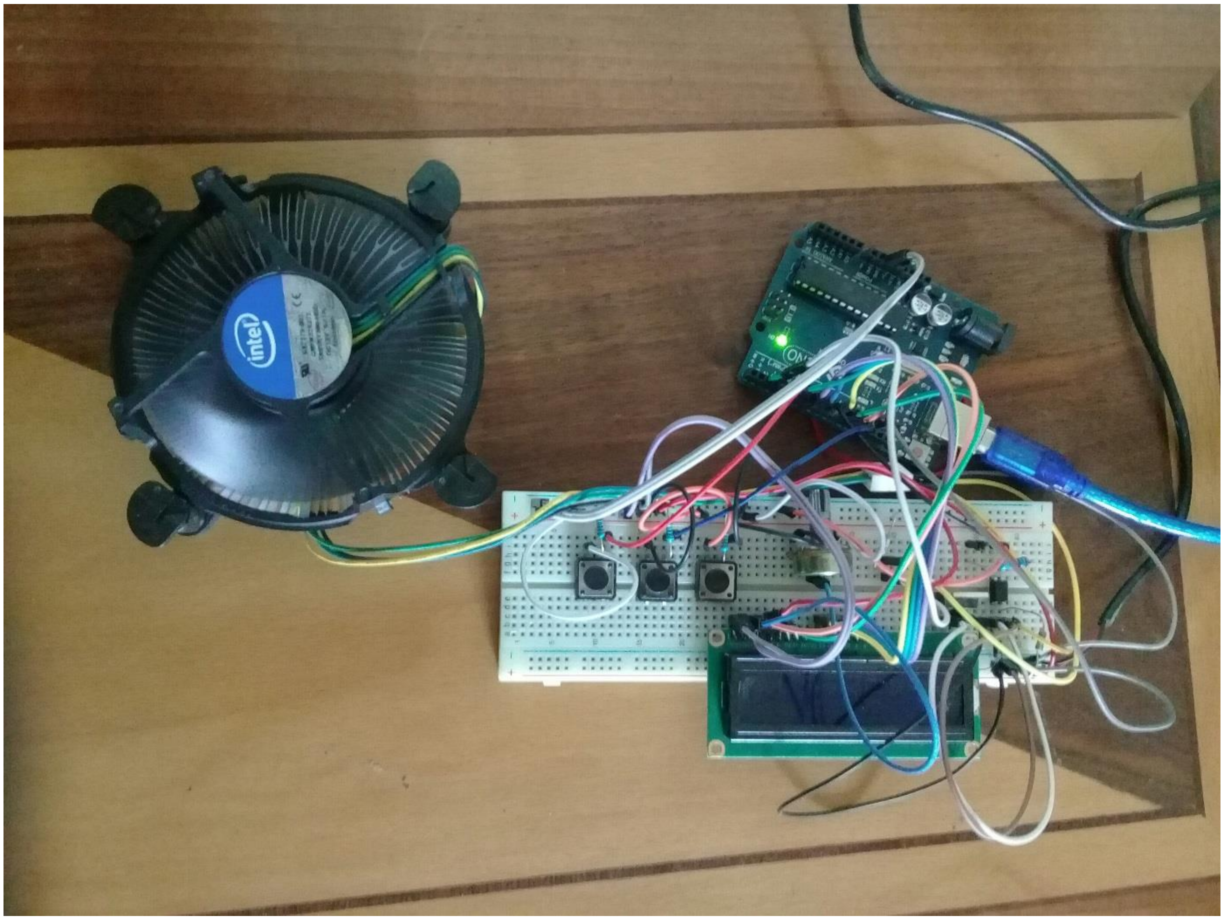


Figure (4.13): Main parts of the system

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The idea of the project is to control the speed of the motor through the PID and fuzzy, and to control the motor speed we studied the control of the PID and the Fuzzy controller in the Chapter two.

In Chapter three we designed the model using the MATLAB program, and explained the Methodology and components of the control circuit.

In Chapter four we explained the work of the circuit that we designed using Proteus program and its application in practice.

5.2 Recommendations

Recommendations are summarized below:

- ✓ We encountered a problem in the Proteus program where the response was slow and became faster when implemented in practice.
- ✓ Improved performance of the motor responses controlled by pid and fuzzy practically became over-shoot and its value is low.
- ✓ Improve the idea of the project by adding control of the start current and the direction of the motor's rotation.

REFERENCES

[1]N. S. Nise, Control systems engineering. [Hoboken, NJ]: Wiley, 2004.

[2]K. Ogata, Modern control engineering. Englewood Cliffs, N.J.: Prentice-Hall, 1970.

[3]Principles of Electrical Machines and Power Electronics- P C Sen.

[4]Y. Tipsuwan and M.Y. Chow, "Fuzzy logic microcontroller

implementation for DC motor speed control," in *Industrial Electronics Society. IECON '99 Proceedings. The 25th Annual Conference of the IEEE*, San Jose, CA , USA, pp. 1271 - 1276 vol.3, 1999.

[5] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, June 1965.

[6] B. Allaoua, A. Laoufi, B. Gasbaoui, and A. Abderrahmani, "Neuro-Fuzzy DC Motor Speed Control Using Particle Swarm Optimization," *Leonardo Electronic Journal of Practices and Technologies*, no. 15, pp. 1-18, Dec. 2009.

[7] W. Banks and G. Hayward, *Fuzzy logic in embedded microcomputers and control systems*. Canada: Byte Craft Limited, 2001.

[8] B. Hamed and M. Al-Mobaied, "Fuzzy PID Controllers Using FPGA Technique for Real Time DC Motor Speed Control," *Intelligent Control and Automation*, vol. 2, pp. 233-240, 2011.

[9] W. Abd El-Meged El-Badry, "Design and Implementation of Real Time DC Motor Speed Control using Fuzzy Logic," Faculty of Engineering, MUST, Research Journal June 2008

[10] O. Peter, D. Szabolcs, C. Andor, and B. Nandor, "Fuzzy Logic Motor Control with MSP430x14x," Texas Instruments, Texas, Application Report SLAA235, February 2005.

[11] J. Charais and R. Lourens, "Software PID Control of an Inverted Pendulum Using the PIC16F684," Microchip Technology Inc., Application Note DS00964A, 2004.

[12] S. Lankton. (2005, Mar.) shawnlankton.com. [Online]. http://www.shawnlankton.com/2005/03/pic-pid-controller/. [Accessed: Mar 22, 2012]

[13] Khoei, A. Hadidi, Kh, Microprocessor Based Closed-Loop Speed Control System For DC Motor Using Power MOSFET, Elctronics Circuits and Systems, IEEE International Conference ICECS 96, Vol. 2, 1247-1250, 1996.

[14] K. J. Åström, T. Hägglund, and K. J. Åström, PID controllers. Research Triangle Park,N.C.: International Society for Measurement and Control, 1995

[15] Leonid Rezink,"Fuzzy Controllers",VictoriaUniversity of Technology, 1997.

[16] Hung T. Nguyen- Nadipuram R. prasad-carol L.Walker-Elbert A.Walker

[17] D. Ibrahim, "Advanced PIC Microcontroller Projects in C," 2008.

[18] S. Monk, 30 Arduino projects for the evil genius. New York: McGraw-Hill, 2010.

[19] J. Boxall, Arduino workshop: A Hands-On introduction with 65 projects: No Starch Press, 2013.

[20] C. Valenti, "Implementing a PID Controller Using a PIC18 MCU," Microchip Technology Inc., Application Note DS00937A, 2004.

APPENDIX

The Programing Code

```
//  The circuit:
// * LCD RS pin to digital pin 12
// * LCD Enable pin to digital pin 11
// * LCD D4 pin to digital pin 10
// * LCD D5 pin to digital pin 8
// * LCD D6 pin to digital pin 5
// * LCD D7 pin to digital pin 7
// * LCD R/W pin to ground
// * LCD VSS pin to ground
// * LCD VCC pin to 5V
// * 10K resistor:
// * ends to +5V and ground
// * wiper to LCD VO pin (pin 3)

// include the library code:
#include <LiquidCrystal.h>
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>
#include <PID_v1.h>

// Step 1 -  Instantiating an object library
Fuzzy* fuzzy = new Fuzzy();

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 10, 5, 8, 7);

// initiate input pins;
int ypin = 4, npin = 3, enpin = 9, flippin = 13;

// initiate program flow variables
```

```
int setSpeed = 50;

// it has to be tested as it may change acording on the
// distance the leds are placed.
int nPalas = 7; // the number of blades of the propeller
boolean done=false;

// fuzzy vars
float err=0,derr=0,lasterr=0,fout=0;

// set pid controls
double setpoint, input, output;
//Define the aggressive and conservative Tuning Parameters
double kp=2, ki=0.5, kd=0.1;

//Specify the links and initial tuning parameters
PID pid(&input, &output, &setpoint, kp, ki, kd, DIRECT);


// tacho variables
byte cnt = 0;
unsigned long past_time = 60000000,last_pulse_time = 0,lastmillis=0;

void counter(){
  cnt++;
  if(cnt >= 14){
    past_time = micros()-last_pulse_time;
    last_pulse_time = micros();
    cnt = 0;
  }
}

void setup()
{
  Serial.begin(9600);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Welcome ...");
  analogWrite(enpin,255);
  pinMode(ypin, INPUT);
  pinMode(npin, INPUT);
```

```
pinMode(flippin, INPUT);
pinMode(2,INPUT);

// initilizing fuzzy papmeters
FuzzyInput* e = new FuzzyInput(1);
FuzzySet* enl = new FuzzySet(-3200, -3200, -1200, -800);
e->addFuzzySet(enl);
FuzzySet* enm = new FuzzySet(-1200, -800, -800, -400);
e->addFuzzySet(enm);
FuzzySet* ens = new FuzzySet(-800, -400, -400, 0);
e->addFuzzySet(ens);
FuzzySet* eze = new FuzzySet(-400, 0, 0, 400);
e->addFuzzySet(eze);
FuzzySet* eps = new FuzzySet(0, 400, 400, 800);
e->addFuzzySet(eps);
FuzzySet* epm = new FuzzySet(400, 800, 800, 1200);
e->addFuzzySet(epm);
FuzzySet* epl = new FuzzySet(800, 1200, 3200, 3200);
e->addFuzzySet(epl);
fuzzy->addFuzzyInput(e);

// initilizing fuzzy papmeters
FuzzyOutput* KP = new FuzzyOutput(1);
FuzzySet* kpsm = new FuzzySet(0, 0, 0.1, 0.3);
KP->addFuzzySet(kpsm);
FuzzySet* kpmd = new FuzzySet(0.1, 0.3, 0.3, 0.5);
KP->addFuzzySet(kpmd);
FuzzySet* kpbg = new FuzzySet(0.3, 0.5, 1, 1);
KP->addFuzzySet(kpbg);
fuzzy->addFuzzyOutput(KP);

// initilizing fuzzy papmeters
FuzzyOutput* KI = new FuzzyOutput(2);
FuzzySet* kism = new FuzzySet(0, 0, 0.02, 0.04);
KI->addFuzzySet(kism);
FuzzySet* kimd = new FuzzySet(0.02, 0.05, 0.05, 0.08);
KI->addFuzzySet(kimd);
FuzzySet* kibg = new FuzzySet(0.06, 0.08, 0.1, 0.1);
KI->addFuzzySet(kibg);
fuzzy->addFuzzyOutput(KI);

// initiating fuzzy rules
```

```
  FuzzyRuleAntecedent* if_e_pl = new FuzzyRuleAntecedent();if_e_pl-
>joinSingle(epl);
  FuzzyRuleConsequent* then_kpsm_and_kism = new FuzzyRuleConsequent();
  then_kpsm_and_kism->addOutput(kpsm);
  then_kpsm_and_kism->addOutput(kism);
  FuzzyRule* fuzzyRule01 = new FuzzyRule(1, if_e_pl, then_kpsm_and_kism);
fuzzy->addFuzzyRule(fuzzyRule01);

  FuzzyRuleAntecedent* if_e_pm = new FuzzyRuleAntecedent();if_e_pm-
>joinSingle(epm);
  FuzzyRuleConsequent* then_kpmd_and_kimd = new FuzzyRuleConsequent();
  then_kpmd_and_kimd->addOutput(kpmd);
  then_kpmd_and_kimd->addOutput(kimd);
  FuzzyRule* fuzzyRule02 = new FuzzyRule(2, if_e_pm,
then_kpmd_and_kimd); fuzzy->addFuzzyRule(fuzzyRule02);

  FuzzyRuleAntecedent* if_e_ps = new FuzzyRuleAntecedent();if_e_ps-
>joinSingle(eps);
  FuzzyRuleConsequent* then_kpbg_and_kibg = new FuzzyRuleConsequent();
  then_kpbg_and_kibg->addOutput(kpbg);
  then_kpbg_and_kibg->addOutput(kibg);
  FuzzyRule* fuzzyRule03 = new FuzzyRule(3, if_e_ps, then_kpbg_and_kibg);
fuzzy->addFuzzyRule(fuzzyRule03);

  FuzzyRuleAntecedent* if_e_ns = new FuzzyRuleAntecedent();if_e_ns-
>joinSingle(ens);
  FuzzyRule* fuzzyRule04 = new FuzzyRule(4, if_e_ns, then_kpbg_and_kibg);
fuzzy->addFuzzyRule(fuzzyRule04);

  FuzzyRuleAntecedent* if_e_nm = new FuzzyRuleAntecedent();if_e_nm-
>joinSingle(enm);
  FuzzyRule* fuzzyRule05 = new FuzzyRule(5, if_e_nm,
then_kpmd_and_kimd); fuzzy->addFuzzyRule(fuzzyRule05);

  FuzzyRuleAntecedent* if_e_nl = new FuzzyRuleAntecedent();if_e_nl-
>joinSingle(enl);
  FuzzyRule* fuzzyRule06 = new FuzzyRule(6, if_e_nl, then_kpsm_and_kism);
fuzzy->addFuzzyRule(fuzzyRule06);

  FuzzyRuleAntecedent* if_e_ze = new FuzzyRuleAntecedent();if_e_ze-
>joinSingle(eze);
```

```
  FuzzyRule* fuzzyRule07 = new FuzzyRule(7, if_e_ze, then_kpsm_and_kism);
fuzzy->addFuzzyRule(fuzzyRule07);

  // initiate fuzzty PID controller
  setpoint = setSpeed*3200.0/255.0;
  lcd.clear();
  lcd.setCursor(0,0);
  attachInterrupt(0,counter,FALLING);
  pid.SetMode(AUTOMATIC);
}

void loop()
{
    // button press chance window-----------------------------------------------------------
--------------------------------
  if (!done){
    if (digitalRead(ypin) == HIGH ) {
      while (digitalRead(ypin) == HIGH);
      setSpeed += 5;
    }
    else if (digitalRead(npin) == HIGH ) {
      while (digitalRead(npin) == HIGH);
      setSpeed -= 5;
    }
    if (digitalRead(flippin) == HIGH){
        while (digitalRead(flippin) == HIGH);
        done = true;
    }
    if (setSpeed > 255) setSpeed = 255;
    if (setSpeed < 1) setSpeed = 0;

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Set Speed: ");
    lcd.print(map(setSpeed, 0, 255, 0, 3200));
    delay(100);
  }

  else{
   //------------ calculate RPM ---------------------------------------------
//   if (millis() - lastmillis >= 10){
    double rpm = round(60000000.0/past_time * 2);
```

```
    input = rpm;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Set Speed: ");
    lcd.print(map(setSpeed, 0, 255, 0, 3200));
    lcd.setCursor(0, 1);
    lcd.print("Speed: ");
    lcd.print(round(rpm));

    //feed fuzzy logic controller ---------------------------------------------------
    setpoint = (double)map(setSpeed,0,255,0,3200);
    err = setpoint - rpm;

    fuzzy->setInput(1, err);
    fuzzy->fuzzify();
    // Set output vlaue: outpu
    kp = (double)fuzzy->defuzzify(1);
    ki = (double)fuzzy->defuzzify(2)*3;
    pid.SetTunings(kp, ki, kd);
    pid.Compute();

    Serial.print("set speed: ");Serial.print(setpoint);Serial.print(",
");Serial.print(err);Serial.print(" output: ");Serial.print(output);Serial.println("");

    // control motor speed ----------------------------------------------------------------
---------------
    analogWrite(enpin,255-output);

    // reset scope control parameters---------------------------------------------------------
------------------------------
    if (digitalRead(flippin) == HIGH){
         while (digitalRead(flippin) == HIGH);
         done = false;
    }
//    lastmillis = millis();
//   }
  }
}
```