# DESIGN AND ANALYSIS OF A SPEED DRIVE FOR A DC MACHINE USING PROPORTIONAL INTEGRAL (PI) CONTROLLERS FOR FOUR QUADRANT MOTOR OPERATION

**OSINOWO PAUL IYINOLU**

**2015/3947**

A PROJECT SUBMITTED TO

THE DEPARTMENT OF ELECTRICAL/ELECTRONICS AND COMPUTER

ENGINEERING

COLLEGE OF ENGINEERIING

**BELLS UNIVERSITY OF TECHNOLOGY, OTA,**

**NIGERIA**

IN PARTIAL FULFILLMENT FOR THE AWARD OF BACHELOR OF ENGINEERING (B. ENG) DEGREE IN

**ELECTRICAL/ELECTRONICS ENGINEERING**

**OCTOBER, 2020.**

# DECLARATION

I, Osinowo Paul Iyinolu, a graduating student of the Department of Electrical/Electronics and Computer Engineering, College of Engineering, Bells University of Technology, Ota, hereby declares that this dissertation titled “DESIGN AND ANALYSIS OF A SPEED DRIVE FOR A DC MACHINE USING PROPORTIONAL INTEGRAL (PI) CONTROLLERS FOR FOUR QUADRANT MOTOR OPERATION” submitted by me in partial fulfilment of the requirements for the Degree of Bachelor of Engineering (B.Eng.) is my original work. It has never been previously submitted in part or in whole for the award of a degree. Wherever contribution of others are involved, effort have been made to indicate them clearly with due reference to the literature.

Signature                                                  Date

# CERTIFICATION

This is to certify that he project title DESIGN AND ANALYSIS OF A SPEED DRIVE FOR A DC MACHINE USING PROPORTIONAL INTEGRAL (PI) CONTROLLERS FOR FOUR QUADRANT MOTOR OPERATION was carried out by Osinowo Paul Iyinolu with matriculation number 2015/3947 of the department of Electrical/Electronics and Computer Engineering, College of Engineering, Bells University of Technology, Ota, Ogun state Nigeria.

______________________________ ______________________

Dr J. O. Aibangbee Date

Supervisor’s Signature

______________________________ ______________________

Dr A. O. Amole Date

Head of Department’s Signature

# DEDICATION

This project is dedicated ultimately to My Lord Jesus Christ, my parents: Mr and Mrs Osinowo for their unending love, support and guidance throughout the preparation of this thesis which took place mostly at their home.

# ACKNOWLEDGEMENTS

I will like to specially thank God Almighty for this unending wisdom knowledge and understanding that he gives me to be able to put together this research work and also for the strength and relentlessness to never give up or settle for less while this research was underway.

I would also like to specially thank my supervisor, Dr J. O. Aibangbee for his continuous support and kind gestures in guiding the process of accurately documenting this research work.

My gratitude goes to the HOD Electrical/Computer Engineering, Dr A.O. Amole and to all my Professors and Lecturers in Bells University of Technology Electrical/Computer Engineering Department for their relentlessness in delivering quality knowledge, most of which have fuelled my understanding to put together this research work.

My gratitude goes to my parents Mr and Mrs Osinowo for their kindness at providing the resources and comfortable study space that was very critical for undertaking this research project.

Finally, I am also extremely grateful to Professor Marko Hinkkanen of Aalto University, Espoo, Finland for his kind gestures in replying my mails on some technical questions related to some of his lecture notes I found online. I also thank my colleagues for their support and willingness to listen and give advices as regards the decision making that fuelled this research work.

.

# TABLE OF CONTENT

# LIST OF SYMBOLS

$R_a$ = Armature Resistance (ohms)

$L_a$ = Armature Inductance

$I_a$ - Armature current

$E_b$ = armature Back EMF (V)

$V$ = applied voltage (V)

$T_M$ = Motor Torque (N-m)

$J$ = Equivalent moment of inertia of motor shaft (Kg/m$^3$)

$B$ = Coefficient of friction of motor shaft (Nm/rad/sec)

$K_F$ = Motor constant, Back EMF constant

$T_{sw}$ = Switching period

$K_p$ = proportional gain

$K_i$ = integral gain

$K_d$ = derivative gain

$\alpha_c$ = Current loop bandwidth

$\alpha_s$ = Speed loop bandwidth

$r$ = active resistance

$b$ = active damping

$e_{ss}$ = Steady state error

$t_s$ = Settling time

$t_r$ = Rise time

## LIST OF ABBRIVATIONS

| | |
|---|---|
| PWM | Pulse Width Modulation |
| PID | Proportional Integral Derivative |
| DC | Direct Current |
| PMDC | Permanent Magnet Direct Current |
| PSO | Particle Swarm Optimization |
| MATLAB | Matrix Laboratory |
| GUI | Graphic User Interface |
| IMC | Internal Model Control |
| DC -DC | Direct Current to Direct Current |

# ABSTRACT

This thesis focuses on the design and analysis of a speed drive for a Permanent Magnet DC (PMDC) Machine using MATLAB/SIMULINK as a simulation aid. The speed drive is designed for fast dynamic speed and current response in all four quadrant of the motor’s torque-speed plane.

The DC motor’s mathematical model is used for charaterising the system, PID/PI controllers are designed and tuned with methods including (MATLAB tuning, particle swarm optimization (PSO) and Internal Model Control). Two control strategies, single loop PID and cascaded PI loops, were studied. The cascaded PI control was used for developing the speed drive of the PMDC machine which was tunned for a current loop bandwith of 2π.600 rads/s and a current limiting logic was implented in the current loop of the controller. The PMDC machine’s speed was controlled using voltage control method with the use of Full bridge DC-DC power converter. Metal–oxide–semiconductor field-effect transistor (MOSFET) was used as the swtich. The switching was done using Unipolar Pulse Width Modulation technique due to its positive effect on the motor’s current ripples. The use of active damping and active resistance in the speed and current loop respectively was done to improve the drive performance.

After tuning the controllers, the PSO-tunned single loop PID controller had comparately better response than the Matlab tuned single loop PID controller. PSO-PID gave a well damped response with minimal overshoot as compared to the Matlab tuned PID. Likewise, the cascade PI controller gains were obtained and the controller yielded a well damped response with negligible overshoot. The drive current limiting mechanism also ensured the rated continous current of the motor was not exceeded during continous operation. The cascade PI controller is shown to have excellent load disturbance rejection capacity with zero steady state error. With the Full Bridge converter, the motor was made to operate in both forward and reverse direction.

# CHAPTER ONE

# INTRODUCTION

## 1.1 BACKGROUND

After the first sources of direct current (DC) were invented, DC machines were one of the first types of electro-mechanical machines used. DC machines are more advantageous over AC machines as regards to Speed regulation and versatility. (Ali A. Hassan, 2018). DC motor is a highly controllable electrical actuator which is widely used for applications including but not limited to robotics manipulators, guided vehicles, steel rolling mills, cutting tool, overhead cranes, electrical traction and other application etc. DC motors are used extensively in industries in variable speed demanding applications because of its speed-torque characteristics and its simplicity in control. Muhammad Rafay Khan, (2015).

Process control industry has seen many advances in the past two decades in terms of controller design and its implementation. The need for automatic controllers that are responsive and accurate in carrying out precision tasks in the industry is of great demand. (Ali A. Hassan, 2018). For achieving the desired performance in the control of most systems, the feedback loop is an indispensable tool in the control of systems. In order to get fast dynamic response of the system, many control strategies have been devised in various feedback control systems. The importance of controls in a drive system includes precise and quick tracking for reference speed with minimum overshoot or undershoot and having little or no steady-state error. Muhammad Rafay Khan, (2015)

Proportional Integral and Derivative (PID) controller is one of the earliest and best understood controllers which is integrated into almost every industrial control application due to its efficiency, high reliability, good robustness, easy to operate stabilization and elimination of steady-state error. Sabir & Khan, (2014) The PID controller has optimum control dynamics including zero steady state error, fast response (rise time), little or no oscillation and higher stability. The purpose of using the derivative gain component in addition to the Proportional-Integral (PI) controller is to eliminate the overshoot and the oscillation occurring in the output response of the system. A key advantage of the PID controller is that it can be used in the control of higher order processes including more than one energy storage element. Rao, (2013)

PID type controllers are used to attain the speed-torque control of a DC motor. However, the optimization and tuning task of these controllers is very difficult and time consuming, mainly under the varying load conditions, parameter variations in abnormal operating modes etc. Muhammad Rafay Khan, (2015)

This Research-Project discusses the use of PID Controllers in achieving stable speed control of a Permanent Magnet DC (PMDC) Machine at a specified value. The DC motor model is developed by using the dynamic equation of the system and MATLAB/SIMULINK is used for simulating and analysing the response of the system, the PID tuning and analysis is done on the Simulink platform. The operation and tuning rules of the PID controller is also discussed. At the end, experimental results are presented and discussed.

## 1.2 AIM OF PROJECT

The Aim of this research project is to design and analyse a speed drive for a Permanent Magnet DC (PMDC) Machine using MATLAB/SIMULINK as a simulation aid.

## 1.3 OBJECTIVES OF PROJECT

The objectives of this project are:

a) To design and simulate a PID controller for controlling the speed of a DC motor.

b) To control the speed response of Brushed PMDC Motor for accurate reference speed tracking at the selected value during all four quadrant of operation.

c) To improve the dynamic performance of the motor's outputs i.e. the speed and torque output and to ensure safe operating conditions for the motor.

d) To simulate and show using MATLAB/SIMULINK software that PI/PID controllers are very powerful at stabilizing a DC Motor's speed at a specified value for agile load disturbance rejection.

## 1.4 METHODOLOGY

The methods to be adopted include;

a) Data collection: Data collected on previous research will be collected and analysed in the project research, which will provide further insight to implementing an efficient DC motor speed controller. DC motor parameters will be obtained from a PMDC motor datasheet

b) MATLAB tuning and Particle Swarm optimization (PSO) algorithm will be adopted as a preliminary step for obtaining optimal values for a single loop PID controller gains $(K_p, K_i, K_d)$

c) Internal Model Control (IMC) for determining Proportional Integral (PI) controller gains for cascaded PI controller.

d) Current limiting logic for ensuring safe operating condition of the controlled DC motor.

e) Switch mode DC-DC converter for ensuring efficient control of voltage to the DC motor.

f) Unipolar PWM for switching control signal.

## 1.5 SCOPE

a) Development of the Speed Drive for a Brushed Permanent Magnet Direct Current (PMDC) machine is done for handling Passive loads (loads incapable of driving the motor at stall condition i.e. zero speed).

b) The DC motor power input is from a DC voltage source or a rectified Alternating Current (AC) source.

c) Analysis carried out in this research are done only on the linear region of operation

## 1.6 SIGNIFICANCE OF PROJECT

DC machines are used for tons of activities in many industries for cutting, milling, driving, spinning and many other functions. This is due to the speed torque characteristics and the wide range of sizes and configurations than can be realized. Dc motor are also the major source of movement and rotation in robotics.

The numerous applications of DC electric machines gives rise to the need for automatic control of DC machines with greater performance and precision within safe limits in a sustainable fashion.

Having control over the speed of your motor is vital for you to be able to get the full potential for your tools and systems. Accurate, precision control of DC motors in industrial processes ensures a no-chatter and perfect working conditions at all times.In industries, for every machining job, there is a cutting, rotating or sliding speed that is optimum. With a robust speed drive integrated to the system we can achieve the desired output of the motors at all times without causing damage in the system.

This research-project focuses on the use of PID/PI controller in achieving the desired speed of DC motors automatically with a quick and accurate response, which is of high demand in industries and processes that employs DC motors as a major actuator.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 GENERAL

This chapter presents the literature review that was conducted as part of the research reported in this thesis. It covers pertinent established concepts and techniques related to the DC motor speed control system design, for applications that require actuators with accurate speed characteristics. Simulation and real time implementation results employed for DC motor speed control systems in various literature are analysed and discussed.

## 2.2 DC MACHINE

For many decades (since 1980), the Brushed DC machine has been the automatic choice where speed torque control was very necessary. Figure 2.1 shows a schematic of a brushed DC motor showing all its important part consisting of the Commutator, Brushes, Armature conductor, and the Field windings. The applications of a DC motor ranges from steel rolling mills, electric traction, center winders to a very wide range of industrial drives, robotics printers and precision servos. The range of power outputs is wide, which varies from several mega-watts to a few watt of power. Also the speed of DC machines can be easily varied and controlled which makes them very suitable for precision industrial activities. (Hughes, 2006). Their speed-torque characteristics also makes them very useful in various applications. A few major components of a DC motor are described in turn. These parts provide a major contribution in the operation of a DC machine. They are also used to determine the kind of DC machine in use.

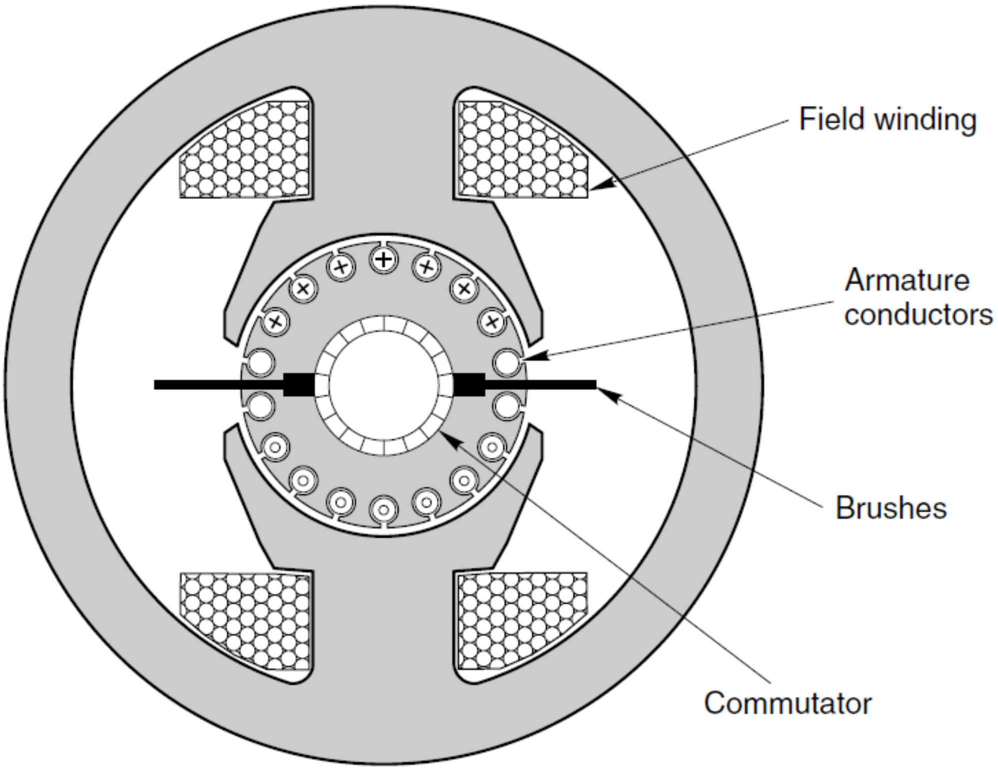


Figure 2.1: Conventional Brushed DC Motor (Austin Hughes, 2006)

**PART OF A DC MACHINE**

1. COMMUTATOR: The commutator forms a ring like shape mounted on the motor shaft. It is connected to all the armature windings of the motor. The purpose of the commutator is to ensure that the pattern of current flow in motor regardless of the position of the motor always remains constant, with positive current facing the North magnetic pole and negative current facing the south magnetic pole. It constantly reverses the current flow in the machine
2. BRUSHES: The brushes are used for supplying (Motoring) and extracting (Generating) dc current form the motor during operation. During motoring, voltage is applied across the brushes of the motor and current flows from one end of the brush to the other end through the armature windings. During generation, current is taking from the brushes and supplied to the mains

3. ARMATURE CONDUCTORS: This consist of copper conductors wound on a laminated former. The armature conductors interact with the magnetic field in the air-gap of the motor. Current flowing through the armature windings is responsible for generating the torque required for acceleration and load handling during motoring. During generation, electrical energy is generated in the armature windings. The generated current is an AC current then rectification is done by the commutator and brush combination.
4. FIELD WINDINGS: The field windings are responsible for developing magnetic flux during motor operation. Field windings configuration are used to group dc machines into two types namely, separately excited dc machine and self-excited dc machine, then there is PMDC machine. Self-excited machines develop their field magnetic flux from the current applied to the motor terminals i.e. the field and armature windings are excited from the same source, separately excited machines relies on an external voltage source for production of magnetic flux. PMDC machines rely on magnets for production of field magnetic flux.

### 2.2.1 TYPES OF DC MOTORS

There are three types of self-excited DC machines which are characterized by the connection of field winding in relation to the armature which are described as follows.

*I.* *Shunt-wound motor*

A motor in which the field windings is connected in parallel with the armature is called a shunt wound motor. The current through the shunt field windings is not the same as the armature current. The shunt field windings are designed to produce the necessary MMF by means of relatively large number of turns of wire having high resistance. This means the shunt field is relatively small compared with the armature current. Mwahib Mohamed et al, (2016).

II. *Series-wound motor*

In this type of motor the field windings is connected in series with the armature. The series field windings carries the armature current. Since the armature current passing through the series field winding is of equal magnitude, the series field windings must be designed with much fewer turns than the shunt field windings for the same MMF. As a result the series field winding has a relatively small number of turns of thick wire and therefore will have a low resistance. Mwahib Mohamed et al, (2016).

III. *Compound-wound motor*

A compound-wound motor has both a series and a shunt field windings, (i.e. one winding in series and one in parallel with the armature circuit). When the shunt field winding is connected directly across the armature terminals it is called short shunt connection. Otherwise, when the shunt windings is connected so that it shunt’s the series connection of armature and series field, it is called a long shunt connection. The flux produced by the shunt field windings of the compound motor is considerably larger than the flux produced by the series field windings. Mwahib Mohamed et al, (2016).

## 2.3 SPEED CONTROL SYSTEMS

In recent times, the need for accurate speed control in industries as led to the demand for highly controllable actuators. DC machines are versatile and robust machines used extensively in industries as actuators for a wide range of operations. Torque-speed characteristics of DC machines is highly demanded in most industrial applications. This has led to extensive research interest in the fabrication of robust control systems for DC machines for carrying out more precision tasks in industries.

### 2.3.1 DIRECT CURRENT SPEED CONTROL METHODS

*i. Flux control method*

Anurag Dwivedi, (2013) has proposed a DC motor speed control using Field and Armature control simultaneously. It was discovered that to increase the speed of a Shunt DC motor the resistance of the windings on the field circuit should be varied, as the resistance increases the current flow in the field circuit is reduced. This action thereby leads to reduction in flux produced by the field windings which results in increased speed of the motor. Also for decreasing the speed of the motor, firstly the resistance of the field windings should be kept at minimum, the resistance of the armature winding should be increased. This action leads to high voltage drop across the armature. Due to constant supply voltage, by this method we can deal with ranges of speed of the motor and the system can be analysed using equation 2.1.

$$N = KV - \frac{Ia(Ra + R)}{\varphi} \qquad 2\text{-}1$$

*ii. Armature voltage control method*

Muniracad, (2019) Described controlling the speed of a DC motor by changing the terminal voltage of the machine. It is the most used method of DC motor speed control in which the armature voltage can be varied by varying the terminal voltage. This effect can be described in the equation ($I_A = V_T - E_A/R_A$). It can be deduced from the equation that by increasing/decreasing the terminal voltage of the DC motor leads to an equivalent increase/decrease in the armature current. Considering an increase in the armature voltage i.e. a corresponding increase in the armature current, will lead to increase in the generated torque which is evident from the equation ($T_{ind} = K\emptyset I_A$). At constant load the increased torque leads to an increase in the angular displacement (speed) of the motor shaft (ω). The increase speed increases the internally generated EMF ($E_A$) also known as back EMF ($E_A = K\emptyset\omega$). Due to increased torque the armature current will decrease which will further decrease the torque induced. The induced torque is reduced until it becomes equal to the load torque. At the point the motor will come in steady-state and working at a higher speed.

*iii. Armature current control method*

Muniracad, (2019) Described the speed control of a DC motor by changing the armature resistance which will lead to a corresponding change in the armature current. Since increasing the armature current leads to the increasing in speed. Hence, it is evident that increasing the resistance of the armature leads to low armature which leads to decrease in speed of the DC motor. This method can be used to decrease the speed of a motor below its base speed. The problem faced by this method is that the increase in resistance will increase power losses. This method can be used if the motor runs at the base speed most of the time and only for a short time slow speed is required.

### 2.3.2 PULSE WIDTH MOULATION (PWM) METHOD

Ms.S.R.Bhagwatkar et al, (2015) proposed a fixed speed control of a DC motor with high precision and reliability. The implementation was done with a microcontroller (ATmega128 microcontroller) and an LCD display was used for physically monitoring the performance of the motor. Manual input are applied to the motor, which cause the motor to rotate at a particular speed. IR sensors (which are contactless) are used to monitor the speed of the motor. The output of the IR sensor is fed to the microcontroller which in turn uses PWM signals is to control a MOSFET transistor (drive circuit for the DC motor), which supplies voltage directly to the Motor. The PWM signal from the microcontroller ensures that the motor will rotate at a fixed speed. Pulse width modulation (PWM) works by making a square wave with a variable on-off-ratio, the average on-time of the pulses may be varied from 0 to 100 percent. This effect leads to a variable amount of power transferred to the load. The main advantage of PWM circuit over resistive circuit is the efficiency, at a 50% level the PWM will consume about 50% of full power, almost all of which is transferred to the load. A resistive controller at 50% load power will consume up to 71% of full power, 50% of the power goes to the load and the other 21% is wasted on heating the series resistor. The design using PWM for control of DC motor with fixed speed control was found reliable and has comparatively high precision.

Joshi & Thakare, (2012) has proposed a method of speed control of a DC motor using analog PWM technique. In this method an analog PWM is developed, that drives the DC motor by switching four H-bridge connected MOSFETs. The H-bridge configuration results in the realization of a bidirectional full bridge circuit capable of operating in four quadrant of the I-V graph. For effectively controlling the motor, a systematic approach was laid for determining the switching frequency of the PWM pulses, as the switching frequency has a direct effect on the motor current ripples due to the fact that a dc motor works like a low pass filter and has a bandwidth of allowable frequencies. A method of choosing the PWM frequency based on the motor characteristics was presented. It works by setting an allowable percent for the current ripples, then the minimum frequency to attain that goal is worked out mathematically. Firstly the average power of the motor during switching is determined using equation 2-2.

$$P = 2I^2R \qquad 2\text{-}2$$

The DC motor current response to a PWM pulse at 50% duty cycle is shown in figure 2-2

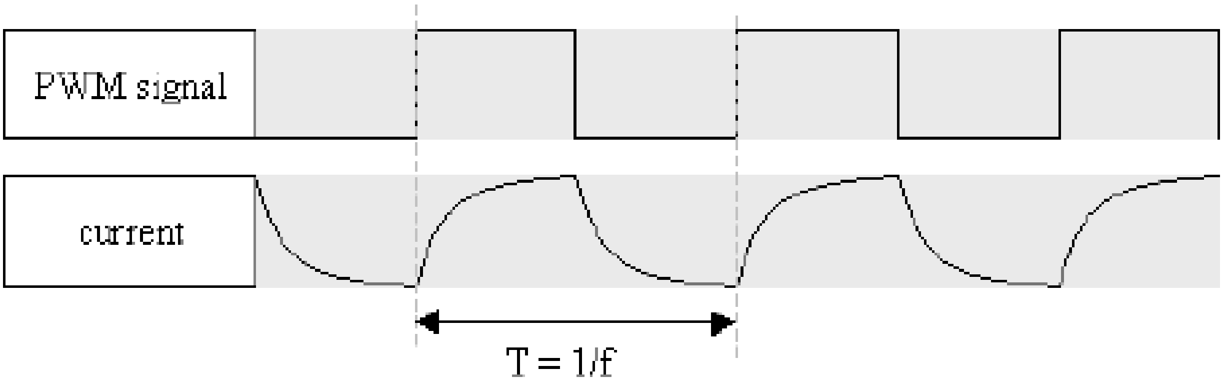


Figure 2.2: Current Response of a DC motor to a PWM Pulse at 50% Duty Cycle

T is given as the switching period of the pulses. Figure 2-2 shows the motor at stall condition which is the highest operating current in the DC motor i.e. the worst case scenario. The current at the falling edge is found in equation 2-3, where τ is the time constant of the motor (L/R). The current at time t = T/2 which is given as $i_1$ must no less than P% lower than at t = 0 ($i_2$).

$$i = Ie^{-t\frac{R}{L}} \qquad 2\text{-}3$$

This limiting condition is determined with the use of equations 2-4 and 2-5.

$$i_1 = \left(1 - \frac{P}{100}\right) i_0 \qquad 2\text{-}4$$

$$Ie^{-\frac{TR}{2L}} = \left(1 - \frac{P}{100}\right) Ie^0$$

$$e^{-\frac{TR}{2L}} = \left(1 - \frac{P}{100}\right) e^0$$

$$-\frac{TR}{2L} = \ln(1 - \frac{P}{100})$$

$$T = -\frac{2L}{R}\ln(1 - \frac{P}{100})$$

Hence, since f = 1/T

$$f = -\frac{R}{2L\ln\left(1-\frac{P}{100}\right)} \qquad 2\text{-}5$$

These formulas were used to prove that the optimal frequency for the PWM pulse is not the highest possible frequency, but somewhere between high and low in the KHZ range.

#### 2.3.2.1 CONTROL SCHEMES

##### i. SINGLE LOOP FEEDBACK CONTROL

A control system using a feedback mechanism maintains a prescriptive relationship between the process output and the set point by constantly comparing them using the error signal as a means of control. Feedback control is the simplest form of closed loop control scheme. Feedback control system has many daily routine application e.g. Automobile speed control or air conditioner temperature control system which uses the difference between the actual and the desired speed or temperature to change the manipulated variable. A system in which the output is used to regulate the output is called a Closed-loop system. The block diagram is shown in Figure 2-3 represents a simple feedback control system. S.K. Singla, (2013)

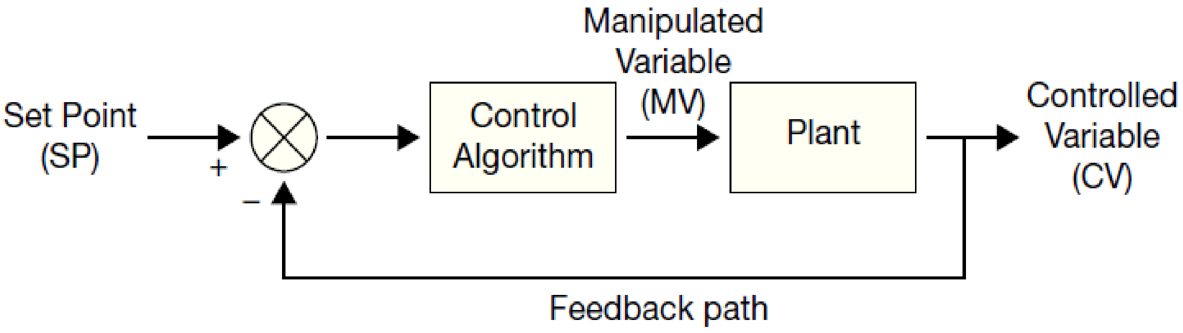


Figure 2.3: Feedback Control System (S.K. Singla et al.)

Different variation of a feedback loop are used in the design of speed control of DC motor, due to its simple and robust structure

## ii. CASCADE FEEDBACK CONTROL

This is the most commonly used variation in Speed control drives and steam process industry. A cascade control comprise of two loops an inner loop and an outer loop. Optimal performance of a cascade control scheme is achieved when the inner loop has a faster dynamics as compares to the outer loop. Figure 2-4. Shows the structure of a cascade scheme. The inner loop controls the secondary process e.g. armature terminal voltage of a DC motor. While the outer loop controls the primary process e.g. the angular displacement of a DC motor. Initially the inner loop is tuned first while keeping the outer loop controller in manual mode. The methods for tuning the inner loop includes the direct synthesis, Ziegler-Nichols, RA methods, metaheuristic techniques (PSO, GA etc.). Thereafter the outer loop controller is tuned to complete the tuning process. A variation to the cascade feedback system is the Cascade plus Feed-Forward Scheme. S.K. Singla, (2013)

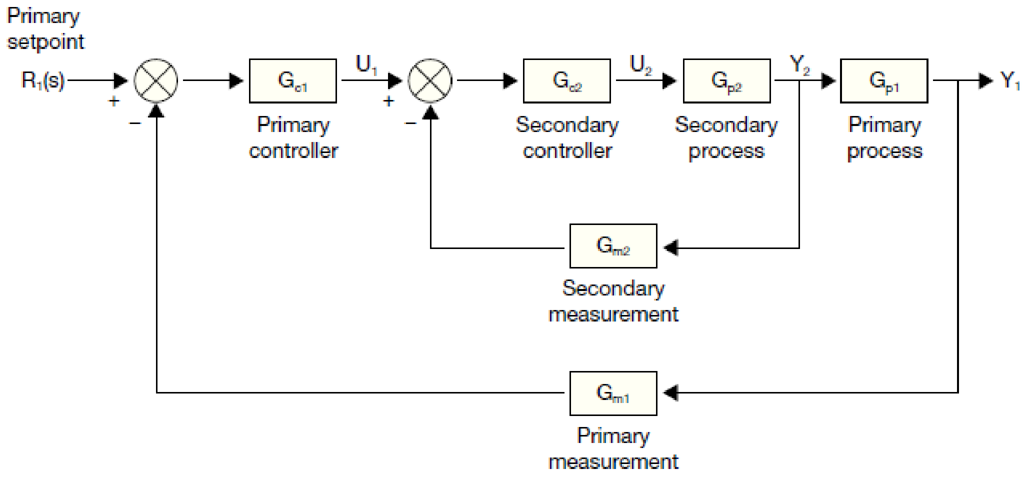


Figure 2.4: Cascade Control System (S.K. Singla et al.)

Most literature look into the speed control of a DC motor as a feedback control loop otherwise known as a closed loop control system. The research activities carried out on these areas shows the need for continuous improvements on the simple feedback loop in controlling the speed of industrial actuators (DC motors). Literature based on various kinds of controllers used for DC motor speed control are presented and discussed.

#### 2.3.2.2 PID CONTROLLERS

The PID controller is by far the most common control algorithm. Most feedback loops are controlled by this algorithm or its minor variations. Figure 2-5 shows the structure of a PID controller. It is implemented in different forms, such as a stand-alone controller in PC, PLC, DCS and micro-controller or as a part of a Direct Digital Control (DDC) package or a hierarchical distributed process control system. The PID algorithm can be approached in many different direction. It can be viewed as a device that can be operated with a few rule of thumb, but it can also be approached analytically. Gajanan, (2011) the various components in a PID controller and their effect on the performance of the control system are discussed in the following section. The "textbook" version of the PID algorithm is given in equation 2-2:

$$u(t) = K\left(e(t) + \frac{1}{Ti}\int_0^t e(\tau)d\tau \; + Td \; \frac{de(t)}{dt}\right) \qquad (2.2)$$

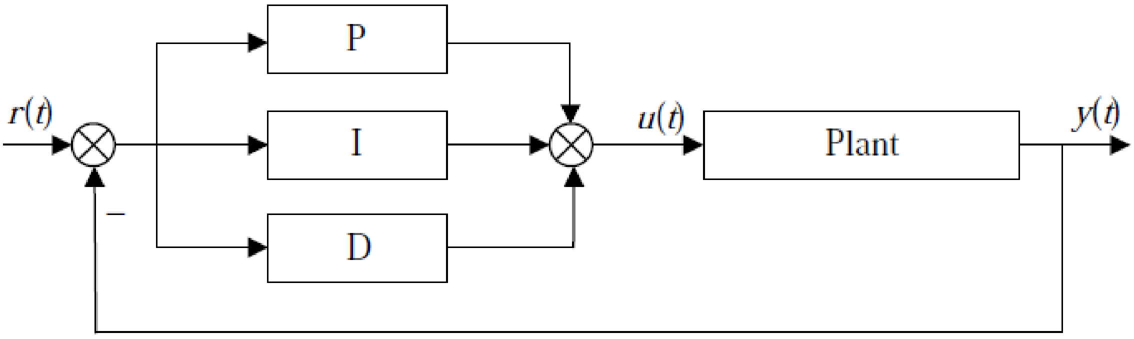


Figure 2.5: Diagram of a PID Control System (Chang-liang Xia, 2012)

Where “u” is the control variable and e is the control error ($e = y_{sp} – y$) where $y_{sp}$ is the Set-point and y is the Feedback control signal. The control variable is thus a sum of three terms: the P-term (which is the proportional to the error), the I-term (which is proportional to the integral of the error), and the D-term (which is proportional to the derivative of the error). The controller parameters are proportional gain K, integral time $T_i$ and the derivative time $T_d$. Karl Johan Astrom, (1995). The PID parameters are futher discussed as follows.

### I. Proportional Action

The proportional action takes into account the present state error only. The proportional signal is a measure of difference between the present value (PV) and set point (SP). The magnitude of the proportional signal increases with this difference. When the PV approached the SP, the error becomes so small so that the controller cannot trigger the PV to meet up with the SP. This implies that there is always a steady state error, which is an offset value form the set point in the system. This effect leads to the choice of higher value for the gain of the proportional term. This higher value, however, makes the system unstable with oscillations and overshoots, which makes the behaviour similar to that of an on-off controller. It can be concluded that the P-term cannot sufficiently satisfy accurate system control especially in higher order systems with more than one energy storage elements. The P-term is usually in conjunction with the Integral or Derivative term. The Mathematical equation of the output of a P-controller is given by the equation 2-6. Bista (2016)

$$u(t) = K_p e(t) \qquad (2.6)$$

### II. Integral Action

The integral action of a PID controller is proportional to the integral of the control error. In terms of equation, it is given in equation 2-7

$$u(t) = K_i \int_0^t e(r)dr \qquad (2.7)$$

Where $K_i$ is the integral gain. It turns out that the integral action is related to the past and present values of the control error. The presence of a pole at the origin of the complex plane allows reduction to zero of the steady-state error when a step reference signal is applied or a step load disturbance occurs. This means the integral action is able to automatically set the term $u_b$ to the correct value automatically so that the steady state error is zero. The integral term is used when it is required that the controller correct for any steady state offset from a constant reference signal value. Integral control overcomes the shortcoming of proportional control by eliminating offset without the use of excessively large controller gain. Johnson & Moradi, (2005)

### III. Derivative Action

The derivative action is based on the rate of change of control error. An ideal derivative control law can be expressed mathematically as given in equation 2-8:

$$u(t) = K_d \frac{de(t)}{dt} \qquad (2.8)$$

The derivative action has a great potential in improving the control performance as it can anticipate an incorrect trend of the control error and counteract for it. Since the controller can use rate of change of an error signal as an input, then this introduces an element of prediction into the control action. When using derivative control care is needed unlike for P and I control. In most application pure Derivative control cannot be implemented due to possible measurement noise amplification. However derivative control is still an essential feature of some real-world control applications e.g. tachogenerator feedback in DC motor control. Johnson & Moradi (2005)

### 2.3.3 DC MOTOR SPEED CONTROL USING PID ALGORITHM

The need for accurate speed control of a DC motor is very important and that leads to the use of accurate control techniques. PID controllers are one of the most popular controls used in industries nowadays, due to its accuracy and robustness in achieving precision speed control. The performance of a PID controller depends heavily on the gain parameters for the proportional, integral and derivative term. Various tuning methods have been devised and will be analysed and discussed in this section. Due to the difficulty involved in the tuning process of a PID controller, there has been considerably lot of effort done in researching on ways to make these controller even more accurate and easy to use.

N. D. Mehta et al, (2017) discussed the DC motor drive combined with a controller for speed control under varying load condition. The DC drive consists of the input source, input and output filters which perform signal processing functions to remove unwanted frequency component form the signal, power electronic modulators (PEM) which converts electrical energy from the source in a suitable form for the motor and a controller that controls the power modulator. The various controllers including P (proportional), PI (proportional integral) and PID (proportional integral and derivative) were analysed. The proportional controller was found to be only suitable for 1st order systems, it offers smaller steady state error, faster dynamics and smaller amplitude and phase margin. The PI controller was found to have a poor performance when the plant controller is highly non-linear in nature. It is only used when fast response of the system is not required, large disturbances and noise are present during operation of the process. The PID controller was found to a more suitable control algorithm for controlling the PEM, with improved rise time, settling time, steady state, and better stability if the $K_d$ is small. The overall system made used of a closed loop speed control scheme that employs an inner loop current loop within an outer speed loop. The inner current loop is provided to limit the converter and motor current and torque below safe limit. It is also beneficial in reducing the effect on drive performance of any non-linearity present in converter-motor system. The simulation result gotten using Simulink shows that the motor speed slows down only for about 270rpm (9%) in 980 milliseconds under the effect of full load. Also the motor is hunting about 200rpm (6.66%) in 900 milliseconds on unloading condition.

Muhammad Rafay Khan et al, (2015) proposed the use of a separately exited DC motor drive for the speed control application due to its accurate speed control, controllable torque, high reliability and simplicity. The dynamic model of the separately exited DC motor was derived and this model was used to develop a transfer function for the relationship between the angular position and the applied voltage. The system consists of a control loop for speed and armature current and a PID controller was employed in order to get the desired response. The block diagram of the system with and without the PID controller was developed and simulated using Matlab/Simulink software. The first step was to analyse the step response of the system with and without PID controller. The PID controller greatly reduced the rise time by around 20%, the settling time was also improved by the 20%. The system was tested under full load condition (load disturbance) with and without PID controller. It was observed that the speed was not attained to the desired level when a PID controller is absent. The PID controller was used to overcome the current problem of speed control under the effect of load. The first step of parameter tuning for the PID controller was done using Ziegler-Nichols tuning method which overcomes the overshoot problem of reference speed and actual speed under no-load condition and also eliminates steady-state error. The next step was to use manual tuning to overcome the undershoot problem of the motor speed under the loading effect. The PID controller is applied as an input voltage to the armature terminals of the DC motor. The ultimate parameters for the gain values were found and simulated.

The selection flowchart shown in Figure 2-6 shows that a PID controller is consists of a family of controller with labels (P, PI, PD, and PID). The flowchart gives a guide to selecting an optimal controller strategy for various application that requires accurate control

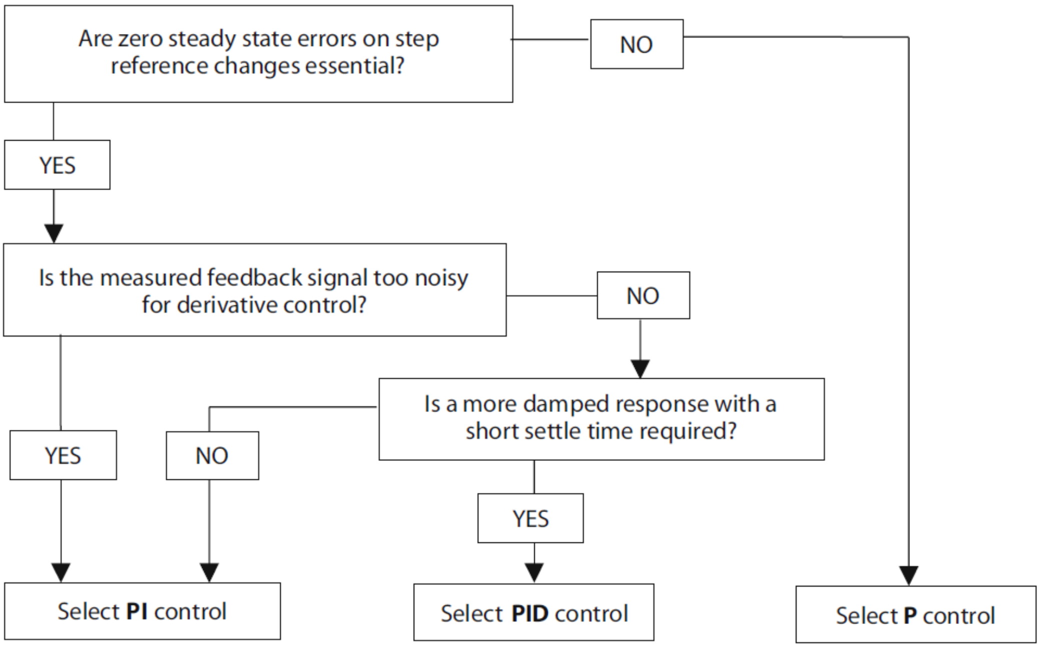

Are zero steady state errors on step reference changes essential?
NO
YES
Is the measured feedback signal too noisy for derivative control?
NO
Is a more damped response with a short settle time required?
YES
NO
YES
Select **PI** control
Select **PID** control
Select **P** control

Figure 2.6: PID Term Selection Flowchart. Johnson & Moradi, (2005)

### 2.3.4 PID TUNNING METHODS

After choosing the PID design to be used in a control process as shown in Figure 2-6, the second step in setting up the PID controller is to tune or choose numerical values for the PID coefficients. Tuning is the adjustment of control parameters to the optimum value for the desired control response. Stability is an important requirements for various control systems. However, different systems have different behaviour, different applications have different requirements which may conflict with one another. PID tuning is difficult problem, even though there are only three parameters which is simple to describe in principle. This is because parameters must satisfy complex criteria within the limitation of PID control. Mwahib

Mohamed Harron Elhag, (2016). There are various types of PID tuning methods, some of which includes:

**Classical Techniques**:

1. Manual Tuning method
2. Ziegler-Nichols method
3. Internal Model control (Model-based)
4. Pole-Placement method

**Heuristic Algorithms:**

1. Genetic Algorithm (GA)
2. Particle Swarm Optimization (PSO)
3. Simulated Annealing

#### 2.3.4.1 MANNUAL TUNNING METHOD

Manual tuning method is achieved by adjusting the system's parameters and watching the systems responses. The parameters ($K_{P,}$ $K_i$, $K_d$) are changed until the desired system response is obtained. Although this method is simple it should be used by experienced personnel.

Mwahib Mohamed et al, (2016) described an example of manual tuning method in which the parameters $K_i$ and $K_d$ are first set to zero. $K_p$ is then increased until the output of the loop oscillates. After obtaining optimum $K_p$ value, it should be set to approximately half of that value for a "quarter amplitude decay" type response. The $K_i$ is then increased until any offset is corrected in sufficient time for the process. However, too much $K_i$ will cause instability. Finally $K_d$ parameter is increased, until the loop is acceptably quick to reach its reference after a load disturbance. However, a very high value of $K_d$ also will cause excessive response and overshoot. To reach the set point more quickly, a fast PID loop tuning usually have slight overshoots. Meanwhile some systems cannot tolerate this overshoot. In this case an over-damped closed-loop system is required, which will require a $K_P$ setting significantly less than half of the $K_p$ setting causing the oscillation. The effect of the PID gains are summarized in Table 2-1.

Table 2-1: Effect of Changing Control Parameters

| Parameter | Rise Time | Over shoot | Settling Time | Steady State Error |
|---|---|---|---|---|
| **$K_P$** | Decrease | Increase | Minor Change | Decrease |
| **$K_I$** | Decrease | Increase | Increase | Eliminate |
| **$K_d$** | Minor Change | Decrease | Decrease | Minor Change |

Manual tuning can also be done using the MATLAB software PID tuner which allows to tune the response of the system according to the desired transient response and loop stability criteria.

Getu, (2019) presented a method for tuning the gains of a PID controller optimally using MATLAB Simulation Software. The design criteria was first set be Rise time <3seconds, Settling time <10 seconds, Overshoot of <5% and a steady state error <1%. The plant to be controlled was a temperature controlling system, which could be used in a furance, industrial chemical process etc. A first order model and a second order model approximation was derived for the system. For the first order system, a Proportional controller with gain of 12 satisfied the desired performance criteria. Unlike the first order system, a propotional controller was not sufficient for achiving the set criteria. Hence gains of $K_p = 110$, $K_i = 10$, $K_d = 100$ was found satisfying all the control requirements. Manual tuning using the root locus approach was also presented which makes used of the MATLAB SISO tool Graphic User Interface (GUI) and enables tuning of the PID controller by viewing the root locus of the system. The step response of the system is also observed in the GUI until the dersired response is achieved. The PID controller achieved on MATLAB can be integrated with Arduino or any other type of microcontroler. The controller can then be used for generating PWM pulses through its digital pin, which inturn can be used to effectively control the plant.

### 2.3.4.2 ZIEGLER-NICHOLS TUNNING METHOD

The Ziegler-Nichols methods use and on-line process experiment followed by the use of rules to calculate the numerical values of the PID coefficients. It is frequency domain method which is based on experimentally determining the point of with the system becomes marginally stable. Van der Zalm, G. M., (2004).

Johnson & Moradi (2005) in his work according to John G. Ziegler and Nathaniel B. Nichols in the 1940s developed two methods for tuning a PID controller parameters, two methods includes the Ziegler-Nichols' closed loop method and the Ziegler-Nichols open loop method. Ziegler-Nichols method uses the definition of acceptable stability as basis for their controller tuning rules. The ratio of the amplitudes of the subsequent peaks in the same direction due to disturbance or a step change of the set point in the control loop is approximately ¼. It has become a common point of view that the ¼ decay ratio of the step response corresponds to poor stability of the control loop. The first aid to improve the system if the stability of the control loop becomes too poor is to decrease the $K_P$ value. The tuning procedure is as follows:

I. Bring the process to (or as close as possible) the specified operating point of the control system to ensure that the controller during the tuning is "experiencing" representative process dynamics and to minimize the chance that variables during the tuning reach limits. Process is brought to the operating point by manually adjusting the control variable, with the controller in manual mode, until the process variable is approximately equal to the set-point.

II. Turn the PID controller into a P controller by setting $T_{i} = \infty$ and $T_d = \infty$. Initially, gain $K_P$ is set to “0”. Close the control loop by setting the controller in automatic mode.

III. Increase $K_P$ until there are sustained oscillations in the signals of the control system, e.g. in the process measurement, after an excitation of the system. The sustained oscillation corresponds to the system being on the stability limit. The value of $K_p$ at this point is referred to as the ultimate (or critical) gain, $K_{pu}$. The excitation can be a step in the set-point, this step must be small (e.g. 5% of the maximum set-point range) so that the process is not driven too far away from the operating point where the dynamic properties of the process may be different. Also the step must not be too small, else it may be difficult to observe the oscillations due to the inevitable measurement noise.

Kushwah & Patra, (2014) proposed two variations of the Ziegler Nichols tuning method, the traditional method and the modified Ziegler Nichols method. The step response of the kind of plants in which the Traditional Ziegler Nichols tuning method is applied to is shown in figure 2-7. The response is a typical response of a first order system that has a transportation delay. Two parameters characterises the response of such systems, which are the delay time, L and the time constant, T. This points are derived by drawing a tangent to the step response at its point of inflection and identifying its intersections with the time axis and steady state value. The plant model is represented by equation 2-9.

$$G(S) = \frac{Ke^{-SL}}{Ts+1} \qquad (2.9)$$

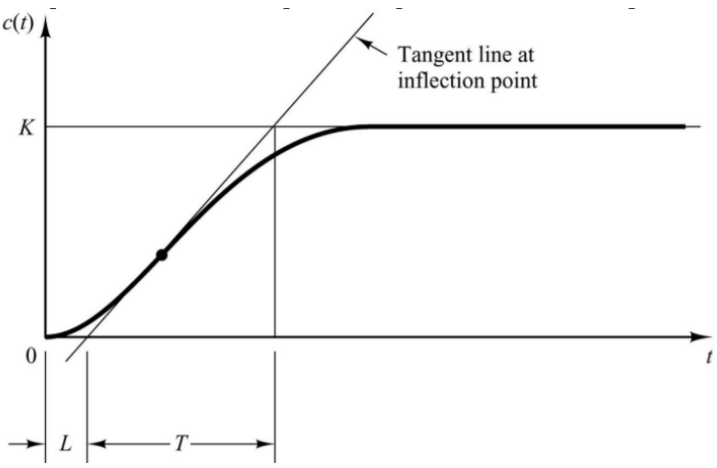


Figure 2.7. Response Curve for Ziegler Nichols Method

When the step response of the system is derived, the output signal can be noted as shown in figure 6, from this figure the parameters k, L, T can be extracted by the straight forward approach shown in Figure 6. With these parameter and the Ziegler Nichols formula shown in Table 2-2, the controller parameters of the system can be derived.

Table 2-2: Ziegler-Nichols Tuning First method

| **Controller** | **$K_P$** | **$T_i$** | **$T_d$** |
|---|---|---|---|
| P | T/L | - | - |
| PI | 0.9T/L | L/0.3 | - |
| PID | 1.2T/L | 2L | 0.5L |

The Modified Ziegler-Nichols tuning method proposed by Manoj Kushwah et al. (2014) uses Chien Hrones Reswick (CHR) tuning algorithm which emphasizes on set-point regulation. The modified Ziegler Nichols Tuning method unlink the Traditional method, uses the Time constant T of the plant explicitly. The controller tuning formulas are given in Table 2-3

Table 2-3: Modified Ziegler-Nichols tuning method

| **Controller** | **$K_P$** | **$T_i$** | **$T_d$** |
|---|---|---|---|
| P | 0.7/a | - | - |
| PI | 0.6/a | T | - |
| PID | 0.95/a | 1.4T | 0.47T |

The parameters K, L and T are derived from the response curve of figure 6. The values of Kp, Ki and Kd are derived from the table and used to form the transfer function used to model the dc motor. The results gotten from computer simulation done on MATLAB Simulink, the Modified Ziegler-Nichols tuned PID controller efficiently is better than the traditional Ziegler-Nichols tuning method, with better performance for the control objectives (i.e. Minimum rise time, Minimum overshoot, and Minimum settling time).

Walaa M Elsrogy & Naglaa K Bahgaat, (2018) proposed that, in Ziegler- Nichols tuning method only the proportional constant is varied and other two gain parameters are kept constant. Kp is increased by the factor of 2 until the system becomes unstable. The period of oscillation and gain at that point is known as ultimate period and gain. From this method the maximum overshoot Mp and settling time Ts can be derived.

Dani & Ingole, (2017) suggested that, in a PID controller, the values of $K_P$, $T_I$ and $T_d$ can be determined by using Ziegler-Nichols method. LQR is used in linear plants for obtaining optimum control. Model Predictive control (MPC) is an optimal control method used in the control of linear and nonlinear systems. Based on the process model, MPC it predicts responses of the system in moving or receding horizon.

Antanio, (2014) discussed that, Ziegler Nicholas method is straight forward method used for tuning of PID controller. Ziegler Nicholas and Modified Zeigler Nicholas tuning method are used for tuning of PID controllers for DC motor speed control. The transient response parameters such as rise time, settling time, percentage overshoot can be minimized for better speed response of the DC motor.

Walaa M Elsrogy. Naglaa K Bahgaat, (2018) explained that in Ziegler Nicholas method, speed is measured and it gives a feedback response to the system using closed loop system. Initially, the integral and derivative parameters of the PID will be set to zero. The proportional term $K_p$ are slowly increased from zero to such a point at which the system begins to oscillate continuously. The value of the proportional coefficients at this point is called the ultimate gain (Ku) and the period of oscillation at this value is known as ultimate period (Tu). When the controller is in proportional mode (i.e. integral and derivative term is set to zero), the gain of controller (Kp) has a small value which gives rise to sluggish response of the system. By increasing Kp by factor of 2 the response become oscillatory and unstable. The proportional gain is finally adjusted until the response produces a continuous oscillation.

#### 2.3.4.3 PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization algorithm was proposed by Kennedy and Eberhart in 1995. PSO algorithm was based on the social behaviour of swarms such as bird flocking and fish schooling. Here, each individual bird is called the particle, and in the case of a PID controller, each individual particle has three attributes which are $K_p$, $K_i$ and $K_d$. Sabir & Khan, (2014). In PSO, the individuals are referred to as particles which are “flown” through hyper-dimensional search space. The changes in position of each individua1 particle within the search space is based on the social-psychological tendency of individuals to emulate the success of other individuals i.e. the changes to a particle within the swarm are influenced by the experience, or knowledge of its neighbours.

The collective behaviour that emerges from this simple behaviour is that of discovering optimal regions of a high dimensional search space. The PSO algorithm maintains a swarm of particles, where each particle represents a potential solution. Let $x_i(t)$ be the position of particle *i* in the search space at time step t. The i'th particle is represented as a d-dimensional vector i.e. ($x_{i1}$, $x_{i2}$, $x_{i3\ldots}$ $x_{id}$). The position of each particle in the swarm is changed by adding a velocity, $v_i(t)$, to the current position as expressed in equation 2.10.

$$X_i(t+1) = X_i(t) + v_i(t+1) \qquad (2.10)$$

The velocity vector $v_i(t)$ drives the optimization process, and reflect both the experiential knowledge of the particle and socially exchanged information from the particle's neighbourhood. The experiential knowledge of a particle is called the *cognitive component*, which is proportional to the distance of the particle from its own best position, found since the first time step. The socially exchanged information is known as the *social component* of the velocity equation.

For global best PSO, otherwise known as gbest PSO, the neighbourhood for each particle is the entire swarm. The practices all together form a star topology kind of social network. In this type of topology the social component of the particle velocity update shows information obtained from all the particles in the swarm, which is the best position found by the swarm (collection of all the particles). The velocity update of particle i is calculated as in equation 2.8

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_2(t)[\hat{y}_j(t) - x_{ij}(t)] \qquad (2.11)$$

Where $v_{ij}$ (t) is a vector that represents the velocity of particle *i* in *j* dimensions at time step *t*, $x_{ij}$ (t) is the position of particle *i* in dimension *j* at step *t*, $c_1$ and $c_2$ are positive acceleration constants used to scale the contribution of the cognitive and social components respectively, and $r_{1j}$ and $r_{2j}$ are random values in the range of [0, 1], sampled from a uniformly distribution. These random values ($r_{1j,}$ $r_{2j}$) introduce a stochastic element to the algorithm and causes the particles to behave randomly. The personal best, $y_i$ is particle i's best position ever since the first step. As with PID tuning the aim is to minimize the gain values, therefore the personal best for particles in PSO will be the coordinates of the particles that minimizes the fitness function $f$ .This is calculated as;

$$y_i(t+1) = \begin{cases} y_i(t) & if\ f\big(\ x_i(t+1)\big)\ \geq f(y_i(t)) \\ x_i(t+1) & if\ f\big(\ x_i(t+1)\big)\ < f(y_i(t)) \end{cases} \qquad (2.12)$$

Where $f : \mathbb{R}^n = \mathbb{R}$ is the fitness function. The fitness function is a measure of performance or quality of a particle (or solution). The global best position, ŷ(t) at time step t is defined as

$$\hat{\mathrm{y}}(\mathrm{t}) = \min\ \{f(x_0(t)), \dots, f(x_n(t))\}. \qquad (2.13)$$

Arnisa Myrtellari et al, (2016) proposed an optimal control of DC motor using PSO algorithm for PID tuning. The DC motor data was imputed to the transfer function on SIMULINK, then differential equations for the model along with the state space representation was derived, also open loop transfer function and closed loop system response was derived and analysed respectively. Finally in their research, the PID for the speed control was first tuned by Ziegler Nichols PID tuning method, then the results (settling time, overshoot, rise time, steady, state error) were recorded. The same was carried out using PSO algorithm and the results were recorded. The Step response of the closed loop control system was analysed with Simulink for both the controller tuned with ZN (Ziegler Nichols) method and PSO. The performance of PSO algorithm method of tuning was proven to be a lot better than traditional method like Ziegler – Nichols method in terms of system overshoot, settling time and rise time.

Fatah & Ibrahem (2014) has explained and analysed the use of PSO for the PID controller tuning of a separately excited DC motor. The fitness function in PSO was used to minimize the steady state error, rise time, maximum overshoot and settling time. Also the Integrated square error (ISE) is which is represented as the cost function of the system was minimized. The proposed controller showed that an optimized speed response is always obtained with change in reference input speed, it also, as well demonstrates that the excellent performance of the PID controller when tuned with a PSO algorithm.

Harun Yazgan et al, (2019) discussed the performance comparison of PSO and Genetic algorithm in tuning a PID controller for a DC motor. An output signal was obtained for tuning the PID controller with parameters which based on some criteria and a unit step signal was applied as an input reference signal of the DC motor. An overshoot, a rise time, a settling time and a steady-state error were found by using MATLAB platform and the output signals were analysed at the same time. Manual tuning was first applied to the DC motor PID controller. Then PSO and GA was applied for tuning the PID. The results of these algorithm for IAE, ITAE, ISE, and ITSE for the parameters, Max overshoot, Rise Time, Settling Time, and Steady state error were substantially decreased. Applying PSO decreased the Max overshoot from 60.9959 to 25.4250 for IAE index, for ITAE index 12.9682. Also special fitness function was obtained and fitness function values ware decreased. The special fitness function, W(k) produced more satisfactory values than other fitness function in terms of overshoot, saturation time, and rise time.

In this research project, a PID controller will be analysed and implemented for the speed control of a DC motor. Ziegler Nichols and PSO tuning with be implemented for getting optimal values of the controller gain.

#### 2.3.4.4 MODEL BASED TECHNIQUE

Gucin, et al., (2015) discussed a classical Proportional Integral (PI) tuning technique which is model based in nature and requires just the motor parameters and the cross-over frequency or bandwidth of the controller to be specified the gains of the controller are achieved by pole cancellation. Although it required lot of in depth understanding about the plant and also some pre-calculation might have to be done, it is comparatively faster to set up. This method was applied to tune a cascaded PI controller for a PMDC motor. Firstly, the current loop was determined. In the case whereby the mechanical inertia of the motor is big enough, the back-EMF factor in the current loop acting as disturbance can be neglected. A simplified current loop was derived as shown in figure 2.8.

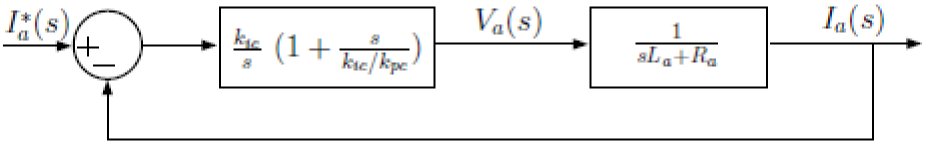


Figure 2.8: Simplified Current Loop Gucin, et al., (2015)

From the open loop transfer function of the current loop the PI controller gains was determined by equation 2-14 and 2-16.

$$\frac{k_{ic}}{k_{pc}} = \frac{1}{\tau_e} \qquad 2\text{-}14$$

$$G_{col} = \frac{k_{ic}}{s.R_a} \qquad 2\text{-}15$$

$$\frac{K_{ic}}{R_a} = \omega_{cc} \qquad 2\text{-}16$$

$k_{pc}$ = proportional gain in current loop, $k_{ic}$ = integral gain in current loop, $G_{col}$ = current open loop transfer function, $\omega_{cc}$ = crossover frequency of $G_{col}$. The speed loop was then designed in turn. The transfer function of the speed loop is given in equation 2-17

$$G_{sol}(s) = \frac{K_T\,.k_{is}}{s}\,(1+\frac{s}{\frac{k_{is}}{k_{ps}}})\frac{1/B_m}{1+\frac{s}{1/\tau_m}} \qquad 2\text{-}17$$

$\tau_m$ = mechanical time constant, $k_{is}$ = integral gain in speed controller, $k_{ps}$ = proportional gain in speed controller.

$$\omega_{cs} = \frac{k_{is}K_T}{B_m} \qquad 2\text{-}18$$

$$k_{is}k_{ps} = \frac{1}{\tau_m} \qquad 2\text{-}19$$

$$G_{sol}(s) = \frac{k_{is}.K_T}{s.B_s} \qquad 2\text{-}20$$

The mechanical time constant is given as $\tau_m = \frac{1}{B_m}$ . Equation 2-19 is used to cancel the poles of the open loop transfer function as shown in equation 2-17 which yields equation 2.20. Hence, the gains of the speed loop was calculated with equation 2-18 and 2-19 choosing the cross-over frequency of the speed loop, $\omega_{cs}$ to be up to 10 times of the current loop crossover frequency. Figure 2.9 shows the speed loop. The current loop is assumed to be ideal and equals to 1.

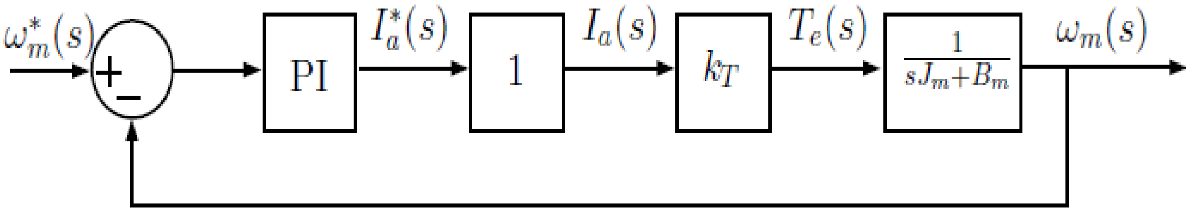


Figure 2.9: Speed Loop

## 2.4 DIRECT CURRENT TO DIRECT CURRENT (DC-DC) CONVETER

DC-DC converters are often used in regulated switch-mode dc power supplies and also in DC motor speed drive applications. Ideally, the input to DC-DC converters is an unregulated voltage gotten by rectifying the line AC voltage which is prone to fluctuation as a result of changes in the magnitude of the line voltage. As a result, switched-mode DC-DC power electronics are used to convert this unregulated dc input into a controlled dc output a desired voltage level for the motor Mohan, Undeland, & Robbins, (2003). Power electronics dc-dcc converters like Step-down (buck), Step-up (boost), Step-down/step-up (buck-boost), Cuk, and Full-bridge converters. Full bridge converters are derived from buck (step-down) converters. Figure 2.10 shows summary of how switched-mode operation is carried out in a dc-dc converter using switching operation.

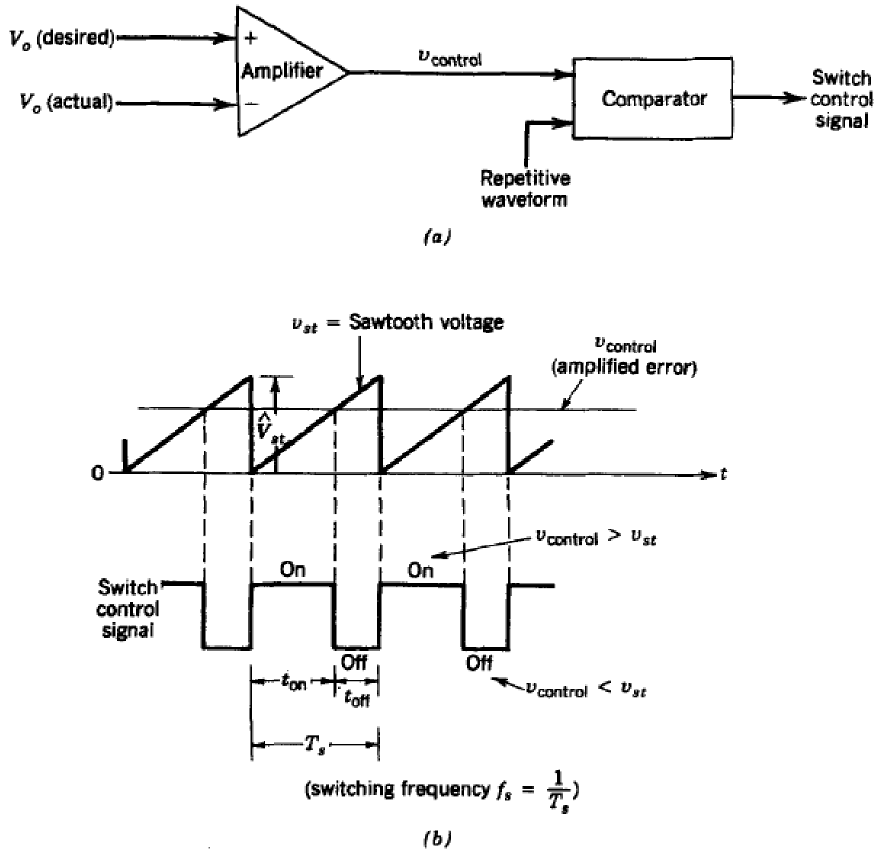


Figure 2.10: Switching Pulse Generation: (a) block diagram; (b) comparator signal

Buck converters are mainly used in dc motor speed drive applications. The basic idea of a step-down converter like a Buck converter is to produce a lower average output voltage than the dc input voltage $V_d$. A simple buck converter circuit is shown in figure 2.11 for a purely resistive load. The instantaneous output voltage is a function of the switch position which is shown in figure 2.11. The average output voltage to the load $V_0$ is derived in terms of the duty cycle to the switch, expressed mathematically in equation 2-14 and 2-19

$$V_o = \frac{t_{on}}{T_s} V_d = DV_d \qquad \text{2-17}$$

Where D is the duty ratio as can be expressed as

$$D = \frac{t_{on}}{T_s} = \frac{V_{control}}{V_{st}} \qquad \text{2-28}$$

Where $V_{st}$ is the sawtooth waveform peaks as shown in figure 2-8. Substituting D into equation 2-17 gives

$$V_o = \frac{V_d}{V_{st}} V_{control} = kV_{control} \qquad \text{2-19}$$

Where k $= \frac{V_d}{V_{st}} = constant$

These relationships shows that by varying the duty cycle ratio $\frac{t_{on}}{T_s}$ of the switch, $V_0$ (voltage to the load can be controlled. Also another observation is, by the average voltage $V_0$ varies linear with the control voltage. When the associated load connected to the buck converter is an inductive load, the switch will have to be equipped with some mean of absorbing or dissipating the inductive energy during the off position of the switch or else the switch could be destroyed. This problem is addressed by connecting a diode antiparallel to the switch as shown in figure 2.11. Also a low pass filter used to eliminate the switching frequency ripple. Mohan, Undeland, & Robbins, (2003)

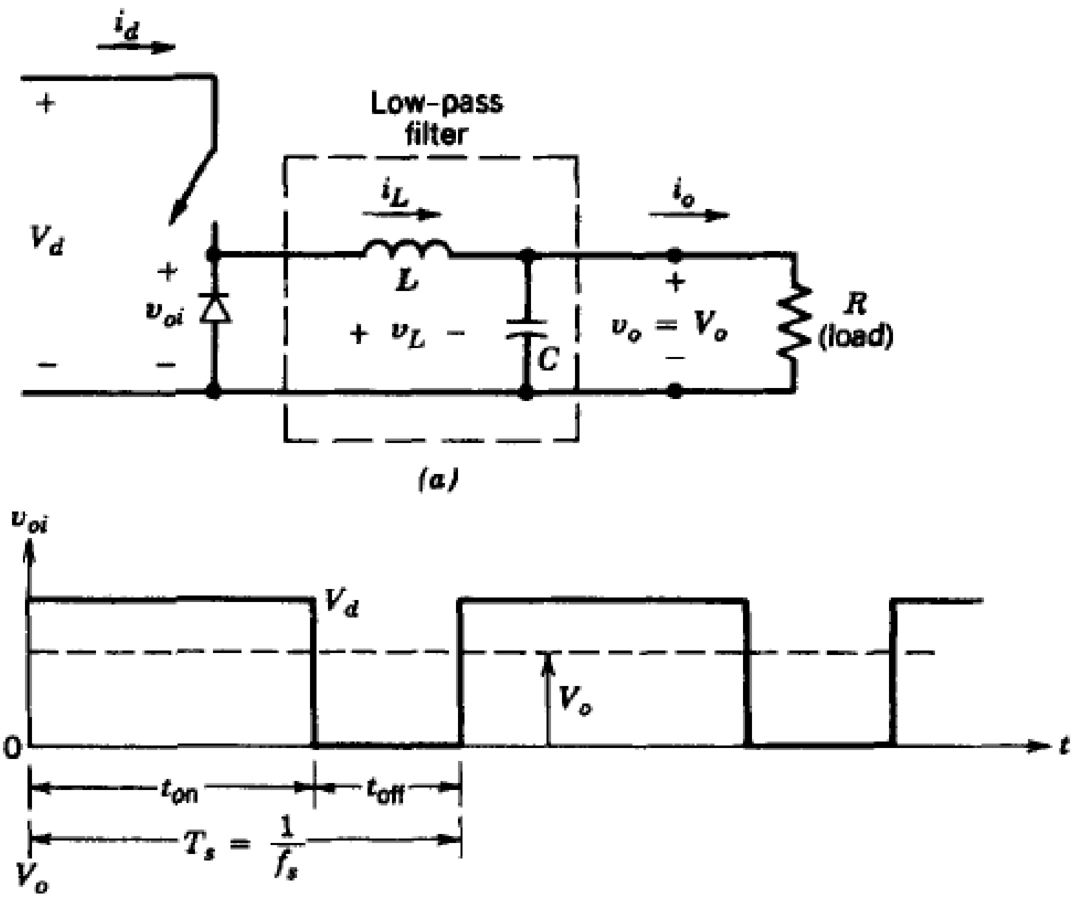


Figure 2.11: Buck Converter Configuration for a Resistive Load

Patil, Raju et al.(2016) has proposed a four quadrant closed loop speed control of a DC motor, which uses similar principle of operation as a buck converter. Their proposition offered a low-cost approach to high performance chopper based four quadrant closed loop speed control for a separately excited DC motor. The drive system was modelled and simulated using Scilab/XCOS. The control strategy used ensures that the output voltage of the dc-dc converter can be controlled both in magnitude and direction. The four quadrant converter consists of two switches on a leg. The switching scheme was made to avoid turning on both switches on the same leg which would lead to shorting the source and causing damage. Also the switching of in the dc-dc converter was done using pulse width modulation PWM. A variation of PWM technique called Unipolar Switching also referred to as double PWM switching was used. This method is described in figure 2-10. Here a high frequency triangular waveform is compared with control voltages +V$_{control}$ and -V$_{control}$ and is used for determining the switching signals for the switches on Leg 1 and 2 on the four quadrant dc-dc converter.

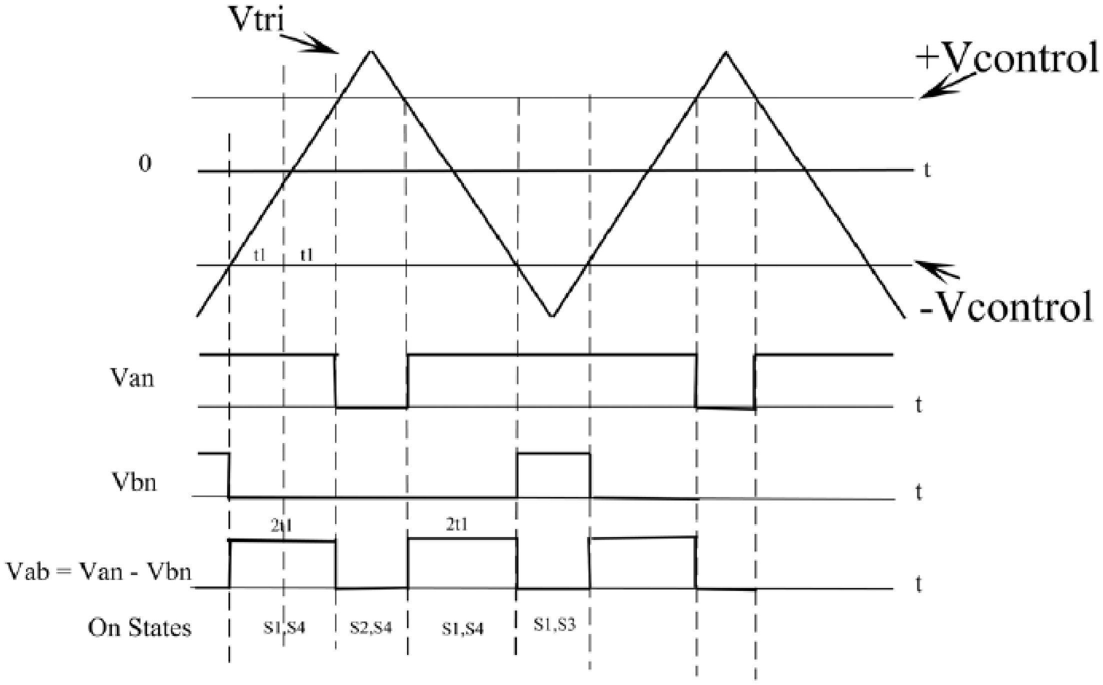


Figure 2.12: PWM Using Unipolar Voltage Switching

The DC motor mathematical model was established using the dynamic equations of the electrical and mechanical part of the motor. A PI controller with gains of 0.25 and 0.001 for the proportional and integral respectively was derived using Trial and Error tuning method which yielded the best result. The hardware implementation consisted of a 3-phase AV supply which was to a three phase bridge rectifier and a chopper circuit. The controller circuit was realized using a L28069M Launchpad with OP-amp based level shifting circuit using a LM339. The four quadrant chopper was realized by using 4 high power IGBT transistors with two connected on each arms. It was established that the four-quadrant dc-dc converter alongside the PI controller was robust for effectively controlling the speed of the motor in both anti-clockwise and clockwise direction.

# CHAPTER THREE

# METHODOLODY

## 3.1 SYSTEM DESCRIPTION

In this section the overall system is described in details. Firstly the mathematical model of the DC motor is presented, the transfer function for the Electrical and Mechanical parts of the model are modelled into a block diagram using MATLAB Simulink, mathematical derivations for the PID controller are also analysed and stated. The later part of the control is divided into two parts. The first part involves modelling the DC motor in a single speed feedback loop in which the response of the system to a step input is simulated, analysed and the PID controller is tuned for better performance. The second part involves modelling a Cascade feedback control loop for a DC motor in which the Primary loop (outer loop) is for speed control, while the Secondary (inner loop) is for current control. The cascade controller is modelled for improving load disturbance and for ensuring safe operation of the DC motor. The PID controller are tuned for optimal performance.

## 3.2 MOTOR PARAMETERS AND SPECIFICATION

The DC motor drive developed is used to drive a Permanent Magnet DC (PMDC) Machine. The Structural details and the machine parameters of the PMDC machine is gotten form a 24V DC motor datasheet by Idcmotion shown in figure 3-1 below. The wiring configuration is also shown in figure 3-2

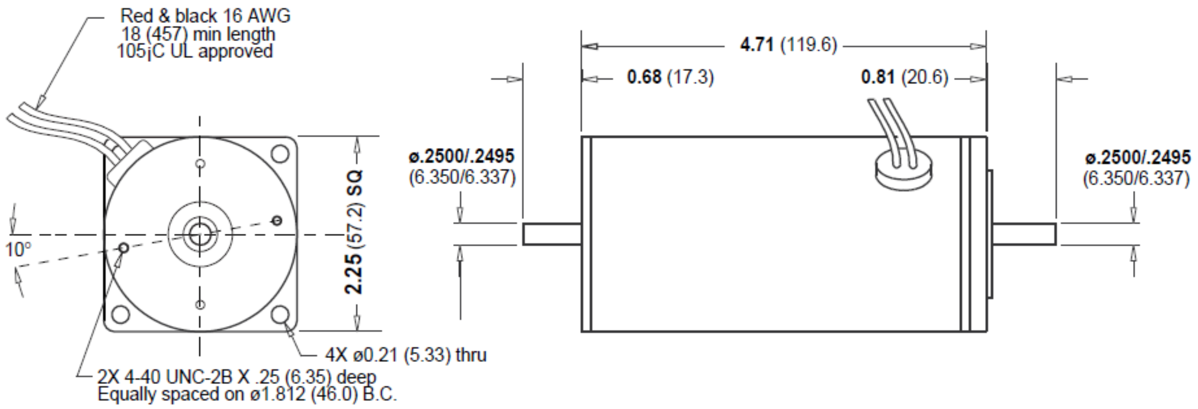


Figure 3.1: 24V DC Motor Inline (Double shaft). Source: D Motor Data Sheet, PCW-4885 http://www.idcmotion.com

The motor electrical and mechanical parameters are shown in Table 3-1 and Table 3-2 respectively.

Table 3-1 Electrical Data of a DC Electric Motor

| | | |
|---|---|---|
| Rated voltage | V | 24 |
| Max. Continuous Current | A | 4.5 |
| Max. Operating Voltage | V | 36 |
| Inductance | mH | 2.0 |
| Kt Torque Constant | Nm/A | 0.062 |
| Winding Resistance @ Ambient | ohms | 1.0 |

Table 3-2: Mechanical Data of a DC Electric Motor

| | | |
|---|---|---|
| Continuous stall torque | N-m | 24 |
| No load speed at rated voltage | RPM | 4.5 |
| No load current | A | 36 |
| Rotor Inertia | Kg-cm$^2$ | 2.0 |

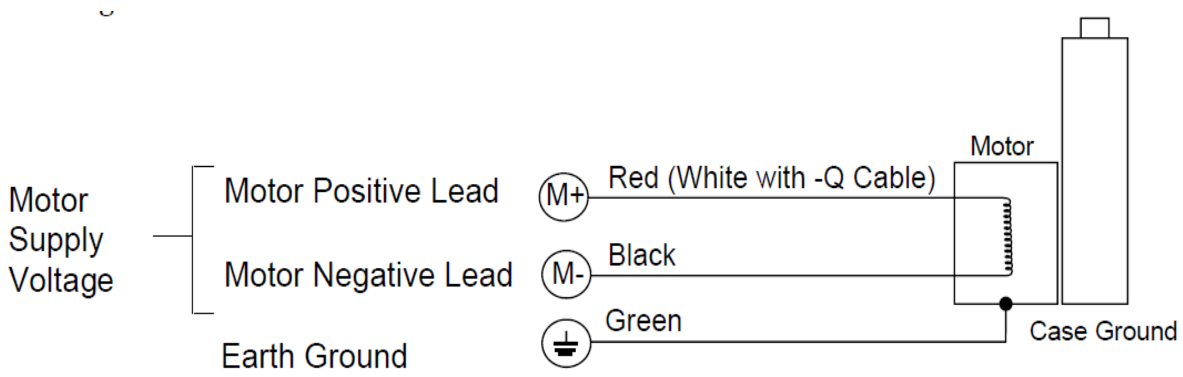


Figure 3.2: DC motor wiring. Source: D Motor Data Sheet, PCW-4885 http://www.idcmotion.com

## 3.3 DESIGN CRITERIA

The motor drive developed in this chapter should meet the following criteria

1. Rise Time <0.05seconds
2. Settling Time <0.1seconds
3. Overshoot <5%
4. Zero Steady state error
5. Current Limiting capacity
6. Fast load disturbance rejection

Basically, a DC motor consists of a synergy between the electrical part and the mechanical part as shown in figure 3-1 to cause rotation of a shaft at some desired speed or angle. In order to implement an efficient control design for any plant, a model depicting the mathematical relations between the various parts – in the case of a DC motor, the mechanical and electrical parts – of the plant are derived.

## 3.4 DC MOTOR MATHEMATICAL MODEL

Assuming a DC voltage source is applied to the terminal of a DC motor, the voltage equation according to Kirchhoff's law, of the dc motor is given as

$$L_a \frac{di_a}{dt} = V_a + R_a i_a + e_a \quad \text{3-1}$$

Where $e_a = K_f\, \omega_m$ is the back EMF of the DC motor.

The Mechanical equation (Motion equation) of a DC motor is given as:

$$J\frac{d\omega_m}{dt} + B\omega_m = \mathrm{T}_m - \mathrm{T}_L \qquad \text{3-2}$$

Where $T_m = K_f\, i_a$ is the motor torque (electromagnetic torque), the motor flux is assumed to be assumed to be constant. A schematic of the DC motor electrical and mechanical parts is shown in figure 3-3

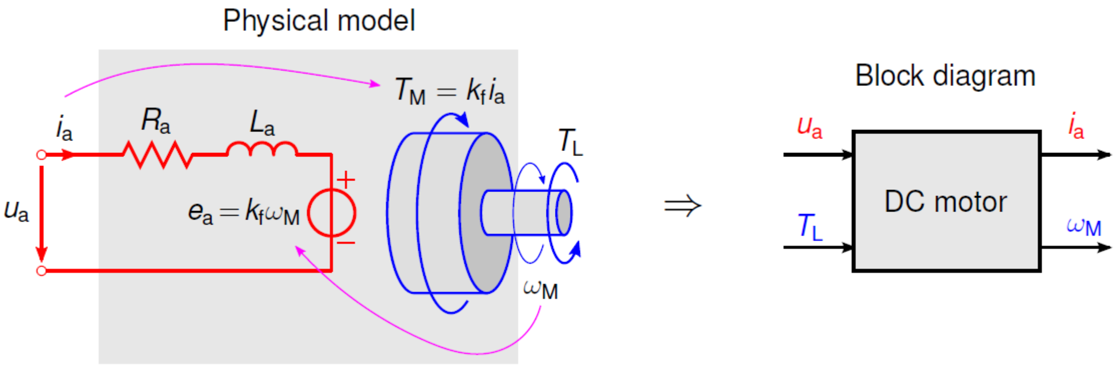


Figure 3.3: Equivalent Circuit and Block Diagram of a DC Motor - Load system

DC motors are mostly used in the linear range of the magnetization curve. In the linear range, the air gap flux Φ which is proportional to the field current i.e.

$$\emptyset \propto i_f$$

$$\emptyset \propto k_f i_f$$

Where $k_f$ is a constant

The torque generated by the motor shaft is directly proportional to the armature current and the air gap flux i.e.

$$T_m \propto \emptyset\, i_a$$

$$T_m = k_a\, \emptyset\, i_a$$

$$T_m = k_m \, i_a \qquad 3\text{-}3$$

Where k = motor torque constant

The back EMF generated by the motor is proportional to its speed i.e.

$$E_b \propto \omega \, \emptyset$$

$$E_b = \, k_b \, \omega \qquad 3\text{-}4$$

Applying Kirchhoff Voltage law (KVL), the electrical equation of the DC motor is given by,

$$V_a - \, L_a \frac{di_a}{dt} - \, i_a R_a - E_b = 0$$

$$L_a \frac{di_a(t)}{dt} + \, R_a i_a(t) = V_a(t) - E_b(t) \qquad 3\text{-}5$$

And the dynamic equation of the mechanical part of the motor, i.e. the relationship between the moment of inertia and coefficient of friction

$$T_m = \, T_l + j \frac{d\omega(t)}{dt} + b\omega(t) \qquad 3\text{-}6$$

Taking the Laplace Transform of equations (4.1), (4.2), (4.3) and (4.4)

$$T_m(s) = k_m \, i_a(s)$$

$$E_b(s) = k_b \omega(s)$$

$$sL_a i_a(s) \, + \, R_a i_a(s) = V_a(s) - E_b(s)$$

$$i_a(s)(R_a + \, sL_a) = V_a(s) - E_b(s)$$

$$i_a(s) = \frac{V_a(s) - k_b \Omega(s)}{(R_a + \, sL_a)} \qquad 3\text{-}7$$

$$T_m(s) - B\Omega(s) = Js\Omega(s)$$

$$k_m\, i_a(s) - B\Omega(s) = Js\Omega(s) \qquad 3\text{-}8$$

Substituting equation (3-5) in (3-6)

$$k_m \frac{V_a(s) - k_b\Omega(s)}{(R_a + sL_a)} - B\Omega(s) = Js\Omega(s)$$

$$k_m \frac{V_a(s)}{(R_a + sL_a)} = k_m \frac{k_b\Omega(s)}{(R_a + sL_a)} + Js\Omega(s) + B\Omega(s)$$

$$k_m \frac{V_a(s)}{(R_a + sL_a)} = (R_a + sL_a)\left(k_m \frac{k_b}{(R_a + sL_a)} + Js + B\right)\Omega(s)$$

$$k_m V_a(s) = (k_m k_b + (R_a + sL_a)(B + Js))\,\Omega(s)$$

The transfer function of the dc motor is presented by comparing the relationship between the armature input voltage and the angular velocity of the motor shaft. The block diagram of the input output relationship is shown in figure 3.4

$$G(s) = \frac{\Omega(s)}{V_a(s)} = \frac{k_m}{(R_a + sL_a)(B + Js) + kk_b} \qquad 3\text{-}9$$

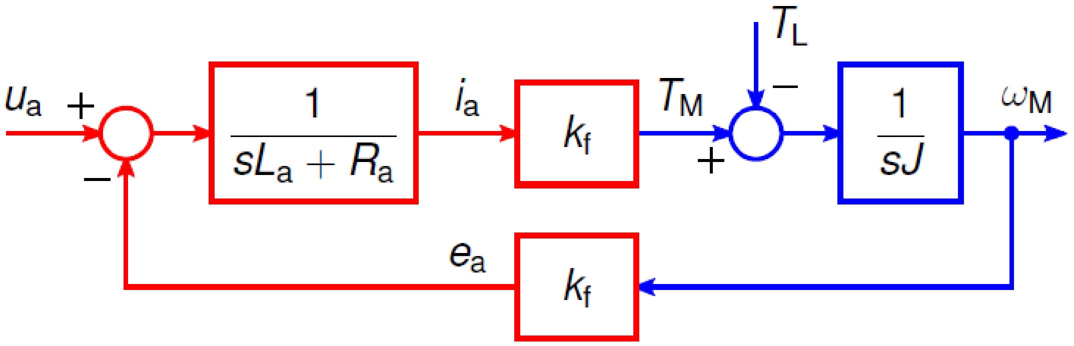


Figure 3.4: Block Diagram of a DC motor (s-domain transfer function)

**KEY OBSERVATIONS:**

1. The armature current is proportional to the armature voltage
2. The motor speed is directly proportional to the armature voltage
3. Motor torque is directly proportional to the armature current.

The DC motor is modelled in using MATLAB Simulink. The block diagram for the DC motor in open loop configuration is shown in figure 3.5.

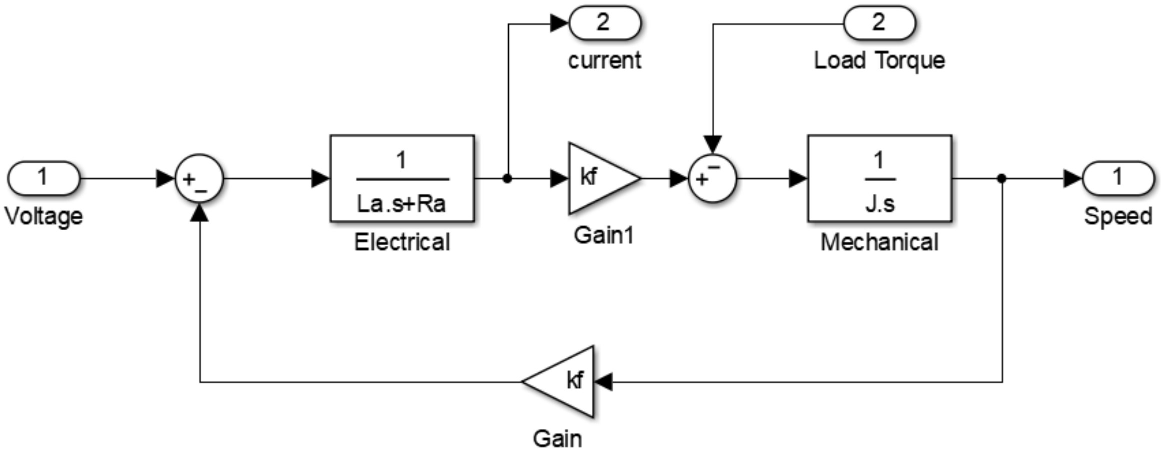


Figure 3.5: DC Motor Simulink Model

When a voltage source is simply applied to the terminals of a DC motor with no form of speed control infrastructure other than the voltage level, this action is referred to as Open loop control. But when a feedback loop is applied to this system in which the motor characteristics (speed) is being monitored and fed to back to the input, such system is a closed loop control and will be detailed in later section of this research paper.

## 3.5 OPEN LOOP SPEED RESPONSE

The open loop response to a step change in input of the DC motor transfer function is shown in figure 3-6, which shows the scope output in SIMULINK. The plot was derived from the simulation data as presented in table 3.4 and the response characteristics is given in table 3.3. Which was generated for a total of one second with sampling step of 0.001s.

Table 3-3: Primary Data of Open Loop Transient characteristics of a DC motor

| | |
|---|---|
| Rise time (seconds) | 0.7549 |
| Settling time (seconds) | 1.2193 |
| Peak time (seconds) | 1.7460 |

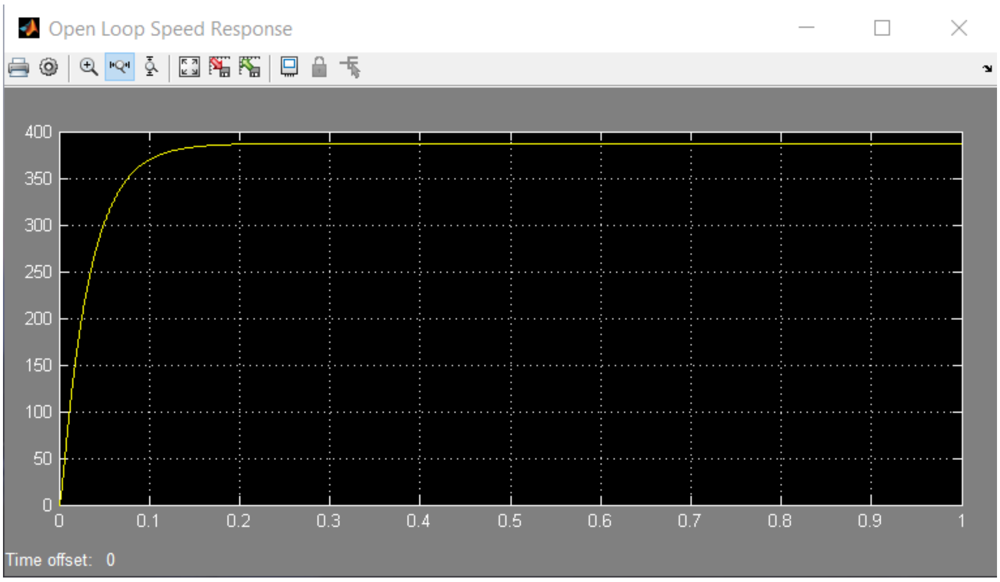


Figure 3.6: DC Motor Open Loop Response from Simulink Scope Output

Table 3-4: Open Loop Speed Data in rads/seconds

| TIME (seconds) | SPEED (rads/seconds) |
|---|---|
| **0.0000** | 0.0000 |
| **0.0010** | 11.2683 |
| **0.0031** | 34.7168 |
| **0.0050** | 55.0017 |
| **0.0082** | 86.1357 |
| **0.0147** | 142.2483 |
| **0.0304** | 239.8840 |
| **0.0711** | 348.2304 |
| **0.1265** | 378.1324 |
| **0.1819** | 385.8755 |
| **0.2917** | 387.0900 |
| **0.4482** | 387.0961 |
| **0.5128** | 387.0966 |
| **0.7616** | 387.0968 |
| **1.0000** | 387.0968 |

The starting current characteristics at no load can be seen in figure 3.7. The starting current was large for a short period, the starting current could reach more than ten time the normal operating current of the DC motor, but within an allowable range. The starting current is useful during acceleration of the DC motor. The motor current data are presented in table 3.5

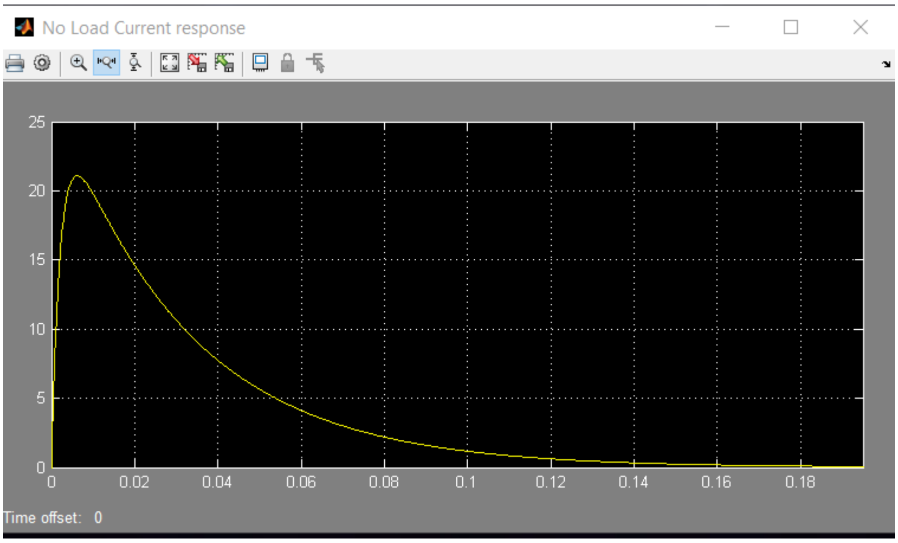


Figure 3.7: Starting Current Response of DC Motor (zoomed in)

Table 3-5: DC Motor Open Loop Response

| TIME (seconds) | Current (Amps) |
|---|---|
| **0.0000** | 23.0000 |
| **0.0010** | 23.9983 |
| **0.0031** | 22.9341 |
| **0.0050** | 21.8172 |
| **0.0082** | 19.8736 |
| **0.0147** | 16.2020 |
| **0.0304** | 9.7459 |
| **0.0711** | 2.5720 |
| **0.1265** | 0.5931 |
| **0.1819** | 0.0808 |
| **0.2917** | 0.0018 |
| **0.4482** | 0.0000 |
| **0.5128** | 0.0000 |
| **0.7616** | 0.0000 |
| **1.0000** | 0.0000 |

The transient characteristics of the DC motor speed in open loop configuration is given in Table 3.3. It evident that for effective control of a DC motor, there’s need for control strategy in form of a feedback loop for the DC motor system. The feedback control alongside a robust controller play a major role in controlling a DC motor for robust reference tracking and load rejection with good transient and steady state characteristics.

## 3.6 DESIGN OF FEEDBACK CONTROLLERS

In order to design effective controllers for a system, a robust control strategy must be implemented. For controlling the speed of a DC motor the main controller objectives of this research project are stability, reference tracking and improved load disturbance rejection. Two controller will be designed in this section. The first controller is designed uses a single PID feedback control loop for fast reference tracking, while the second controller utilizes a cascade PI control strategy, designed fast reference tracking with improved load disturbance.

A block diagram of a simple control loop is shown in figure 3-8. The system consists of two major components namely the controller and the plant which is denoted as blocks with arrows denoting the causal relationship between input and output.

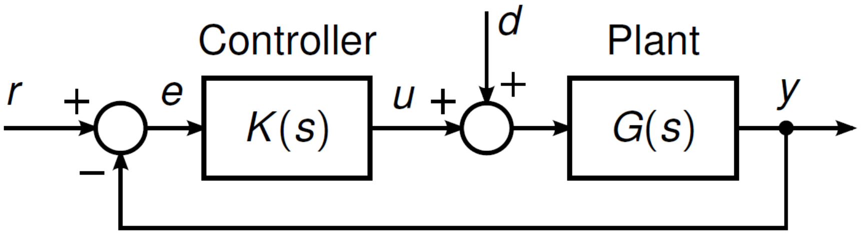


Figure 3.8: Feedback Controller Structure

The Plant has one input called the manipulated variable or the control variable, *u* as shown in figure 3.8. The Plant’s output is called the Process variable, *y*. The Process variable is measure by a sensor and feedback to the summing junction at the controller’s input. The error, *e* is the difference between the set-point and the process variable, the error is the input to the controller. The purpose of the overall system is to cause the process variable close to the reference value in spite of disturbances.

Due to critical advantages such as the functionality, simplicity, and ease of use, PID controllers have been preferred in many control applications for improving system dynamic response and reducing steady state error. The general transfer function of a parallel PID structure in the Laplace domain is given in equation 3-10.

Where E(s) is the error signal. $K_p$, $K_i$, $K_d$ are the controller gains.

$$U(s) = \left(K_p + \frac{K_i}{s} + K_d s\right) E(s) \tag{3-10}$$

## 3.7 SINGLE FEEDBACK LOOP

A single Feedback loop for the control of a DC motor is model using Simulink. The Block diagram is show in figure 3-9. The single feedback control loop has the speed error fed back to the input, the difference between the reference and the actual speed makes up the error signal and it is fed to the PID controller in order to determine more accurate control signal for effective reference tracking.

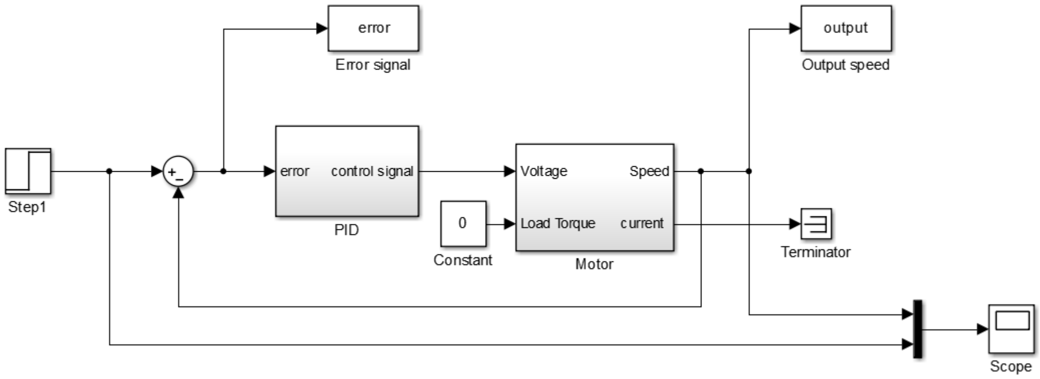


Figure 3.9: Single Feedback DC motor Control with PID Controller

### 3.7.1 TUNING PID CONTROLLER PARAMETERS

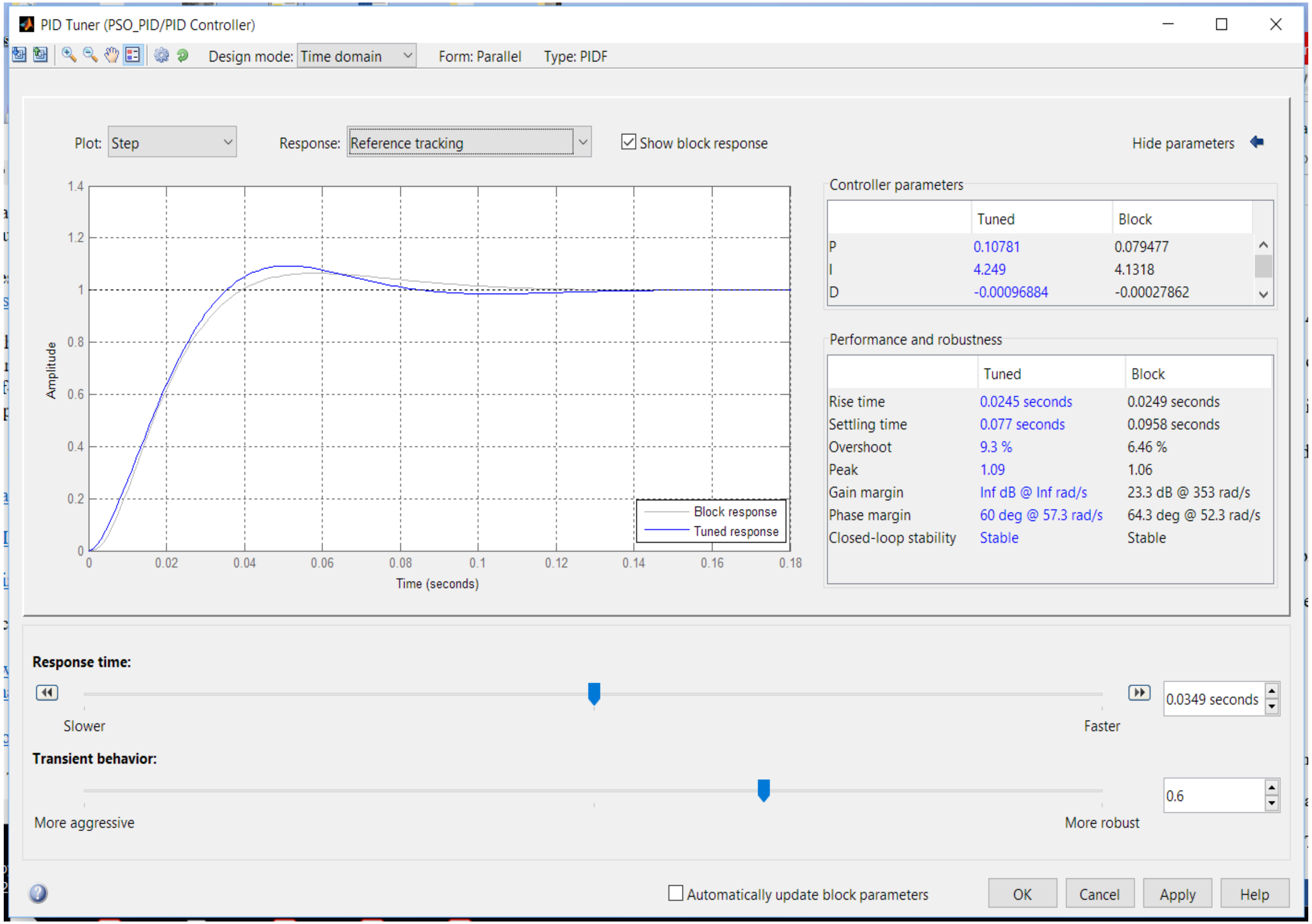
After the plant modelling and choice of PID structure has been determined, the next step in setting up the controller (PID) is tuning the controller to get the system to behave in a desired fashion. There are lot of ways to tune a PID/PI controller, a heuristic and a classical approach will be established.

MATLAB tuning and PSO tuning will be considered as a preliminary step in determining a control parameters for the single loop controller used by the DC motor drive system being developed in this research project.

#### 3.7.1.1 MATLAB PID TUNER

The first approach used to tune the single loop PID controller was done directly via MATLAB. The software provides an interactive Graphic User Interface (GUI) for automatically tuning PID controllers to achieve fast and stable results. The PID tuner automatically linearizes the plant in the model. It considers the plant to be all the blocks between the PID controller output and input excluding the PID block

An initial design than balances performance and robustness is automatically derived. Hence the tuner provides a GUI to help interactively refine the performance of the controller to meet the design requirement. Figure 3-10 shows the PID tuner GUI on MATLAB. The flexibility of determining the behaviour of the system during transient and steady state condition is utilized by toggling the slider mechanism provided in the interactive GUI. Also the desired response time of the system could be achieved using similar procedures.



Design mode: Time domain  Form: Parallel  Type: PIDF

Plot: Step  Response: Reference tracking  ☑ Show block response

Controller parameters

| | Tuned | Block |
|---|---|---|
| P | 0.10781 | 0.079477 |
| I | 4.249 | 4.1318 |
| D | -0.00096884 | -0.00027862 |

Performance and robustness

| | Tuned | Block |
|---|---|---|
| Rise time | 0.0245 seconds | 0.0249 seconds |
| Settling time | 0.077 seconds | 0.0958 seconds |
| Overshoot | 9.3 % | 6.46 % |
| Peak | 1.09 | 1.06 |
| Gain margin | Inf dB @ Inf rad/s | 23.3 dB @ 353 rad/s |
| Phase margin | 60 deg @ 57.3 rad/s | 64.3 deg @ 52.3 rad/s |
| Closed-loop stability | Stable | Stable |

Response time: 0.0349 seconds

Transient behavior: 0.6

☐ Automatically update block parameters

Figure 3.10: MATLAB PID TUNER GUI

### 3.7.1.2 PARTICLE SWARM OPTIMIZATION (PSO)

The single feedback loop demonstrated in this section will PSO for achieving optimal parameter tuning. Particle Swarm Optimization is a stochastic evolutionary optimization algorithm with is based on simulating the behaviour of swarms, like birds flocking or fish schooling. This simulation can be done by using the PSO equation that models the movement of the swarm, the position and the velocity update. This equation for velocity and position is given in equation 3-11 and 3-12.

$$V_i^{K+1} = w\,^k V_i^k + c_1 r_1 \left(P_{best}^k - X_i^k\right) + c_2 r_2 (G_{best}^k - X_i^k) \qquad \text{3-11}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \qquad \text{3-12}$$

Where *k* is the iteration progression number, *i* is the number of particle vectors in n-dimensional search space, *w* is the inertia weight which directly influences the velocity of each particle. *c1* and *c2* are the acceleration factors also known as the cognitive and social constants, *r1* and *r2* are random numbers that influences the stochastic distribution of the particle in the search space. *P<sub>best</sub>* is the best local solution of each particle, *G<sub>best</sub>* is the global solution i.e. overall best position/solution in the swarm for each iteration.

The inertia weight w defines the relationship between the past and current velocity of the swarms i.e. it affects the flying abilities of swam population to either a narrow or wide range. Larger inertia weight indicates a wider the “flying” range of the swarms (Global exploration), while a smaller inertia weight value indicates narrower “flying” range (Local exploration). Typically the inertia weight is decreased linearly from 0.9 to 0.4 depending the on the maximum and current iteration number. The formula used to calculate the inertia weight is given in equation 3-13.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \; iter_{current} \qquad 3\text{-}13$$

Steps for simulating the PSO algorithm is described below:

**Step 1.** Initialize the particles with random velocities and positions.

**Step 2.** Evaluate and compare fitness values of the particles in the population and obtain the local best value (*Pbest*) of the population for current iteration, keep the *Pbest* value in memory.

**Step 3.** Compare the *Pbest* value to global best (*Gbest*) value, which is initially assigned to *Pbest* value, and assign global best (*Gbest*) value to the position of the particle with the best fitness function value.

**Step 4.** Update the velocities of the particles by using equation 3-11

**Step 5.** Move each particle to their new position by using equation 3-12

**Step 6.** Increase iteration number, go to step 2 and repeat the steps until the stopping criterion is met.

The purpose of the controller in in feedback loop is to minimize the error produced by certain reference input. Performance criteria mostly used in optimization is based on the system error over time. The overall performance of a PSO process depends to a large amount on the objective function which ‘monitors’ the search process. This optimization function is chosen to maximize or minimize some constraints. The following optimization function (fitness function) is used to derive the optimal values for the PID gains for the single loop DC motor speed controller setup. The fitness function used is given in equation 3-14.

$$min\, F = \left(1 - e^{-\zeta}\right)\left(M_p + e_{ss}\right) + e^{-\zeta}(t_s - t_r) \qquad 3\text{-}14$$

Where:

$M_p$ = Maximum overshoot

$e_{ss}$ = Steady State Error

$t_s$ = Settling time

$t_r$ = Rise time

$\zeta$ = Weighing factor

The second fitness function is based on the inherent transient response of the feedback system. PSO optimization routine tries to find the set of controller gain that minimizes these undesired transients. The weighing factor can be varied in there steps 0.5, 1.0, 1.5.

The parameter used for the optimization algorithm is given in the Table 3-6.

Table 3-6: PSO Parameters

| **PARAMETERS** | **VALUES** |
|---|---|
| Swarm Population Size | 100 |
| Acceleration Constant, C1 | 2 |
| Acceleration Constant C2 | 2 |
| Number of Iterations | 50 |

The overall block diagram of a single loop PID controller tuned using particle swarm optimization is shown in figure 3-11.

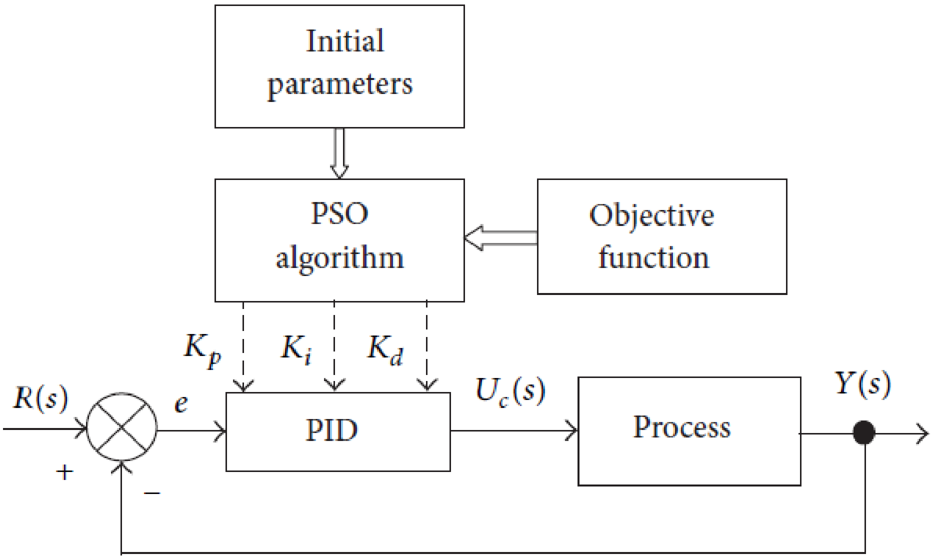


Figure 3.11: PSO-based PID Controller Structure

## 3.8 CASCADED FEEDBACK LOOP CONFIGURATION

Most DC motors used in industry are mostly subjected to some form of load for motor to do work on. Due to the inherent nature of a DC motor, application of any form of load on its rotating shaft causes speed retardation. This is undesirable when it comes to the use of DC motors for tasks that requires great precision and fast response.

PID controllers can take on more robust forms in order to offer even better and improved load rejection characteristics. A commonly used PID structure is the cascade PID configuration, which involves using more than one PID block all together. This configuration gives better dynamic performance, disturbance rejection and reduced starting current as compared to a single feedback loop configuration. Figure 3.12 shows the structure of a cascaded feedback loop control mechanism showing both the speed and current controller.

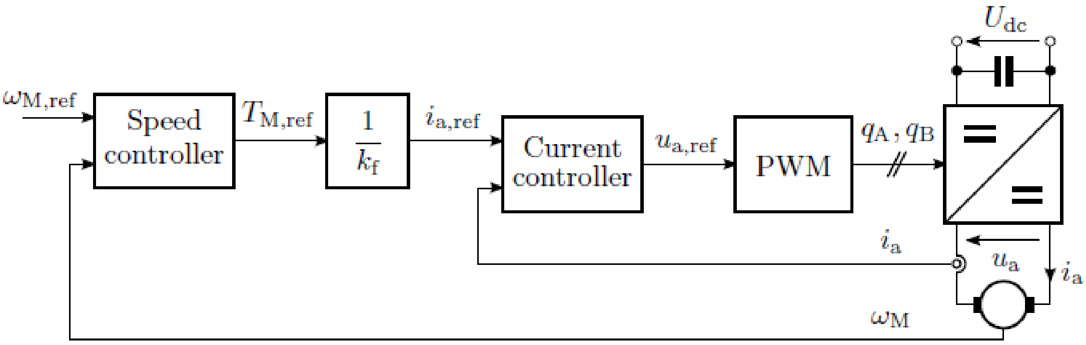


Figure 3.12: Cascaded Feedback Loops in DC Motor Drive

### 3.8.1 MODEL BASED TUNING APPROACH

Model based method of tuning provides a straightforward yet robust technique for determining the gains of the controllers based on the model parameters of the motor and the desired closed loop bandwidth of the system.

Firstly, the current loop is considered. This is the innermost loop responsible for delivering accurate current with respect to the torque requirement of the drive. The torque and current are proportional in a Permanent Magnet Dc motor (PMDC), since the field flux is constant, the armature current can be a perfect control element. The current loop is shown in figure 3.13, the load torque is neglected for simplicity.

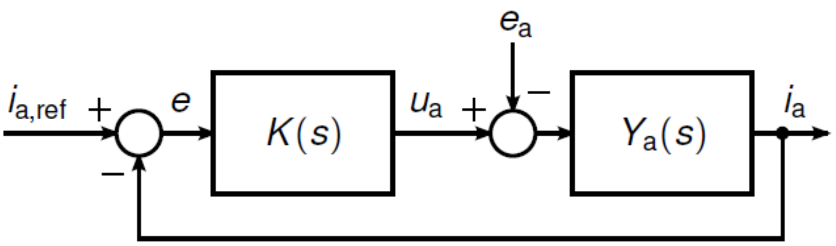


Figure 3.13: Simplified Current Loop

The closed loop current control enables current limitation and precise and fast current control. With precise current control the motor can respond to the torque demand of the load, thereby ensuring fast load rejection.

The current controller used in this configuration is Proportional Integral (PI) controller with the Laplace domain transfer function given as:

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

And the electrical equation given as:

$$Y_a(s) = \frac{1}{sL_a + R_a}$$

Therefore the closed loop transfer function is given as;

$$\frac{i_a(s)}{i_{a,ref}(s)} = \frac{K(s)Y(s)}{1 + K(s)Y(s)}$$

The desired closed-loop system is given as:

$$H(s) = \frac{\alpha_c}{s + \alpha_c}$$

$\alpha_c$ is the bandwidth of the loop (i.e. the closed loop bandwidth)

Since the desired closed loop bandwidth of the closed loop is $\alpha_c$, then the time constant of the closed loop is $\tau_c = \frac{1}{\alpha_c}$

By equating the current closed loop transfer function to the desired one, H(s), we get equation 3-15.

$$\frac{K(s)Y(s)}{1 + K(s)Y(s)} = \frac{\alpha_c}{s + \alpha_c} = K(S)Y(s) = \frac{\alpha_c}{s} \qquad 3\text{-}155$$

$K(S)Y(s)$ is the loop gain of the current loop.

The expression for the controller can be derived using the principle of Internal Model Control (IMC) a model based approach.

$$K(s) = \frac{\alpha_c}{sY_a(s)} = \frac{\alpha_c}{s}\,(sL_a + R_a) = \alpha_c L_a + \frac{\alpha_c R_a}{s}$$

Internal Model Control principle was introduced by Garcia and Morari. It involves modelling a controller based on the explicit properties of the process model. In that the controller is governed internally by the process model.

The result after solving for the controller using internal model principle is a PI controller with gains given as:

$$K_p = \alpha_c L_a \quad and \quad K_i = \alpha_c R_a$$

The bandwidth parameter $\alpha_c$ should be at least one decade smaller than the angular sampling frequency $2\pi/T_s$

In the current loop, the back EMF acts as a form of load disturbance that is inherent in the system, this disturbance causes error during acceleration i.e. during increase in angular speed of the motor shaft since the back EMF of the motor is directly proportional to the angular velocity of the motor.

The loop can be designed for improved disturbance rejection by the use of Active resistance r, which acts as a physical resistance without causing any losses as shown in figure 3-14.

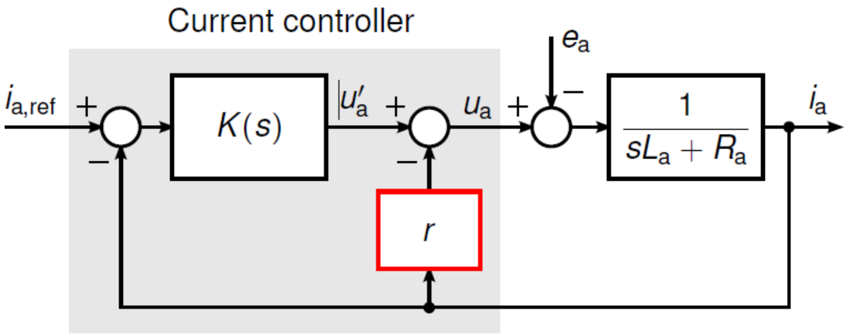


Figure 3.14: Use of Active Resistance in the Current Feedback Loop for Improved Load Rejection

Factoring the active resistance r, into the current feedback loop, the new electrical transfer function gives equation 3-16.

$$Y'_a(s) = \frac{1}{sL_a + \ L_a + r} \qquad 3\text{-}166$$

r is the active resistance.

Therefore the bandwidth of the plant $Y'_a(s)$ can be equals to $\alpha_c$ then active resistance can be calculated as follows;

$$Y'_a(s) = \frac{1/L_a}{s + \ (R_a + r)/L_a}$$

$$\frac{R_a + r}{L_a} = \alpha_c$$

Therefore, $$r = \ \alpha_c L_a - \ R_a$$

Using internal model principle, the controller parameters can be derived as follows:

$$K(s) = \frac{\alpha_c}{sY'_a(s)} = \ \alpha_c L_a + \ \frac{\alpha_c(R_a + r)}{s}$$

Hence, controller gains are given as:

$$r = \alpha_c L_a - \ R_a$$

$$k_p = \ \alpha_c L_a$$

$$k_i \ = \ \alpha_c (R_a + r)$$

The current loop is designed to have a desired bandwidth of $2\pi$×600 rad/s and the controller gains can be calculated as shown in the MATLAB code.

```
Ra = 1.0;                % Armature resistance
La = 2.0e-3;             % Armature inductance
kf = 0.062;              % Flux constant
J = 1.3e-4;              % Moment of inertia

%% Gains of the PI current controller
alphac = 2*pi*600;       % Closed-loop bandwidth
kpc = alphac*La;         % Proportional gain
kic = alphac^2*La;       % Integral gain
r = kpc - Ra;            % Active resistance
```

The internal current control loop designed makes it possible for the DC motor current to be controlled as fast as possible ensuring high performance in delivering the necessary torque required to drive the load. This in effect makes it possible for the motor current to be limited to a certain value and hence prevents the motor current from exceeding the rated value. Hence, this is done by limiting the reference current of the current control loop as shown in figure 3-15. We provide this flexibility keeping in mind that, during starting period the motor draws its stall current which is far above the max rated current of the motor. It was studied that this transient current spike only last for about 0.03 seconds from start time, although this varies with the type of DC motor being controlled. Therefore a conditional switch is made to toggle between a limited (saturated) current reference and an unlimited (unsaturated) current reference with respect to time. At time 0.03 seconds the logic switches to a limited (saturated) current reference signal ensuring safe operation of the dc motor.

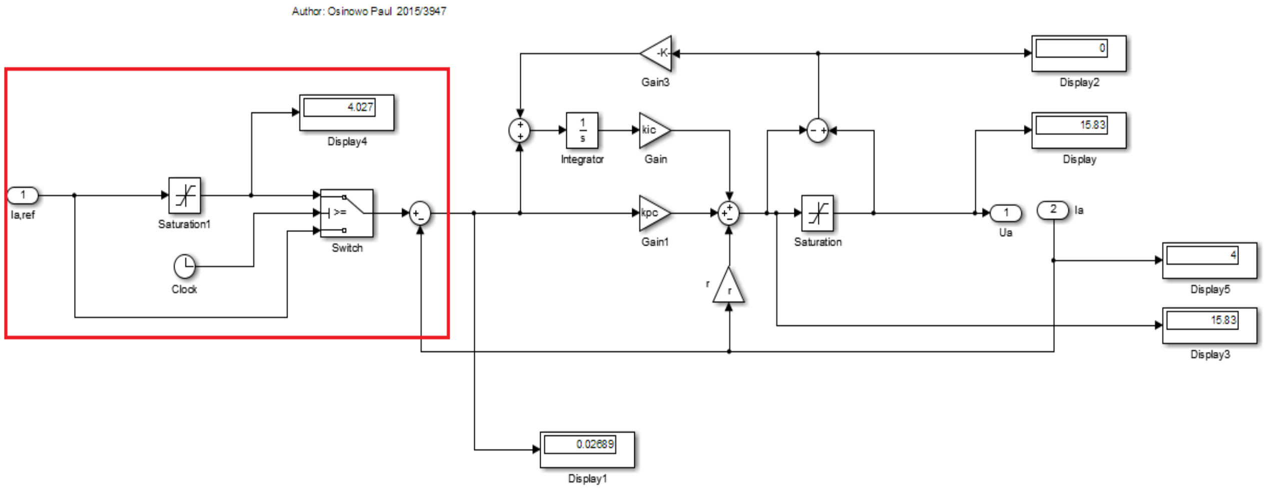


Figure 3.15: Limited Current Reference Logic in Current Control Loop on Simulink

The current is limited to 120% of the rated current of the DC motor in control. This is done to allow for quick acceleration during change in reference speed during normal operation.

Secondly, the speed controller is designed. The speed controller is cascaded with the current controller, with the later acting as the primary loop and the former, the secondary loop. The bandwidth of the speed loop (secondary loop) should be made at least ten times the current loop (primary loop) for stable coordination of the both loops.

The speed loop as shown in Figure 3-16 provides the torque reference for the inner current loop, which responds as fast as possible to supply the required amount of current in order to provide necessary torque to drive the motor to its reference speed which eliminates the errors in the system at steady state. The speed loop ensures a well damped speed response of the motor for elimination of speed overshoot. This is done with an active damping feedback from the motor output speed.

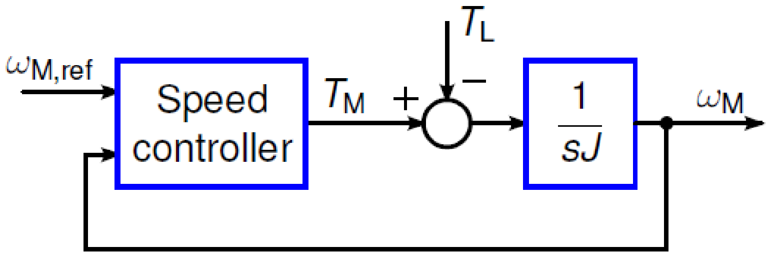


Figure 3.16: Speed Control Loop

The speed loop discussed in this section is designed with similar concepts introduced in the current loop (Anti-windup, Torque saturation, Active Damping, Internal model principle).

The current loop provides the current feedback for the DC motor while the speed loop provides the torque reference. This kind of orchestration between the current and speed loop make the cascade controller structure for DC motors very ideal for processes that require proper reference tracking and load rejection characteristics. The electrical and mechanical part of a DC motor are analogous as shown the equation 3-17.

$$\frac{1}{\mathrm{sJ}+\mathrm{B}} \overset{analogous}{\Longleftrightarrow} \frac{1}{\mathrm{sLa}+\mathrm{Ra}} \qquad 3\text{-}177$$

Therefore the following relationships can be derived

J ↔ La, B ↔ Ra , TM ↔ Va, ωM ↔ ia

The speed loop is modelled by the assumption that reference torque is equal to the motor torque i.e. neglecting the effect of the control loop. The mechanical transfer function is given in equation 3-18

$$Y(s) = \frac{1}{\mathrm{sJ}+\ \mathrm{Ra}+\mathrm{b}} \qquad 3\text{-}188$$

b = active damping

By transforming equation 3-18 to the desired format yields:

$$Y(s) = \frac{1}{sJ + B + b} = \frac{\alpha_s}{s + \alpha_s} \tag{3-199}$$

$\alpha_s$ = Speed loop bandwidth (for proper operation, this should be set to ten times the current loop bandwidth)

Solving for the speed controller using internal model principle yields equation 3.19

$$K(s) = \frac{\alpha_s}{sY(s)} = \alpha_s J + \frac{\alpha_s(B + b)}{s} \tag{3-200}$$

The speed controller parameters are therefore;

$$b = \alpha_s J - B$$

$$k_p = \alpha_s J$$

$$k_i = \alpha_s(B + b)$$

The entire cascade control system is modelled on MATLAB/SIMULINK in the continuous domain. The speed control parameters is defined as follows.

```
%% Gains of the PI speed controller
alphas = 0.1*alphac; % Closed-loop bandwidth
kps = alphas*J; % Proportional gain
kis = (alphas^2)*J; % Integral gain
b = kps - B; % Active damping
```

## 3.9 DC-DC CONVERTER DESIGN

A major component of DC motor drive is the power electronics used for transferring electrical energy to the motor in the most efficient way possible. To control the speed of a separately exited or a permanent magnet brushed DC motor, the voltage across the motor armature has to be varied accurately for robust speed control. In DC motor drive applications, dc-dc switched-mode converters are most commonly used. These drives are often fed by a rectifying an AC line voltage which is unregulated and is prone to fluctuation as its magnitude changes. Hence switched mode converters are used to convert unregulated dc input into a controlled dc output for a desired voltage level. The alternation current rectification circuit will not be covered in this research. The dc-dc converter system is shown in figure 3.17.

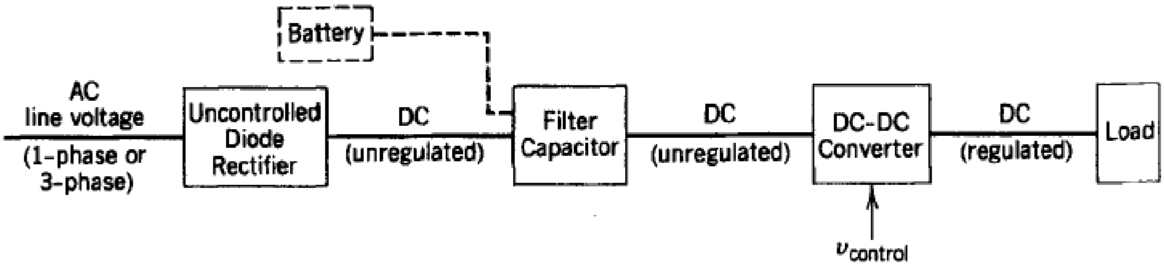
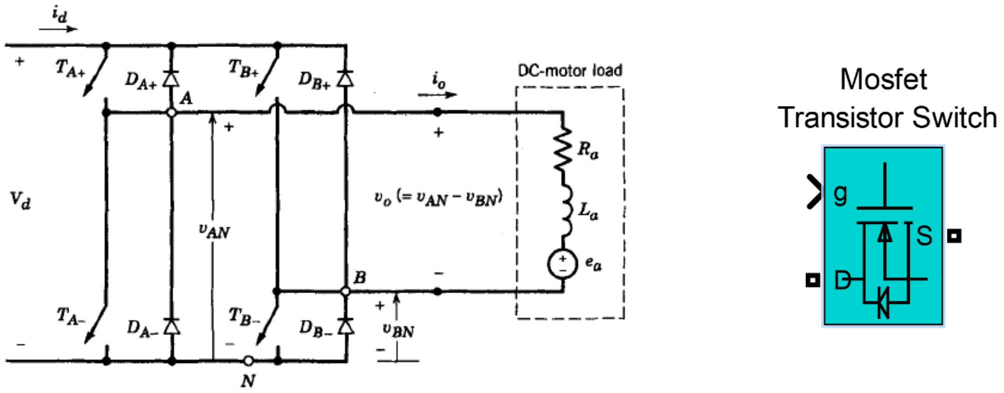


Figure 3.17: DC-DC Converter System

### 3.9.1 FULL BRIDGE DC-DC CONVERTER

Full Bridge dc-dc converter are typical for dc motor drive applications. The topology and mechanism of operation is similar to that of a dc-dc Buck converter. Schematic of a full bridge converter is shown in the figure 3-18(a).



(a)

(b)

Figure 3.18 (a) Full Bridge Schematic (b) MOSFET Switch Block on Simulink

The full bridge dc to dc converter, also known as an H-bridge is modelled and simulated on Simulink software. The input $V_d$ is a fixed dc voltage source, commonly the same as the rated voltage of the Dc motor load. The output voltage, $v_0$ is also of dc type and is varied from 0 to $V_d$ with respect to the switching period of the four switches and it can be controlled in magnitude as well as in polarity. Likewise, the magnitude and direction of output current can also be controlled, thereby giving the full bridge circuit the ability to operate in all four quadrants of the i-v plane. The switches are modelled as MOSFET (metal oxide silicon field effect transistor) switch which is shown in figure 3-18(b). MOSFETs enables high voltage fast switching with very efficiency. A diode is connect antiparallel to all the MOSFET devices on the full bridge converter as shown in figure 3.18(b). This in effect means that the switch (MOSFET) can be turned on and will or will not conduct current, depending on the direction of output current flow. The switch is said to be in conducting state when it is closed and conducting. The full bridge (H-bridge) circuit model on Simulink is shown in figure 3.19.

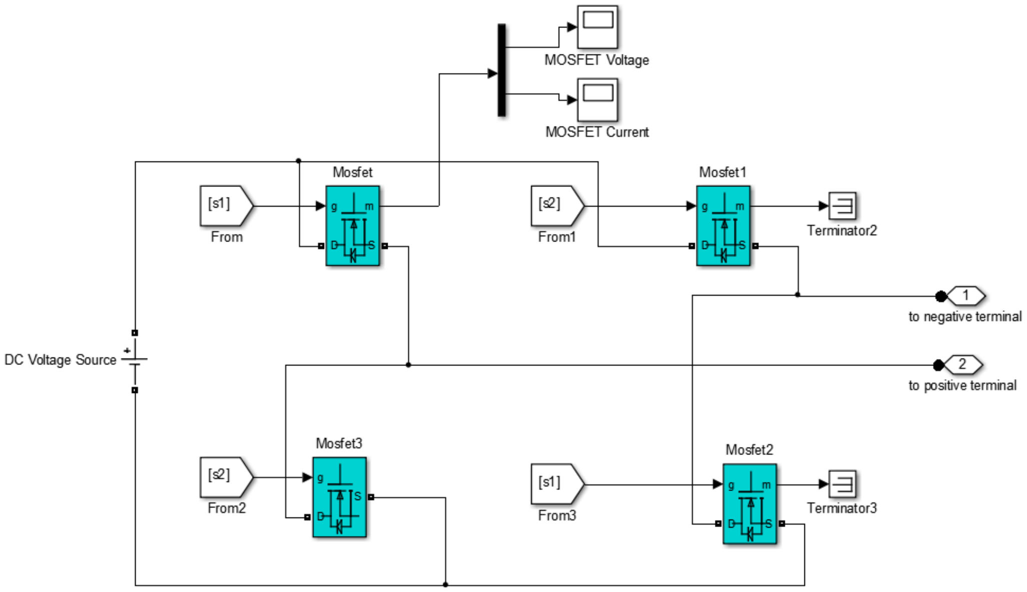


Figure 3.19: Full Bridge Converter Circuit on Simulink

There are two legs in a full-bridge circuit, left leg and right leg. The switching mechanism in both legs are done in a way that both switches are not open (off-state) simultaneously, when ideal switching is considered. This result in continuous flow of the output current to the motor load. Therefore the voltage across the dc motor connected to a full bridge circuit is solely determined by the switching states of the four MOSFET devices.

Moreover, the output voltage of both legs (legs A and leg B) averaged over one switching frequency time period $T_s$ depends only on the dc bus voltage $V_d$ and the duty ratio of $T_{A+}$ and $T_{B+}$ as shown in figure 3.18(a). This gives rise to the expression for the output voltage $V_0$ given in equation 3-21

$$V_0 = (V_{AN} - V_{BN}) \qquad 3\text{-}211$$

This in effect means the output voltage to the dc motor load can be controlled by controlling the duty cycle of the switching pulses and is independent of the magnitude and direction of the output current flow.

### 3.9.2 UNIPOLAR VOLTAGE SWITHCHING

The pulses used for the switching mechanism in a full bridge dc-dc converter is generated using pulse width modulation (PWM). Two variations to PWM exist, Unipolar switching and Bipolar switching. A unipolar voltage switching is considered in this research project.

The mechanism makes use of a triangular waveform as shown in figure 3.20 that is compared with the average duty cycle (straight line) as shown in figure 3-21, for both legs of the full bridge circuit. The duty cycle is generated by the equation 3-22 and 3-23 with respect to the control voltage coming from the current controller as modelled in section 3.5.2.

$$d_A = \frac{1}{2}\left(1 + \frac{u_{a,ref}}{U_{dc}}\right) \qquad 3\text{-}22$$

$$d_B = \frac{1}{2}\left(1 - \frac{u_{a,ref}}{U_{dc}}\right) \qquad \text{3-23}$$

The resulting pulses is shown in figure 3.22, generated from comparing the average duty cycle $d_A$ and $d_B$ with the triangular waveform is fed to the legs A and B of the full bridge circuit respectively. This process is modelled on Simulink and shown in figure 3.23. The switching frequency $f_s$ is determined by the frequency of the triangular wave carrier, and should be made much greater than the crossover frequency of the controller. The resulting Unipolar PWM pulses during simulation is shown in figure 3-22.

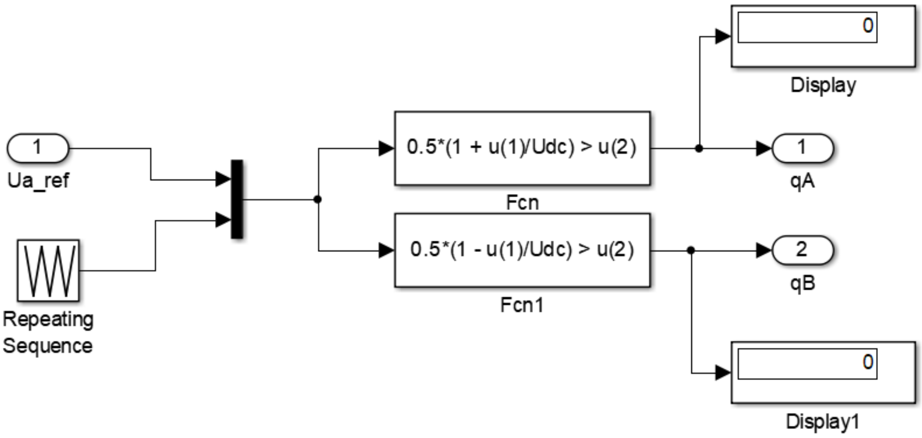


Figure 3.20: Comparator System for Pulse Generation

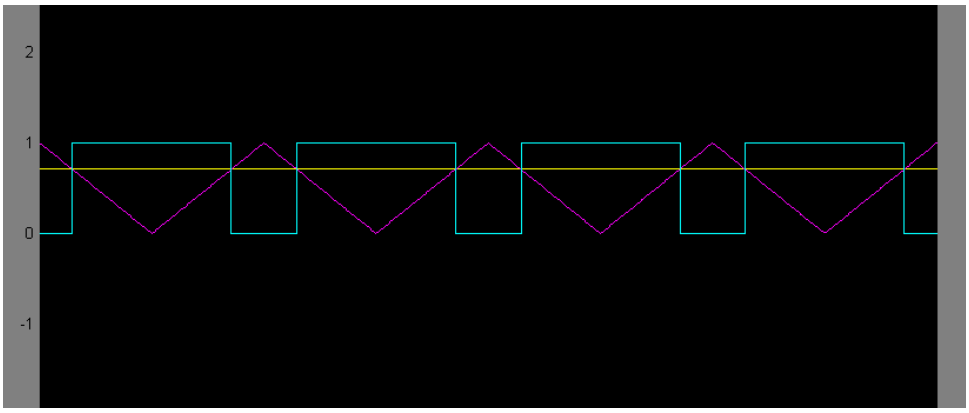


Figure 3.21: Simulink Scope showing One of Two Pulses Generated by Comparing against a Triangular Waveform

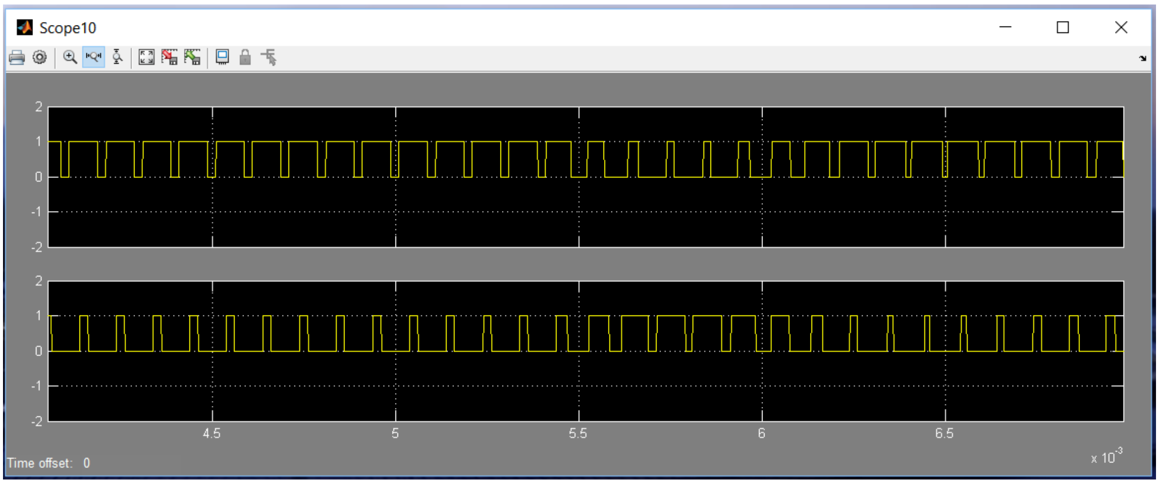


Figure 3.22: Pulses fed to both arms of Full Bridge Converter

The overall system is modelled and simulated in continuous time as shown in figure 3-23. The dc motor drive designed in this project is capable of operating all four quadrant of i-v plane. The operation and consideration will be discussed further in the following chapter. The simulation parameters are given below.

```
Ra = 1.0; % Armature resistance (Ohmns)
La = 2.0e-3; % Armature inductance (H)
kf = 0.062; % Flux constant (N-m/A)
J = 1.3e-4; % Moment of inertia (Kg-m^2)

%% Gains of the PI current controller
alphac = 2*pi*600; % Closed-loop bandwidth
kpc = alphac*La; % Current Proportional gain
kic = alphac^2*La; % Current Integral gain
r = kpc - Ra; % Active resistance
Umax = 50; % Saturation: upper limit
Umin = -50; % Saturation: lower limit
%% Gains of the PI speed controller
alphas = 0.1*alphac; % Speed Closed-loop bandwidth
kps = alphas*J; % Speed Proportional gain
kis = (alphas^2)*J; % Speed Integral gain
b = kps; % Active damping
TN = 2; % Rated torque (N-m)
Tmax = 2*TN; % Saturation: upper limit
Tmin = -2*TN; % Saturation: lower limit
%% PWM Parameters
Vdc = 24; % Full bridge converter bus voltage (V)
Tsw = 1.667e-4; % Triangular carrier wave switching period
(HZ)
```

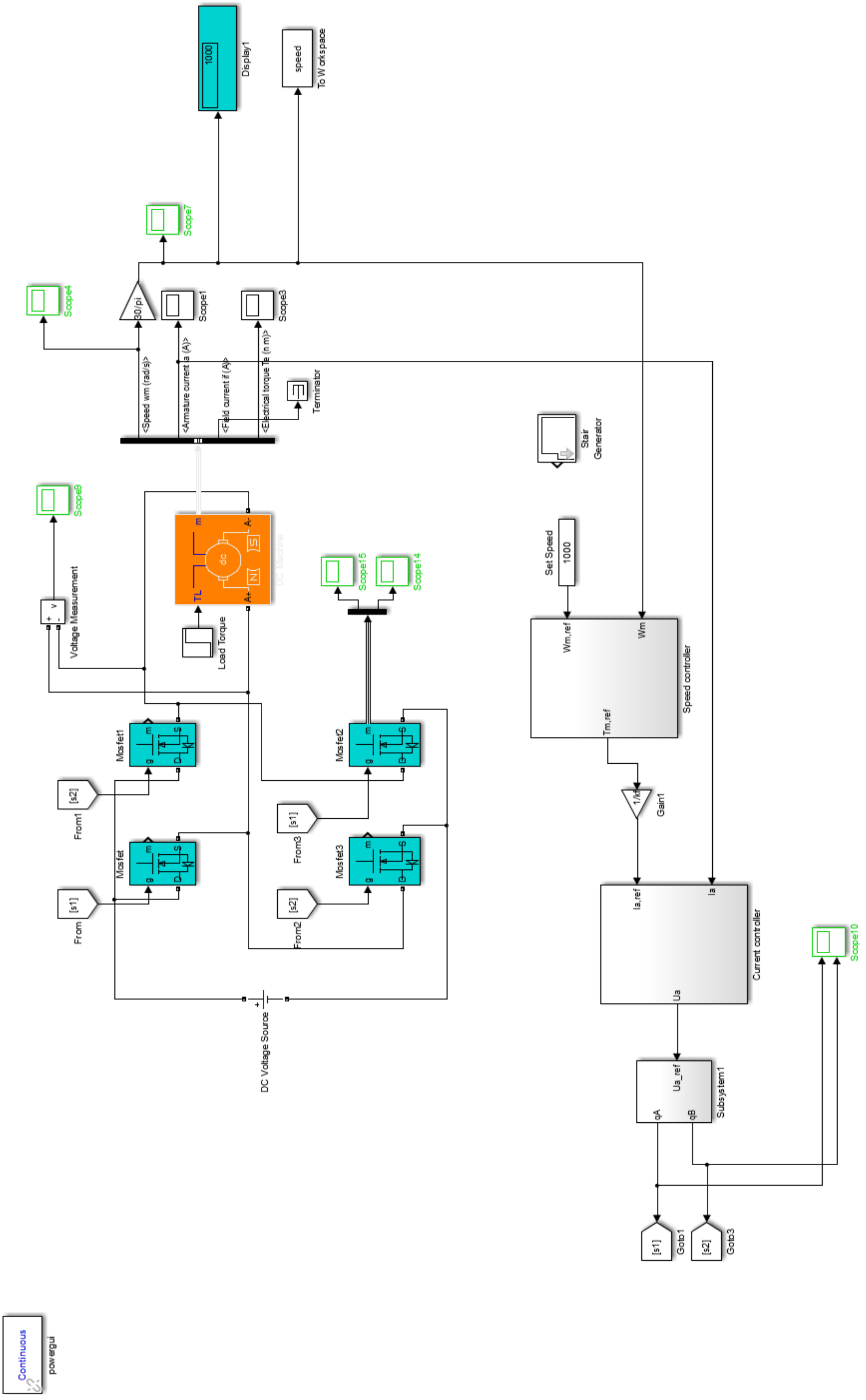


Figure 3.23: PMDC Motor Speed Drive Simulation on Simulink

# CHAPTER FOUR

# RESULTS AND DISCUSSIONS

This section organization presents the response of the controllers designed in the previous chapter. Two controllers were realized, Single loop PID controller (tuned with MATLAB and PSO) and a Cascaded loop PI controller (tuned with a Model-Based method). The PID Single loop controller was done as preliminary step in showing the effectiveness of the tuning methods used (MATLAB automatic tuning and particle swarm optimization). The cascaded PI controller was then used for the proper DC Machine drive simulation using PWM on Simulink software. The cascaded PI controller was tuned using a model based method which offers quick determination of the PI controller gain with respect to the desired bandwidth of the system.

## 4.1 MATLAB TUNED PID (SINGLE LOOP PID CONTROLLER)

The speed response of the single loop PID controller tuned using MATLAB is shown in figure 4-1. The figure shows the response to a set reference of 24 rads/s. The characteristics of the response is summarized in Table 4-1.

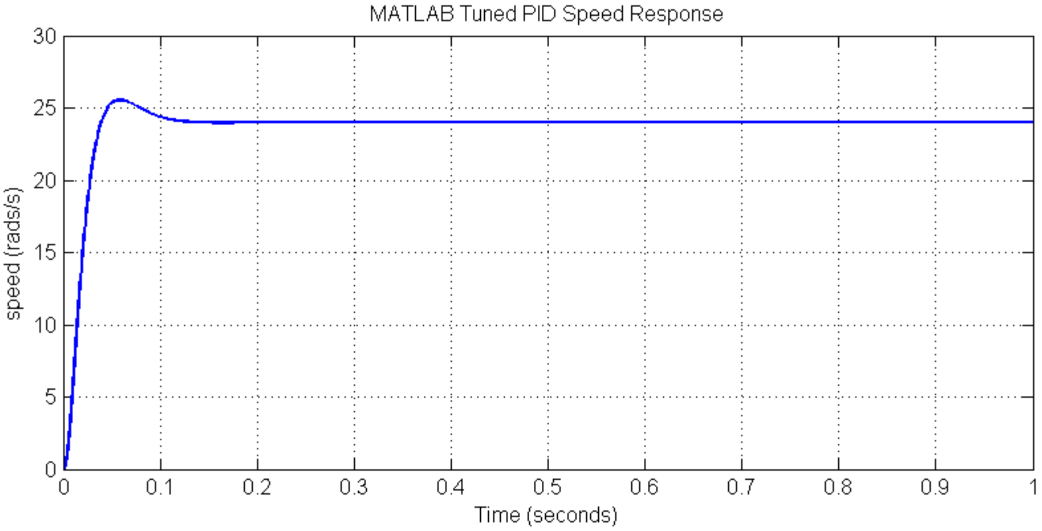


Figure 4.1: Speed Response of MATLAB Tuned PID

Table 4-1: Speed Response Data of MATLAB tuned PID

| TIME (seconds) | SPEED (rads/seconds) |
|---|---|
| **0.0000** | 0.0000 |
| **0.0001** | 0.0023 |
| **0.0006** | 0.0548 |
| **0.0029** | 1.0326 |
| **0.0077** | 4.6844 |
| **0.0138** | 9.7236 |
| **0.0223** | 15.8163 |
| **0.0323** | 21.3126 |
| **0.0384** | 23.1899 |
| **0.0526** | 25.5779 |
| **0.0624** | 26.1331 |
| **0.0688** | 25.8730 |
| **0.1229** | 24.2421 |
| **1.0000** | 24.0756 |

Table 4-2: Unit Step Response Characteristics of MATLAB tuned PID

| | |
|---|---|
| Rise Time (seconds) | 0.0249 |
| Settling Time (seconds) | 0.0958 |
| Overshoot (%) | 6.46 |
| Gain Margin | Inf db @ Inf rads/s |
| Phase Margin | 180 deg @ 41.9 rads/s |
| Steady state error | 0 |

## 4.2 PSO TUNED PID (SINGLE LOOP PID CONTROLLER)

Tuning the PID single loop controller was done using particle swarm optimization (PSO), which is an evolutionary algorithm for providing optimal solution to optimization problems when the algorithm is well configured. Figure 4-2 shows the response of the system when exited with a step change in reference speed from 0 to 24rads/s.

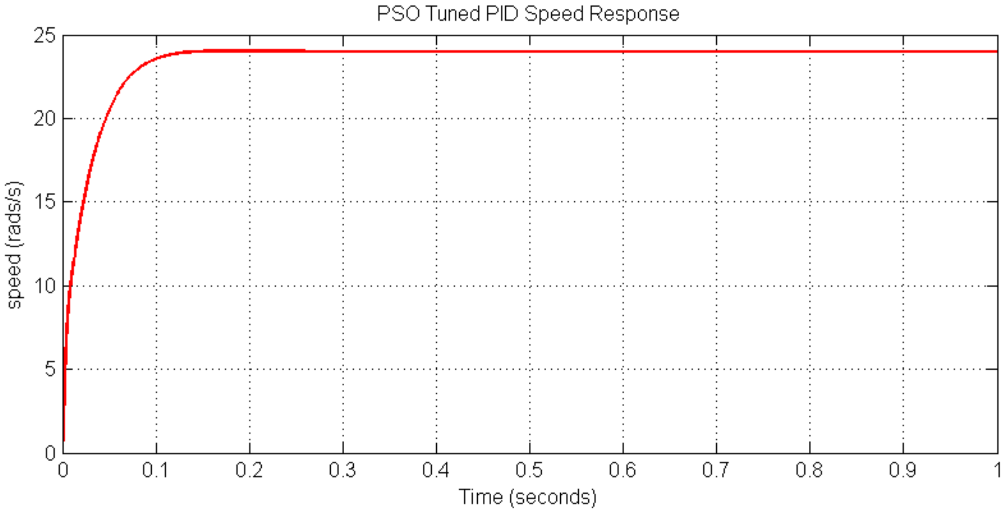


Figure 4.2: PSO-PID Response

Table 4-3: Current Response Data for MATLAB tuned PID

| TIME (seconds) | SPEED (rads/seconds) |
|---|---|
| **0.0000** | 0.0000 |
| **0.0002** | 0.0113 |
| **0.0041** | 3.3246 |
| **0.0093** | 8.9411 |
| **0.0156** | 13.3841 |
| **0.0243** | 16.7538 |
| **0.0357** | 19.3242 |
| **0.0434** | 20.1859 |
| **0.0510** | 20.9069 |
| **0.0604** | 21.7288 |
| **0.0861** | 23.3479 |
| **0.1391** | 24.0702 |
| **0.7042** | 24.0782 |
| **0.8972** | 24.2081 |
| **1.0000** | 24.1109 |

The transient characteristics of the PSO-PID are summarized as follows in Table 4-2

Table 4-4: Unit Step Response Characteristics of PSO Tuned PID

| | |
|---|---|
| Rise Time (seconds) | 0.0585 |
| Settling Time (seconds) | 0.098 |
| Overshoot (%) | 0.111 |
| Gain Margin | Inf db @ Inf rads/s |
| Phase Margin | 180 deg @ 41.9 rads/s |
| Steady state error | 0 |

An indicator of proper functioning of the algorithm is shown in figure 4-3. This plot shows the minimization of the fitness function which is a function of the DC motor speed's steady state error, rise time, settling time and overshoot. The downward trend in the plot shows that that the algorithm helps in improving the transient characteristics of the DC motor response by minimizing the fitness function, hence better PID gains for the single loop controller to efficiently control the DC motor.

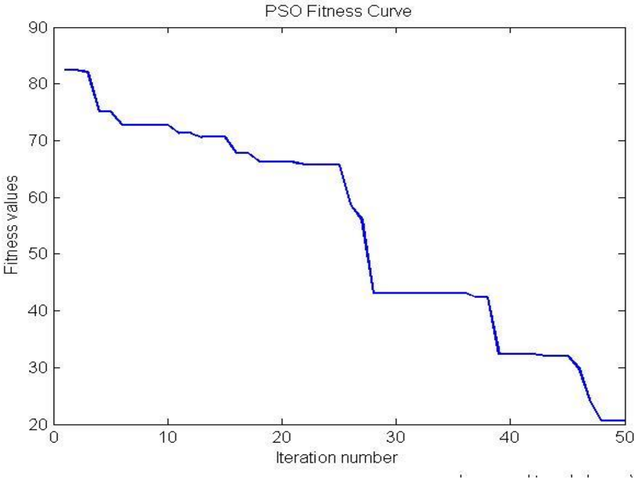


Figure 4.3: Plot of PSO Fitness value against Iteration Number

The comparison between the response of the MATLAB-TUNED and PSO-TUNED PID Controllers is shown in figure 4-4

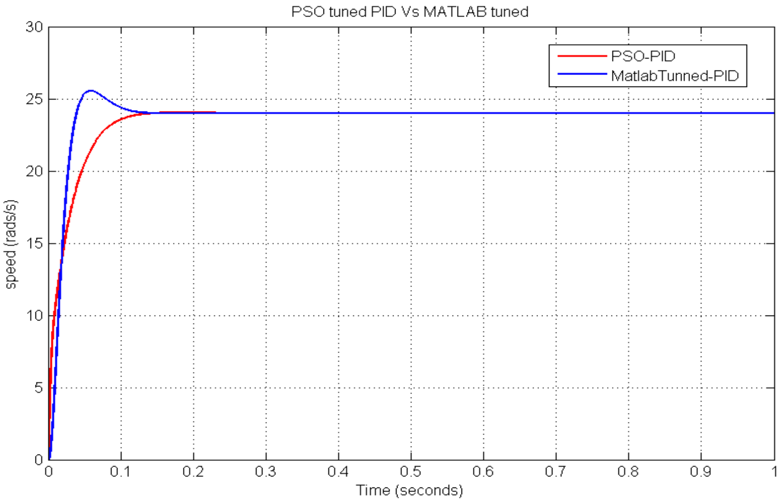


Figure 4.4: MATLAB tuning VS PSO tuning

## 4.3 DC MOTOR SPEED DRIVE (CASCADE PI CONTROLLER)

After the single loop PID controller structure has been studied the cascaded PI controller are then used for designing the DC Motor Speed Drive. The results gotten are presented and discussed in this section. Before presentation and discussion of the results gotten from the simulation of the DC motor drive developed in chapter 3, we present a verification result to show that the DC motor model used in the simulation matches the behaviour of the subject DC motor being studied as specified in the motor's datasheet.

The DC motor is connected to its rated voltage of 24 V and a load torque equal in magnitude to that of the continuous stall torque (0.28 Nm) given in the motor datasheet is applied to the motor. The open loop speed, current and torque response is shown in figure 4-5

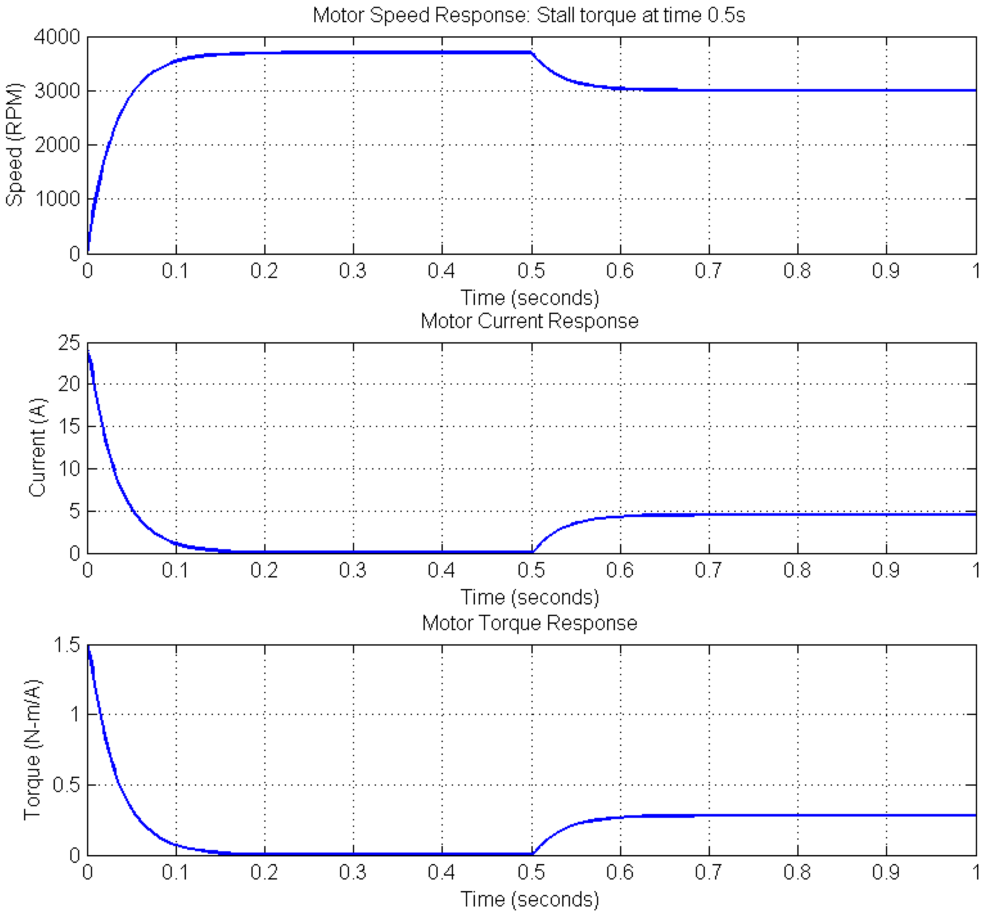


Figure 4.5: DC motor Verification Response

As shown in figure 4-5, the speed of the motor ramps up to approximately the rated speed given in the motor datasheet (3600 RPM). When the continuous stall torque is applied as a load torque, the speed reduces to approximately 3000 RPM and the motor draws a current of approximately equal magnitude to that of the maximum continuous current (4.5A) given in the motor datasheet. This current is required to sustain the load within safe limit.

Hence, the DC motor can be properly driven by the developed speed drive taking into consideration all the necessary ratings and safe limits of the motor. The response of the DC motor when controlled by a Cascade PI Controller with a staircase reference speed of 1000, 1500 and 2000RPM is shown in figure 4.6.

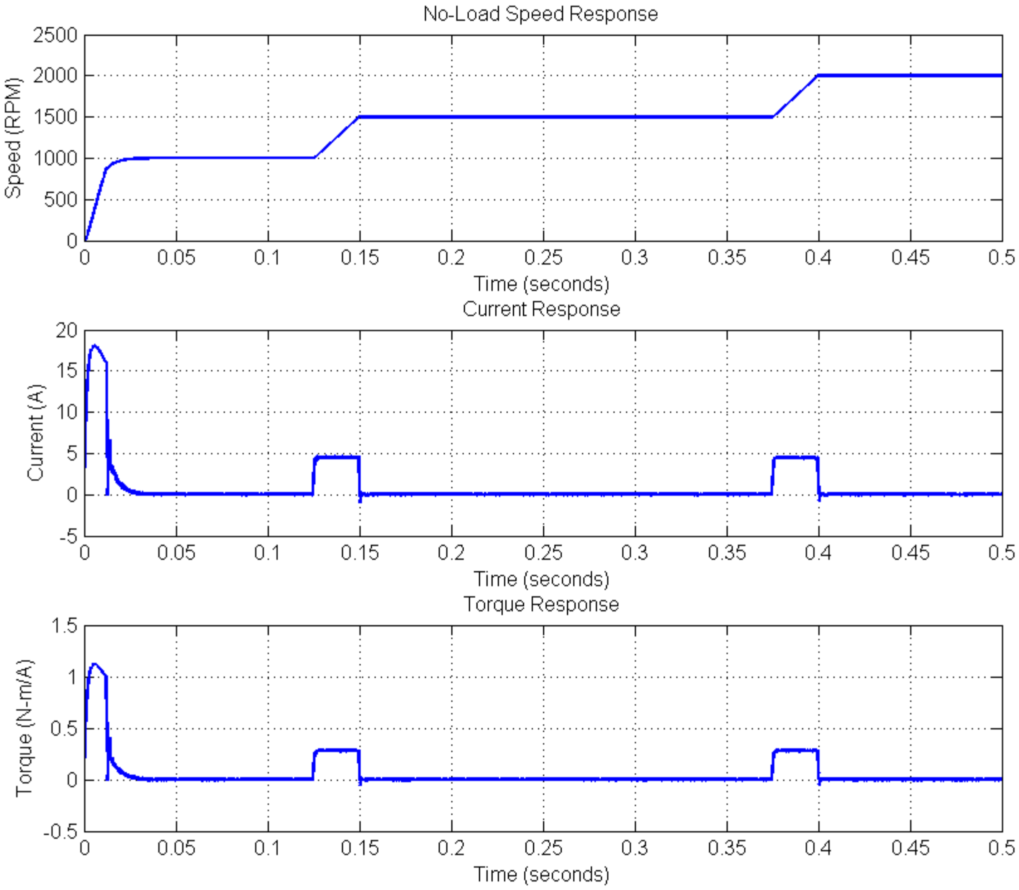


Figure 4.6: No-load Response to Staircase Input of 1000, 1500 and 2000 RPM

According to the motor datasheet, the continuous stall torque is given as 0.28 Nm. To study the load disturbance rejection capacity of the DC motor speed drive, a load torque of 0.25Nm is introduced to the DC motor at time 0.1s when motor speed is 1000RPM, the speed response is shown in figure 4.7, and the graph is zoomed in to show the impact of the load toque on the motor speed.

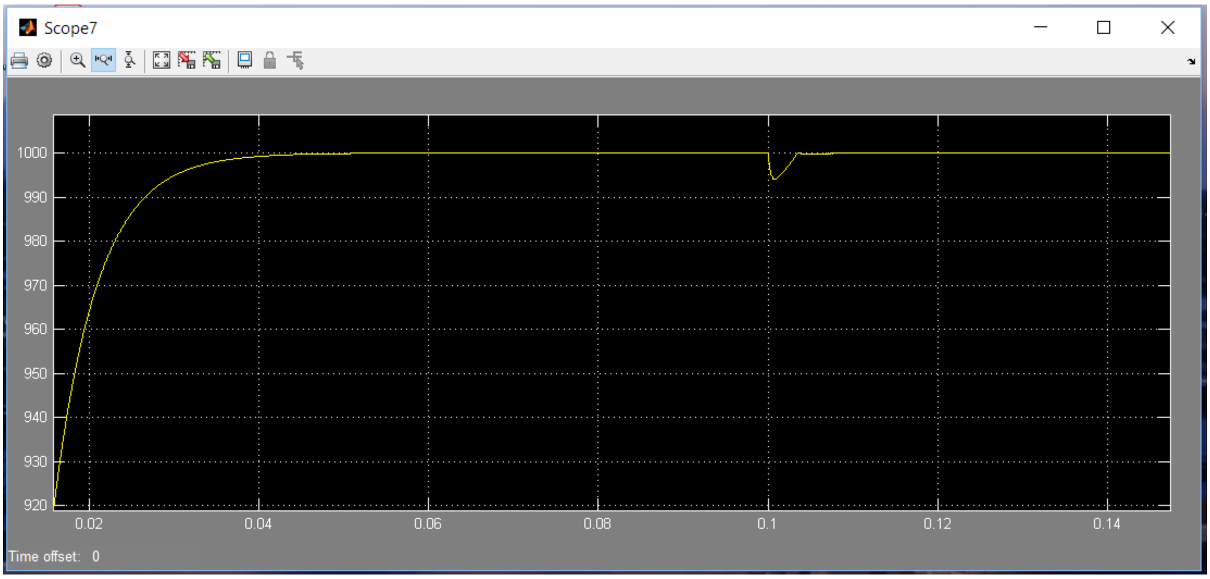


Figure 4.7: Load Rejection Capacity of Speed Drive to Load Torque of 0.25 Nm from Simulink Scope Output

The y-axis and x-axis in figure 4.7 represents the speed (RPM) and time (seconds) respectively. The load causes the motor shaft speed to drop to approximately 994 RPM, then the speed drive ensures the motor regains its reference set speed of 1000RPM in 0.004seconds. The graph showing the motor current during this phase is shown in figure 4.8. The motor supplies more current to compensate for the additional demanded shaft torque imposed on it by load disturbance.

The load torque of 0.25 Nm is below the continuous stall torque of the dc motor given in its datasheet. This means load torque of 0.25 Nm is within the safe operating range of the motor.

The high current drawn by the motor at starting period are possible and are allowed. The current spike is due to the inherent nature of the dc motor at stall condition i.e. speed = 0. At this point the largest current flows through the motor, which is necessary for acceleration purposes during starting. It is also due to the torque-speed characteristics of dc motors. It is possible to significantly exceed the current and torque limits on a short term basis, but at steady state operation i.e. during normal ON time of the dc motor, the current and torque is expected to stay within the rated limits, for safe continuous operation.

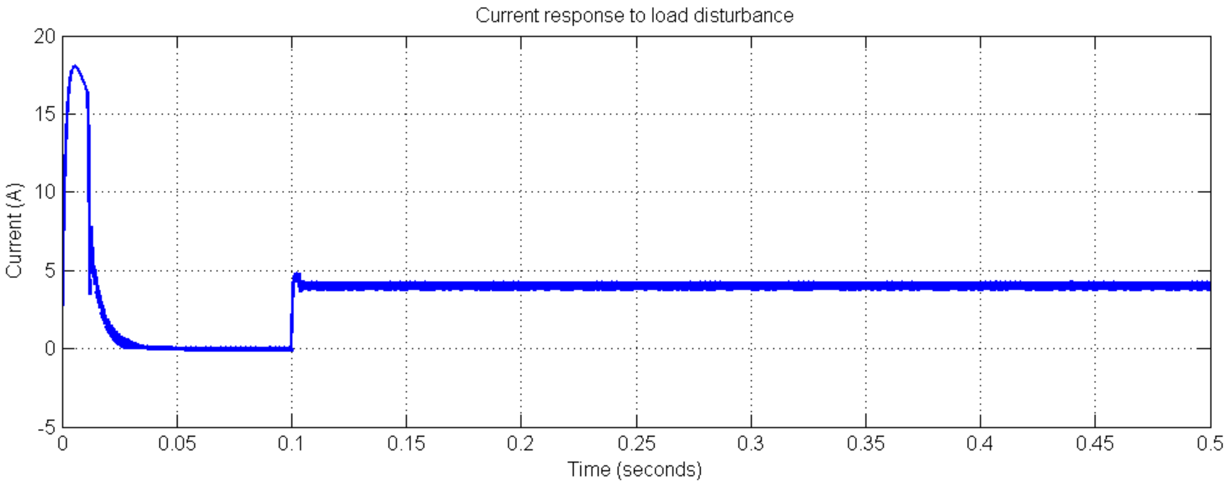


Figure 4.8: Current response to load disturbance of 0.25 Nm

The cascade controller used in the dc motor speed drive provides the flexibility of a current and torque loop for effectively controlling the dc motor. In the current loop, a current limiting feature was implemented in chapter 3. The result of this feature is simulated and tested by adding a load disturbance to the motor. The magnitude of the load disturbance is done such that it is higher than the rated continuous torque of the DC motor as specified in the motor’s datasheet. A load torque of 0.8 Nm is applied at time 0.05 seconds when the motor reference speed is set to 1500 RPM. The current response of the motor during this period is simulated and shown in figure 4.9

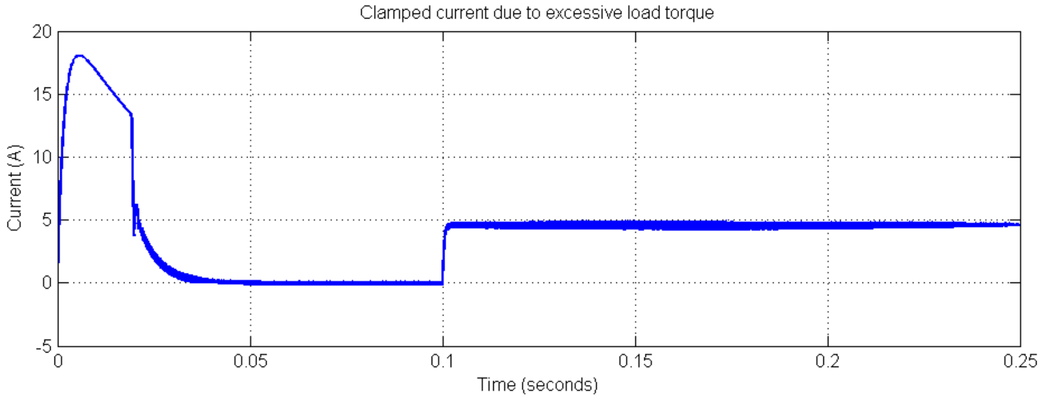


Figure 4.9: Current Limited response of DC motor

Another situation where the motor current can exceed its continuous current limit is when there is sudden drop is speed reference of a fully loaded motor i.e. during breaking. Current clamping also helps to prevent excessive motor current during this period. This is simulated on Simulink by loading the motor with a torque of 0.28 Nm (rated continuous torque) at 0.04 seconds from the starting time of the motor when the motor is running at 1500 RPM. The reference speed is suddenly dropped from 1500 to 200 RPM while the motor is loaded. The speed and current response of the motor during this period is shown in figure 4-10.

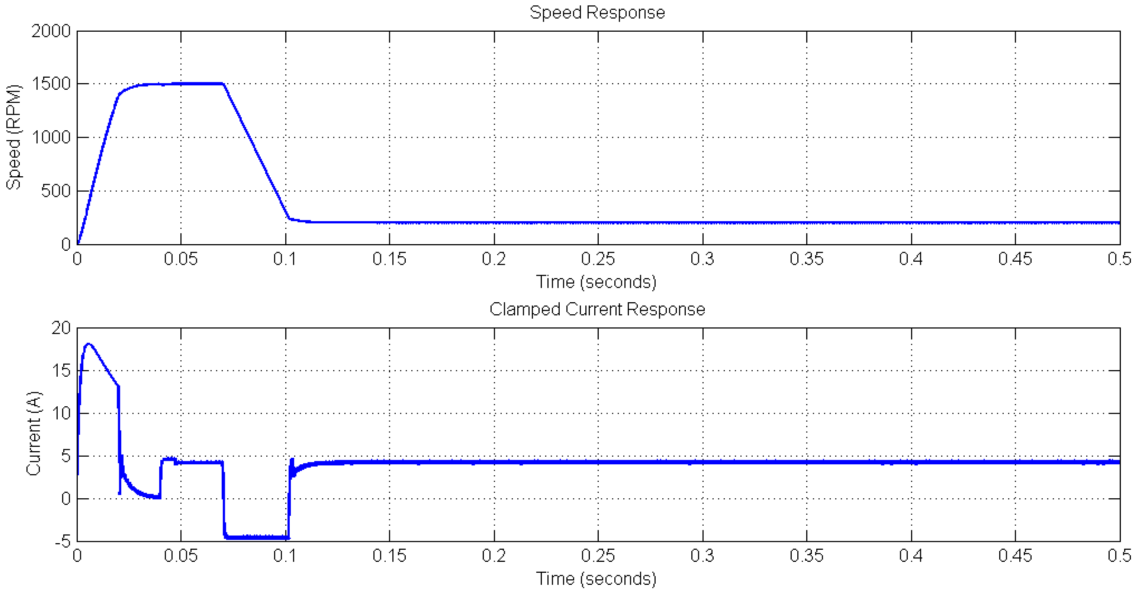


Figure 4.10: Motor response to sudden reduction in reference speed while keeping motor current clamped

It can be observed from figure 4-10 that at time 0.07 seconds the motor performs a breaking action which occurs in the 2nd quadrant (breaking in forward direction) of the speed-current graph, in which the motor acts as a generator, with negative current flow. Ideally, dropping the reference speed causes large negative current to flow across the motor terminal which at times exceeds the motor's rated current. The current clamping implemented helps to keep the negative current within save limit of -4.5A shown in figure 4.10.

The result for the motor running in both clockwise and anti-clockwise direction is simulated. The full-bridge dc-dc converter used for supply voltage to the motor enables four quadrant operation which makes it possible for the motor to work in both direction. Reference speed of 1000 RPM and -1000RPM is used for the simulation and the motor's speed response and current response is shown in figure 4.11.

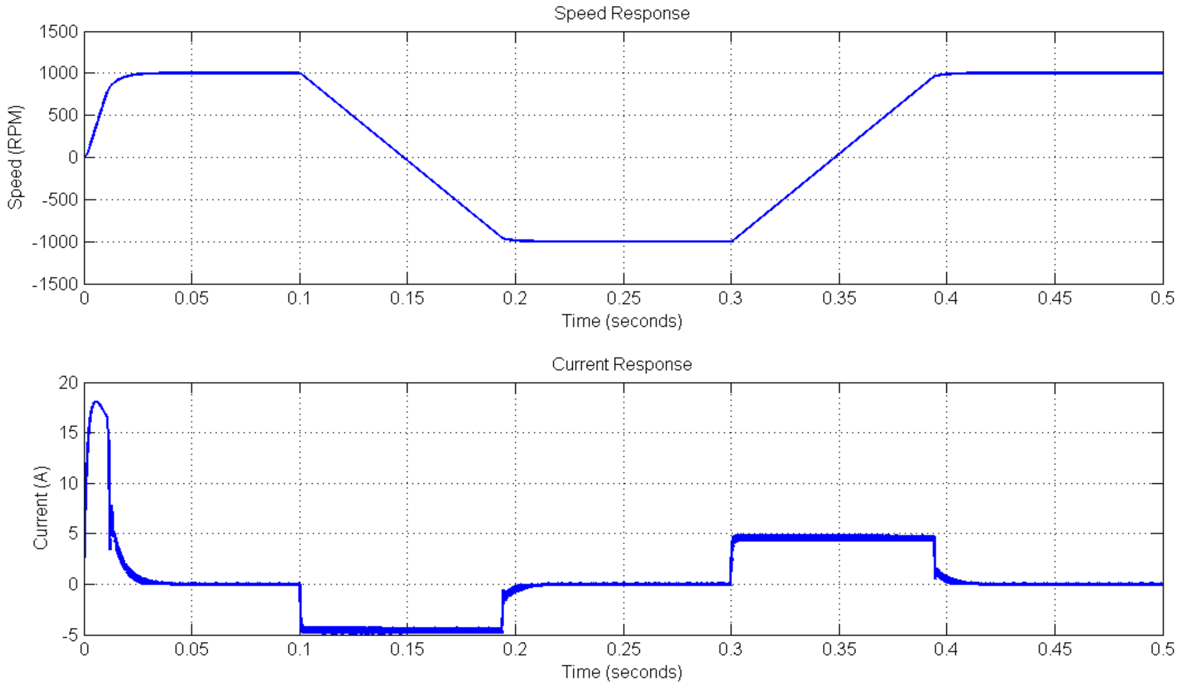


Figure 4.11: Speed and Current response to change in direction of reference speed

Finally, the effect of active damping in feedback in the speed loop of the cascade controller is shown in figure 4.12. When a reference speed of 1000RPM is set, the motor drive causes the motor's speed to ramp-up quickly to its set reference value. This is carried out without very little speed response overshoot which is a highly desired feature in precise servo drives. This minimal overshoot is caused by the action of a feedback active damper in the speed loop of the cascade controller. The transient response characteristics is given in table 4.5.

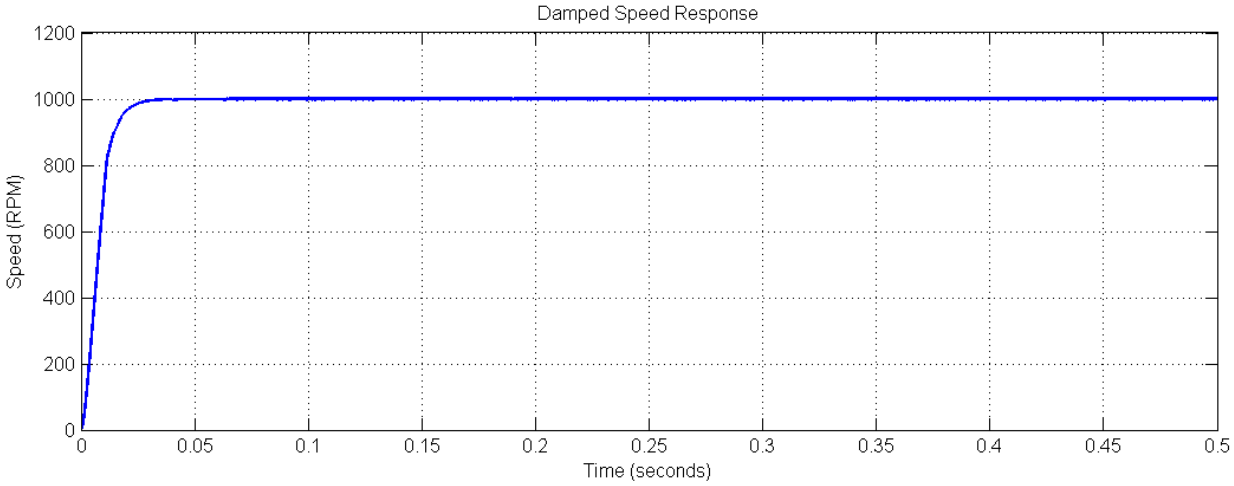


Figure 4.12: Simulation Output Showing Damped Speed Response with negligible Overshoot

Table 4-5: Transient Response Characteristics of the motor at 1000 RPM Reference Speed

| | |
|---|---|
| Rise  Time (seconds) | 0.0123 |
| Settling Time (seconds) | 0.0230 |
| Overshoot (RPM) | 0.0057 |
| Steady State Error | 0 |

# CHAPTER FIVE

# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 CONCLUSIONS

This research project has shown that PID/PI controllers play a major role in the design of speed drives for DC machines. The PID control implemented helps in delivering effective control signals for fast and robust response to changing reference speed and improved load disturbance rejection in the DC motor.

Among the three PID tuning methods that were presented, Particle Swarm optimization (PSO) method of tuning has shown to be a useful method for finding optimal gains for the PID controller and requires less technical knowledge on the PID controller and the plant. However, a method requiring more technical knowledge of the control system namely Internal Model Control (IMC) principle was used to tune the gains of the PI cascade controller for a predetermined speed and current loop bandwidth which enabled accurate orchestration of the cascaded loops.

With the use of a Full Bridge DC-DC converter, four quadrant of operation of the motor was achieved which enabled bi-directional operation of the motor in the most efficient and sustainable way. Also the implemented current limiting feature of the motor speed drive has proved to be effective for driving the motor in safe conditions at all times.

The DC motor current ripples has been minimized by the use of Unipolar Pulse width modulation technique. Also the active resistance feedback scheme provided in the current controller of the DC motor drive ensures a better performing system with improved load disturbance rejection. Likewise for the mechanical part of the motor, active damping feedback scheme implemented in the speed controller of the DC motor drive ensures a well damped speed response of the motor with minimal (almost negligible) overshoot. These characteristics are very important for servo applications requiring accurate reference tracking.

The controller design done on Simulink can be easily converted to C++ and then implemented on a dedicated microcontroller chipset for real life implementation. This approach provides a comparatively cost effective method for implementing DC machine speed drives with features such as current limitation, which normally is implemented using analog component and raises the price of DC machine drives.

## 5.2 RECOMMENDATIONS

It has been studied overtime that the heating (due to resistive power loss in the armature windings) of the motor during normal operation when controlled by PWM is proportional to the switching frequency and the inductance of the motor. Therefore it is recommended that the switching frequency of the PWM pulses should be chosen carefully in order to minimize the current ripple which is proportional to the motor heating. The PWM switching frequency should be equals to the sampling frequency for digital implementation. Also careful selection of the PWM frequency is also necessary for improving the form factor of the motor.

Lastly, to further improve the performance of the speed drive for better performance in high precision servo applications, it is recommended that a position loop should be cascaded with the existing speed and current feedback loop.

# REFERENCE

Ali A. Hassan, N. K.-S. (2018). Comparative Study for DC Motor Speed Control Using PID Controller.

Arnisa Myrtellari, P. M. (2016). Optimal Control of DC Motors Using PSO.

Ayasa, M. S., & Sahinb, E. (2019). Parameter effect analysis of particle swarm optimization algorithm. *An International Journal of Optimization and Control: Theories & Applications*.

Bista, D. (2016). Understanding and Design of an Arduino-based PID controller.

Dani, S., & Ingole, D. (2017). Performance evaluation of PID, LQR and MPC for DC motor speed control.

Engelbrecht, A. P. (2007). *Computational Intelligence. An Introduction.* South Africa: John Wiley and Sons, Ltd.

Fatah, I. S. (2014). PSO-BASED TUNNING OF PID CONTROLLER FOR SPEED CONTROL OF DC MOTOR.

G¨ucin, T. N., Biberoˇglu, M., Fincan, B., & G¨ulbahc¸e, M. O. (2015). Tuning Cascade PI(D) Controllers in PMDC Motor Drives: A Performance.

Gajanan, M. (2011). Design of PID controller for improved performance of higher order systems.

Getu, B. N. (2019). Tuning the Parameters of the PID Controller Using Matlab. *Journal of Engineering and Applied Sciences*.

Gucin, T. N., Biberoglu, M., Fincan, B., & Gulbahce, M. O. (2015). Tuning Cascade PI(D) Controllers in PMDC Motor Drives: A Performace Comparison for Different Types of Tuning Methods.

Hagglund, K. J. (1995). *PID COntrollers, Theory, Design and Tuning.* International society for measurement and control.

Harun Yazgan, F. Y. (2018). Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor.

Hinkkanen, M. (2017). *Control of a DC Motor Drive.*

Hinkkanen, M. (2017). *State Feedback Current Control: Continuous-Time Design.* Politecnico di Torino.

Hughes, A. (2006). *Electric Motors and Drives fundamentals, types and applications.*

Johnson, M., & Moradi, M. H. (n.d.). *PID CONTROL New identification and Design Methods.* 2005: Springer.

Joshi, N. P., & Thakare, A. P. (2012). Speed Control Of DC Motor Using Analog PWM Technique. *International Journal of Engineering Research & Technology (IJERT).*

K. Venkateswarlu, D. C. (2013). Comparative study on DC Motor Speed Control using various contorllers.

Karl Johan Astrom, T. H. (1995). *PID controllers: Theory, Design and Tunning.* International Society for Measurement and Control.

KUSHWAH, M., & PATRA, A. (2014). PID Controller Tuning using Ziegler-Nichols Method for Speed Control of DC motor. *International Journal of Scientific Engineering and Technology Research*.

Latha, K., Rajinikanth, V., & Surekha, P. M. (2013). PSO-Based PID Controller Design for a Class of Stable and Unstable Systems. *Hindawi Publishing Corporation*.

Mohan, N., Undeland, T. M., & Robbins, W. P. (2003). *POWER ELECTRONICS Converters, Applications and Design.* John Wiley & Sons, Inc.

Ms.S.R.Bhagwatkar, M. D. (2015). A Review on Automatic Closed Loop.

Muhammad Rafay Khan, A. A. (2015). Speed Control of a DC Motor under varying load using PID controller.

Muniracad. (2019, January 26th). *Speed control of DC Motor*. Retrieved from Munir Academy: https://muniracademy.com/speed-control-of-dc-motors/

Mutalid, M. A. (2008). Speed control of DC Motor uisng PI controller.

Mwahib Mohamed Harron Elhag, R. M. (2016). *Speed control of a DC motor using fuzzy logic and PI controller.* SUDAN UNIVERSITY OF SCIENCE & TECHNOLOGY.

Patil, V. S., Angadi, S., & Raju, A. B. (2016). Four Quadrant Close Loop Speed Control of DC Motor.

Prof. N. D. Mehta, P. A. (2017). Modeling and simulation of P, PI and PID controller for speed control of DC Motor Drive.

Rao, V. M. (2013). Performance Analysis of Speed Control of DC Motor Using P, PI, PD and PID Controllers.

S.K. Singla, R. K. (2013). Comparison among some well known control schemes.

Sabir, M. M., & Khan, J. A. (2014). Optimal Design of PID controller for the speed control of DC motor by using Metaheuristic Techniques.

Senawi, A., Yahya, N. M., & Yusoff, W. A. (2010). Tuning of Optimum PID Controller Parameter Using Particle.

Walaa M Elsrogy. Naglaa K Bahgaat, M. E.-S. (2018). speed contol of a DC motor using PID controller based on changed intelligent techniques.

Xia, C.-l. (2012). *Permanent Magnet Brushless DC Motor drives and controls.* Singapore: John Wiley & Sons.

Zalm, G. v. (2004). TUNING OF PID-TYPE CONTROLLERS: LITERATURE OVERVIEW.

# APPENDIX

## PARTICLE SWARM OPTIMIZATION ALGORITHM

```
%% Tunning of PID controller for a DC Motor control using
Particle Swarm Optimization
%
%
% Author: Paul Osinowo (osinowo62@gmail.com)
%
% BSc Student, Electrical/Electronics Enginering Dept,
% College of Engineering Bells University of Technology,
Nigeria
%

%% Initialization

clear
clc

fitness_values = ones(1, 50);


n = 100;           % Size of the swarm " no of birds "
bird_setp = 50;   % Maximum number of "birds steps"
dim = 3;          % Dimension of the problem

c2 = 2;          % PSO parameter C1
c1 = 2;        % PSO parameter C2

w_max = 0.9;  % inertia weights
w_min = 0.4;

w = 0.01;     % inertia weights initialize


%    initialize the parameter %

a = 1; % Lower Bound
b = 100; % Upper Bound


R1 = rand(dim, n);  %2 x 50
R2 = rand(dim, n);  %2 x 50
current_fitness =0*ones(n,1);  %50 x 1

% Initializing swarm and velocities and position %
% 2 x 50 (Initial particle random positions)
current_position = (b-a).*rand(dim, n)+ a;

velocity = zeros(dim, n); % Initial Particle Velocities

local_best_position  = current_position ;   %2 x 50


%  Evaluate initial population  %

for i = 1:n

% Evaluates with the fitness function, the fitness of each
points (PID gains).
    current_fitness(i) = fitness(current_position(:,i));
end

local_best_fitness = current_fitness;
[global_best_fitness, position] = min(current_fitness);

global_best_position = ones(size(current_position));
fitness_values(1) = global_best_fitness;


for i=1:n
    global_best_position(:,i) = local_best_position(:,
position);
end

%  VELOCITY UPDATE  %

velocity = w * velocity + c1*(R1.*(local_best_position-
current_position))...
    + c2*(R2.*(global_best_position-current_position));

%   SWARM UPDATE    %

current_position = current_position + velocity;

%  evaluate a new swarm   %


%% Main Loop
iter = 0 ;         % Iterations’counter
while  ( iter < bird_setp )
    iter = iter + 1;

    for i = 1:n
        current_fitness(i) =
fitness(abs(current_position(:,i)));
    end
    for i = 1:n
        if current_fitness(i) < local_best_fitness(i)
           local_best_fitness(i)  = current_fitness(i);
           local_best_position(:,i) = current_position(:,i);
        end
    end


    [current_global_best_fitness,position] =
min(local_best_fitness);



    if current_global_best_fitness < global_best_fitness
       global_best_fitness = current_global_best_fitness;

       for i=1:n
            global_best_position(:,i) =
local_best_position(:,position);
       end

    end

    w = w_max - ((w_max - w_min)/bird_setp)*iter;

    velocity = w *velocity + c1*(R1.*(local_best_position-
current_position))...
        + c2*(R2.*(global_best_position-current_position));


    R1 = rand(dim, n);  %2 x 50
    R2 = rand(dim, n);

    current_position = current_position + velocity;

    sprintf('The value of interation iter %3.0f ', iter)
    fitness_values(iter) = global_best_fitness;
    fitness_values(iter)



end % end of while loop (Particles positions have converged to
an optimal location).
```