INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

HASSIBA BENBOUALI UNIVERSITY OF CHLEF

VELOCITY CONTROL OF A DC MOTOR USING PID AND CDM METHOD BASED ON MATLAB/SIMULINK AND ARDUINO

Student:

Khalid Boudane - a46640

Master Degree In Industrial Engineering

Supervised by:

Dr. José Gonçalves
Dr. João Paulo Coelho
Dr. Souad Tahraoui

Bragança 2021

Dedication

It is with great pleasure that I dedicate this work to my dearest parents and sisters,

Your love, the values transmitted, your support in the most difficult moments and your constant attention for your children. I boast, every day since the beginning of my life, even without words, of the pride I take in being your child. No matter what happens, making you proud is also one of my goals.

To the whole mechatronics and Industrial family,

Our paths crossed for the first time when we entered the IPB, keeping the friendship and international relation that unites us and the memories of all the moments we spent together, nothing to say, I treasure each and every one of the moments we spent with you.

To everyone who has contributed in any way to running of my work.

Acknowledgements

I am pleased to express my sincere thanks to all the IPB team for their welcome and the trust they have placed in me.
My sincerest thanks to our dears Dr ***José Gonçalves*** *and Dr* ***Joào Paulo Coelho****, my university supervisors, for the enriching and interesting experience they gave me during the period of the project and for the autonomy they gave me throughout the project.*

Our thanks also go to all the IPB staff who contributed to the running of my final project. I have the honour to express my special thanks to the members of the jury for giving me the honour of judging this modest work. Finally, I would like to thank the entire

teaching team of the IPB for having ensured a good training during the period of research master in the department of Technology and Management (ESTIG) of IPB Bragança.

Abstract

The control of the speed of a (direct current) DC motor is very important as any change can lead to instability of the closed loop system. The aim of this project is to show how a DC motor can be controlled using a PID controller and the CDM Coefficient Diagram method in MATLAB/Simulink. The DC motor will be interfaced to MATLAB using an Arduino Mega 2560. The speed of the motor will be set by creating a Simulink model for the PID controller and CDM Coefficient Diagram Method in MATLAB. This last will send a serial command to the DC motor using the PWM pins on the Arduino board. The DC motor will run at the speed defined by the user. The velocity of the DC motor will be measured using the encoder. From the encoder, the output is sent to the controller (PID/CDM) in Simulink via Arduino. The controller compares the actual velocity of the motor with the setpoint velocity.

Keywords:

Matlab/Simulink, Arduino, Regulation PID, CDM Coefficient Diagram Method, Encoders, PWM

Resumo

O controlo da velocidade de um motor CC (corrente contínua) é muito importante, uma vez que uma estratégia de control errada pode levar à instabilidade do sistema em malha fechada. O objectivo deste projecto é mostrar como um motor de corrente contínua pode ser controlado utilizando um controlador PID e o método CDM em MATLAB/Simulink. O motor CC será interligado ao MATLAB usando um Arduino Mega 2560. A velocidade do motor será controlada através da criação de um modelo Simulink para o controlador PID e para o CDM. O software enviará um comando série ao motor DC usando os pinos PWM na placa Arduino fazendo com que o motor de corrente contínua opere à velocidade definida pelo utilizador. A velocidade do motor de corrente contínua será medida utilizando um codificador incremental. A partir dos pulsos gerados pelo codificador, a velocidade do motor é derminada sendo usada pelo controlador (PID/CDM) em Simulink via Arduino..

Palavras-chave:

Matlab/Simulink, Arduino, Regulamento PID, Método do Diagrama de Coeficiente CDM, Codificadores , PWM

Contents

List of Tables

List of Figures

Introduction

Control is one of the most fundamental factors that enables us to use devices to their maximum potential and with the highest accuracy, where we can achieve high efficiency, total control of the dc motor, and improved precision.

In this project, we would use a PID controller and CDM method to manipulate the velocity of a (direct current) DC motor using an encoder signal. The encoder allows us to read the direction and the speed of the shaft and calculate the actual speed, which we will enter the result into the controller to get the desired speed depending on the PID controller's feedback. Then, as a theoretical analysis, we will simulate the modilisation of DC motor mathematically in Matlab/simulink software and preview the effects before continuing through the practical experience.

Our project will have three chapters, each of those will be as follows:

The first chapter presents an overview of the generality of DC motors, followed by an introduction to the types, construction, and implementations of DC motors, while moving on to the advantages and disadvantages of DC motors.

The second chapter have the equations and mathematical models that describe our DC motor to represent speed, field, and armature current to derive mathematical model for implement PID controller and CDM Coefficient Diagram Method and by using Ziegler-Nichols procedure to specify the parameters of PID controller to implement in our model and analyze the theoretical result.

The third chapter, we will go over to present the equipment used to conduct this control, from electrical to mechanical components. We would also discuss the properties and technological requirements, interface and use an L298N Dual H-Bridge Motor with

Arduino board and Matlab/Simulink software. After that, we have to go over the model of simulink that control DC motor via Arduino and compare the real velocity with the desired velocity, and Test results of control is presented and performance is explained in detail.

After that, we'll go through the results of the speed control testing in the graph and analyze them, and then we'll go over the Speed Regulation of DC Motor with PID Controller and CDM Coefficient Diagram Method conclusions.

For future study, we should add different controller methods to this work, such as (Neuro-Fuzzy controller, LQR, and LGG optimal controller), and compare the precision, feedback, and characteristics of each method which provide best control and high precise.

Chapter 1

State Of The Art

The history of (direct current) DC motors extends way back into the 19th century. From initial testing and development to widespread use across global industries, DC gear motors have evolved significantly into the present day.

Being a trusted electric motor manufacturer for over 70 years means have seen how the DC motor has changed and adapted to new technologies over time. Always intrigued about how it will continue to evolve and succeed in new industries in the future [1].

The invention of the DC motor came about in the early 1800s, with initial developments made in 1832 by British scientist William Sturgeon. Sturgeon created the very first commutator DC motor, with the ability to turn machinery, however sturgeon's idea was developed and built upon by Thomas Davenport, an American inventor. Davenport is more widely known to have officially invented a working DC motor, which he went onto patent a few years later in 1837. Initially, davenport struggled with issues surrounding the expensive costs of battery power whilst running the motors, which made the very first DC motor fairly incapable of withstanding the test of time [1].

After Davenport's initial invention of the DC motor came to light, many other inventors and engineers became inspired to develop their own concepts. In 1834, russian engineer Moritz Von Jacobi went on to invent the very first rotating DC motor. His invention became famous for being incredibly powerful, which later set a world record. Incredibly, he broke his own world record in 1838 with a new and improved version of his DC motor invention. This motor motivated others to produce DC motors of the same powerful standard, with the ability to drive a boat with a capacity of 14 people across a river. The year of 1864 saw an amazing breakthrough in the history of DC motors, with the very first recognition of the ring armature by Antonio Pacinotti. This has become a vital piece of equipment within the DC motor's design, which carries current through coils grouped together [1].

With all of these developments in mind, the 1800's had still not seen a more practical DC motor with greater speed control. This came in 1886, where Frank Julian Sprague invented a motor that could maintain constant speed under variable loads. His invention lead to wider commercial use of the DC motor, such as the first electric elevator and powered trolley system. The practicality of this DC motor caused high demand to surge in both commercial and more residential settings, such as in factories and within the homes [1].

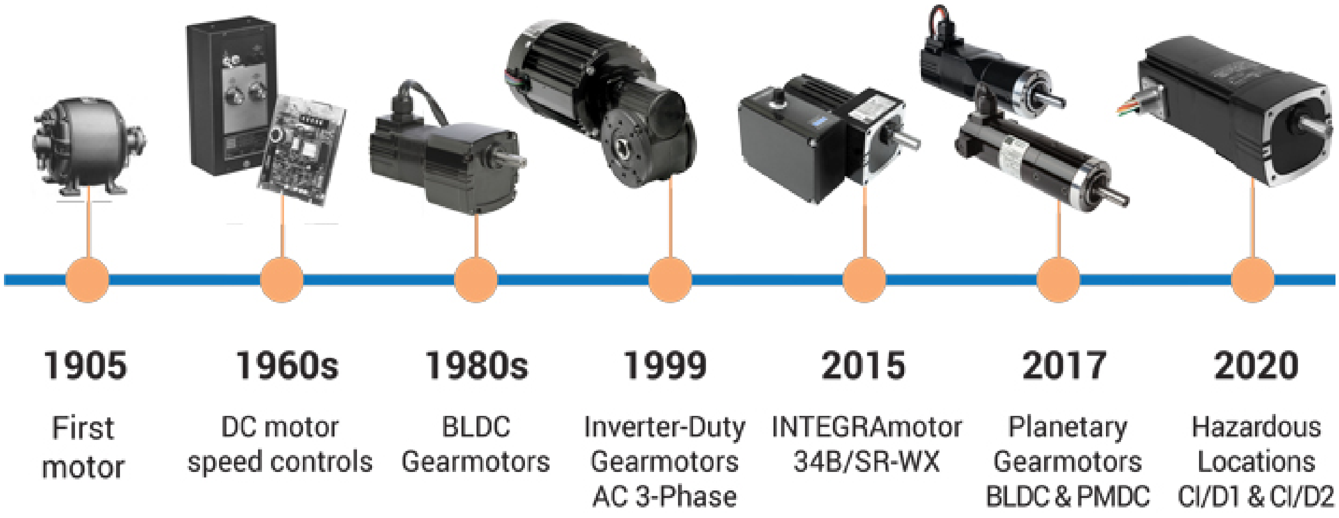


Figure 1.1: Evolution of DC motor

1.1 DC motor

A direct current (DC) motor is a fairly simple electric motor that uses electricity and a magnetic field to produce torque, Which causes it to turn. At its most simple, It requires two magnets of opposite polarity and an electric coil, which acts as an electromagnet. The repellent and attractive electromagnetic forces of the magnets provide the torque that causes the motor to turn Anyone who has ever played with magnets knows that they are polarized, With a positive and a negative side. The attraction between opposite poles and the repulsion of similar poles can easily be felt, even with relatively weak magnets. A DC motor uses these properties to convert electricity into motion. As the magnets within the motor attract and repel one another, the motor turns [2].

Figure 1.2: DC motor

A DC motor is powered by direct current, which converts electrical energy into mechanical energy. This mechanical energy causes rotational movement within the motor, helping to power a vast majority of industry applications [1].

1.1.1 Types of DC motors

Permanent magnet motors

The permanent magnet motor uses a magnet to supply field flux, permanent magnet DC motors have excellent starting torque capability with good speed regulation, a disadvantage of permanent magnet DC motors is they are limited to the amount of load they can drive, these motors can be found on low power applications [3].

Another disadvantage is that torque is usually limited to 150%, of rated torque to prevent demagnetization of the permanent magnets [3].

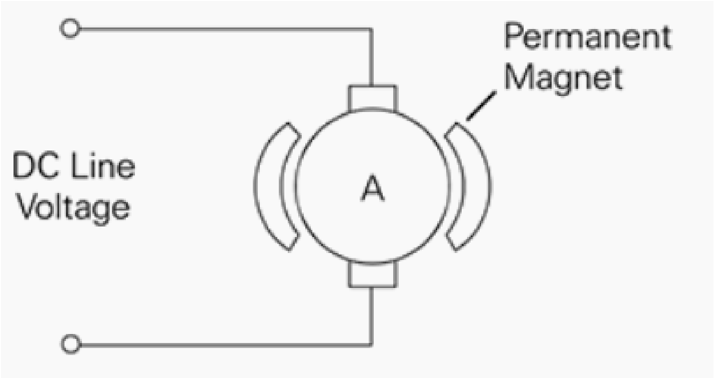


Figure 1.3: Permanent magnet motor [3]

Series motors

In a series DC motor the field is connected in series with the armature, the field is wound with a few turns of large wire because it must carry the full armature current. A characteristic of series motors is the motor develops a large amount of starting torque. However, speed varies widely between no load and full load. Series motors cannot be used where a constant speed is required under varying loads. Additionally, the speed of a series motor with no load increases to the point where the motor can become damaged, Some load must always be connected to a series-connected motor. Series-connected motors generally are not suitable for use on most variable speed drive applications [3].

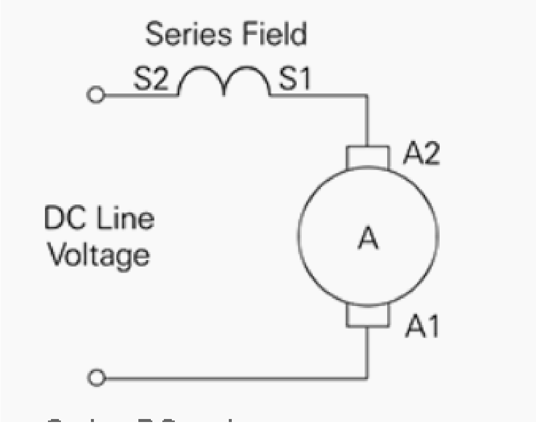


Figure 1.4: Series DC motor [3]

Shunt motors

In a shunt motor the field is connected in parallel (shunt) with the armature windings, the shunt-connected motor offers good speed regulation, The field winding can be separately excited or connected to the same source as the armature. An advantage to a separately excited shunt field is the ability of a variable speed drive to provide independent control of the armature and field, the shunt-connected motor offers simplified control for reversing. This is especially beneficial in regenerative drives [3].

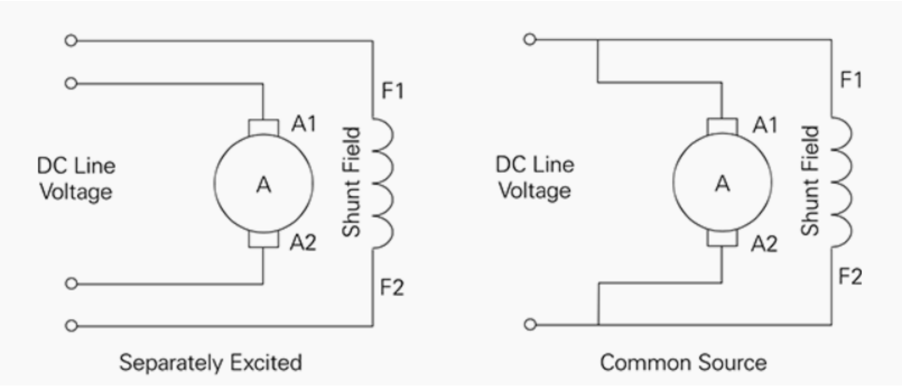


Figure 1.5: Shunt DC motor [3]

Compound motors

Compound motors have a field connected in series with the armature and a separately excited shunt field, the series field provides better starting torque and the shunt field provides better speed regulation, however the series field can cause control problems in variable speed drive applications and is generally not used in four quadrant drives [3].

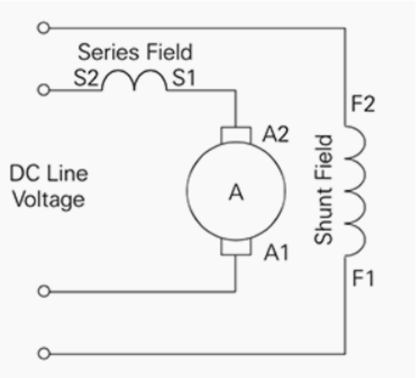


Figure 1.6: Compound DC motor [3]

1.1.2 Construction of DC motor

Across-section of a 4-pole DC motor is shown in figure below:

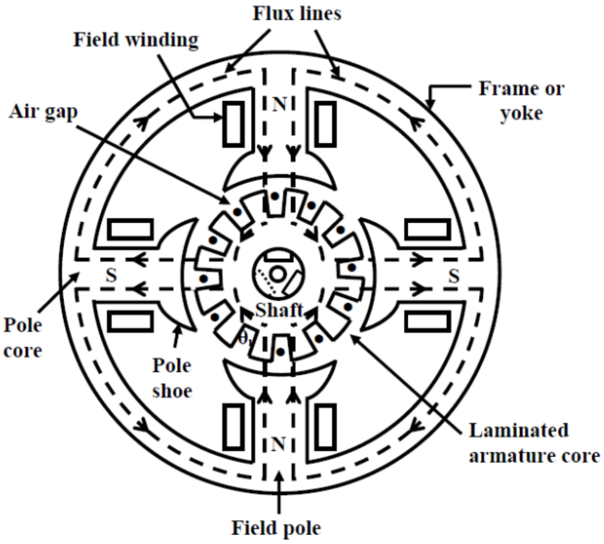


Figure 1.7: Construction Of DC Motor [4]

The construction generally consists of:

Yoke

This is the outer part of the DC motor. It provides the mechanical supports for the poles and acts as a protecting cover for the whole machine. It carries the magnetic flux produced by the poles. Yokes are made out of cast iron or cast steel [4].

Field pole

The field poles are mounted inside the yoke, are made of thin lamination stacked together, It consist of pole cores and pole shoes as shown in figure below [4].

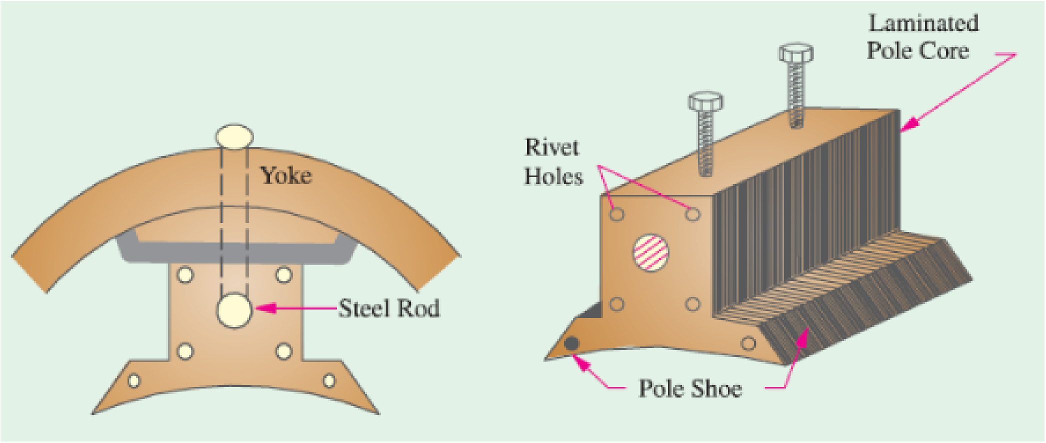


Figure 1.8: Pole Cores And Pole [4]

Field winding

The field coils are wound on the poles. There are two types of field windings:

- **a: Shunt field winding**: large number of turns of small section copper conductor is used (fine wire), It is connected in parallel with the armature windings [4].
- **b: Series field winding**: few turns of heavy cross section conductor is used. It is connected in series with the armature windings, A DC motor may have both field

windings wound on the same pole.

- Shunt motor: motor with only shunt field windings.
- Series DC motor: motor with only series field windings.
- Compound DC motor: motor with both field windings (shunt series) [4].

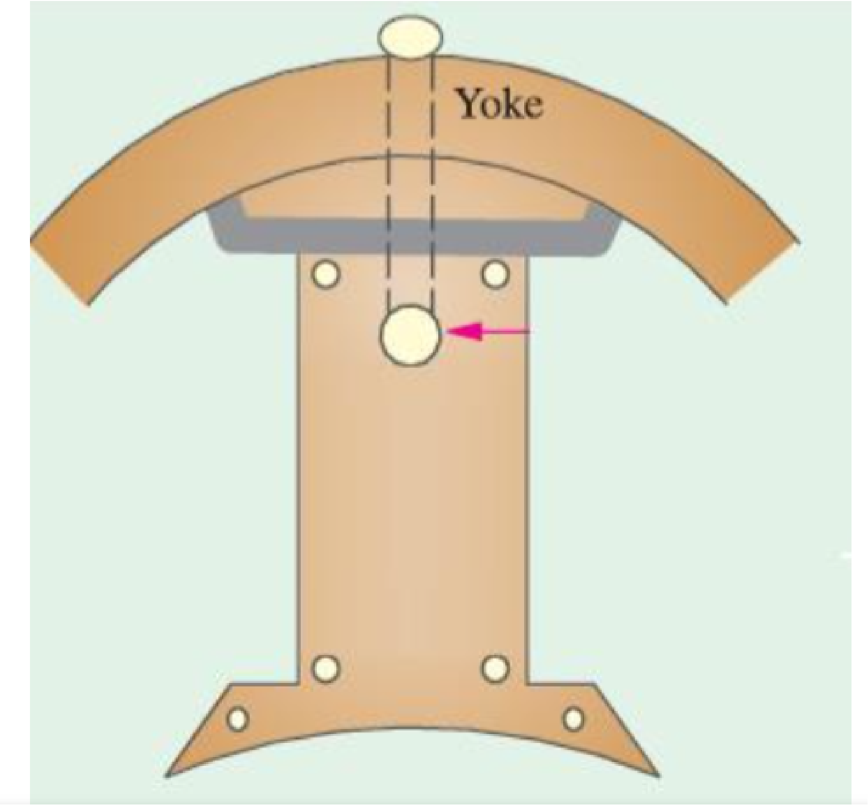


Figure 1.9: Field winding Of DC motor [4]

1.1.3 Armature

Armature core: it carries the armature winding, is made of sheet-steel lamination's. The lamination's are stacked together to form a cylindrical structure as shown in figure below [4].

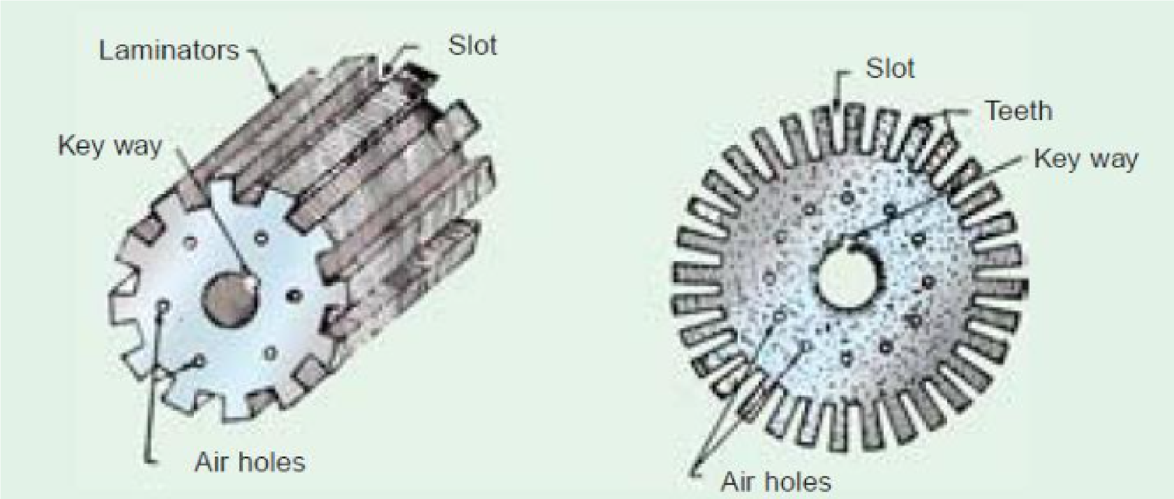


Figure 1.10: Armature core [4]

b. Armature windings: is the heart of the DC motor in which the torque is developed [4].

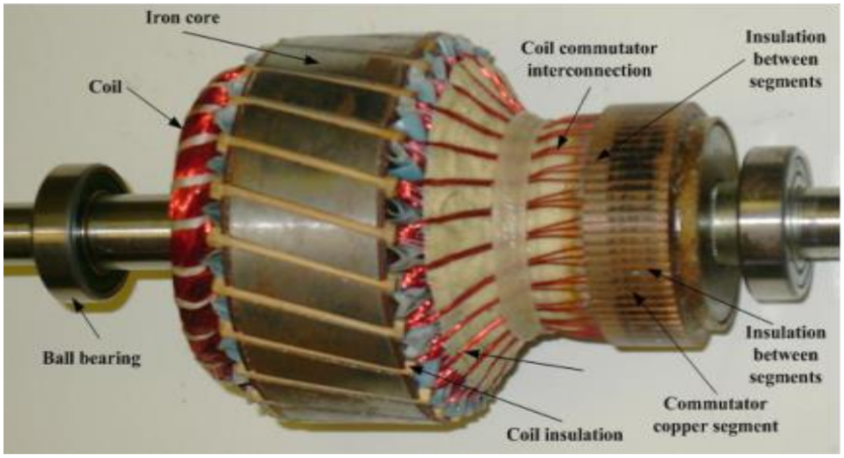


Figure 1.11: Armature windings [4]

1.1.4 Commutator

The commutator, whose function is to facilitate collection of current from the armature, it consists of copper segments tightly fastened together with mica/micanite insulating separators on an insulated base. The whole commutator forms a rigid and solid assembly of insulated copper strips and can rotate at high speeds commutator segment is provided with a riser where the ends of the armature coils get connected [4].

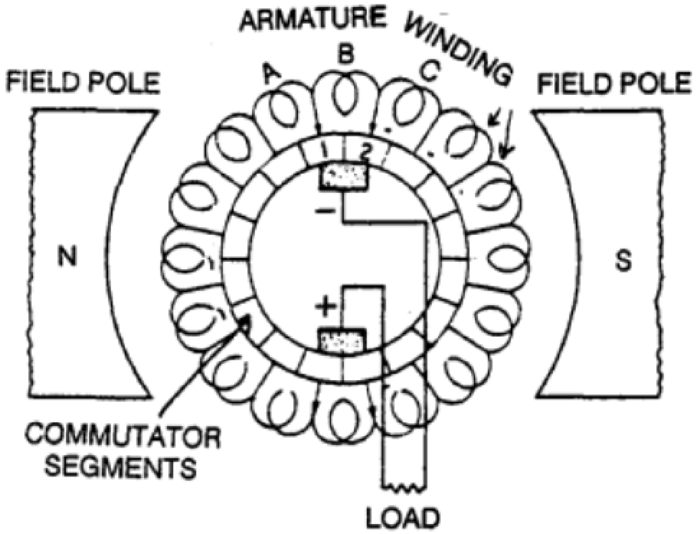


Figure 1.12: Armature and commutator in a two-pole DC motor [4]

1.1.5 Brushes

Are the elements which are connecting the armature windings (through commutator) to the external terminal of the motor. The brush pressure on the commutator should be just right, because low pressure leads to poor contacts which results in excessive sparking and burning the commutator. While high pressure lead overheating the commutator [4].

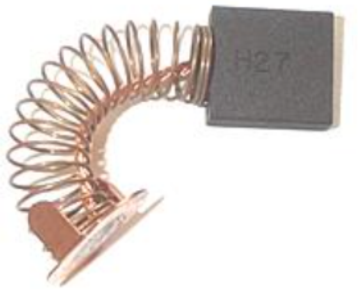


Figure 1.13: Rushe form of DC motor [4]

1.1.6 Working principle of DC motor

Start with the simplest DC motor possible. It looks like as shown in the Fig 1.14 the stator is a permanent magnet and provides a constant magnetic field. The armature, which is the rotating part, is a simple coil [5].

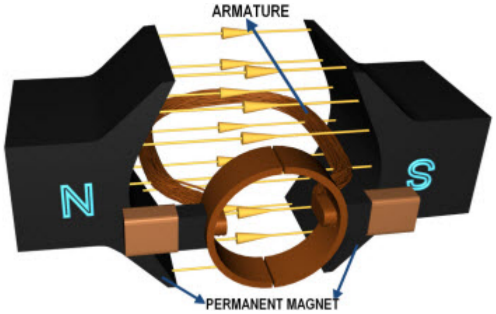


Figure 1.14: The working of DC motor [5]

The armature is connected to a DC power source through a pair of commutator rings. When the current flows through the coil an electromagnetic force is induced on it according to the Lorentz law, so the coil will start to rotate. The force induced due to the electromagnetic induction is shown using 'red arrows' in the Fig 1.15 [5].

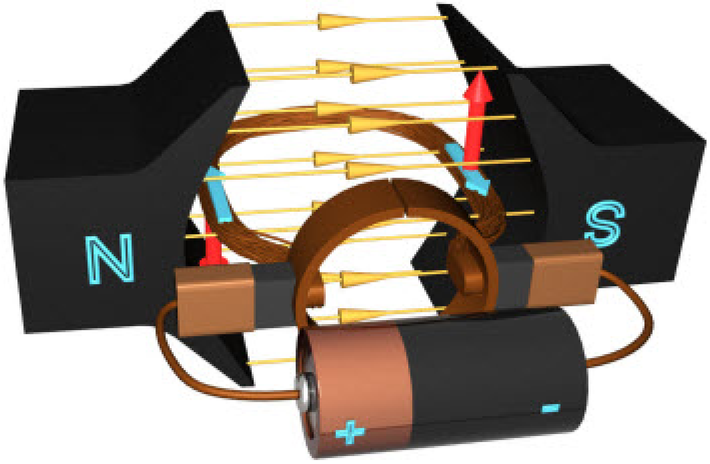


Figure 1.15: The electromagnetic force induced on the coils make the armature coil rotate [5]

You will notice that as the coil rotates, the commutator rings connect with the power source of opposite polarity. As a result, on the left side of the coil the electricity will always flow 'away' and on the right side, electricity will always flow 'towards '. This ensures that the torque action is also in the same direction throughout the motion, so the coil will continue rotating [5].

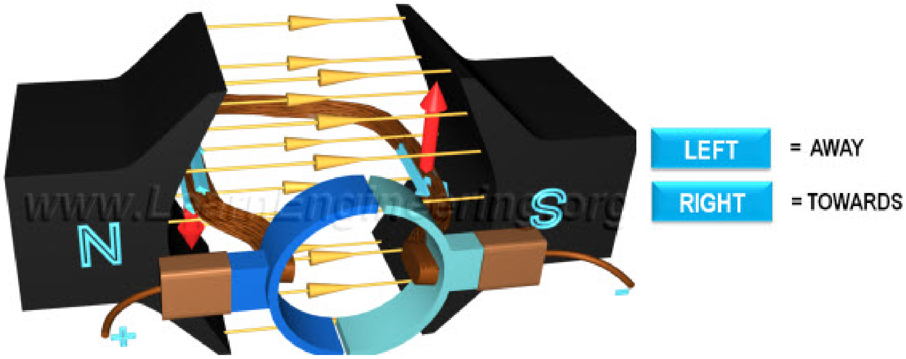


Figure 1.16: The commutator rings make sure a uni-directional current flows through the left and right part of the coil [5]

Improving the torque action

But if you observe the torque action on the coil closely, you will notice that, when the coil is nearly perpendicular to the magnetic flux, the torque action nears zero [5].

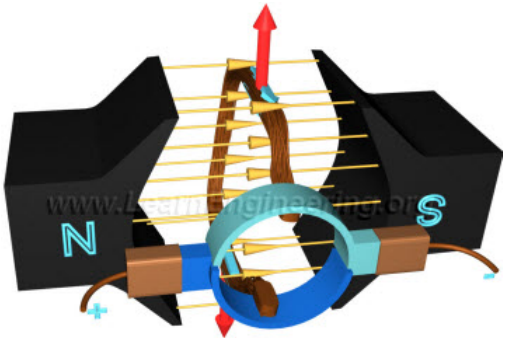


Figure 1.17: When the coil nears perpendicular to the magnetic flux, the torque produced nears zero [5]

As a result there will be irregular motion of the rotor, if you run such a DC motor. Here is the trick to overcome this problem add one more loop to the rotor, with a separate commutator pair for it. In this arrangement when the first loop is in the vertical position, the second loop will be connected to the power source. So a motive force is always present in the system [5].

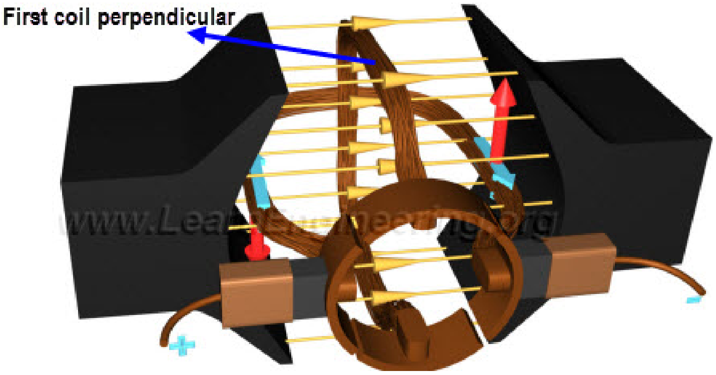


Figure 1.18: The 2 coil rotor arrangement [5]

when the first coil is perpendicular to the magnetic flux, second coil is connected to the power source

Moreover, the more such loops, the smoother will be the motor rotation. In a practical motor, the armature loops are fitted inside slots of highly permeable steel layers. This will enhance magnetic flux interaction. Spring loaded commutator brushes help to maintain contact with the power source [5].

Figure 1.19: More the number of the coils, smoother will be motor rotation; to enhance magnetic flux interaction, coils are put between steel layer poles [5]

Use of electromagnet

A permanent magnet stator pole is used only for very small DC motors. Most often an electromagnet is used, the field coil of the electromagnet is powered from the same DC source [5].

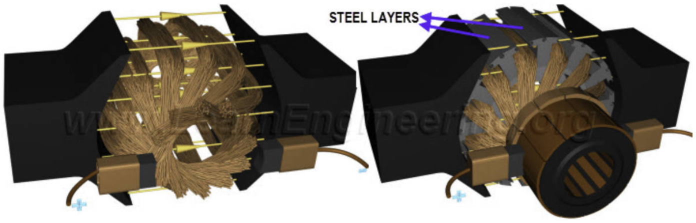


Figure 1.20: An electromagnet is used most of the time in DC motor [5]

Shunt and series motors

The field coils can be connected to the rotor windings in 2 different ways, parallel or series. This results into 2 different kinds of DC motor constructions, a shunt and a series motors [5].

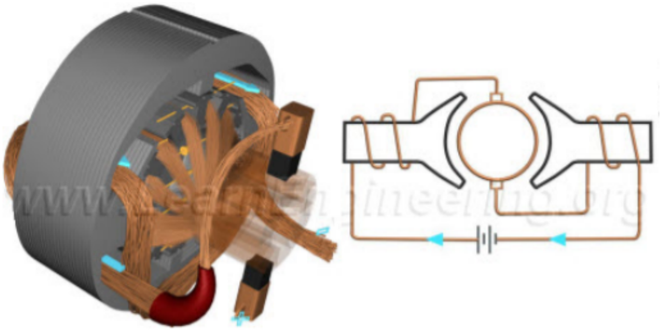


Figure 1.21: A shunt and a series motors [5]

The series wound motor has good starting torque, but its speed drops drastically with the load. This nature is shown in the Fig the shunt motor has a low starting torque, but it is able to run almost at a constant speed, irrespective of the load acting on the motor. This is an attractive operation characteristic of a shunt wound motor, the nature of the speed-torque variation is shown in the Fig 1.22 [5].

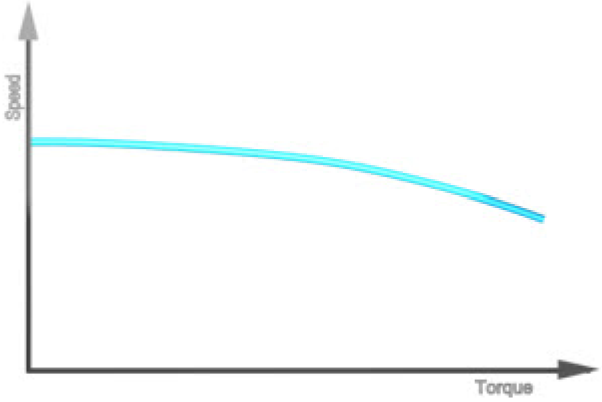


Figure 1.22: A shunt motor provides a constant speed-torque characteristics [5]

The Concept of Back E.M.F

Unlike the other electrical machines DC motors exhibit a unique characteristic, the production of back EMF. A rotating loop in magnetic field will produce an EMF according to the principle of electromagnetic induction [5].

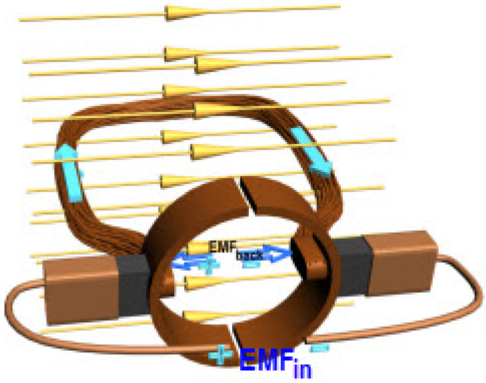


Figure 1.23: The Back E.M.F [5]

The case of rotating armature loops is also the same. An internal EMF will be induced that opposes the applied input voltage. The back EMF reduces armature current by a large amount. Back EMF is proportional to the speed of the rotor. At the starting of the motor, back EMF is too low, thus the armature current becomes too high, leading to burnout of the rotor. Thus a proper starting mechanism that controls the applied input voltage is necessary in large DC motors [5].

1.1.7 velocity control of DC motor

Speed control means intentional change of drive speed to a value required for performing the specific work process. This concept of speed control or adjustment should not be taken to include the natural change in speed which occurs due to change in the load on the shaft. Any given piece of industrial equipment may have its speed change or adjusted mechanically by means of stepped pulleys, sets of change gears, variable speed friction clutch mechanism and other mechanical devices. Historically it is proved to be the first step in transition from nonadjustable speed to adjustable speed drive. The electrical speed control has many economical as well as engineering advantages over mechanical speed control. The nature of the speed control requirement for an industrial drive depends upon its type. Some drives may require continues variation of speed for the whole of the range from zero to full speed or over a portion of this range, while the others may require two or more fixed speeds [6].

1.1.8 Speed control methods

There are 5 main ways to achieve speed regulation in DC motors as follow [6]:

- Armature or rheostat control method.
- Flux control method. It is seen that speed of the motor is inversely proportional to flux.
- Armature control method.
- Voltage control method.

- Variable resistance in series with armature.

Flux control method

It is known that ($N\alpha * 1/\phi$) by the field flux ϕ and (Na) is speed decreasing the flux, thus speed can be increased and vice versa. Hence, name flux or field control method [6].

The flux of DC motor can be changed by changing (Ish) with help of a shunt field rheostat. Since (Ish) in relatively small, shunt field rheostat has to carry only a small, so that rheostat is small in size. This method therefore very efficient in non-interpolar machines the speed can be increased by this method in the ratio 2:1 any further weakening of flux Φ adversely affect the communication [6].

And hence puts a limit to the maximum speed obtainable with this method in machines fitted with interlopes in ratio of maximum to minimum speeds of 6:1 is fairly common. The connection diagram for this type of speed control is shown in figure bellow [6].

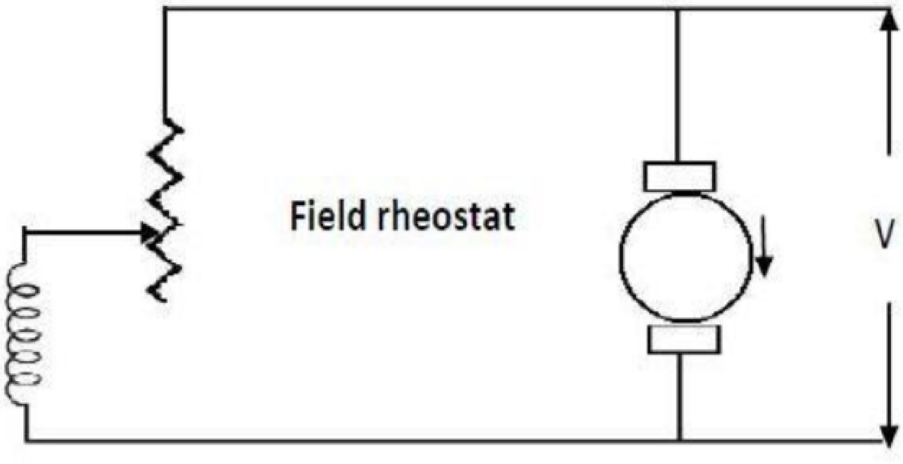


Figure 1.24: flux control method [6]

Armature or rheostat control method

The armature resistance control is based on the principle that the speed of the motor is directly proportional to the back EMF. So, if the supply voltage and the armature resistance are kept at a constant value, the speed of the motor will be directly proportional to the armature current [7].

Rheostat control method is used when speeds below the no load speed are required. As the supply voltage is normally constant, the voltage across the armature is varied by inserting a variable rheostat or controller resistance in series with the armature circuit as shown in fig 1.4 as controller resistance is increased, potential difference across the armature is decreased, thereby decreasing the armature speed. For a load of constant torque, speed is approximately proportional to the potential difference [6].

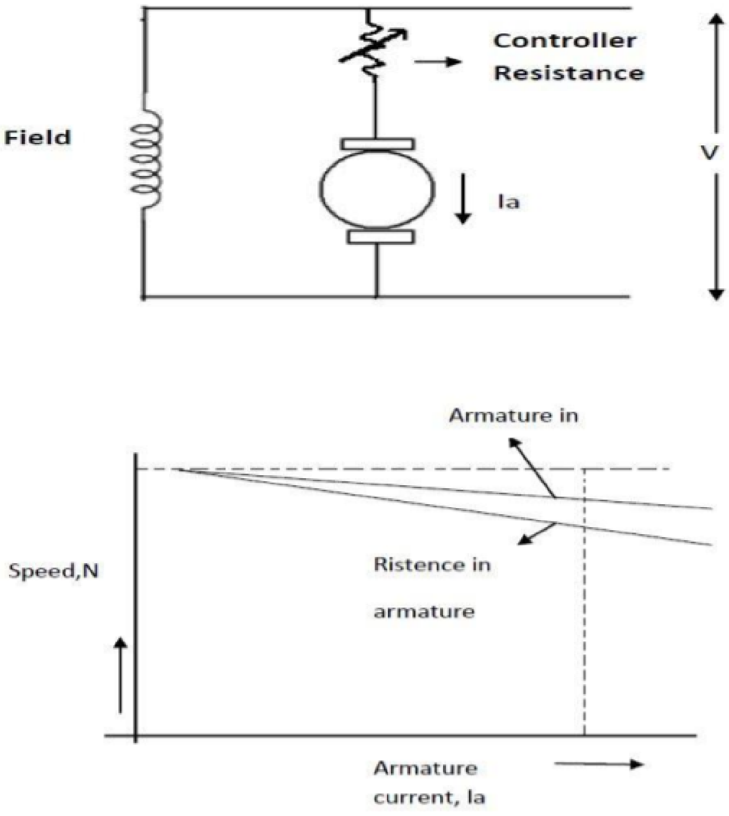


Figure 1.25: Rheostat control method [6]

Voltage regulation method

The variable regulation method is typically used in shunt dc motors. There are, again, two ways to achieve voltage regulation control [7]:

- Connecting the shunt field to a fixed exciting voltage while supplying the armature with different voltages (aka multiple voltage control)
- Varying the voltage supplied to the armature (aka the Ward Leonard method)

1.1.9 Drive voltage control techniques

Two ways of adjusting the drive voltage are linear control and PWM control.

Linear control works by placing a variable resistor in series with the motor, and adjusting the resistance to vary the voltage across the motor. While a transistor or other semiconductor device can be used as the serial-connected variable resistor, this approach has poor efficiency due to the large amount of heat generated by the resistance (semiconductor), and therefore it is rarely used these days [8].

An alternative technique is the PWM control. The voltage applied to the motor can be varied by turning a semiconductor switch (such as a transistor or an FET) on and off at high speed, with the voltage being determined by the on and off pulse widths. The high efficiency of this technique makes it most commonly used nowadays [8].

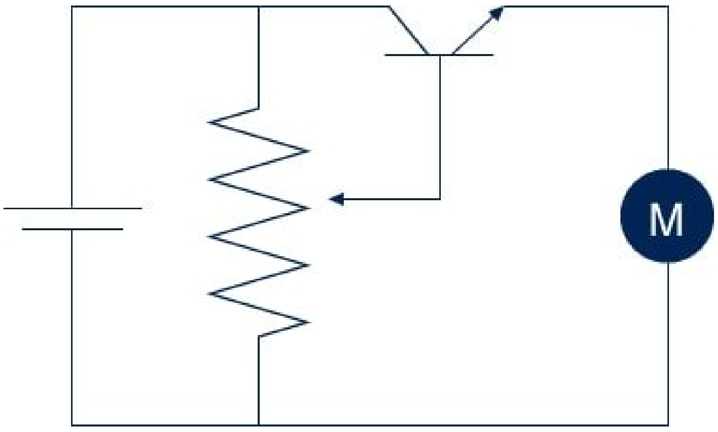


Figure 1.26: Linear control [8]

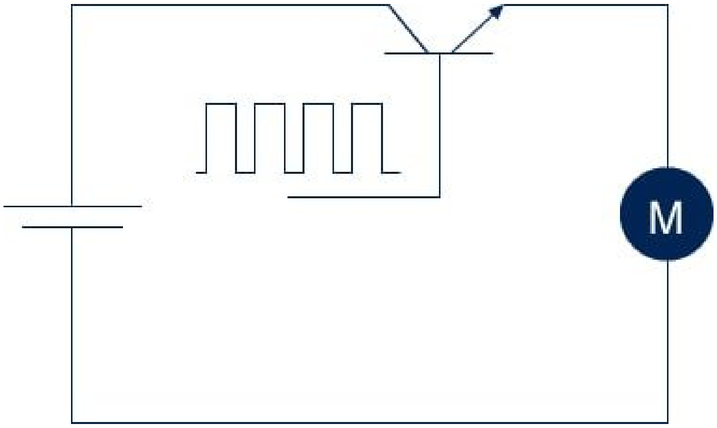


Figure 1.27: PWM control [8]

1.1.10 PWM technique

Pulse width modulation control works by switching the power supplied to the motor on and off very rapidly. The DC voltage is converted to a square wave signal, alternating between fully on (nearly 12v) and zero, giving the motor a series of power “kicks” [6].

Pulse width modulation technique (PWM) is a technique for speed control which can overcome the problem of poor starting performance of a motor. PWM for motor speed control works in a very similar way. Instead of supplying a varying voltage to a motor, it is supplied with a fixed voltage value (such as 12v) which starts it spinning immediately. The voltage is then removed and the motor coasts. By continuing this voltage on/off cycle with a varying duty cycle, the motor speed can be controlled [6].

Pulse-width modulation (PWM) or duty-cycle variation methods are commonly used in speed control of DC motors. The duty cycle is defined as the percentage of digital ‘high’ to digital ‘low’ plus digital ‘high’ pulse-width during a PWM period [6]. Fig.1.7 shows the 5V pulses with 0% through 100% duty cycle. The average DC Voltage value for 0% duty cycle is zero, with 20% duty cycle the average value is 1.2V (20% of 5V). With 50% duty cycle the average value is 2.5V, and if the duty cycle is 80%, the average voltage is 4V and so on. The maximum duty cycle can be 100%, which is equivalent to a DC waveform [6].

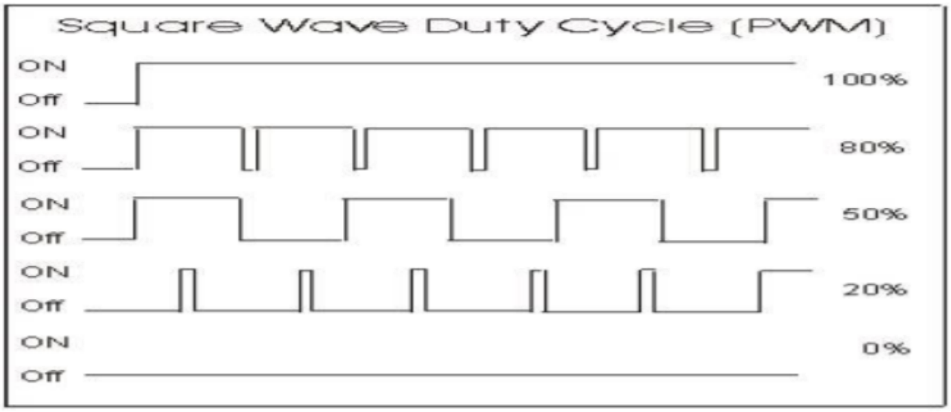


Figure 1.28: : 5V Pulses with 0% Through 100% duty cycle [6]

1.1.11 DC motor speed control using PWM

The major reason for using pulse width modulation in DC motor control is to avoid the excessive heat dissipation in linear power amplifiers. The heat dissipation problem often results in large heat sinks and sometimes forced cooling. PWM amplifiers greatly reduce this problem because of their much higher power conversion efficiency. Moreover, the input signal to the PWM driver may be directly derived from any digital system without the need for any D/A converters [6].

The PWM power amplifier is not without disadvantages. The desired signal is not translated to a voltage amplitude but rather the time duration (or duty cycle) of a pulse. This is obviously not a linear operation. But with a few assumptions, which are usually valid in motor control, the PWM may be approximated as being linear (i.e., a pure gain). The linear model of the PWM amplifier is based on the average voltage being equal to the integral of the voltage waveform [6].

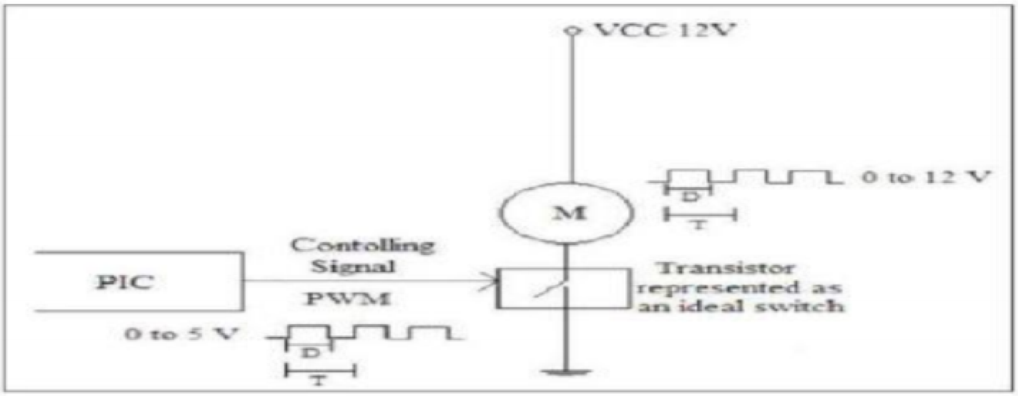


Figure 1.29: DC motor speed control using PWM [6]

1.1.12 Application of DC motors

The DC machine can operate as either a motor or a generator, at present its use as a generator is limited because of the widespread use of ac power. Large DC motors are used in machine tools, printing presses, fans, pumps, cranes, paper mill, traction, textile mills and so forth. Small DC machines (fractional horsepower rating) are used primarily as control device-such as tachogenerators for speed sensing and servomotors for position and tracking, and used in robots,drones,medical applications [4].

DC motors in robotics

A DC motor converts direct current electrical energy into mechanical energy. This differs from an AC motor, which applies alternating current to the electric motor. At the most basic level, DC motors work well in robotics because they allow, the robot to be battery powered, which offers great advantages for a variety of robotic applications, particularly mobile and collaborative robots [9]. that's not to say that other motors are not used for robotics, like AC synchronous brushless servo motors, which facilitate precise control of the robot's movements, the drive electronics of these motors actually work from a DC source, EMG30 motor is very popular at robotics filed and most use at robotic at this days in addition to enabling the use of batteries to facilitate mobility, two key features that make DC motors great for robotics are speed variation and torque [9].

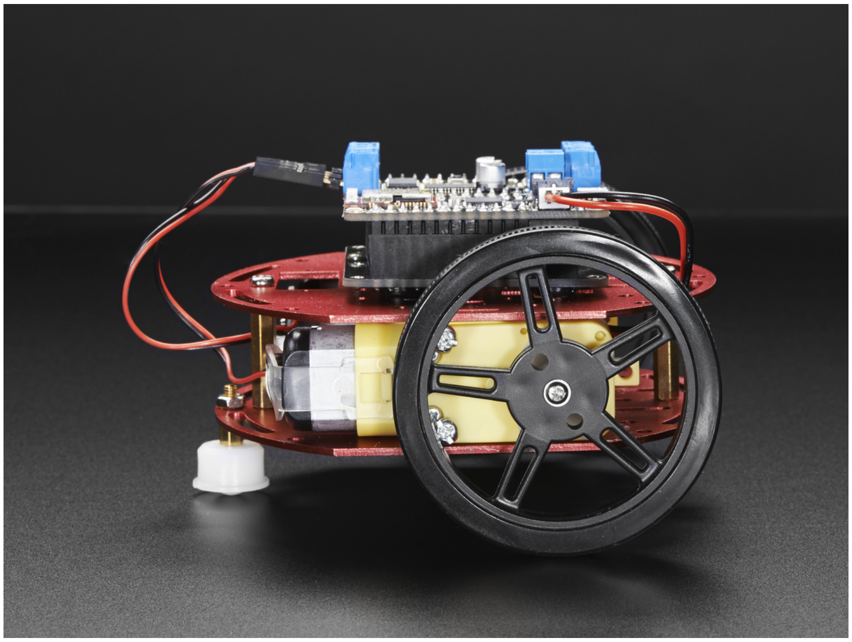


Figure 1.30: Robot with DC motor [9]

Industrial application

For industrial applications, brushless DC motors are primarily used in servo, actuation, positioning, and variable speed applications where precise motion control and stable operation are critical for the satisfactory operation of the manufacturing or industrial process. They are commonly used as [10]:

- Linear motors
- Servomotors
- Actuators for industrial robots
- Extruder drive motors
- Feed drives for CNC machine tools

Figure 1.31: Brushed DC Motor actuators for industrial robots [10]

Medical applications

DC gear motors are professionally designed for use in critical medical applications, whether they are intended for surgical, clinical or laboratory settings, these geared motors are used to give applications greater accuracy and control for medical professionals, such as a robotic surgical arm, joysticks to manoeuvre a delicate tool or an instrument with precision, DC gearmotors are small geared motors which can also be used in small fluid pumps, rotating trays in sterilisers and other automated laboratory functions, for all medical applications, precision microdrives DC gear motors offer not only the variable speed and torque control required in healthcare settings, but they also possess high-quality characteristics, such as reliability, ruggedness and compactness [11].

Figure 1.32: Gear DC motor are used in a variety of medical device [11]

Home applications

DC motor is very suitable for home applications the cause as high-efficiency, low-noise, small size and low-cost are demanded for the motors of home applications, as fans, pumps, cranes, wash machine, toy of kids[12].

Figure 1.33: DC Motors for Home Application [12]

1.1.13 Drone applications

DC motors have small permanent magnets made from iron and cobalt alloys. Using special alloys in the magnets makes them able to be smaller and lighter. Lithium-ion batteries are currently used to power drones as the weight to power storage ratio is better than any other battery. Motors are not just used for the propellers but are also used in the gimbal to steady cameras [13].

Figure 1.34: Drone with DC motors [13]

1.1.14 Advantages of DC motors

Good speed control

DC motors offer highly controllable speed. By changing the armature or field voltage it's possible to achieve wide speed variation and with this level of controllability, DC motors offer the precision required by a wide range of industry applications [1].

High torque

A DC motor also offers a high starting torque, which makes it perfect for use in applications that are designed to move heavier loads, such as wiper systems and in industrial automation applications, such as conveyor systems or materials handling equipment, the consistent drive power that DC motors deliver means they're ideal for maintaining a constant torque whilst an application is in use, making them an excellent choice for a geared motor solution [1].

Seamless operation

As DC motors operate with high levels of controllable power across a range of speeds, they offer the benefit of seamless operation. In some industries, it is vital that DC motors can start and stop efficiently to cope with the requirements of the application. If you are looking for a solution that offers rapid acceleration, an option to reverse direction and start/stop efficiency, a DC motor is a good choice [1].

Free from harmonics

In any electric power system, a harmonic is a voltage or current at a multiple of the fundamental frequency of the system, typically produced by the action of non-linear loads such as rectifiers or saturated magnetic devices. Harmonic frequencies in the power grid can be the cause of power quality problems and harmonics in some AC motors can cause torque pulsations, resulting in a decrease in torque. DC motors are free from issues associated with harmonics [1].

1.1.15 Disadvantages of DC motors

- Needs regular maintenance
- Cannot be used in explosive area.
- High cost

Chapter 2

Problem analysis and mathematical Modelling of a DC Motor

DC Motor plays a crucial role in research and laboratory experiments because of their simplicity and low cost. The speed of the motor can be controlled by many methods namely terminal voltage control, armature rheostat control method and flux control method.

A control system is an interconnection of components forming a system configuration that will provide a desired system response. A controlled DC-motor with PID controller was developed allowing Arduino hardware acts as the interface between the computer and the outside world. It primarily functions as a device that digitizes incoming digitals signals so that the computer can interpret them. The user interface was developed in an Matlab/Simulink environment [14].

2.0.1 Control system theory

The control system is that means by which any quantity of interest in a machine, mechanism or other equipment is maintained in accordance with a desired manner. All our tools and machines need appropriate control to work, otherwise it will be difficult to finish their designated tasks accurately. Therefore, we need control systems to guide, instruct and regulate our tools and machines. Common control systems include mechanical, electronic, pneumatic and computer aided. A component or process to be controlled can be represented by a block as shown in Figure 3.1 [14].

Figure 2.1: Process under control [14]

Process under control :

- Plant: The plant is also known as the process and it is the physical system to be controlled [14].
- Output or the Controlled Variable: It is the signal or the plant output which we need to control. In our case the speed of the DC Motor is the controlled variable which is to be control [14].
- Reference: It is the desired value that we want to see at the output. In this thesis set point or reference is the desired speed of the DC Motor [14].

There are basically two types of control system: The open loop system and the closed loop system. They can both be represented by block diagrams.

2.0.2 Open loop control

Any physical system which does not automatically correct for variation in its output is called an open-loop system. In these systems the output remains constant for a constant input signal provided the external conditions remain unaltered. The output may be changed to any desired value by approximately changing the input signal but variations in external conditions may cause the output to vary from the desired value in an uncontrollable fashion Examples of the open loop control systems include washing machines, light switches, gas etc [14].

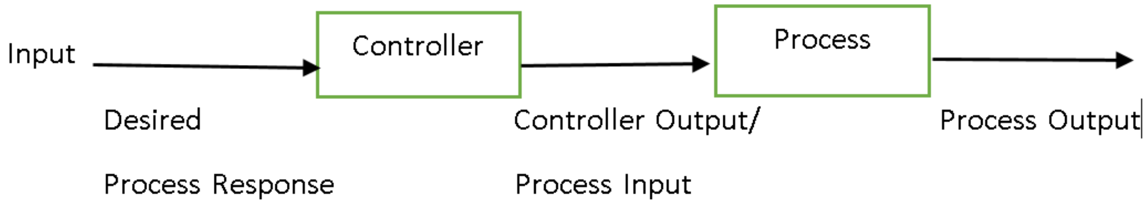


Figure 2.2: General block diagram of open-loop system [14]

2.0.3 Open-loop characteristics

- Shows an open-loop action (controlled chain).
- Can only counteract against disturbances, for which it has been designed, other disturbances cannot be removed.
- Cannot become unstable - as long as the controlled object is stable [14].

2.0.4 Closed loop control

Feedback is a special feature of a closed loop control system. A closed loop control system compares the output with the expected result or command status, and then it takes appropriate control actions to adjust the input signal. Therefore, a closed loop system is always equipped with a sensor, which is used to monitor the output and compare it with the expected result. The output signal is fed back to the input to produce a new output. A well-designed feedback system can often increase the accuracy of the output [14].

Feedback can be divided into positive feedback and negative feedback. Positive feedback causes the new output to deviate from the present command status. Negative feedback directs the new output towards the present command status, so as to allow more sophisticated control. Most modern appliances and machinery are equipped with closed loop control systems. Examples include air conditioners, refrigerators, etc [14].

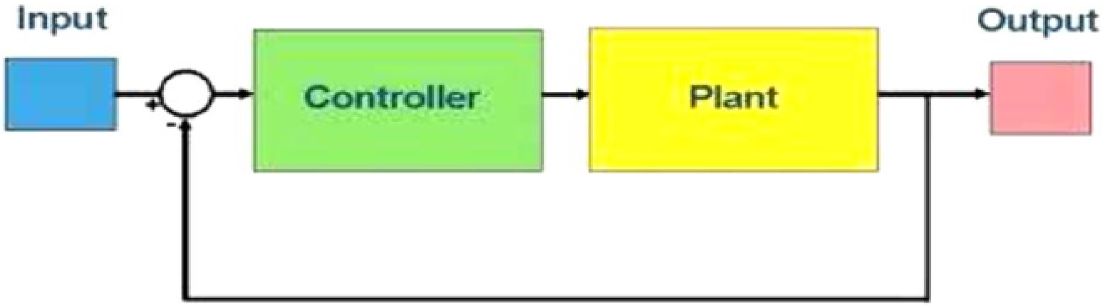


Figure 2.3: General block diagram of closed-loop system [14]

2.0.5 Closed-loop characteristics

- Shows a closed-loop action (closed control loop).
- Can counteract against disturbances (negative feedback).
- Can become unstable, i.e. the controlled variable does not fade away, but grows (theoretically) to an infinite value [14].

2.0.6 Difference between open loop and closed loop operation of DC motor

In open loop control of DC Motor the output (speed of the motor) can not be maintained at a desired value due to external disturbances and system variables, whereas in closed loop operation the output can maintained due to the presence of feedback circuit. Feedback circuit samples the output and gives signals to the error detector which compares the feedback signal with the specified value and produces a modified signal according to the output [14].

2.0.7 Controlling the speed of DC motor

For DC motors, speed is proportional to the voltage applied across its terminals. One way to control voltage applied to the DC motor is to use a potentiometer or another type of variable resistor in a voltage divider. Varying the potentiometer would vary the voltage applied, changing the speed of the motor. The problem with this method is that power is dissipated in the resistors, decreasing efficiency. A better way to control the voltage is to use Pulse Width Modulation (PWM). PWM uses a micro-controller to turn the voltage on and off, effectively changing the average voltage applied to the DC motor. Motor speed is increased by increasing the duty cycle, and is decreased by decreasing the duty cycle. The Arduino mega is capable of PWM for speed control [14].

2.0.8 DC motor speed controller (PWM of ARDUINO)

Function of a DC motor speed controller is to take as input a signal representing the demanded speed and to drive a motor at that speed. The controller may or may not measure the speed of the motor to use it as a feedback for the purpose of error reduction. If it does so, it is called a closed loop system else an open loop system [14].

The DC motor speed in general is directly proportional to the supply voltage, so if reduce the voltage from 12 volts to 6 volts then our speed become half of what it originally had. But in practice, for changing the speed of a DC motor we cannot go on changing the supply voltage all the time. So how to change the speed of the motor with input voltage fixed led to the development of systems known as speed controllers [14].

The speed controller for a DC motor works by varying the average voltage supplied to the motor (PWM is one such speed controller were we can get varying voltage according to the duty cycle of the PWM output signal). Rather than simply adjusting the voltage sent to the motor, we can switch the motor supply on and off where switching is done so much fast that the motor only notices the average effect [14].

The time it takes a motor to speed up slow down under switching conditions depends on the inertia of the motor. The graph below shows the speed of a motor that is switched on and off fairly slowly. If switching is done fast enough the motor does not have time to change speed and it gives an almost constant speed. Thus the speed is set by Pulse Width Modulation (PWM) [14].

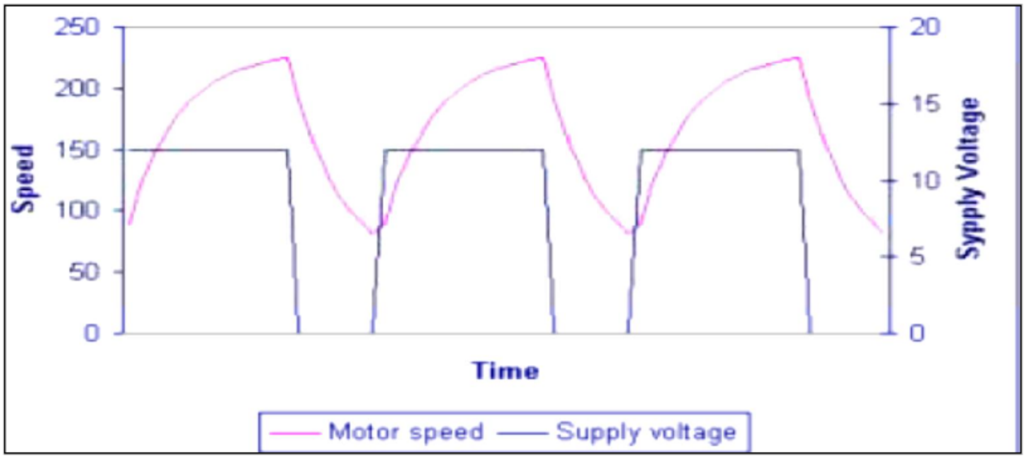


Figure 2.4: DC motor speed versus supply voltage [14]

2.0.9 The Impact of DC Motors Control

DC motor speed control and control of DC motors is a fundamental problem. It require to obtain a desire speed and full control of motor. The speed should be control automatically, but the velocity of DC motor should always be controlled and the desire velocity should be regulated [14].

control the speed of DC motors are mostly use in many machines in industry. This is a daunting task because it can affect both the performance and the flow of the process. It is therefore important to keep the speed at the set point. In most industries, control of motors pose many problems due to their non-linear dynamic behavior. Too high speed can upset the process equilibrium, damage equipment or result in the spillage of hazardous or valuable materials. If the speed is too low, it can have negative consequences for sequential operations.Therefore, the speed control and control of dc motors is an important and common element in the perfect process industries [14].

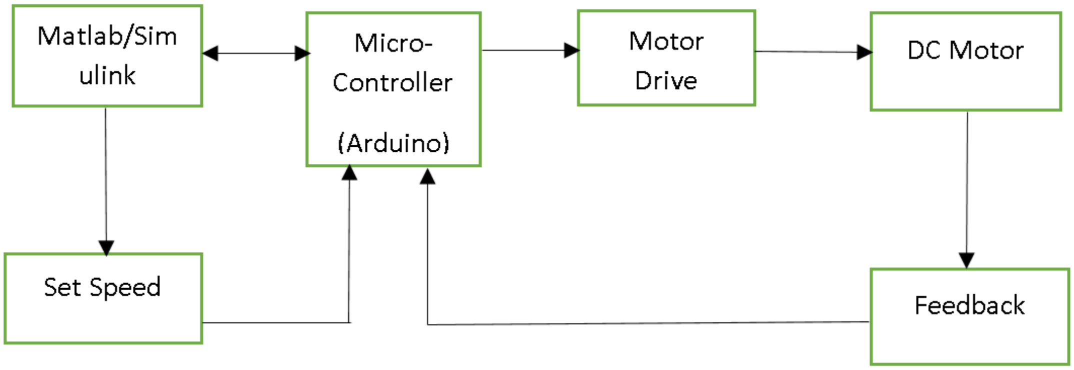


Figure 2.5: procedure of speed control

2.0.10 Mathematical model of DC motor

The response of DC Motor depends on its DC gain, (K), and time constant(t). Both (K) and (t) are function of DC Motor parameters. The objective of this chapter is to model a first-order DC Motor and investigate the effect of DC Motor parameters on its response to a step input. We choose to experiment with an armature controlled DC motor, which behaves as a first-order machine when the armature voltage is the input and the angular speed is the output. We obtain the transfer function of the DC motor and identify specific parameters of the system that affect system response with the PID controller to experimentally verify the change in system response.

2.0.11 The model mathematical of the EMG30 DC motor

The EMG30 is an actuator worldwide popular in the mobile robotics domain, being a low cost 12v motor equipped with encoders and a 30:1 reduction gearbox. The fact that it is equipped with encoders is an important feature because it provides important data to obtain the closed loop velocity control and to obtain relative measurements based on the odometry calculation [15].

The EMG30 model can be defined by the following equation, where u_a is the output voltage, R_a is the equivalent resistor, L_a is the equivalent inductance, (e) is the back emf (electromotive force) voltage, I_a is the motor current as expressed by equation [15].

$$u_a(t) = e(t) + R_a i_a(t) + L_a \frac{di_a(t)}{dt} \tag{2.1}$$

The motor can provide a torque T_l that will be applied to the load, being the developed torque T_d subtracted by the friction torque, which is the sum of the static friction T_c and viscous friction, as shown in equation [15].

$$T_l = T_d - T_c - B\omega \tag{2.2}$$

Current i_a can be correlated with the developed torque T_d through equation (3.1), the back emf voltage can be correlated with angular velocity through equation (3.2) and the load torque T_l can be correlated with the moment of inertia and the angular acceleration through equations.

$$T_d = k_s i_a \tag{2.3}$$

$$e = k_s \omega \tag{2.4}$$

$$t_L = J\dot{\omega} \tag{2.5}$$

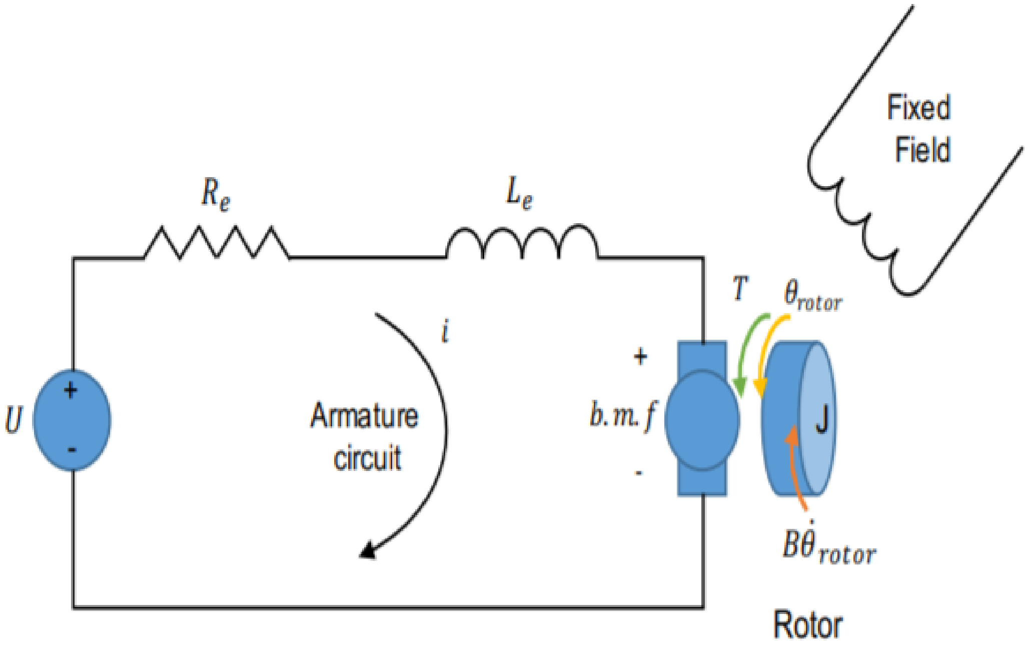


Figure 2.6: The schematic diagram of DC motor [16]

Then,

$$T_{xm} = J_m \cdot \frac{d\theta^2(t)}{dt} + B_{\mathrm{m}} \cdot \frac{d\theta(t)}{dt} \tag{2.6}$$

were the transfer function will be :

$$G(s) = \frac{\theta}{V} = \frac{K}{(Js+b)(Ls+Ra)+K^2}. \tag{15}$$

Where;

- V_a = armature voltage (v)
- R_a = armature resistance Ω
- L_a = armature inductance (H)
- I_a = armature current (A)

- E_b = back emf (V)
- w = angular speed (rad/s)
- T_m = motor torque (Nm)
- θ = angular position of rotor shaft (rad)
- $ginJ_mj_m$ = rotor inertia
- (kg $m^2m^2)B_m$ = viscous friction coefficient (Nms/rad)
- $K_T =$ torque constant (Nm/A)
- K_{b} = back.emf constant (Vs/rad)

The DC motor under study has the following specifications and parameters are given in Table (3.1):

parameters	**value**
Ra	7.101 Ω
La	0.00019 H
Kb (Back emf)	0.360 Vs/rad
Jm (Moment of inertia)	0.0034 kg. m2
Bm (Frictional constant)	0.000600 (Nms/rad)

Table 2.1: The Specification and Parameters of EMG30 DC Motor [15]

2.0.12 System simulation

open loop Simulink model

All models of DC motor blocks which was developed mathematically have been integrated into a blocks in the Simulink environment. The block has the same inputs and outputs as the real system. Thus,it contains:

- input of command speed

- output : output speed

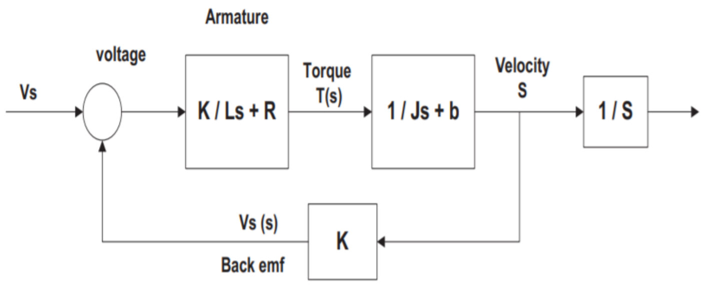


Figure 2.7: Block diagram of DC Motor Transfer function Mathematical Model

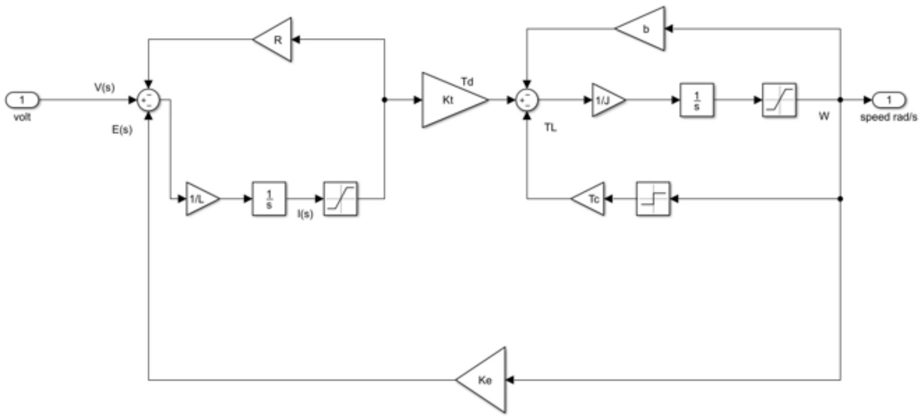


Figure 2.8: Simulink model of EMG30 DC motor

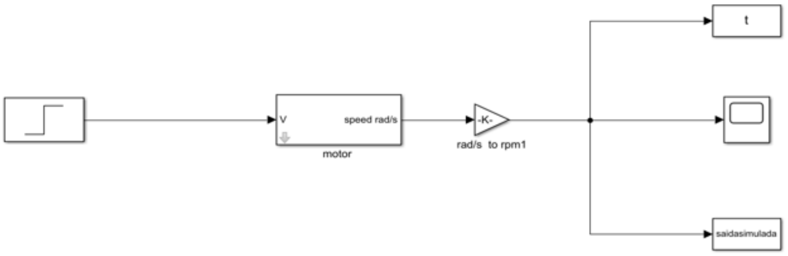


Figure 2.9: Simulink model of open loop

step response of model dc motor in 2s simple time from the plot we see that when 12 Volt is applied to the system the motor can only achieve a maximum speed of 211rpm, Also, it takes the motor 0.8 seconds to reach its steady-state speed from 2s simple time

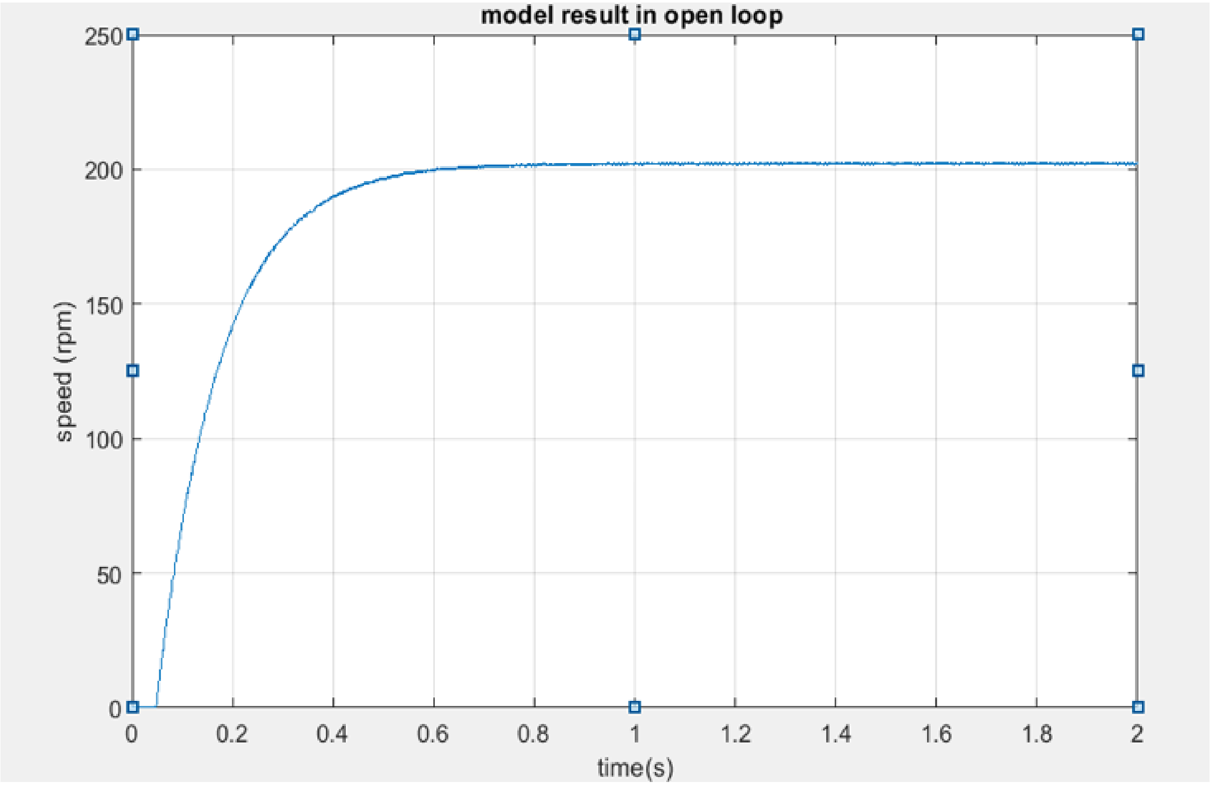


Figure 2.10: step response of dc model dc motor in open loop

2.0.13 PID Controller

In the processing industries, different control strategies are used nowadays, according to their needs and in a practical way, insufficient permanent precision, instability, too high a response time, too great an overrun, with regard to the specifications of a specification. It is therefore often necessary to integrate an organ called a corrector (Regulator) into the loop, the objective of which is to improve the performance of the system. It is the control mechanism that uses the information from the measurements of the controlled variable to manipulate a variable in order to obtain the desired result. The controller is driven by the error between the actual process output and the setpoint [17].

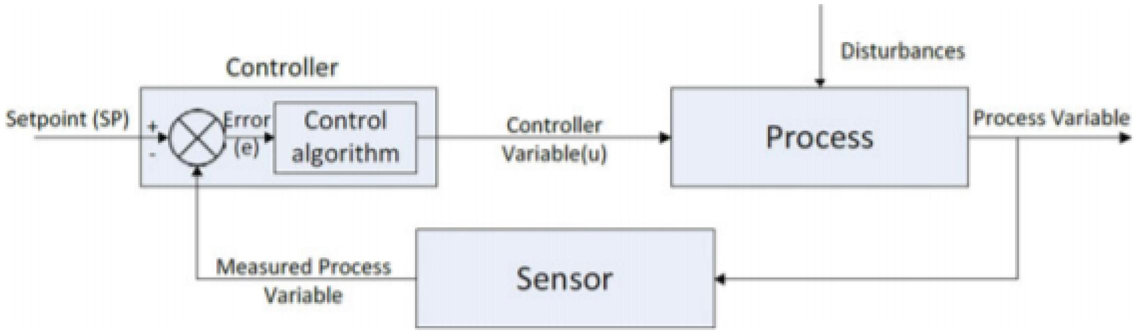


Figure 2.11: Functional diagram of a system with regulator [17]

Proportional-integral-derivative (PID) controllers are widely used in industrial control systems because of the reduced number of parameters to be tuned. They provide control signals that are proportional to the error between the reference signal and the actual output (proportional action), to the integral of the error (integral action), and to the derivative of the error (derivative action) namely [18].

$$\mathrm{U(t) = K_P\left[e(t) + \frac{1}{T_i}\int_0^t e(r)d\tau + T_d\frac{d}{dt}e(t)\right]} \tag{2.7}$$

Where u(t) and e(t) denote the control and the error signals respectively, and Kp, Ti and Td are the parameters to be tuned. The corresponding transfer function is given as [18].

$$\mathrm{K(s) = K_p\left[1 + \frac{1}{T_i(s)} + T_d(s)\right]} \tag{2.8}$$

These functions have been enough to the most control processes. Because the structure of PID controller is simple, it is the most extensive control method to be used in industry so far. The PID controller is mainly to adjust an appropriate proportional gain (Kp), integral gain (KI), and differential gain (KD) to achieve the optimal control performance.

The PID controller system block diagram of this paper is shown in Figure bellow. Transfer function can also be expressed as [18].

$$\mathrm{K(s)} = \frac{\mathrm{U(s)}}{\mathrm{E(s)}} = \mathrm{K_p} + \frac{\mathrm{K_i}}{\mathrm{S}} + \mathrm{K_d S} \tag{2.9}$$

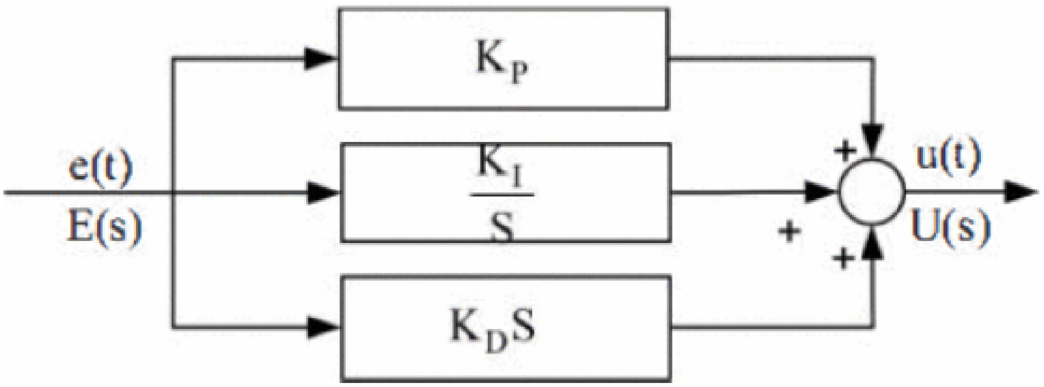


Figure 2.12: PID controller block diagram [18]

The main features of PID controllers are the capacity to eliminate steady-state error of the response to a step reference signal (because of integral action) and the ability to anticipate output changes (when derivative action is employed).

2.0.14 PD Controller

A PD controller is described by the transfer function:

$$K(s) = k_p + k_d s = k_d \left(s + \frac{k_p}{k_d} \right) \tag{2.10}$$

A PD controller thus adds a single zero to the loop transfer function. The closed-loop characteristic polynomial is given as: The phase contribution of the PD controller increases from 0 at low frequencies to 90 at high frequencies. For practical reasons, a pole with a short time constant, Tf, may be added to the PD controller. The pole helps limit the loop gain at high frequencies, which is desirable for disturbance rejection. The modified PD controller is described by the transfer function [2] :

$$K(s) = k_p + \frac{k_d s}{T_f s + 1} \tag{2.11}$$

The modified PD controller is very similar to a first-order phase-lead controller, it is similarly employed to improve the transient response of the system.

2.0.15 PI Controller

A PI controller is described by the transfer function:

$$K(s) = k_p + \frac{k_i}{s} = \frac{k_p(s + k_i/k_p)}{s} \tag{2.12}$$

The PI controller thus adds a pole at the origin (an integrator) and a finite zero to the feedback loop. The presence of the integrator in the loop forces the error to a constant input to go to zero in steady-state, hence PI controller is commonly used in designing servomechanisms [2].

The controller zero is normally placed close to the origin in the complex s-plane. The presence of a pole–zero pair adds a closed-loop system pole with a large time constant. The zero location can be adjusted so that the contribution of the slow mode to the overall system response stays small [2].

2.0.16 Design a PID controller using an Bode plot reshape

From the main problem, the dynamic equations in the Laplace domain and the open-loop transfer function of the DC Motor are the following:

$$s(Js + B) = KI(s) \tag{2.13}$$

$$I_s(Ls + R) = V(s) - k_s\theta_(s) \tag{2.14}$$

$$p(s) = \frac{\theta}{V} = \frac{K}{(Js+b)(Ls+Ra)+K^2}. \qquad (15)$$

The structure of the control system has the form shown in the figure below.

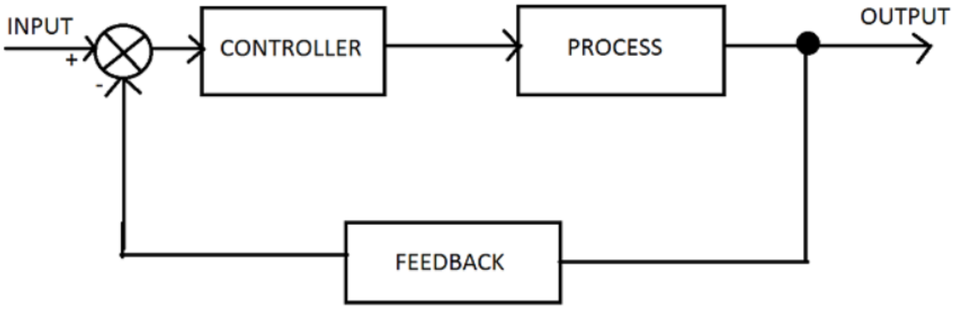


Figure 2.13: The structure of the control system closed loop

- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-state error less than 1%

Now let's design a controller using the Matlab to Create bode blot reshapre with following parameters of dc motor .

- Ra=7.101 resistence of armature (ohm)
- La=0.00019 inductance (H)
- J=0.0034 momente of inercia of rotor (kgm2/s2)
- B=0.0006 coef of fricion on thr motor (Nm/s)
- Kb=0.360 constante of motor (Nm/A)

2.0.17 Drawing the original Bode plot

The main idea of frequency-based design is to use the Bode plot of the open-loop transfer function to estimate the closed-loop response. Adding a controller to the system changes the open-loop Bode plot, thereby changing the closed-loop response. It is our goal to design the controller to shape the open-loop Bode plot in such a way that the closed-loop system behaves in a desired manner.

2.0.18 Adding proportional gain

From the Bode plot above, it appears that the gain margin and phase margin of this system are currently infinite which indicates the system is robust and has minimal overshoot. The problem with this is that the phase margin is infinite because the magnitude plot is below 0 dB at all frequencies. This indicates that the system will have trouble tracking various reference signals without excessive error. Therefore, we would like to increase the gain of the system while still achieving enough phase margin.

A phase margin of 60 degrees is generally sufficient for stability margin. From the above Bode plot, this phase margin is achieved for a crossover frequency of approximately 10 rad/sec. The gain needed to raise the magnitude plot so that the gain crossover frequency occurs at 10 rad/sec appears to be approximately 40 dB. The exact phase and gain of the Bode plot at a given frequency can be determined by clicking on the graph at the corresponding frequency. The bode command, invoked with left-hand arguments, can also be used to provide the exact phase and magnitude at 10 rad/sec as shown below.

mag = 1.3042, phase = -61.0061, w = 10

Therefore, the exact phase margin for a gain crossover frequency of 10 rad/sec is 180 - 61.7729 = 118.9939 degrees. Since the exact magnitude at this frequency is 20 log 1.1.3042 = -17.693 dB of gain must be added to the system. Otherwise stated, a proportional gain of 1/1.3042 = 0.76 will achieve an open-loop gain of 1 at 10 rad/sec. Add the following commands to your m-file to observe the effect of this proportional controller on the system. In this case, we use the margin command instead of the bode command in order to explicitly see the new gain and phase margins and crossover frequencies.

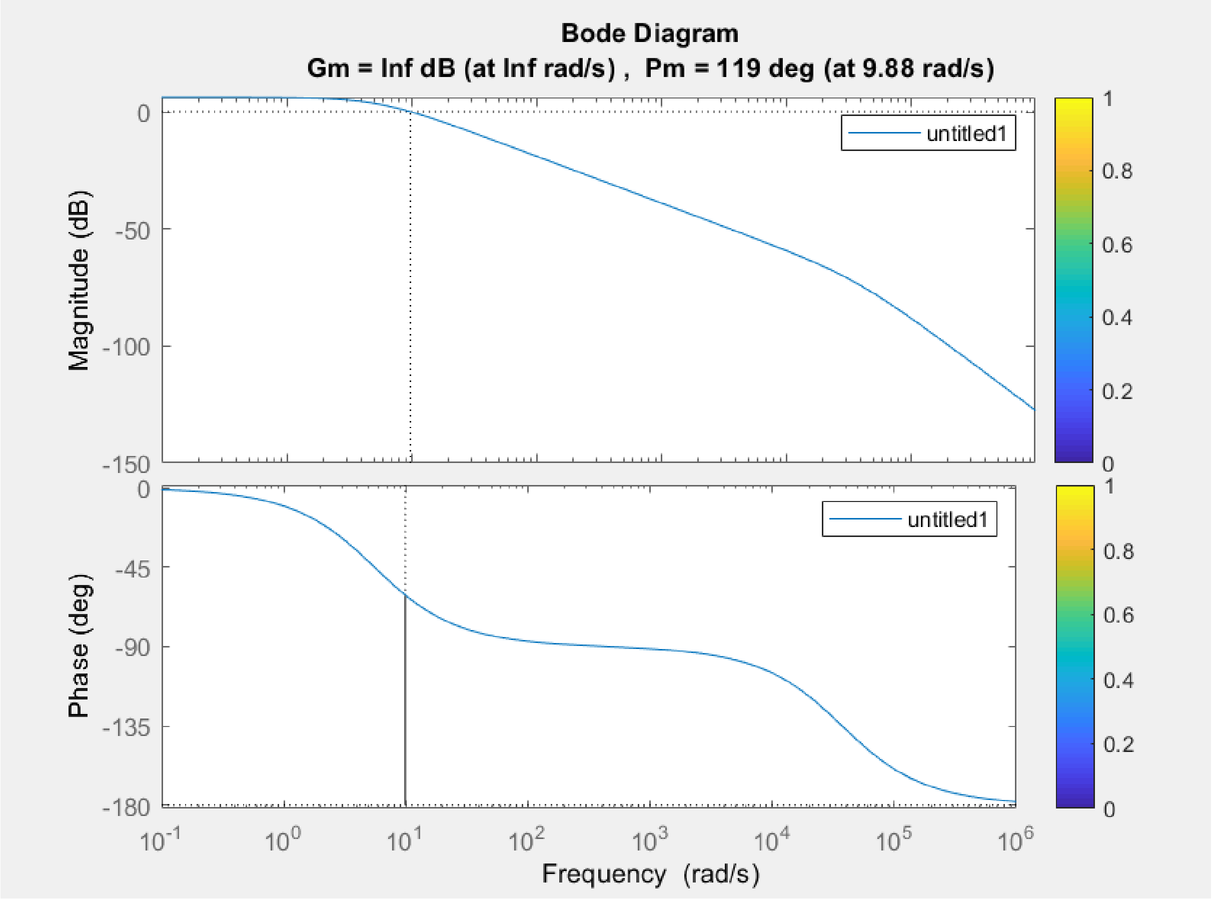


Figure 2.14: Bode diagram

2.0.19 Plotting the closed-loop response

From the plot above we see that the resulting phase margin and gain crossover frequency are as we expected. Let's see what the closed-loop response look like.

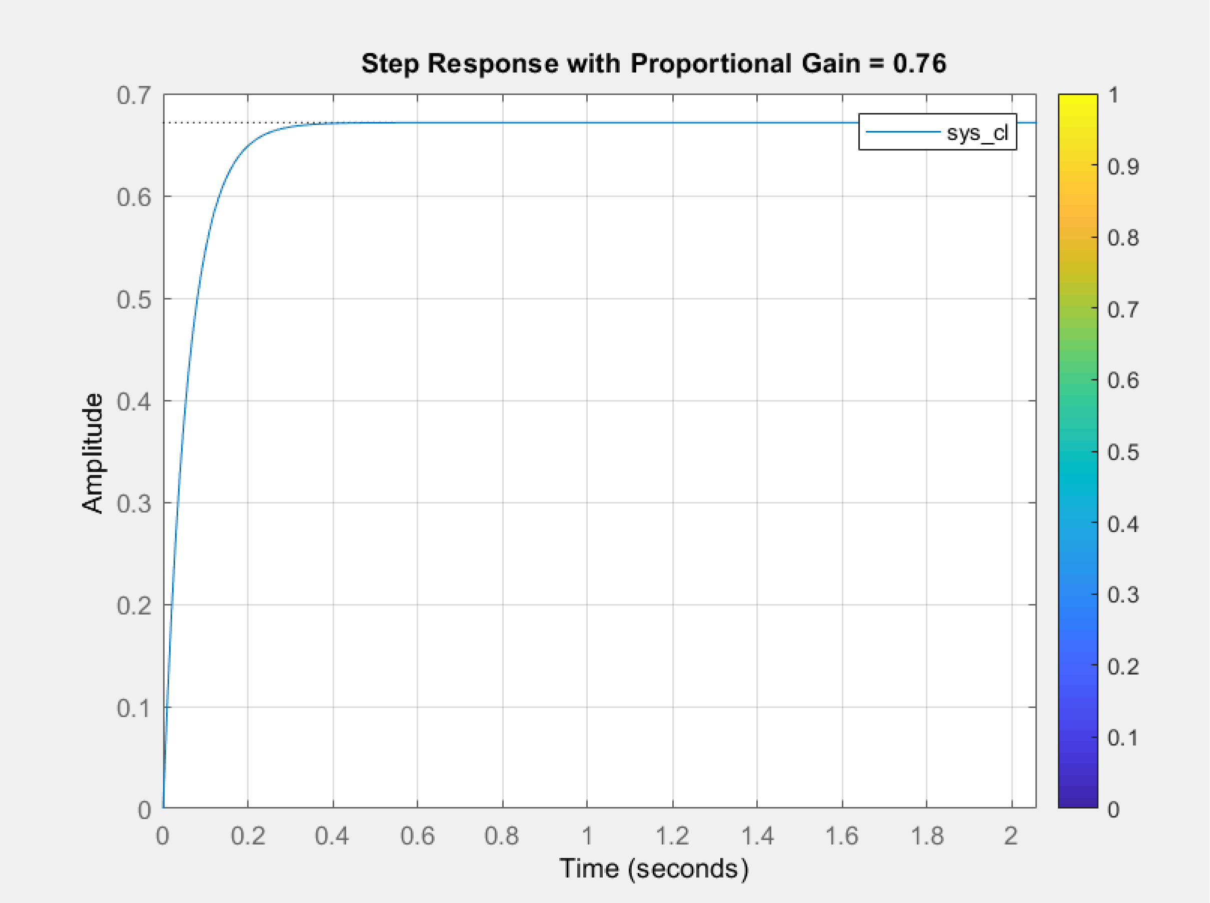


Figure 2.15: Step Response with Proportional Gain = 0.76

Note that the settling time is fast enough, the overshoot and the steady-state error are more appropriate. The overshoot Overshoot less than 5%, the steady-state error become even Steady-state error less than 1%. A lag compensator could not be helpful here in that it can decrease the gain crossover frequency in order to increase the phase margin without decreasing the system's DC gain.

2.0.20 Closed loop simulation

Add a PID to the EMG30 mathematically simulink model to compare it to the open loop system. We use computational tools like MATLAB/simulink to speed up modeling.

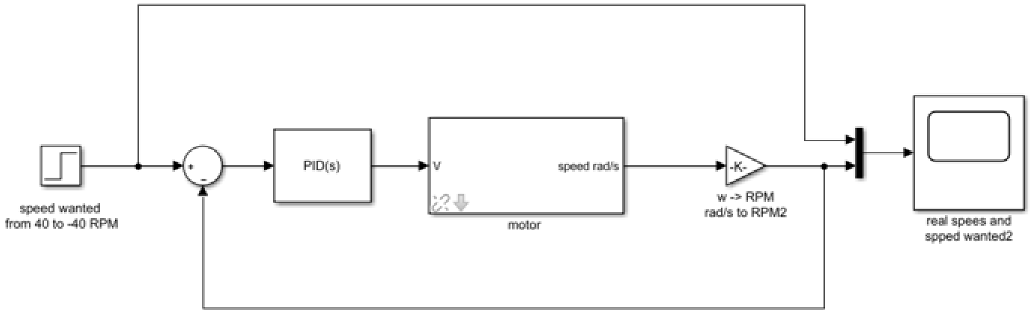


Figure 2.16: Block Simulink of model in closed loop

2.0.21 PID controller tuning

The PID controller tuning refers to the selection of the controller gains: kp, kd, ki to achieve desired performance objectives. Industrial PID controllers are often tuned using empirical rules, such as the Ziegler–Nichols rules [2].

2.0.22 Ziegler-Nichols method

Frequency response with Ziegler-Nichols method

Critical point method of Ziegler-Nichols (second method of Ziegler-Nichols)

This method is based on the knowledge of the critical point of the process. Experimentally, the process is looped on a simple proportional regulator whose gain is increased until the system oscillates permanently. The gain is increased until the system is permanently oscillating, we thus find ourselves at the limit of stability. After having determined the critical gain Kcr of the controller and the period of time.

The parameters of the chosen regulator can be calculated with the help of table. Here again, the proposed values lead to a relatively short rise time, unfortunately with a high overshoot this situation is not always satisfactory, so we may have to correct the proposed coefficients and, in particular, to reduce the gain KP. It should be noted that the parameters Ti and Td proposed by the two Ziegler-Nichols methods are in a constant ratio 'equal' to those proposed by the two methods. Methods are in a constant ratio. The regulator thus has two merged zeros worth -1/(2Td) = -2/Ti .

Controller type	**Kp**	**ki**	**kd**
P	0.5Kcr	-	-
PI	0.4Kcr	0.8Tcr	-
PID	0.6Kcr	0.5Tcr	o.125Tcr

Table 2.2: Ziegler-Nichols second method table of PID parameters

On the frequency response of the process Gp(s), we have measured:

Critical gain:

$$k(\text{cr}) = \frac{1}{G\pi} \tag{2.15}$$

The critical period:

$$T(\text{cr}) = \frac{2\pi}{w\pi} \tag{2.16}$$

The frequency response of an aperiodic process is illustrated.

- Response, the following quantities are defined
- The pulsation for which the phase is -180
- the gain G corresponding to this pulsation
- the critical gain Kcr that must be introduced in the loop system to make it unstable [18].

$$\text{k}(0) = G(0) \tag{2.17}$$

$$w(\text{cr}) = w\pi \tag{2.18}$$

$$k(\mathrm{cr}) = \frac{1}{G\pi} \tag{2.19}$$

$$k = \frac{G\pi}{G0} \tag{2.20}$$

$$k = \frac{G\pi}{G0} = \frac{1}{k(\mathrm{cr})\mathrm{k}(0)} \tag{2.21}$$

From the Ziegler-Nichols table, we get the 3 parameters of the PID controller:

- Kp=0.6*Kcr
- ki=0.5*Tcr
- Kd=0.125*Tcr

From the method and calculation the parameters and tune the parameters and applicate it for our system we found the result: KP=0.045 Ki=0.31 KD=0.0007

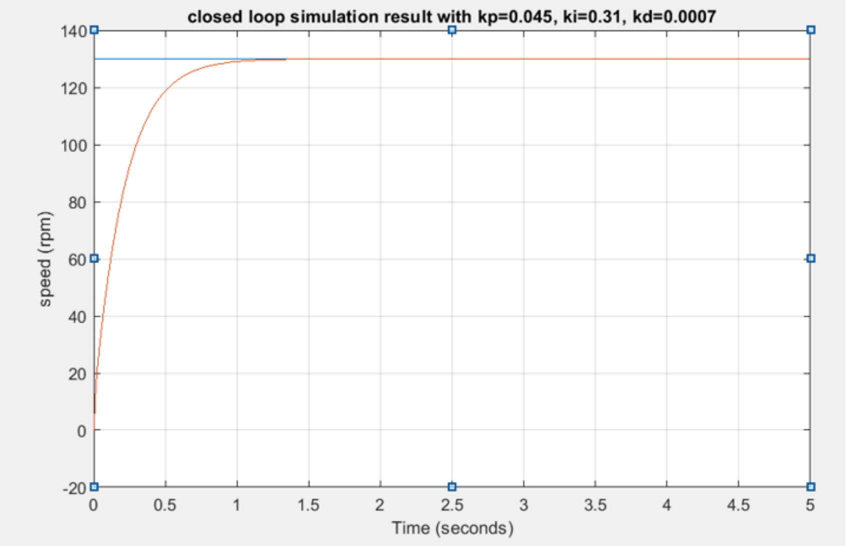


Figure 2.17: Closed loop result of model simulation

2.0.23 Overview of coefficient diagram method (CDM)

CDM is a polynomial algebraic approach and proposed by Manabe in the year 1991. The algebraic approach (CDM) was then said to be an alternative for conventional and modern control theories and uses polynomial expression for the mathematical representation. The advantageous parts of these control theories are combined to form the principles of CDM and it is derived by using the previous experience and knowledge about the controller design. Without confronting with serious difficulties and necessitating much experience, CDM makes possible to design very good controllers with less effort and relative ease when compared with the other existing methods (Manabe, 1998). By comparing the existing method, it is very easy to design a controller under the conditions of stability, time domain performance and robustness. Also, CDM is less sensitive to disturbances and bounded uncertainties resulted from the parameter variations. The important property of CDM is that the designer can have complete control over the transient response by specifying the key parameters namely stability indices (yi) and equivalent time constant (t) at the beginning of the design. The simultaneous design nature exists in CDM, gives advantages to designer to keep good balance between the rigor of the requirements and the complexity of the controller [19].

2.0.24 Mathematical model

The standard CDM block diagram for single input single output system is shown in Fig where, (y) is the output signal, (r) is the reference input, (u) is the controller signal and (d) is the external disturbance signal. N(s) and D(s) are numerator and denominator polynomials of the plant transfer function. A(s) is the forward denominator polynomial while F(s) and B(s) are the reference numerator and the feedback numerator polynomials of the controller transfer function [19].

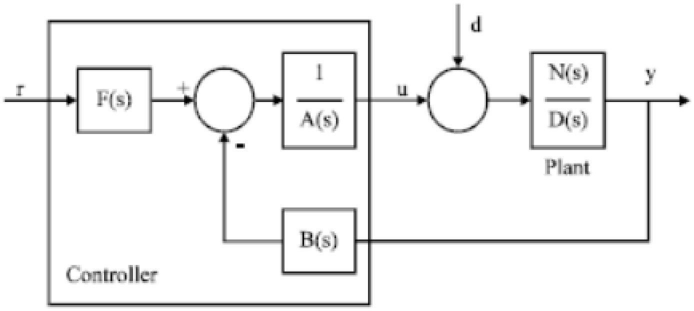


Figure 2.18: Standard CDM block diagram [19]

In Figure 2.19, A(s),B(s), and F(s) are controller polynomials. The system to be controlled is represented by G(s) transfer function given by [20]:

$$G(s) = \frac{N(s)}{D(s)} \tag{2.22}$$

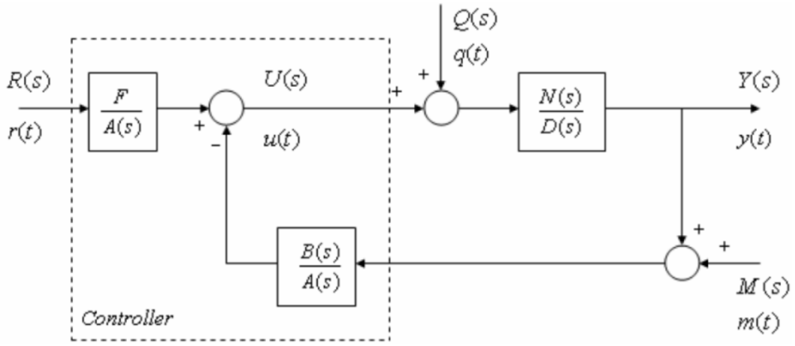


Figure 2.19: Equivalent block diagram of CDM [19]

where, P(s) is the characteristic polynomial of the closed-loop system and defined by [20]:

$$P(s) = A(s)D(s) + B(s)N(s) = \sum_{j=0}^{n} a_i s^i \tag{2.23}$$

CDM needs some design parameters with respect to the characteristic polynomial coefficients such as τ the equivalent time constant, γ_i the stability indices and γ_i^* the stability limits. The relations between these parameters and the coefficients of the characteristic polynomial a_i are given in relations [20]:

$$\gamma_j = \frac{a_i^2}{a_{i+1}a_{i-1}} \quad i = 1\ldots(n-1)\tau = \frac{a_1}{a_0} \text{ and } \gamma_i^* = \frac{1}{\gamma_{i-1}} + \frac{1}{\gamma_{i+1}} \tag{2.24}$$

Using these relations in Equation (2.24), it is possible to formulate the characteristic polynomial P(s) in terms of the design parameters τ and γ_i as follows [20]:

$$P_T(s) = a_0\left[\left\{\sum_{i=2}^{n}\left(\prod_{j=1}^{i-1}\frac{1}{\gamma_{i-j}^{j}}\right)(\tau s)^i\right\} + \tau s + 1\right] \tag{2.25}$$

$P_T(s)$ is the Target Characteristic Polynomial (TCP). Note that the a_{i} coefficients of the characteristic polynomial can be expressed as [20]:

$$a_i = \frac{\tau^i}{\prod\limits_{j=1}^{i-1}\gamma_j^{i-j}}a_0 \tag{2.26}$$

This is a relation between equivalent time constant τ and the settling time T_s, since the time constant can be written as $\tau = T_s/\alpha$, where $\alpha \in [2.5, 3]$ [20].

2.0.25 Mathematical model

The standard block diagram of the CDM design for single-input-single-output system is shown in Fig. 2.21 The extension to multi-input-multi-output can be made with proper interpretation, but it is not discussed here for simplicity. The plant equation is given as [21]:

$$A_p(s)x = u + d, (1a)y = B_p(s)x, (1b) \tag{2.27}$$

where (u) ,(y), and (d) are input, output, and disturbance. The symbol (x) is called the basic state variable. Ap(s) and Bp(s) are the denominator and numerator polynomial of the plant transfer function Gp(s) It will be easily seen that this expression has a direct correspondence with the control canonical form of the state-space expression, and (x) corresponds to the state variable of the lowest order. All the other states are expressed as the derivatives of (x) of high order. Controller equation is given as [21]:

$$A_c(s)u = B_a(s)y_r - B_c(s)(y + n) \tag{2.28}$$

where (yr) and (n) are reference input and noise on the output. Ac(s) is the denominator of the controller transfer function. Ba(s) and Bc(s) are called the reference numerator and feedback numerator of the controller transfer function. Because the controller transfer function has two numerators, it is called two-degree-of-freedom system. This expression corresponds to the observer canonical form of the state-space expression. Elimination of (y) and (u) from (2.28) gives [21]:

$$P(s)x = B_a(s)y_r + A_c(s)d - B_c(s)n \tag{2.29}$$

where P(s) is the characteristic polynomial and given as:

$$P(s) = A_c(s)A_p(s) + B_c(s)B_p(s) \tag{2.30}$$

For CDM design, the following four basic relations are selected as standard, namely [21]:

P(s)x=P(0)y$_r$, (4a)

P(s)y=B$_p(s)B_a(s)y_r$, (4b)

P(s)y=B$_p(s)A_c(s)d$, (4c)

P(s)(-y)=B$_n(s)B,(s)n.$(4d)

Equation (4a) is the response of (x) to (yr) when Ba(s)=P(0), and it is the 0-th order canonical transfer function of P(s), this equation specifies the characteristic polynomial, and it is a very good measure of stability and response speed. Equation (4b) is for the command following characteristics. Eq. (4c) is for the disturbance rejection characteristics. Equation (4d) corresponds to the complementary sensitivity function T(s), and it is useful for checking the robustness. In the CDM design, these four basic relations are used as performance specification. The design of P(s) is first made to satisfy specifications on (4a) (4c) (4d), and then Ba(s) is adjusted to satisfy the specification on (4b) [21].

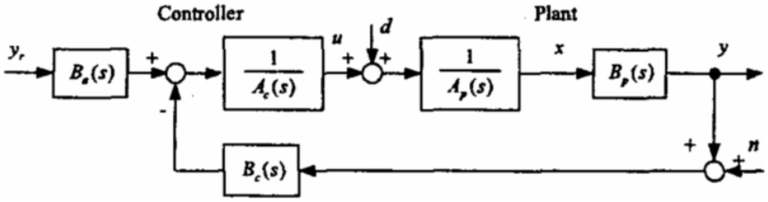


Figure 2.20: Mathematical model [21]

2.0.26 Mathematical Relations

Some mathematical relations extensively used in CDM will be introduced hereafter. The characteristic polynomial is given in the following form [21] .

$$P(s) = a_n s^n + \cdots + a_1 s + a_0 = \sum_{i=0}^{n} a_i s^i \tag{5)(2.31}$$

The stability index γ_i, the equivalent time constant τ and stability limit γ_i are defined as follows [21]:

$\gamma_i = a_1^2/(a_{i+1}a_{i-1}), \qquad i = 1 \sim n-1$

$\tau = a_1/a_0$

$\gamma_i^* = 1/\gamma_{i+1} + 1/\gamma_{i-1},$

i=1$\sim n-1, \gamma_n = \gamma_0 = \infty$

Also the equivalent time constant of the i-th order τ_i is defined as follows:

$\tau_i = a_{i+1}/a_i, \qquad i = 1 \sim n-1$

Then the following relations are derived.

$\tau_i = \tau_{i-1}/\gamma_i = \tau(\gamma_i \cdots \gamma_2\gamma_1)$

a$_i = \tau_{i-1} \cdots \tau_2\tau_1\tau \ a_0,$

=a$_0\tau^i/(\gamma_{i-1}\gamma_{i-2}^2\cdots\gamma_2^{i-2}\gamma_1^{i-1})$

The characteristic polynomial will be expressed by a_0, τ and γ_i as follows [21]:

P(s)=a$_0[\{\sum_{i=2}^{n}(\prod_{j=1}^{i-1} 1/\gamma_{i-j}^{j})(\tau s)^i\} + \tau s + 1]$

Notice here that P(s) is expressed in (s)i, and its coefficients are sole functions of γ_i Thus for given γ_i, the response shape of (4a) is similar irrespective of τ. For different τ, the response speed changes, while the response shape remains similar [21]:

2.0.27 Controller design

The most important feature of CDM is the simultaneous design of characteristic polynomial and controller. The controller structure has a great effect on robustness. The traditional design principle of sticking to the minimum-phase controller wherever possible, with the lowest-possible order, and with the narrowest possible bandwidth is actually found to be a strong guarantee of robustness. If such consideration is to be given, the stability index and equivalent time constant can not be chosen freely, but should be chosen with such care that the specific Diophantine equation is satisfied. The problem is then as follows [21]:

“Given some parts of controller parameter and some parts of characteristic polynomial, Find the rest of controller parameters and characteristic polynomial.” [21].

In solving such vaguely defined problem, human intuition on the basis of graphical representation is found to be most powerful. The coefficient diagram is a very powerful means for this purpose. By intuition, the designer can finish a rough design. For the rest of the refinement, he can rely on computer power [21].

Usually stability index and equivalent time constant are chosen in the following range [21]:

$\gamma_1 = 2.5, \gamma_2 = \gamma_{n-1} = 2,$

$\gamma_i > 1.5\gamma_i^*, \quad i = n-1 \sim 3,$

$\gamma_i = 1.5 \sim 4, \quad i = n-1 \sim 1,$

$\tau \simeq (1/3)$ of settling time

Contrary to pole assignment, the equivalent time constant becomes a trade-off issue as well as stability index and controller parameters. Kessler proposed 2 as the standard stability index for all order . In CDM standard form, $\gamma_1 = 2.5$ and rest of stability indices are 2. For simple system, the CDM standard form suffices. But for more complex system, some adjustment of γ_i is usually required [21].

Application

Consider the second order system transfer function :

$$G(s) = \frac{0.360}{(0.000000646)s^2 + (0.02414)s + (0.1339)}. \tag{15}$$

the continuous and digital characteristic polynomials are obtained as:

$$P_T(s) = 0.000000646s^3 + 0.0024s^2 + 4.458s + 3311.66 \tag{2.32}$$

According to the target characteristic polynomial PT(s) the controller polynomials of the CDM are calculated as :

$$A(z) = 1.486s + 1103.886$$

B(z)=s+5.500 (2.33)

$$k(s) = \frac{1.486s + 1103.886}{(s + 5.500)}.$$

$$f(s) = \frac{1103.886}{(1.486s + 1103.886)}.$$

CDM simulation closed loop

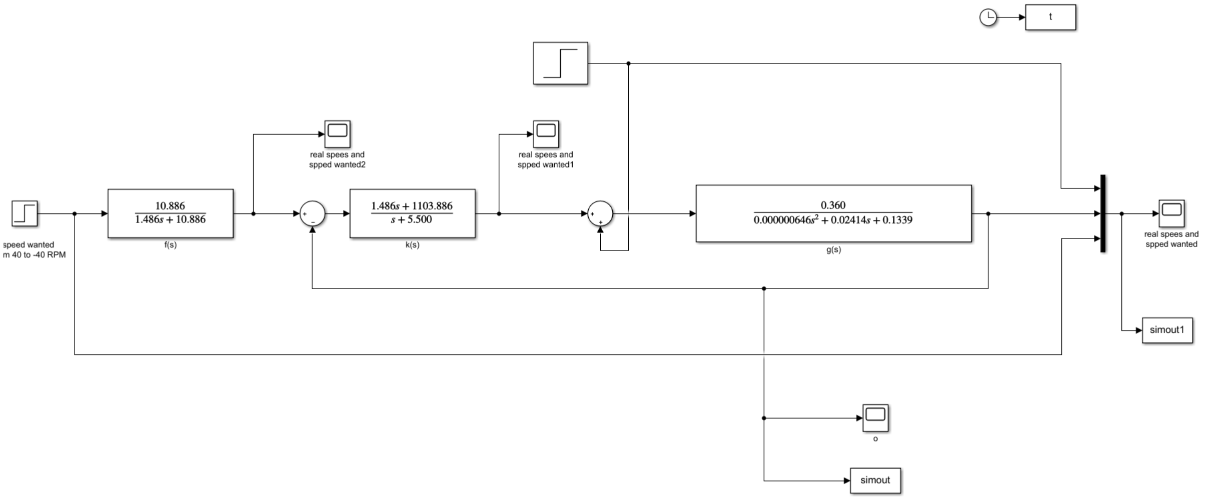


Figure 2.21: CDM simulation of closed loop system

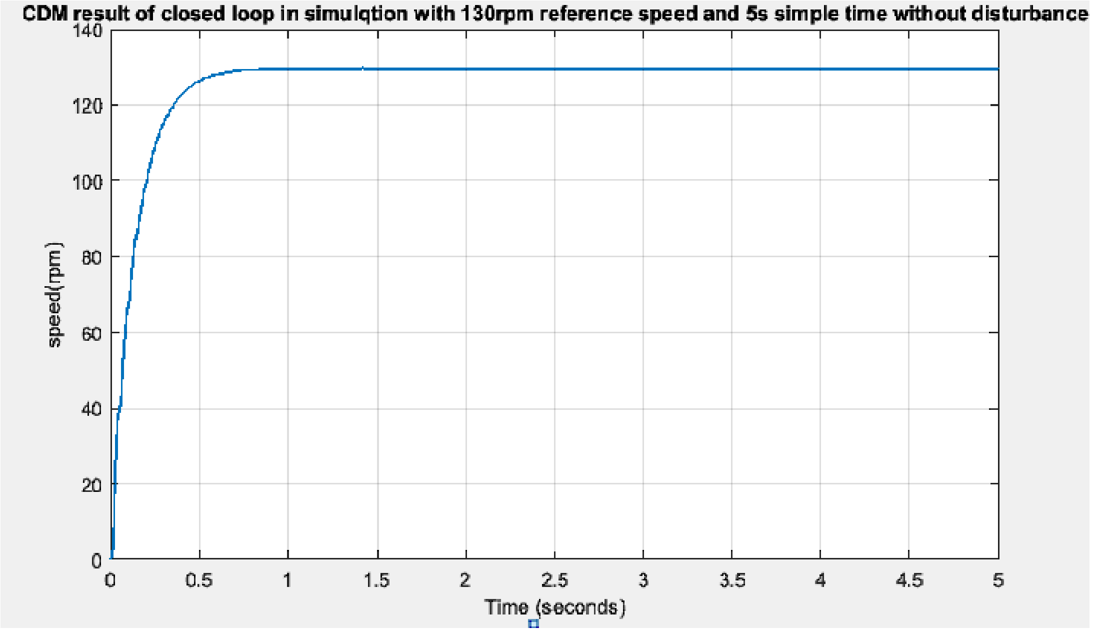


Figure 2.22: CDM simulation result of closed loop without disturbance

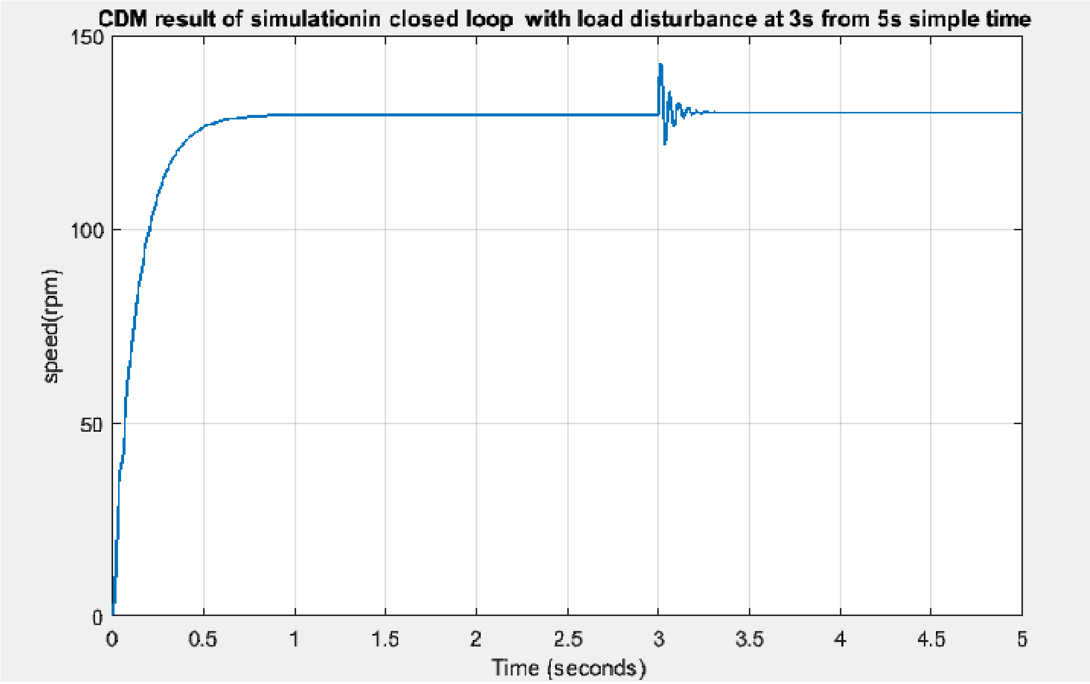


Figure 2.23: CDM simulation result of closed loop with step disturbance

2.0.28 comparison CDM (coefficient diagram method) with PID controller

As seen before in our result of PID controller and of CDM method and from our result the CDM gives fast response of dc motor in steady state (0.5s) then the PID controller(0.75-85s) moreover and step response more fast and robust then PID.

CDM (coefficient diagram method)controller the most considerable advantages of CDM can be design procedure is quiet easily understandable, systematic and useful. Therefore, the coefficients of the CDM controller polynomials can be determined more easily than those of the PID controller. This creates the possibility of an easy realization for a new designer to control any kind of system. There are explicit relations between the performance parameters specified before the design and the coefficients of the controller polynomials. For this reason, the designer can easily realize many control systems having different performance properties for a given control problem in a wide range of freedom. It is needed to develop different tuning methods for the processes with various properties in PID control. But it is sufficient to use a single design procedure in CDM technique. In CDM, the characteristic polynomial and the controller are designed simultaneously with the help of the coefficient diagram. The characteristic polynomial specifies stability and response. The structure of controller guarantees robustness. Thus a simplest controller, which satisfies the stability, response, and robustness requirements, can be designed with ease. The CDM has a wide application besides designing a satisfactory controller. When combined with LQ design, it gives the analytic method of selecting weights (Manabe, 1998b). The adaptive control is another field of application.

PID controller represent a low robustness against CDM method and PID controller are linear and symmetric so in nonlinear systems the performance of system varies if we compare against CDM method that non varies. More then the derivative noise in PID controller.

Chapter 3

Interpreting the realization of a DC Motor Control

Here in this chapter terminal voltage control method is employed. A control system is an interconnection of components forming a system configuration that will provide a desired system response. A controlled DC motor is developed allowing Arduino hardware which acts as the interface between the computer Matlab and the outside world via L298 H Bridge Driver. It primarily functions as a device that digitizes incoming digitals signals so that the Matlab can interpret them. The user interface was developed in an Arduino environment. The aim is to control the speed of the dc motor using the Low Cost data acquisition board i.e. the Arduino board interfaced with PID/CDM Controllers in Matlab.

3.0.1 Development of the synoptic diagram

The different modules that go into the realization of this system are:

- Power: Like all electronic devices, our system needs to be powered by a DC voltage source.
- Sensor (Encoder): which allows to read the real speed of motor. The central processing unit: it is the logical part and the brain of our system, its role is to

direct, manage, process and direct information towards the different blocks that make up our system thanks to the program that has been assigned to him.

- Entrance: speed command given to EMG30 DC motor.

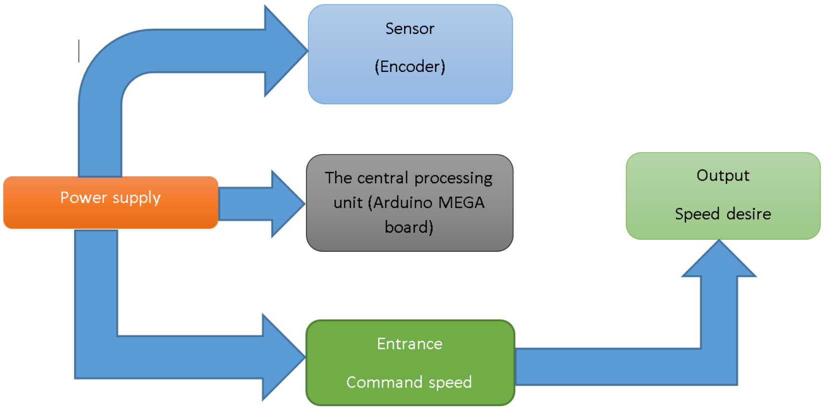


Figure 3.1: System block diagram

3.0.2 Hardware part

Required equipment

For the real experience of our system, we need the following materials:

- EMG30 DC motor
- L298N motor drive
- Arduino MEGA 2560
- Encoder
- Power supply
- wheel 100
- LAPTOP

3.0.3 EMG30 DC motor

The EMG30 (encoder, motor, gearbox 30:1) is a 12v motor fully equipped with encoders and a 30:1 reduction gearbox. It is ideal for small or medium robotic applications, providing cost effective drive and feedback for the user. It also includes a standard noise suppression capacitor across the motor windings [22].

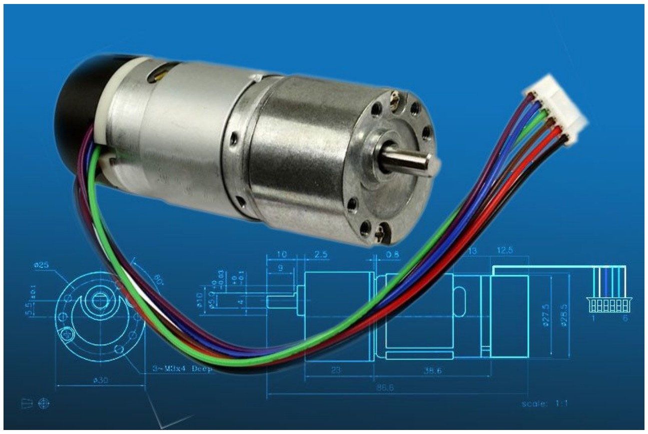


Figure 3.2: EMG30 motor [22]

3.0.4 Measurements

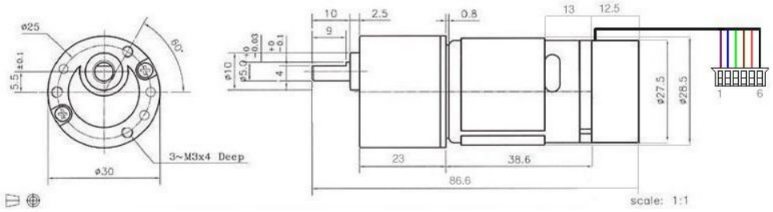


Figure 3.3: Measurements of EMG30 motor [22]

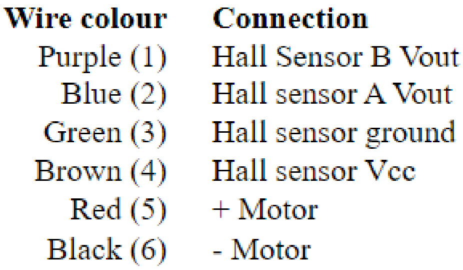
3.0.5 Specification

- Rated voltage 12v
- Rated torque 1.5kg/cm

- Rated speed 170rpm
- Rated current 530mA
- No load speed 216
- No load current 150mA
- Stall Current 2.5A
- Rated output 4.22W
- Encoder counts per output shaft turn 360

3.0.6 Connectors

The EMG30 is supplied with a 6 way JST connector (part no PHR-6) at the end of approx 90mm of cable as standard. The connections are:



Wire colour	Connection
Purple (1)	Hall Sensor B Vout
Blue (2)	Hall sensor A Vout
Green (3)	Hall sensor ground
Brown (4)	Hall sensor Vcc
Red (5)	+ Motor
Black (6)	- Motor

Figure 3.4: Connector

- Wire colours are from the actual cable.
- The hall sensors accept voltages between 3.5v and 20v.
- The outputs are open collector and require pull-ups to whatever signal level is required.
- On the LN298 they are powered from 12v and pulled up to 5v for the signals.

3.0.7 EMG30 mounting bracket

Providing easy mounting of the EMG30 to the robot, the bracket is made from a 2mm thick strong aluminum and finished in blue enamel.

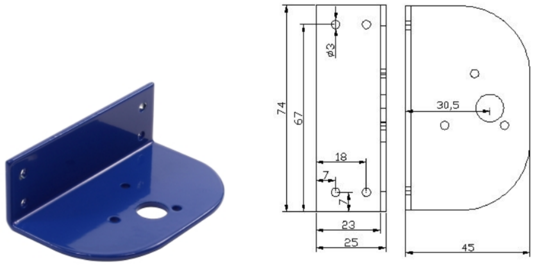


Figure 3.5: EMG30 mounting bracket

3.1 Motor encoder

An encoder is an electromechanical device that provides an electrical signal that is used for speed and/or position control. Encoders turn mechanical motion into an electrical signal that is used by the control system to monitor specific parameters of the application and make adjustments if necessary to maintain the machine operating as desired. The parameters monitored are determined by the type of application and can include speed, distance, RPM, position among others. Applications that utilize encoders or other sensors to control specific parameters are often referred to as closed-loop feedback or closed-loop control systems [6].

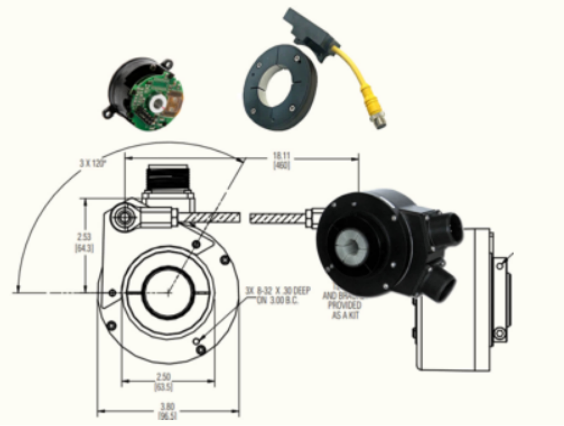


Figure 3.6: DC motor encoder [6]

3.1.1 What is a motor encoder

A motor encoder is a rotary encoder mounted to an electric motor that provides closed loop feedback signals by tracking the speed and/or position of a motor shaft. There are a wide variety of motor encoder configurations available such as incremental or absolute, optical or magnetic, shafted or hub/hollow shaft, among others. The type of motor encoder used is dependent upon a number of factors, particularly motor type, the application requiring closed-loop feedback, and the mounting configuration required [6].

3.1.2 DC motor encoders

DC motor encoders are used for speed control feedback in DC motors where an armature or rotor with wound wires rotates inside a magnetic field created by a stator. The DC motor encoder provides a mechanism to measure the speed of the rotor and provide closed loop feedback to the drive for precise speed control [6].

Encoder works by observing changes to the magnetic field created by a magnet attached the motor shaft. As the motor rotates, the encoder outputs will trigger periodically.

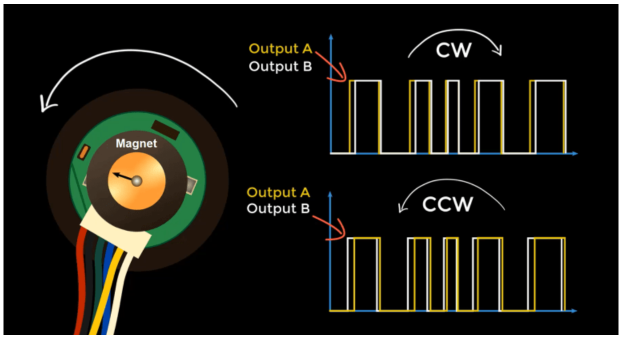


Figure 3.7: Principe of work Dc motor encoder [6]

3.2 Motor driver

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A [23].

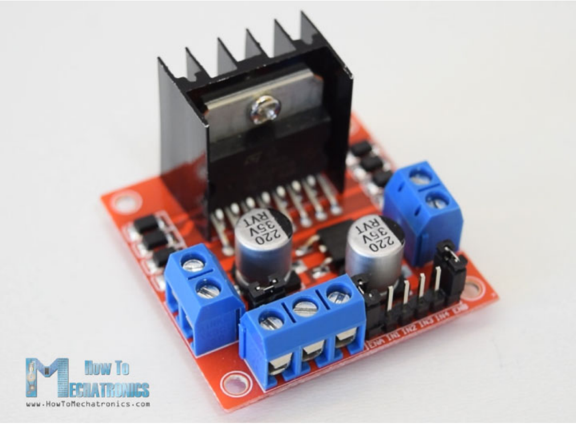


Figure 3.8: L298N-dual H-bridge motor driver [23]

Let's take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output [23].

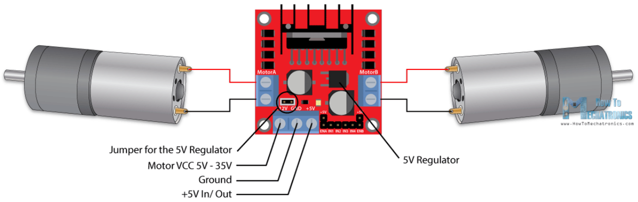


Figure 3.9: L298N-pin out [23]

This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly. We can note here that this IC makes a voltage drop of about 2V. So for example, if we use a 12V power supply, the voltage at motors terminals will be about 10V, which means that we won't be able to get the maximum speed out of our 12V DC motor [23].

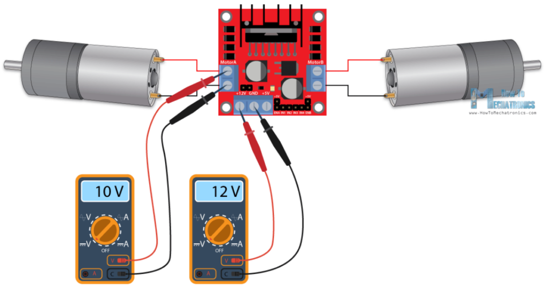


Figure 3.10: L298N-voltage [23]

3.2.1 L298N block diagram

The Input 1 and Input 2 pins are used for controlling the rotation direction of the motor A, and the inputs 3 and 4 for the motor B. Using these pins we actually control the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B [23].

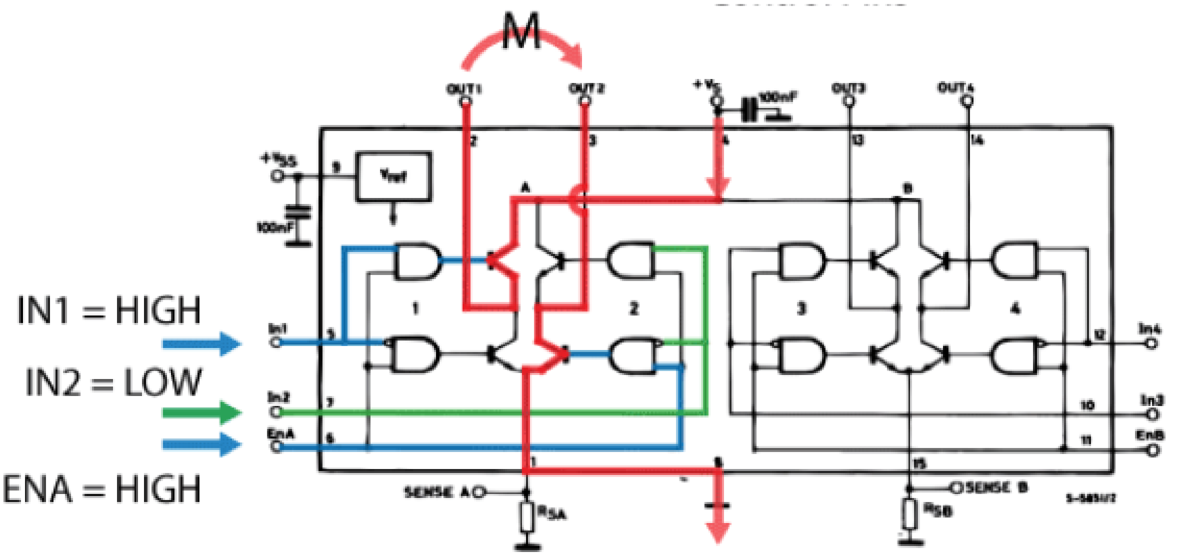


Figure 3.11: L298N block diagram [23]

3.3 Arduino controller

3.3.1 Arduino mega 2560

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller, simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila [24].

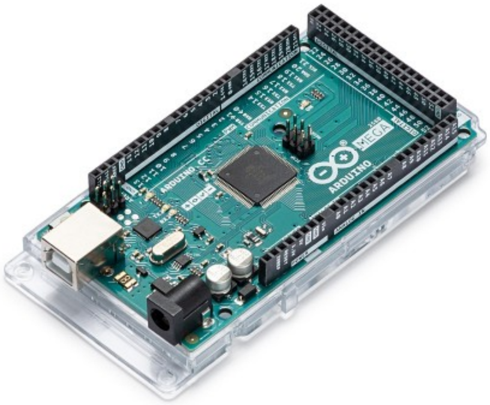


Figure 3.12: Arduino Mega2560 [24]

3.3.2 Arduino Mega 2560 pin diagram

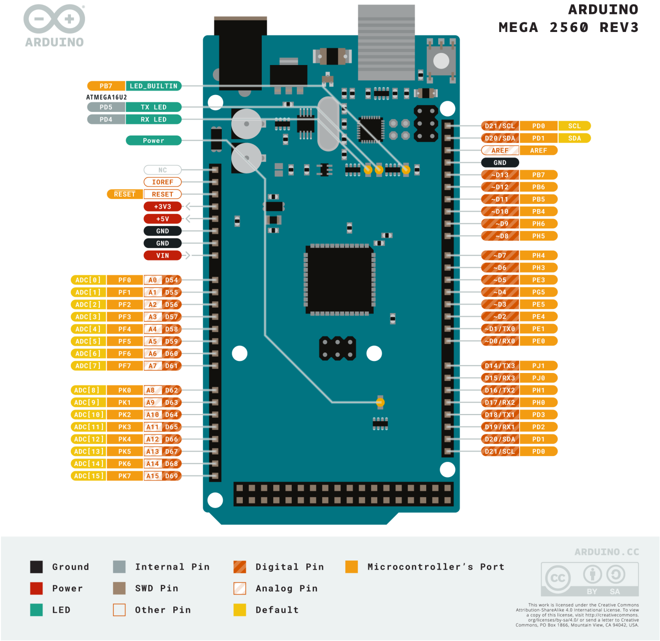


Figure 3.13: Arduino mega 2560 pin diagram [24]

3.4 Power supplies

To powered the system we use power supply of 12v connected to motor drive L298N and from motor drive to actuator (EMG30 motor) 12V power supplies (or 12VDC power supplies) are one of the most common power supplies in use today. In general, a 12VDC output is obtained from a 100v or 220v input using a combination of transformers, diodes and transistors. 12V power supplies can be of two types: 12V regulated power supplies, and 12V unregulated power supplies. 12V regulated power supplies come in three styles, switching regulated AC to DC, Linear regulated AC to DC, and Switching regulated DC to DC.

Figure 3.14: Power Supply 12v

3.5 Mechanical parts

3.5.1 Wheel 100

A 100mm diameter wheel with 5mm diameter hub for easy attachment to the EMG30, the wheel has a 26mm wide rubber tread, will applied in load.

Figure 3.15: wheel 100

3.5.2 Softwere part

3.5.3 The Matlab / Simulink environment

It is a mathematical calculation software for engineers and scientists created by Mathworks. MATLAB is a programming environment for algorithm development, data analysis, visualization, and numerical computation. Using MATLAB, the Solving complex computational problems is done faster than with traditional programming languages, such as C, C ++, and Fortran. SIMULINK is an environment for multi-domain simulation. It provides an interactive graphical environment and a set of block libraries that allow you to design, simulate, implement, and test a variety of systems, such as communications, control, signal processing, visual processing, and image processing systems [25].

3.6 Matlab interface with Arduino

Arduino programming is supposed to be fun but can become frustrating and time consuming for tasks such as plotting sensor data or incorporating advanced math, signal processing, or controls routines into your projects. MATLAB and Simulink address several challenges with traditional Arduino programming. The products support two primary workflows:

- Read, write, and analyze data from Arduino sensors
- Develop algorithms that run standalone on the Arduino device

Read, write, and analyze data from Arduino sensors MATLAB support package for Arduino lets you write MATLAB programs that read and write data to your Arduino and connected devices such as Adafruit motor shield, I2C, and SPI devices. Because MATLAB is a high level interpreted language, programming with it is easier than with C/C++ and other compiled languages, and you can see results from I/O instructions immediately – no compiling. MATLAB includes thousands of built-in math, engineering, and plotting functions that you can use to quickly analyze and visualize data collected from your Arduino [26].

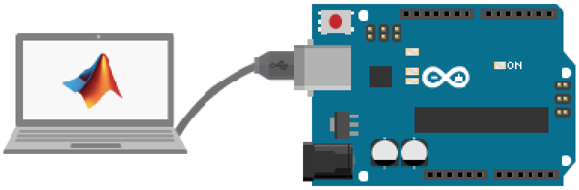


Figure 3.16: Read, write, and analyze data from Arduino [26]

With MATLAB support package for Arduino, the Arduino is connected to a computer running MATLAB. Processing is done on the computer with MATLAB. Benefits of using MATLAB for Arduino programming [26].

- Read and write sensor data interactively without waiting for your code to compile
- Analyze your sensor data using thousands of pre-built functions for signal processing, machine learning, mathematical modeling, and more
- Quickly visualize your data using MATLAB's vast array of plot types

3.6.1 Pre-Loading Of the program in the Arduino board

- Download the ArduinoIO package
- Unzip to the root of the hard disk, example E :arduinoio
- Open the unzipped folder.
- Go to :ArduinoIO: pde : adioes
- Load the file adiosrv.pde to the Arduino software.
- Upload

Note: adioes is the abbreviation for : Analog and Digital Input and Output Server for MATLAB. The Arduino MEGA board is now configured to be used as an input/output interface board [26].

```
/* Analog and Digital Input and Output Server for MATLAB     */
/* Giampiero Campa, Copyright 2012 The MathWorks, Inc        */

/* This file is meant to be used with the MATLAB arduino IO
   package, however, it can be used from the IDE environment
   (or any other serial terminal) by typing commands like:

   0e0 : assigns digital pin #4 (e) as input
   0f1 : assigns digital pin #5 (f) as output
   0n1 : assigns digital pin #13 (n) as output

   1c  : reads digital pin #2 (c)
   1e  : reads digital pin #4 (e)
   2n0 : sets digital pin #13 (n) low
   2n1 : sets digital pin #13 (n) high
   2f1 : sets digital pin #5 (f) high
   2f0 : sets digital pin #5 (f) low
   4j2 : sets digital pin #9 (j) to  50=ascii(2) over 255
   4jz : sets digital pin #9 (j) to 122=ascii(z) over 255
   3a  : reads analog pin #0 (a)
   3f  : reads analog pin #5 (f)

   5j    : reads status (attached/detached) of servo on pin #9
   5k    : reads status (attached/detached) of servo on pin #10
   6j1   : attaches servo on pin #9
   8jz   : moves servo on pin #9 of 122 degrees (122=ascii(z))
   7j    : reads angle of servo on pin #9
   6j0   : detaches servo on pin #9

   E0cd  : attaches encoder #0 (0) on pins 2 (c) and 3 (d)
   E1st  : attaches encoder #1 on pins 18 (s) and 19 (t)
   E2vu  : attaches encoder #2 on pins 21 (v) and 20 (u)
   G0    : gets 0 position of encoder #0
```

Figure 3.17: Ploading Tthe Adioes Program to the Arduino Interface

3.6.2 Develop algorithms that run standalone on the Arduino

Develop algorithms that run standalone on the Arduino Simulink support package for Arduino lets you develop algorithms in Simulink, a block-diagram environment for modeling dynamic systems and developing algorithms, and run them standalone on your Arduino. The support package extends Simulink with blocks for configuring Arduino sensors and reading and writing data from them. After creating your Simulink model, you can simulate it, tune algorithm parameters until you get it just right, and download the completed algorithm for standalone execution on the device. With the MATLAB Function block, you can incorporate MATLAB code into your Simulink model [26].

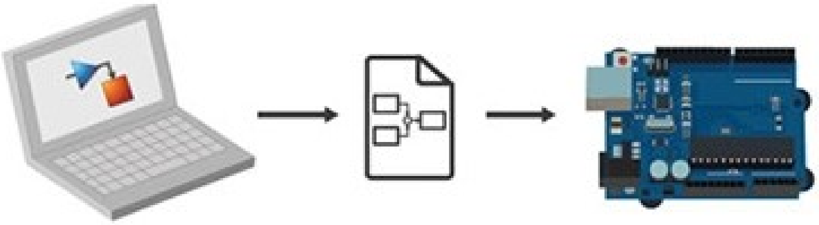


Figure 3.18: Evelop Algorithms That Run Standalone on The Arduino [26]

With Simulink support package for Arduino, you develop the algorithm in Simulink and deploy to the Arduino using automatic code generation. Processing is then done on the Arduino. Benefits of using Simulink for Arduino programming [26].

- Develop and simulate your algorithms in Simulink and use automatic code generation to run them on the device
- Incorporate signal processing, control design, state logic, and other advanced math and engineering routines in your hardware projects
- Interactively tune and optimize parameters as your algorithm runs on the device
- Easily modify algorithms to run on other low-cost and commercial hardware platforms [26].

3.6.3 Connection of the motor drive with the Arduino and with EMG30 motor

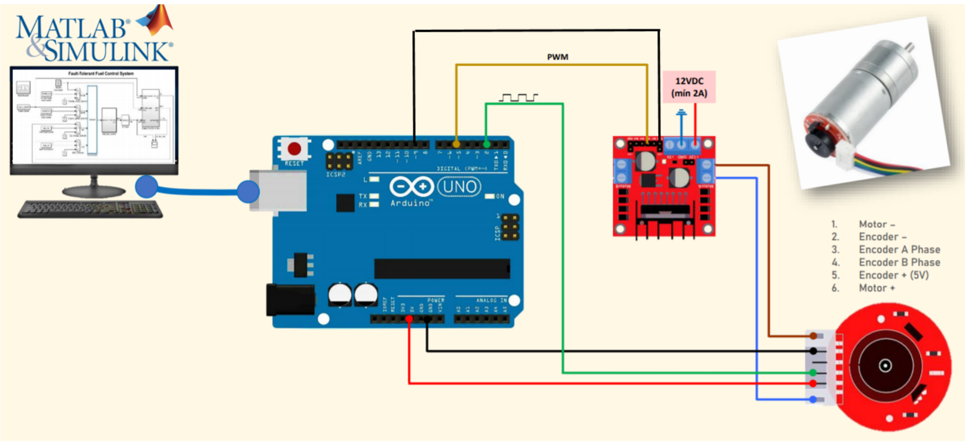


Figure 3.19: Connection Diagram

-The connection is made as in the table bellow :

pin	**connection**
motor+/-	motor drive +/-
pin encoder A of the motor	Pin 2 of the Arduino
pin encoder B of the motor	Pin 3 of the Arduino
GND of the motor (common neutral)	GND of the Arduino
VCC encoder of the motor	Pin +5 of the Arduino
ENA of of motor drive L298N	pin number 5 of the Arduino
IN1 of of motor drive L298N	pin number 13 of the Arduino
+12/- of the motor drive	with power supply

Table 3.1: Connection of arduino with DC motor with motor drive

3.6.4 Diagram of the system

Before going on with the programming, we have to make a diagram that explains the different sequences of our of the different sequences of our system.

The program of simulink give command in arduino to start motor after compare the desire speed with the real speed if the real speed is similar to the speed desire let the motor run, if the speed is lees or more then the speed desire matlab give again commanded to arduino to increase of decrease the speed to be equal to the speed desire.

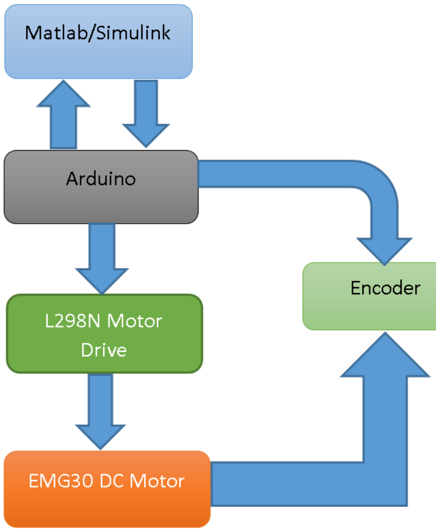


Figure 3.20: Diagram of the System

‘

3.6.5 Real Experience

Simulink model of velocity measurement

To control the speed of DC motor, it needs to construct the speed closed loop control, at the same time, to join the PID controller. PID controller or CDM involves P, I, and D parameters’ setting and f(s) forward controller and k(s) controller of CDM method.

Firstly, the output is measured when the system loaded a given input signal. To establish the system model as shown in Fig, speed command module generates a step signal with a range of 0-255. Adruino mega 2560 digital pin 2 collect encoder signal and then calculate the motor speed. The encoder is the AB phase incremental encoder, each circle outputs 360 pulses.

Where t is the time between adjacent pulses, (v) is the speed of the motor. The Simulink model block is shown in Fig bellow :

Note that the sample time of the pulse signal detection module can’t be too large, because it will lose pulse. And it could not be too small, because it will cause the

Arduino overload.

here we give motor speed reference 130rpm and time of 2s of simulation and 0.02 time for step time

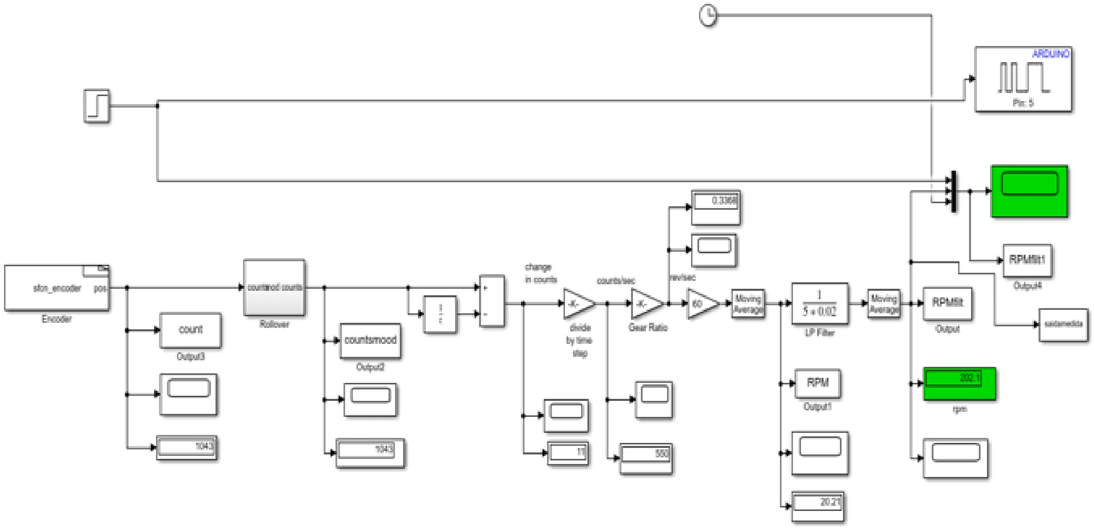


Figure 3.21: Simulink Block of Velocity Measurement without control

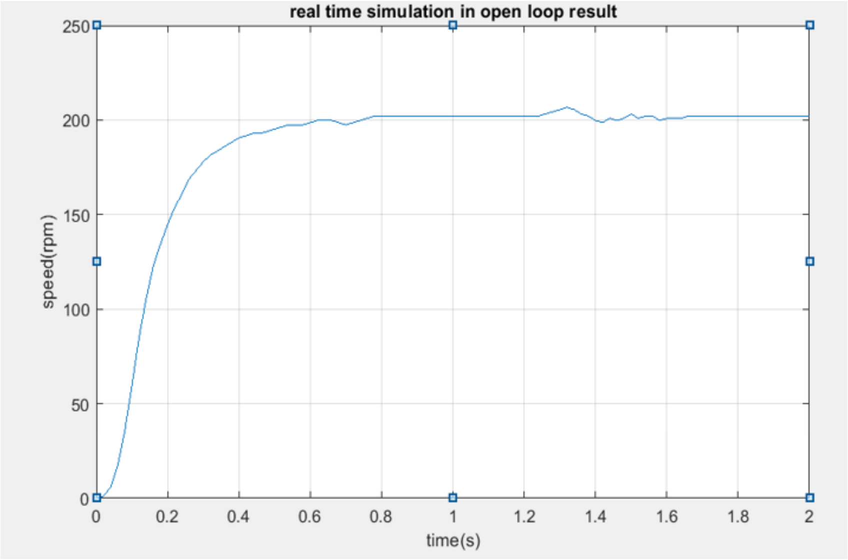


Figure 3.22: Result of real time motor in open loop test in 130rpm qnd 2s simple time

Model simulation and real time motor comparison

Now we plot the model mathematics and real time result of motor in open loop together to validate the motor model with open loop real time

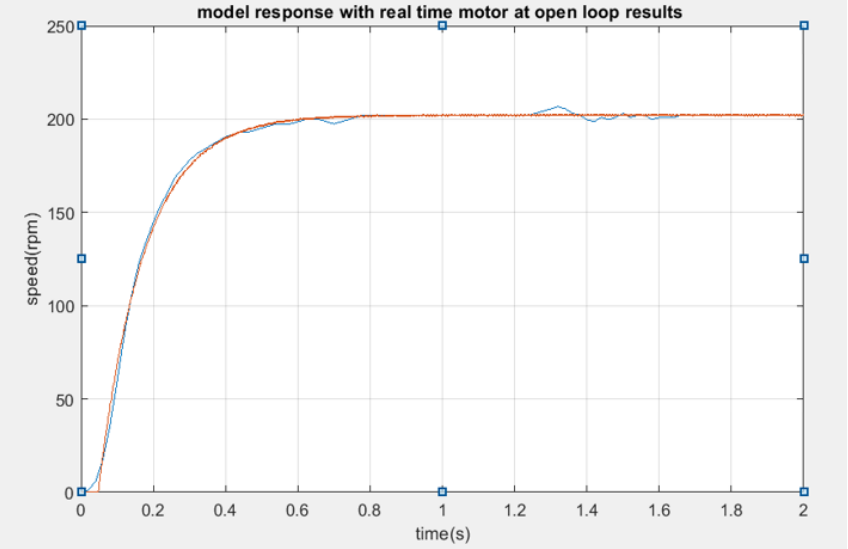


Figure 3.23: Result of model with motor real time in open loop

3.6.6 Global simulink model of system

Computer and Arduino mega 2560 board are connected by USB cable. Serial communication port is set to make sure the connection is fine. The model in external mode is performed as shown in Fig. We can observe real-time input and output waveform of the system by open scope.

In the real experiment, the PID parameters are used to carry out firstly. PID parameters are set as Kp=0.0035, Ki=0.14, Kd=0.00007 In the experiment and the parameters of the forward controller f(s) and k(s) controller of CDM, the speed is stable at 130 r/min, of the system are shown in Figures bellow result of the real experiment firstly with PID controller is shown in Fig bellow.

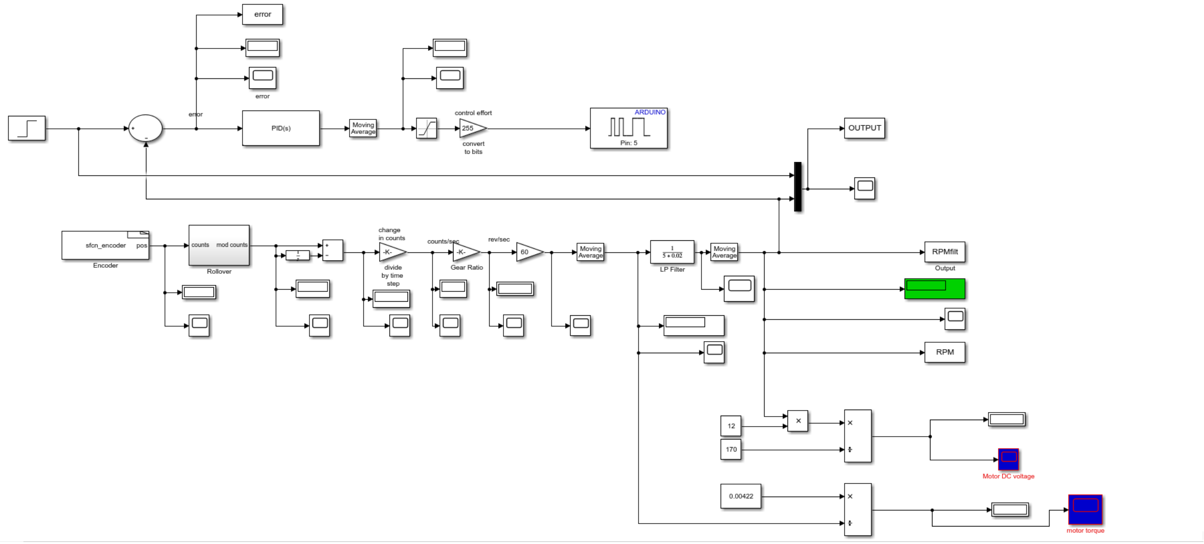


Figure 3.24: PID control system based on Arduino

Figure 3.25: Real experience

3.6.7 Result and discussion

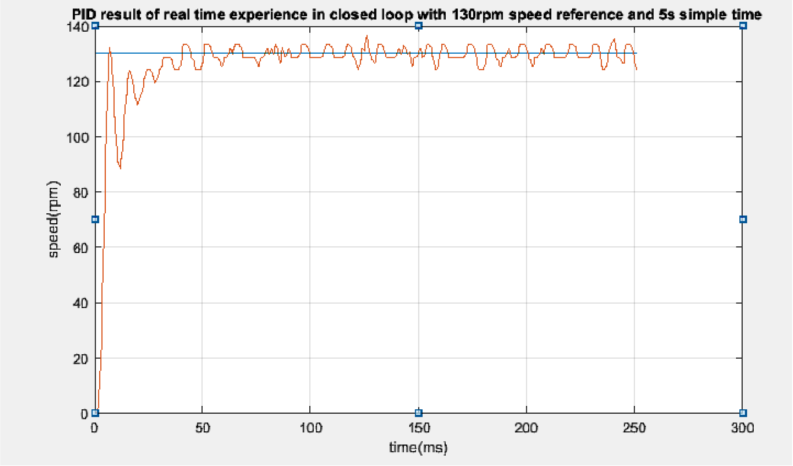


Figure 3.26: PID result of real time experience in closed loop

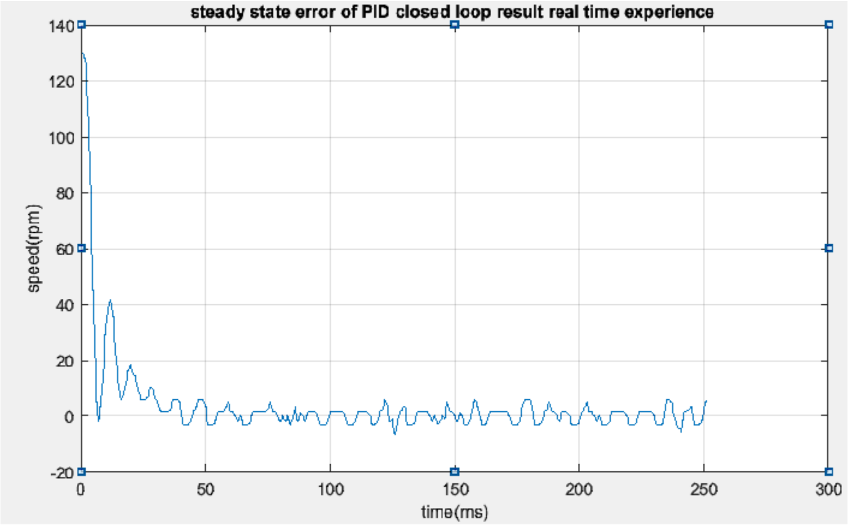


Figure 3.27: steady state error of PID control of real time system in closed loop

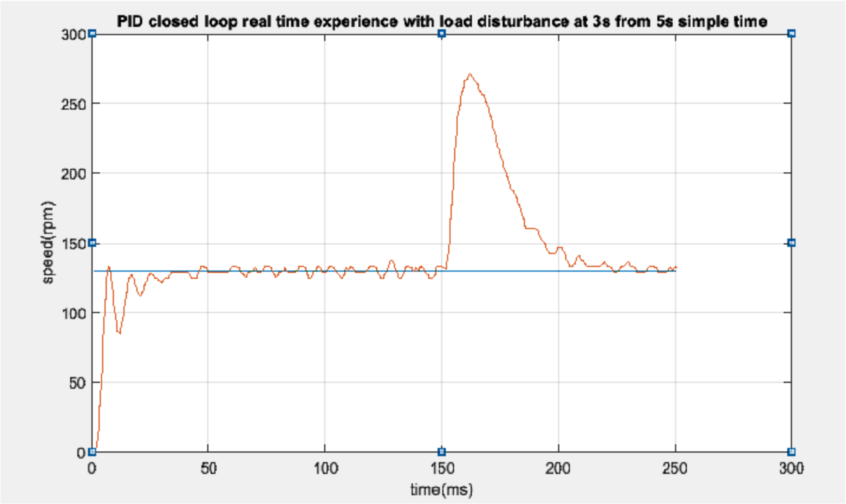


Figure 3.28: PID closed loop result in real time with load disturbance

Figure 3.29: steady state error of PID control in real time at closed loop result with load disturbance

Figure up shows the speed control of DC motor with PID controller. Experimental results are shown in Fig 3.26 and Fig. 3.27 and fig 3.28 and 3.29 for step response for speed change from 0 rpm to 130 rpm . These figures clarify the soft start of the motor and the operation of the current limit as well as the satisfactory transient response. The speed of a DC motor has been successfully controlled by using L298 H bridge driver and PID type Speed controller based on closed loop system model. Initially a simplified closed loop model for speed control of DC motor is considered and requirement of speed controller is studied. Then a generalized modeling of DC motor is done. After that a complete layout of DC drive system is obtained. Then designing of speed controller is done. The optimization of speed control loop is achieved through Ziegler-Nichols method. A DC motor specification is taken and corresponding parameters are found out from derived design approach. The experimental results under varying reference speed, The test shows good results under all conditions employed during simulation.

3.6.8 CDM real time experience closed loop results

Again and as we did before with PID, computer and Arduino mega 2560 board are connected by USB cable. Serial communication port is set to make sure the connection is fine. The model in external mode is performed as shown in Fig. We can observe real-time input and output waveform of the system by open scope.

In the real experiment, in the experiment and the parameters of the forward controller f(s) and k(s) controller of CDM, the speed is stable at 130 r/min, of the system are shown in Figures bellow result of the real experiment secondly with CDM controller is shown in Fig bellow.

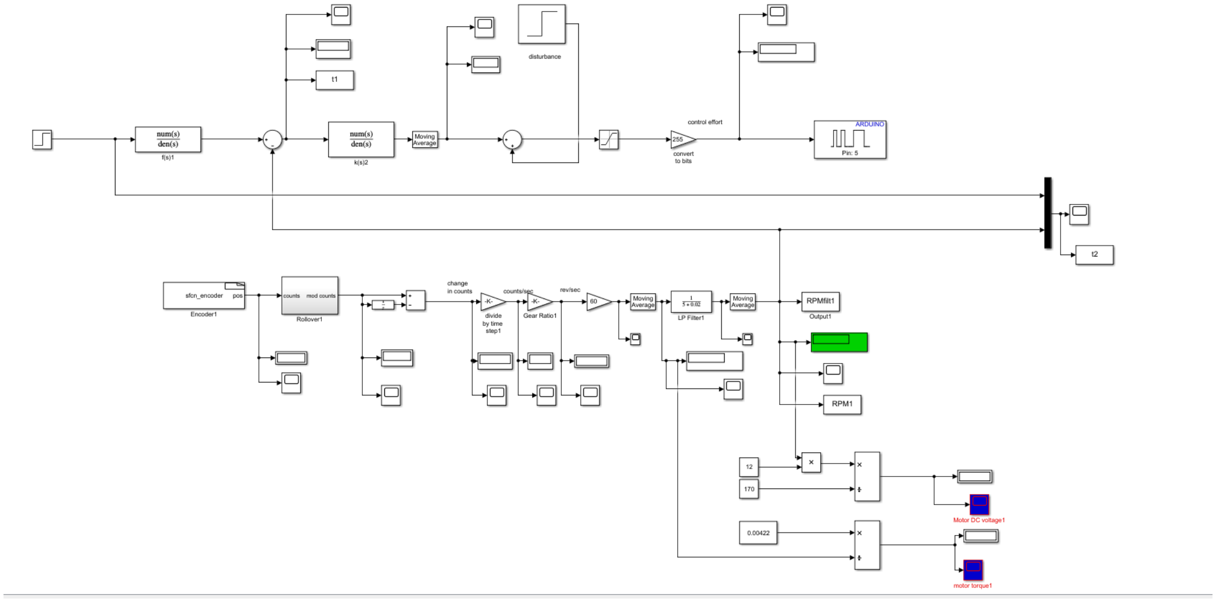


Figure 3.30: CDM real time simulation closed loop

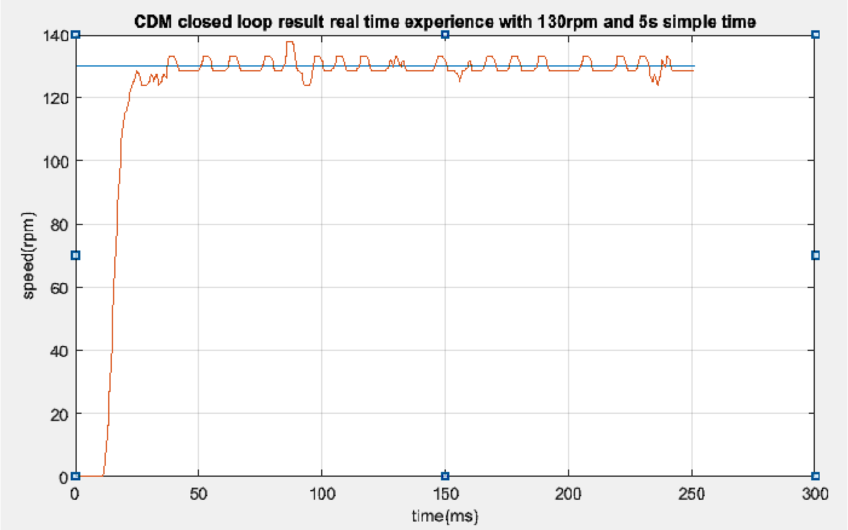


Figure 3.31: CDM closed loop result real time experience

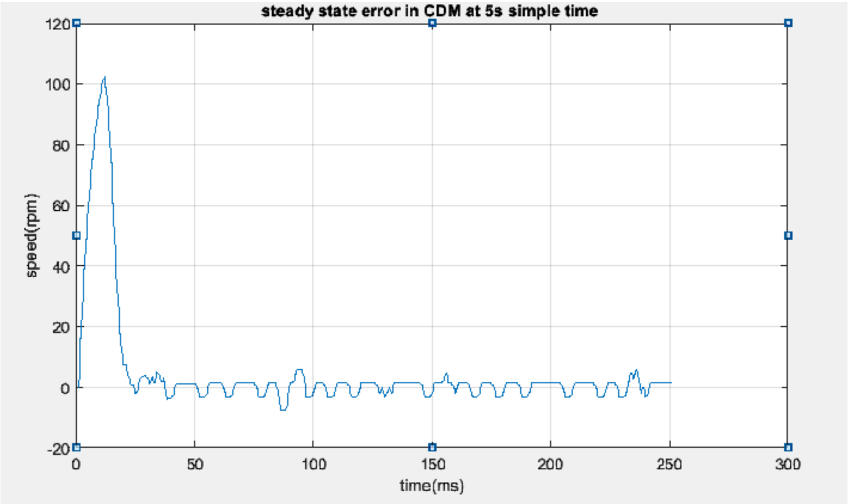


Figure 3.32: CDM steady state error without disturbance

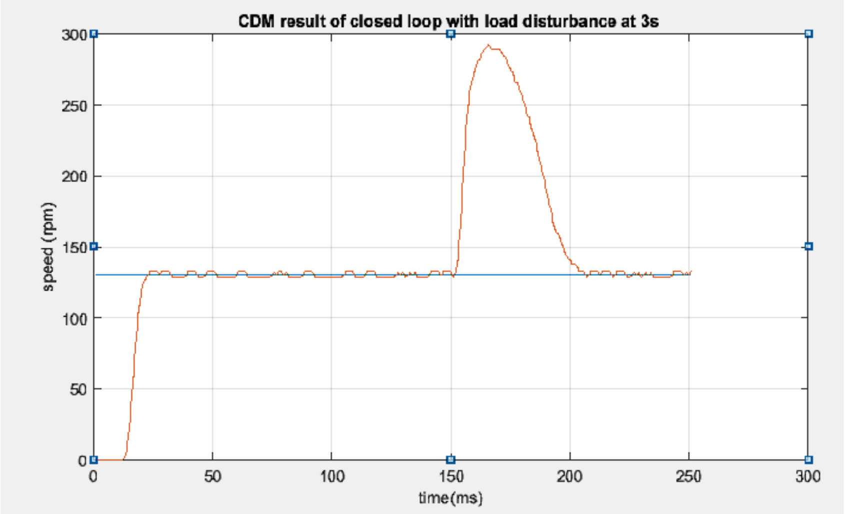


Figure 3.33: CDM result of closed loop in real time experience with disturbance

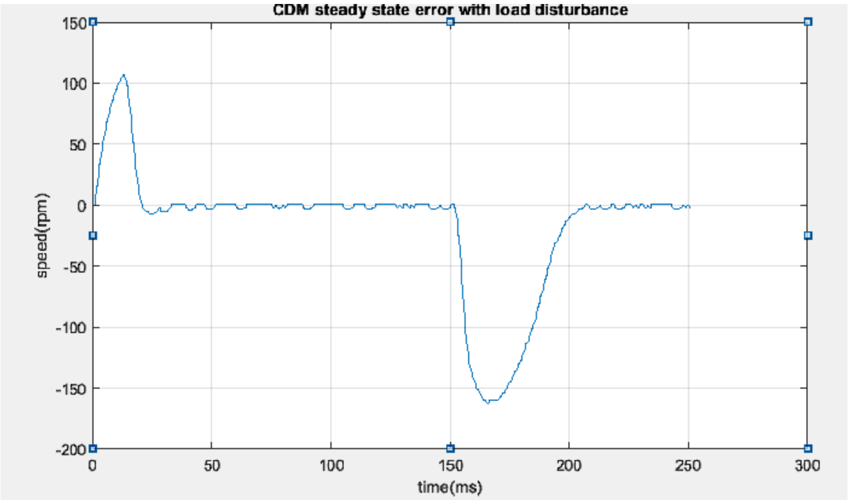


Figure 3.34: CDM steady state result of closed loop in real time experience with disturbance

As before the figure up shows the speed control of DC motor with CDM controller. Experimental results are shown in Fig 3.31 and Fig. 3.32 and fig 3.33 and 3.34 for step response for speed change from 0 rpm to 130 rpm . These figures clarify the soft start of the motor and the operation of the current limit as well as the satisfactory transient response. The speed of a DC motor has been successfully controlled by using L298 H bridge driver and CDM method type Speed controller based on closed loop system model. Initially a simplified closed loop model for speed control of DC motor is considered and requirement of speed controller is studied. Then a generalized modeling of DC motor is done. After that a complete layout of DC drive system is obtained. Then designing of speed controller is done. The optimization of speed control loop is achieved through the forward controller and controller of the CDM method. A DC motor specification is taken and corresponding parameters are found out from derived design approach. The experimental results under varying reference speed, The test shows good results under all conditions employed during simulation.

3.6.9 Advantages of PID and CDM method

In the table bellow we are presenting advantage's of PID and CDM controllers and their features in order to get more knowledge of these two methods of digital controllers.

CDM	**PID**
-Design procedure is quiet easily understandable, systematic and useful.	-Feasibility and easy to be implemented.
-The coefficients of the CDM controller polynomials can be determined more easily.	-PID controller generally has to balance all three-gains impact to the whole system and may compromise the transient response.
-The designer can easily realize many control systems having different performance properties for a given control problem in a wide range of freedom.	-The PID gains can be designed based upon the system parameters if they can be achieved or estimated precisely.
-The characteristic polynomial and the controller are designed simultaneously with the help of the coefficient diagram.	-Easy to implement in hardware (through filters) and also easy to implement though microcontrollers, PLC.
-The structure of controller guarantees robustness. Thus a simplest controller, which satisfies the stability, response, and robustness.	-Not much expertise is needed for tuning, hence, few middle-skilled technicians may easily carry out the tasks.

Table 3.2: Advantages of PID and CDM method

Conclusion and Future Work

Electric motors are applied to provide mechanical work in industries. The DC motors is considered to be a basic electric machine. The DC motors are one of the electrical drives that are rapidly gaining popularity, due to their high efficiency, good dynamic response and low maintenance. The DC motors and drives have grown significantly in recent years in the appliance industry and the automotive industry. DC drives are very preferable for compact, low maintenance, and high reliability system . In this work, an overview of DC motors, including the equations that represent torque, rotor speed, field and armature currents and voltages is developed.Identification and PID Controller of a 2st Order DC Motor Model and CDM method discussed. The Results of the testing about the Speed Control of DC Motor performance are analyzed. A PID controller has been employed for speed control of DC motor by Matlab interfaced with Arduino and we applicate the CDM method for control velocity also. The speed controller has been designed successfully for closed loop operation of the DC motor and the motor runs nearly to the reference speed. The experimental results under varying reference speed. The test shows good results under all conditions employed during simulation.

To continuous this work and for a future work we should control speed of dc motor using Artificial Intelligence techniques of control such as (LQR and LGG optimal controller, Model predictive control, Neuro-Fuzzy controller..) will also allow us to control the DC motor and give best performance and high precision for each method.

Bibliography

[1] B. R. Arnett. (consulted on 10 August 2020 17:13:07 BST). The History of DC Motors, [Online]. Available: `https://www.parvalux.com/news/history-of-dc-motors/`.

[2] K. Iqbal. (consulted on the Mar 5, 2021). ‘PI, PD, and PID Controllers ’, [Online]. Available: `https://chem.libretexts.org/@go/page/28514`.

[3] E. Csanyi. (consulted on January, 7th 2015). 4 Types of DC Motors and Their Characteristics, [Online]. Available: `https://electrical-engineering-portal.com/4-types-of-dc-motors-and-their-characteristics`.

[4] A. M. T. I. Al-Naib. (consulted on May 2019). DC Motors, [Online]. Available: `https://www.researchgate.net/publication/332835517_DC_Motors`.

[5] S. Mathew. (September 23, 2014). DC Motor, how it works ?, [Online]. Available: `https://lesics.com/dc-motor-working.html`.

[6] dynapar. (2021). Motor Encoder?, [Online]. Available: `https://www.dynapar.com/technology/encoder_basics/motor_encoders/`.

[7] Ikimi.O. (February 04, 2020). Understanding DC Motor Operation Modes and Speed Regulation Techniques, [Online]. Available: `https://www.allaboutcircuits.com/news/a-basic-guide-to-dc-motors-types-of-motors-operation-modes-and-their-applications/`.

[8] ASPINA. (11 Feb. 2021). How are DC motors controlled? - Speed control of DC motors, [Online]. Available: `https://www.aspina-group.com/en/learning-zone/columns/what-is/011/`.

[9] B. M. C. O. M. Team. (consulted on 05/14/2019). Benefits of DC Motors and Why They're Great for Robotics, [Online]. Available: `https://www.motioncontrolonline.org/blog-article.cfm/Benefits-of-DC-Motors-and-Why-They-re-Great-for-Robotics/82#:~:text=Many%20say%20high%20torque%20is,shaft%20power%20varies%20with%20speed.`.

[10] J. N. Pires, *Industrial robots programming: building applications for the factories of the future*. Springer Science & Business Media, 2007.

[11] enquiries@precisionmicrodrives.com. (consulted on Mon. – Fri. 9:00 – 18:00 (BST /UTC)). Medical Applications For DC Gearmotors, [Online]. Available: `https://www.precisionmicrodrives.com/gear-motors/medical-applications-dc-gearmotors/`.

[12] P. Dixit. (consulted on Monday, April 20, 2020). DC Motor Applications, [Online]. Available: `https://robu.in/dc-motor-applications/`.

[13] edisontechcenter. (2014). Drone Power, [Online]. Available: `https://edisontechcenter.org/Drones.html`.

[14] R. KUMAR, "SPEED CONTROL OF DC MOTOR USING PULSE WIDTH MODULATION", PhD thesis, University of Technology, 2019.

[15] J. Gonçalves, J. Lima, P. J. Costa, and A. P. Moreira, "Modeling and simulation of the emg30 geared motor with encoder resorting to simtwo: the official robot@ factory simulator", in *Advances in Sustainable and Competitive Manufacturing Systems*, Springer, 2013, pp. 307–314.

[16] C. D. R. Dionısio, "Ground Mobile Vehicle Velocity Control using Encoders and Optical Flow Sensor Fusion", *Técnico Lisboa*, 2016.

[17] B. W. Bequette, *Process control: modeling, design, and simulation.* Prentice Hall Professional, 2003.

[18] P. Meshram and R. G. Kanojiya, “Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor”, in *IEEE-international conference on advances in engineering, science and management (ICAESM-2012)*, IEEE, 2012, pp. 117–122.

[19] M. K. K.Krishnamurthy. (May 22, 2012). Design and Implementation of CDM-PI Control Strategy in a pH Neutralization System, [Online]. Available: `https://scialert.net/fulltext/?doi=ajsr.2012.78.92`.

[20] O. Ocal, A. Bir, and B. Tibken, “Digital design of coefficient diagram method”, in *2009 American control conference*, IEEE, 2009, pp. 2849–2854.

[21] S. Manabe, “Importance of coefficient diagram in polynomial method”, in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, IEEE, vol. 4, 2003, pp. 3489–3494.

[22] robot-electronics. (-). EMG30, mounting bracket and wheel specification, [Online]. Available: `https://www.robot-electronics.co.uk/htm/emg30.htm`.

[23] Dejan. (consulted on 2017). Arduino DC Motor Control Tutorial – L298N | PWM H-Bridge, [Online]. Available: `https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/`.

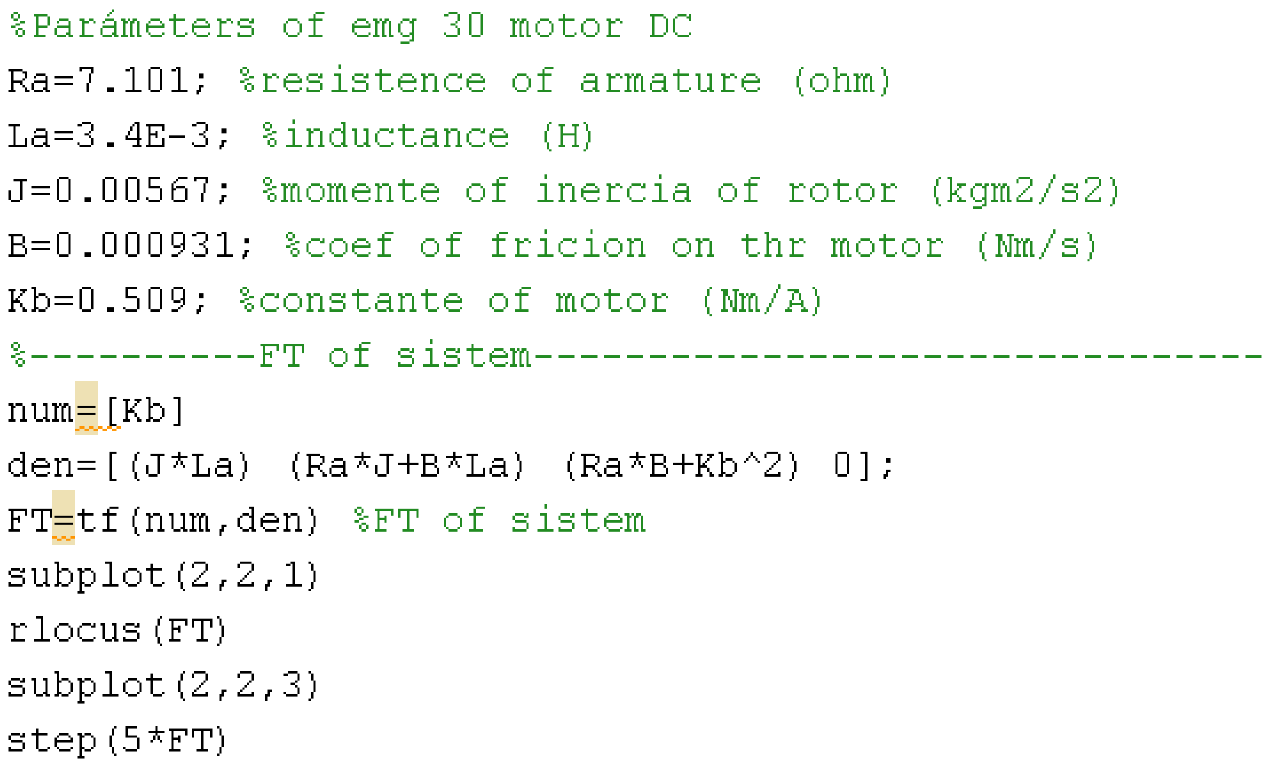
[24] arduino website. (2021). Arduino MEGA 2560 Genuino MEGA 2560, [Online]. Available: `https://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardMega2560`.

[25] C. N. C. Amine. (Hammamet 3/4 Mai 2014). Commande d’un système thermique à l’aide de la carte Arduino UNO, [Online]. Available: `https://chellynizarblog.files.wordpress.com/2015/10/report1.pdf`.

[26] , and, “Contrôle de vitesse du moteur à courant continu basé sur MATLAB Interface avec Arduino et L298N H-Bridge”, 2017.

Appendix A

Simulink/arduino and Codes



```
%Parámeters of emg 30 motor DC
Ra=7.101; %resistence of armature (ohm)
La=3.4E-3; %inductance (H)
J=0.00567; %momente of inercia of rotor (kgm2/s2)
B=0.000931; %coef of fricion on thr motor (Nm/s)
Kb=0.509; %constante of motor (Nm/A)
%----------FT of sistem-------------------------------
num=[Kb]
den=[(J*La) (Ra*J+B*La) (Ra*B+Kb^2) 0];
FT=tf(num,den) %FT of sistem
subplot(2,2,1)
rlocus(FT)
subplot(2,2,3)
step(5*FT)
```

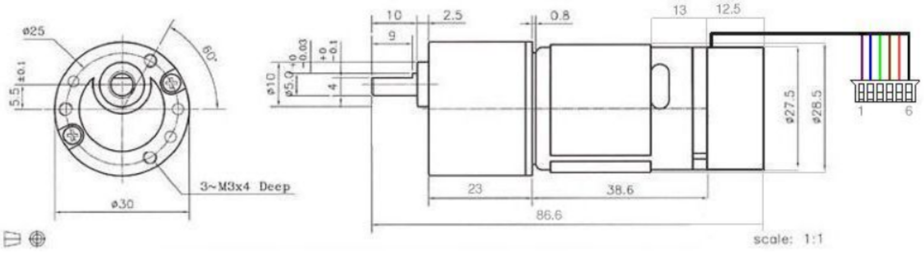
Appendix B

Electronics Components

EMG30, mounting bracket and wheel specification

The EMG30 (encoder, motor, gearbox 30:1) is a 12v motor fully equipped with encoders and a 30:1 reduction gearbox. It is ideal for small or medium robotic applications, providing cost includes a standard noise suppression capacitor across the motor windings.

Measurements



Connector

The EMG30 is supplied with a 6 way JST connector (part no PHR-6) at the end of approx 90
The connections are:

Wire colour	Connection
Purple (1)	Hall Sensor B Vout
Blue (2)	Hall sensor A Vout
Green (3)	Hall sensor ground
Brown (4)	Hall sensor Vcc
Red (5)	+ Motor
Black (6)	- Motor

Wire colours are from the actual cable.
The hall sensors accept voltages between 3.5v and 20v.
The outputs are open collector and require pull-ups to whatever signal level is required.
On the MD25 they are powered from 12v and pulled up to 5v for the signals.

specification

Rated voltage	12v
Rated torque	1.5kg/cm
Rated speed	170rpm
Rated current	530mA
No load speed	216
No load current	150mA
Stall Current	2.5A
Rated output	4.22W
Encoder counts per output shaft turn	360

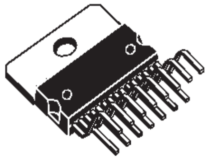
Measured Shaft Speed when used off-load with MD23 and 12v supply.

Minimum Speed	1.5rpm
Maximum Speed	200rpm

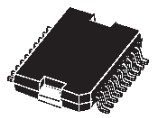
L298

DUAL FULL-BRIDGE DRIVER

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



Multiwatt15



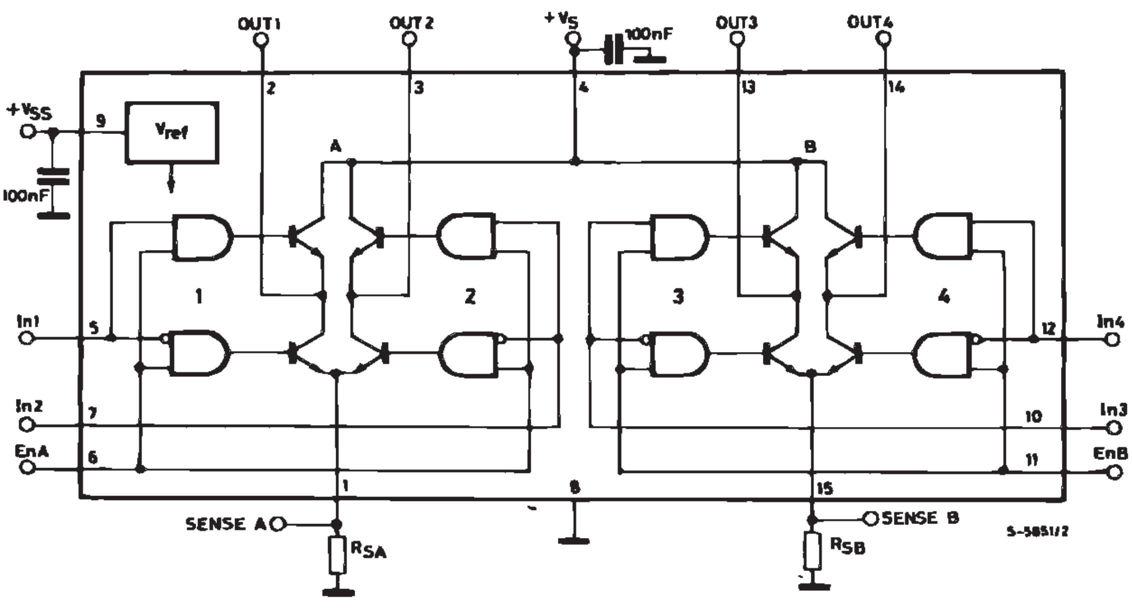
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

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

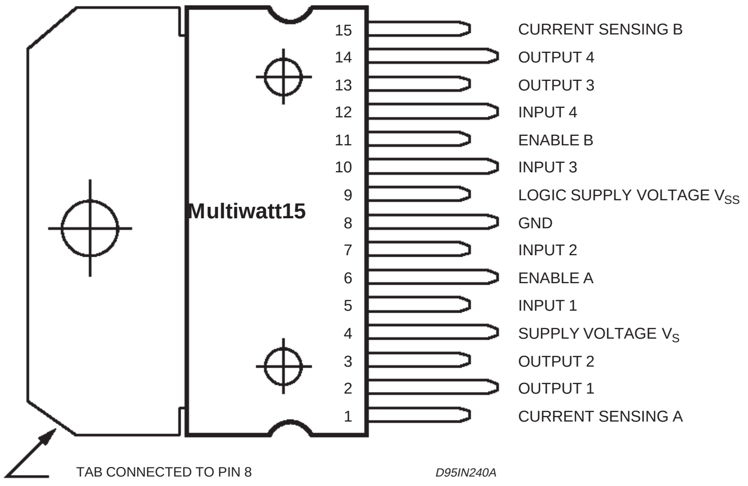
BLOCK DIAGRAM

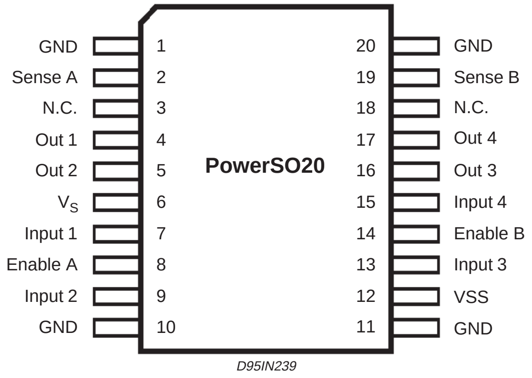


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate