DEVELOPMENT AND TUNING OF A PID CONTROL SYSTEM FOR MOBILE ROBOT DRIVE

by

Syed Muztuza Ali

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

Department of Mechanical Engineering
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY
Dhaka, Bangladesh.

2011

Certificate of Approval

The thesis titled, **"Development and Tuning of a PID Control System for Mobile Robot Drive",** submitted by **Syed Muztuza Ali**, Roll No: **100710030 P**, Session: **October 2007**, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN MECHANICAL ENGINEERING** on **19th January 2011**.

_____

Dr. Md. Zahurul Haq
Professor
Department of Mechanical Engineering
BUET, Dhaka, Bangladesh.

Chairman
(Supervisor)

_____

Dr. Muhammad Mahbubul Alam
Professor & Head
Department of Mechanical Engineering
BUET, Dhaka, Bangladesh.

Member
(Ex-Officio)

_____

Dr. Mohammad Mamun
Associate Professor
Department of Mechanical Engineering
BUET, Dhaka, Bangladesh.

Member

_____

Dr. Md. Shafiqul Islam
Director, Central Engg. Facilities
Atomic Energy Research Establish
Gonokbari, Savar, Dhaka.

Member
(External)

Candidate’s Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

_____

Syed Muztuza Ali

CONTENTS

List of Tables

List of Figures

List of Symbols

Symbol	
emf	Electromotive force
MCU	Microcontroller Unit
NO	Normally Open
NC	Normally Close
PWM	Pulse Width Modulation
SP	Set-point
PV	Process Variable
MV	Manipulated Variable
y	Controller output
y_0	Controller output with no error
e	Controller Deviation
Kp	Proportional Gain
K_I	Integral Gain
K_D	Derivative Gain
Kc	Critical Gain
Tc	Critical Period
T_I	Integral Period
T_D	Derivative Period
DAS	Data Acquisition System
DIP	Dual Inline Package
USART	Universal Synchronous Asynchronous Receiver Transmitter
CCP	Compare-Capture
ADC	Analog to Digital Conversion
INT	Interrupt

TMR	Timer
PS	Prescaler
DC	Duty Cycle

Acknowledgements

The Author would like to express his sincerest gratitude and deepest reverence to his thesis supervisor Dr. Md. Zahurul Haq, Professor, Department of Mechanical Engineering, BUET for his constant gratitude, constructive criticisms, encouragement and careful supervision throughout this research without which this thesis would not come to an end.

The Author would also like to express his sincere gratitude to Dr. Md. Shafiqul Islam, Director, Central Engg. Facilities, Atomic Energy Reseach Establish and Dr. Mohammad Mamun, Associate Professor, Department of Mechanical Engineering, BUET for their valuable advice and sharing of their vast knowledge.

Special thanks to all his friends for their tremendous support and encouragement throughout the work. Finally the author would like express his gratitude to his family for their encouragement and support.

Abstract

In the present study, a motor-drive test-bench is designed and fabricated using locally available hardware components. The test-bench contains a 12V DC motor which is connected to a flywheel to provide inertial load and also to a DC motor to act as a fixed load. The system is controlled by a control system capable of implementing ON-OFF, P, PI, PD and PID control actions. The controller is implemented using PIC18F4550 microcontroller. The system also contains extensive sensor/instrumentation to measure the system response in response to the applied control action. Three different speed profiles are transmitted to the motor control system which generates the control action using the sensor readings. The motor control system is implemented using PIC18F452 microcontroller system. Control actions are implemented using Pulse Width Modulation (PWM) signal sent by the motor control system. H-Bridge is used to control the direction of the motor and MOSFET is used as an electronic switch to provide PWM action. The system responses are analyzed and it is found that optimized PID controllers provide optimum system response.

CHAPTER 1

Introduction

The world of Robotics is one of the most exciting areas that have been under constant innovation and evolution. Robotics is interdisciplinary field of engineering that has become more and more a part of our daily life. These are no longer a vision for the future but a reality of the present. In developed world, robots fill many important parts from mowing the lawn, securing our uninhabited houses, feeding our pets or building our cars. It is visible that these are everywhere and as time passes these will be more and more prevailing. In robotics, special attention is given to mobile robots, since these have the capability to navigate in their environment and are not fixed to one physical location. The tasks these can perform are endless and the possibilities for these to contribute to our lives are countless (e.g., unmanned aerial vehicles, autonomous underwater vehicles). The development of industrial robots is characterized by a multidisciplinary fusion of a large spectrum of technologies. Many of these technologies are not specific for robotics and can be developed from solutions in other much larger product areas. However, robot control and then especially robot motion control, is very specific to the robot product and constitutes one of the most important key competences for the development of industrial robotics. By applying and developing advanced control, it is possible to continuously improve the robot performance, which is necessary in order to increase performance and lower cost of industrial robot automation.

There are many applications where a robotic vehicle is to be automatically driven along some path with high accuracy. The accuracy can be obtained with optimum feedback control system using suitable sensors. Control schemes for motion stabilization and trajectory tracking differ in complexity and flexibility. In order to provide consistent, reliable and stabilized movement during running course, a proportional control parameter confirmation experiments are reported in different literature and these studies demonstrated the

importance of inclusion of additional control parameters and their proper tuning for stabilized motion and tracking [1].

Mobile robots are widely used to autonomously transport work pieces between various assembly stations by following special guide wires or colored strips. In potentially dangerous and inhospitable environments, autonomous robots are gaining widespread popularity. Schulze and Zhao (2007) reported the widespread usage of mobile robots in Europe and China. For the developing countries, industrial applications of robots are not very popular and cheap labor cost plays an important role for such situation. However, when the production speed and accuracy are not to be sacrificed, application of automation employing robots offers a practical solution [2].

Dynamic characteristics of a servo controlled robot was investigated using optimum pulse width modulation where this robot performs extensively well in flat tracks [3]. The present study concentrates on controlling mobile robots in a specified speed profile by optimizing different control parameters.

1.1 Objectives

This study is carried out with the following specific objectives:

(a) To design and fabricate a robot drive control test-bench.

(b) To setup measuring instrumentation, drive system and control electronics.

(c) To setup high speed data acquisition and data communication systems to monitor and log the system response and controller actions under actual operating conditions.

(d) To drive the motor system with fixed and inertial loads and subjected to different control actions.

(e) To analyze system response to obtain control parameters required for control parameter's tuning for optimum control.

1.2 Scope of the Thesis

The thesis reports development and tuning of different control systems for mobile robot drive. System response is analyzed to obtain the optimum control parameter. In this thesis, literature is reviewed in Chapter 2. The design of the robot drive control test bench and experimental procedure is described in Chapter 3. Chapter 4 presents the outcome and conclusions of the thesis.

CHAPTER 2

Review of Servo-control Systems for Mobile Robots

Recent years have witnessed a major increase of mobile robots. Now-a-days these are subject of major research projects being present in industry, military, security and even home environments as consumer products for entertainment and home aid tasks. Mobile robots are generally those robots which can move from place to place across the ground. Mobility gives a robot a much greater flexibility to perform new, complex, exciting tasks. The world does not have to be modified to bring all needed items within reach of the robot. The robots can move where needed. Fewer robots can be used. Robots with mobility can perform more natural tasks in which the environment is not designed specially for them. These robots can work in a human centered space and cooperate with men by sharing a workspace together.

2.1 Motion Control of Mobile Robots

A mobile robot needs locomotion mechanisms that enable it to move unbounded throughout its environment. There is a large variety of possible ways to move which makes the selection of a robot's approach to locomotion an important aspect of mobile robot design. Although a lot of locomotion mechanisms have been inspired by their biological counterparts, nature did not develop a fully rotating, actively powered joint, which is the technology necessary for wheeled locomotion. In general, legged locomotion requires higher degrees of freedom and therefore greater mechanical complexity than wheeled locomotion [4].

The wheel has been by far the most popular locomotion mechanism in mobile robotics and in man-made vehicles in general. It can achieve very good efficiencies, and does so with a relatively simple mechanical implementation. Wheels are extremely well suited for flat surfaces where this type of locomotion is more efficient than a legged one. For example the railway is ideally engineered for wheeled locomotion because rolling friction is minimized on a hard and flat steel surface. When the surface becomes soft, wheeled locomotion

accumulates inefficiencies due to rolling friction whereas legged locomotion suffers much less because it consists only of point contacts with the ground [5].

Drive is the electromechanical component of a mobile robot to produce mobility or locomotion that makes it capable to move effectively in many environments. An efficient drive system is characterized by it's high top speed, acceleration, pushing/pulling ability, maneuverability, accuracy, obstacle handling, climbing ability, reliability/durability, ease of control etc. These features are strongly dependent on the selection of motor, size and weight of base, size and number of wheels, coupling method of motor shaft and the wheel, control method etc [6].

DC motors are one of the most widely used prime movers of Mobile Robot Drive. Years ago a majority of the small servomotors used for control purposes were of AC variety. In reality AC motors are more difficult to control, especially for position control and their characteristics are quite nonlinear which makes the analytical task more difficult. On the other hand though the DC motors are more expensive, because of the brushes and commutators, these are suitable for control applications due to it's compatibility with control applications. Stepper Motors, DC Brushed Servo Motors and DC Brushless Servomotors are mostly used in Mobile Robot Drive [6].

For automation often three wheeler robots are used with two independently controllable wheels at the rare end and a free un-powered caster at the front. These two independently controllable wheels provide forward and backward motion as shown in Fig. 2.1, depending on their relative motion robot turns by spinning on its axle. The caster on the other end provides balance and a pivot for turning. The robot steers by changing the direction and speed of each motor as follows:

> to move forward both motors rotate in forward direction, to go back both motors rotate in reverse direction, to take a right turn either right motor stops, left motor rotates in forward direction or left motor rotates faster than right, to take a left turn either left motor stops, right motor rotates in forward direction or left motor rotates faster than right. But to take a hard right turn right motor rotates in forward direction and left motor rotates in backward direction and to take a hard left turn left motor rotates in forward direction and right motor rotates in backward direction [7].

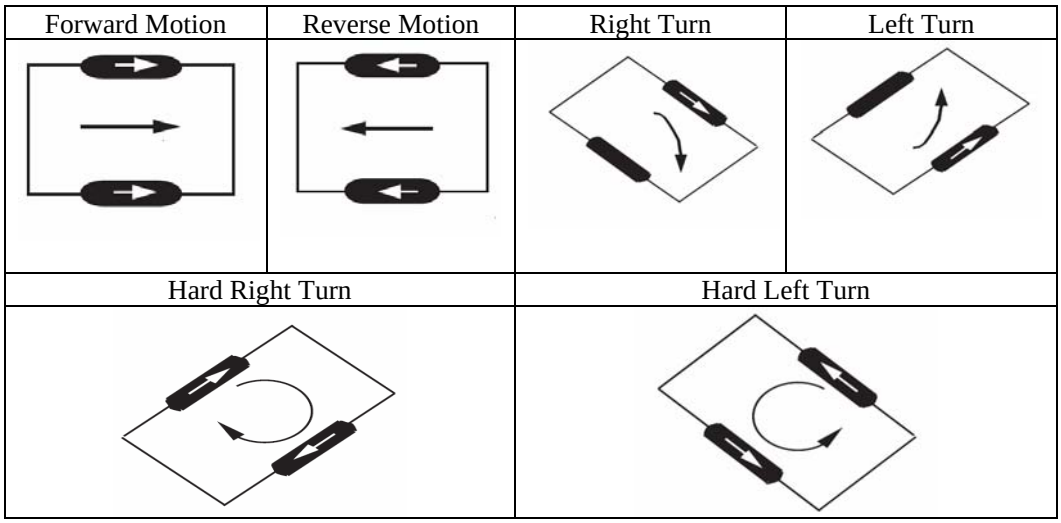


Fig. 2.1: Direction Control of Mobile Robot

To control the motors direction through Micro Controller Unit (MCU) there should be a device that would act like a solid state switch, a transistor, and to hook it up the motor. A diode should be connected across the coil of the relay which will keep the spike voltage and back EMF coming out of the coil of the relay, from getting into the MCU and damaging it. Using the circuits shown in Fig 2.2, it is only possible to get the motor to stop or turn in one direction, forward for the first circuit or reverse for the second circuit.

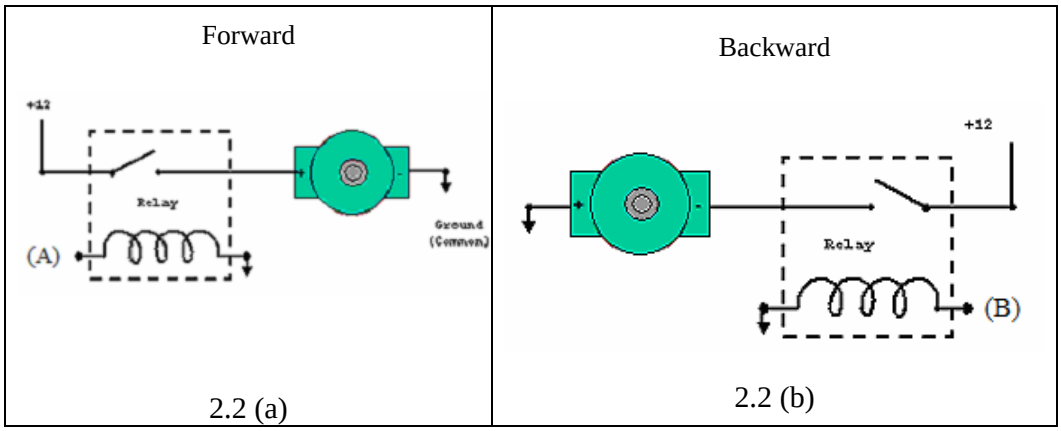


Fig. 2.2: (a) If a logical one (+ 5V) supplied from the MCU to point A causes the motor to turn forward. Applying a logical zero, (ground) causes the motor to stop turning (to coast and stop). (b) If a logical one (+ 5V) supplied from the MCU to point B causes the motor to turn forward. Applying a logical zero, (ground) causes the motor to stop turning (to coast and stop).

To control the motor in both forward and reverse with a processor, more circuitry is needed. H-Bridge circuits are widely used to control direction of mobile robots as shown in Fig 2.3.

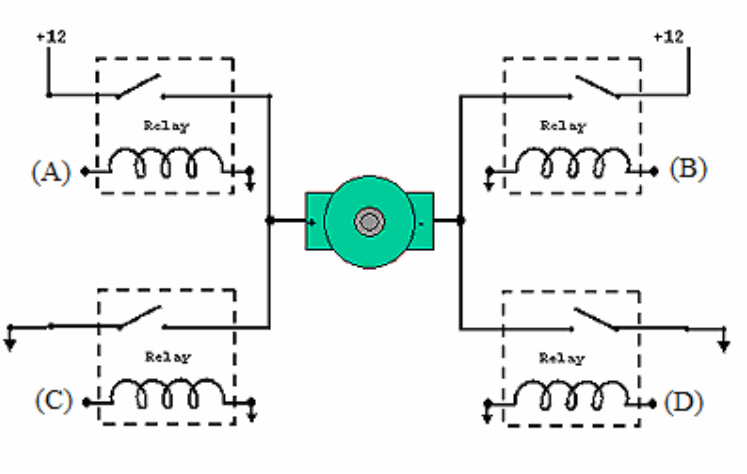


Fig. 2.3: H-bridge Circuit for DC Motor Direction Control

The "high side drivers" are the relays that control the positive voltage to the motor. This is called sourcing current. The "low side drivers" are the relays that control the negative voltage to sink current to the motor. "Sinking current" is the term for connecting the circuit to the negative side of the power supply, which is usually ground. So, turning on the upper left and lower right circuits, flows power through the motor forward to turn motor forward whereas turning on the upper right and lower left circuits will flow power through the motor in reverse, and make the motor turn reverse.

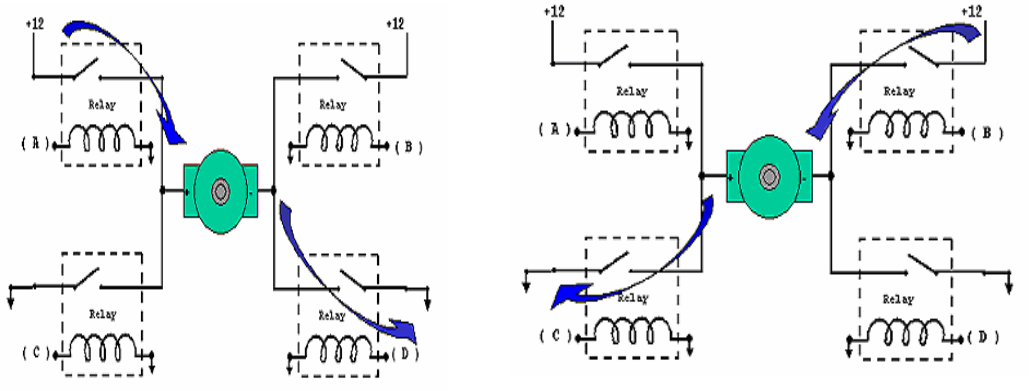


Fig. 2.4: Direction Control of DC Motor

Here relays are shown with a single NO (Normally Open) connection which becomes closed when relay's magnetic coil are energized with microcontroller signal. But in practice every relay has at least two connections, one is NO and the other is NC (Normally Closed). NC connections become opened when relays coil are energized. This is possible to implement the same circuit using two relays instead of using four relays. To do so NC connection of

Relay A can be used as A and NO connection of Relay A as C, whereas it is to use NC connection of Relay B as D and NO connection of Relay B as B. [8]

It seems that stopping a DC motor is a simple, almost trivial operation. Unfortunately, this is not always true. When a large dc motor is coupled to a heavy inertia load, it may take an hour or more for the system to come to a halt. For many reasons such a lengthy deceleration time is often unacceptable and, under some circumstances, a braking torque must be supplied to ensure a rapid stop. One way to brake the motor is by simple mechanical friction, in the same way as a car is stopped. A more elegant method consists of supplying a reverse current in the armature, so as to brake the motor electrically. The method to create such an electromechanical brake is known as dynamic braking. During dynamic braking armature current reverses, armature torque reverses, and the motor tries to reverse. The speed in the forward direction rapidly decreases as does the voltage generated in the armature. At the point of reversal or zero speed, generated voltage is zero. The motor stops at this point since current cannot flow and no more reversing torque is generated [9].

Often speed of DC motors are controlled with a variable resistor or variable resistor connected to a transistor. But it generates heat and hence wastes power. Pulse Width Modulation (PWM) can eliminate these DC motor control problems. It controls the motor speed by driving the motor with short pulses. These pulses vary in duration to change the speed of the motor. The longer the pulses, the faster the motor turns, and vice versa. The speed of a DC motor is directly proportional to the supply voltage, so if the supply voltage is reduced from 12 Volts to 6 Volts, the motor will run at half the speed. But this is to be achieved when the battery is fixed at 12 Volts. A better way is to switch the motor's supply on and off very quickly. If the switching is fast enough, the motor doesn't notice it, it only notices the average effect. As the amount of time that the voltage is ON increases compared with the amount of time that it is OFF, the average speed of the motor increases. So this PWM signal is apparently a digital signal as it has only two stages when the supply voltage is completely on or completely off. This ON/OFF switching is performed by power MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor) that can turn very large currents on and off under the control of a low signal level voltage (Fig 2.5) [10].

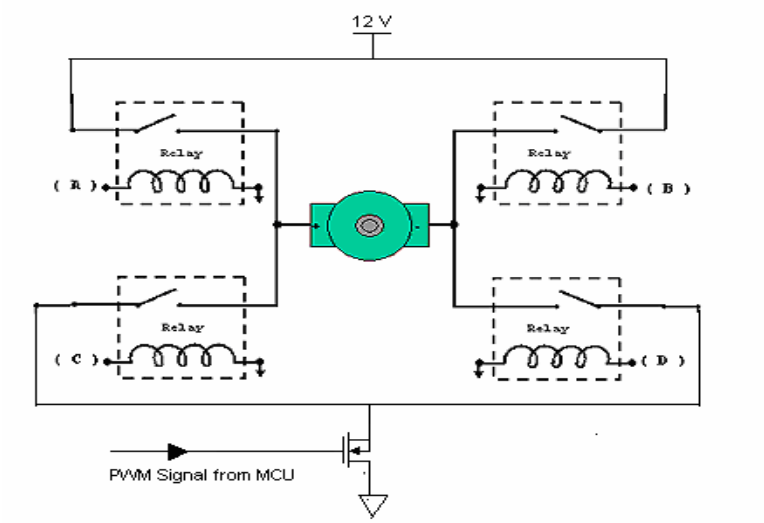


Fig. 2.5: Speed Control of a Mobile Robot

In a nutshell, PWM is a way of digitally encoding analog signal levels. Advantages of using digitally encoded PWM signal over analog signal is that it is less effected by interference even it does not require digital to analog conversion. Through the use of high-resolution counters, the *duty cycle* of a square wave is modulated to encode a specific analog signal level. The voltage or current source is supplied to the analog load by means of a repeating series of ON and OFF pulses. The ON-*time* is the time during which the DC supply is applied to the load, and the OFF-*time* is the period during which the supply is switched off. Given a sufficient bandwidth, any analog value can be encoded with PWM. The ratio of the 'On period' to the 'Total period' of a single pulse (Shown in Fig. 2.6) is termed as duty cycle which is defined as:

$$\text{Duty Cycle} = \frac{ON\ \ Time}{Total\ \ Period} \text{ X } 100\% \qquad (2.1)$$

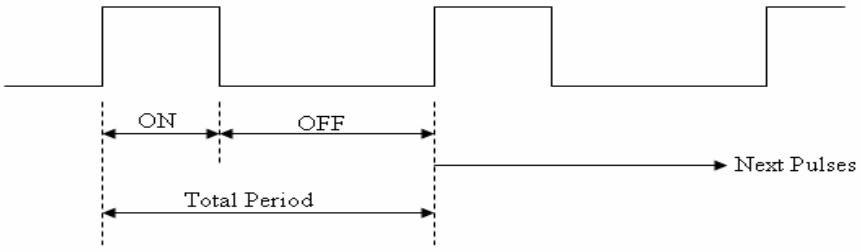


Fig. 2.6: PWM Signal

Path traveled by a mobile robot is determined by the number of rotation of wheel and it's circumference. To determine the no of rotation of wheel, an 'Optical Encoder' and an encoder disk is used which is circular in size and with equally spaced holes on it's periphery mounted on the shaft of wheel. This optical encoder allows to pass light of a led through these holes when this led is in front of these holes and blocks it when led is not aligned with the holes. This light is sensed by a phototransistor and it sends a signal of two levels high (5V) or low (0V) to a microcontroller unit through a signal conditioning circuit. Microcontroller unit counts these signals from which it decides the number of rotations the wheel rotated. From which controller calculates the number of rotations and distance traveled by the robot. [2]

2.2 Motor Drive Control Fundamentals

A control system is an interconnection of components connected or related in such a manner as to command, direct, or regulate itself or another system [10]. In an open loop control system, one or more input variables of a system act on a process variable. The actual value of the process variable is not being checked, with the result that possible deviations e.g. caused by disturbances are not compensated for in the open loop control process. Thus, the characteristic feature of open loop control is an open action flow. In a closed loop or feedback control system, the variable to be controlled (controlled variable x) is continuously measured and then compared with a predetermined value (reference variable w). If there is a difference between these two variables (error e or system deviation x - w), adjustments are being made until the measured difference is eliminated and the controlled variable equals the reference variable. Various terminology used in closed-loop control system is shown in Fig. 2.7.

A controller generates a control signal to the final element, based on a measured deviation of the controlled variable from the set-point. Controller responds in different ways to the deviation. Sometimes control system only turns on or off the actuator depending on the deviation, on the contrary in some cases control system actuates the final control element a little or lot or fast or slowly with the variation in deviation. So the operating modes of control system are of two types:

i. Discontinuous or ON-OFF controller
ii. Continuous controller

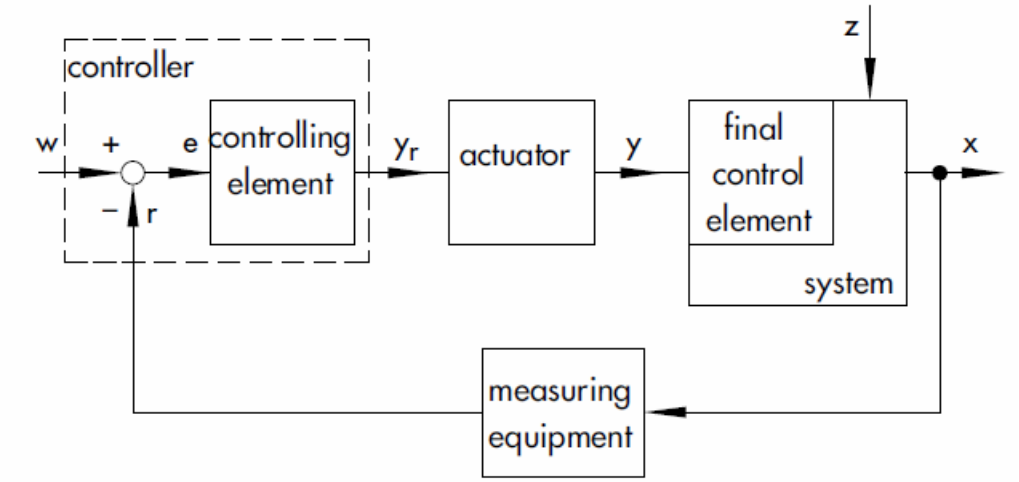


Fig. 2.7: Elements of a closed loop control system

Variables indicated in Fig 2.7 are defined here as follows:

w : *Reference Variable*
r : *Feedback variable*
e : *Error*
y_r : *Controller output variable*
y : *Manipulated variable*
z : *Disturbance Variable*
x : *Controlled variable or actual value* [12]

2.2.1 ON/OF Controller

The most elementary control mode is ON/OFF or two-position control mode. This is an example of a discontinuous control mode. It operates on the manipulated variable when the controlled variable crosses the set-point (SP). The output has thus two states, usually fully ON or fully OFF. Equation for the controller output can be generally written as:

$$Y = 0\% \quad \text{when } e < 0 \tag{2.2}$$

or

$$Y = 100\% \quad \text{when } e > 0$$

This relation shows that when the measured value is less than the set-point, full controller output results. When it is more than the set-point, the controller output is zero. So for a motor drive controller when the actual speed drops bellow the reference speed, motor is turned on and when the speed rises above the reference speed it turns off the motor. One drawback of this control system is that the rate at which the final control element switches on

and off can be very high. This condition can result in regular oscillations of measured value over the reference variable. This condition would be detrimental for most control devices, such as contactors and valves. To prevent this, an ON-OFF differential or hysteresis is added to the controller function [13] . An example of temperature control using such control action is shown in Fig. 2.8. However, oscillations in controlled variable (temperature) is observed even with the control system functioning.

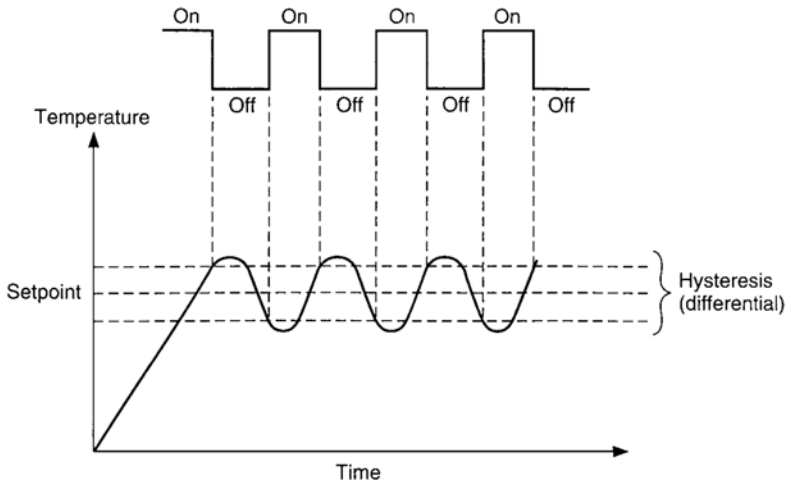


Fig. 2.8: ON-OFF Control

2.2.2 Continuous Controllers

To reduce the oscillations in ON/OFF controller, continuous controllers are widely used. Several continuous controllers are used in motion control, such as:

i) P (Proportional) Controller
ii) PI (Proportional-Integral) Controller
iii) PD (Proportional-Derivative) controller
iv) PID (Proportional-Integral-Derivative) controller

In a P (Proportional) controller the control deviation is produced by forming the difference between the process variable PV and the selected set-point SP; this is then amplified to give the manipulating variable MV, which operates a suitable actuator.

The control deviation signal has to be amplified, since it is too small and cannot be used directly as the manipulating variable. The gain (Kp) of a P controller must be adjustable, so

that the controller can be matched to the process. This control mode can be expressed by the expression

$$y = k_p e + y_0 \tag{2.3}$$

where y : Manipulated Variable, MV.

k_p : Proportional gain between error and controller output

y_0 : Controller Output with no error.

The output signal is directly proportional to the control deviation, and follows the same course; it is merely amplified by a certain factor. A step change in the deviation e, caused for example by a sudden change in setpoint, results in a step change in manipulating variable.

A P controller only produces a manipulating variable when there is a control deviation, as it is already known from the controller equation. This means that the manipulating variable becomes zero when the process variable reaches the set-point. So a P controller always has a permanent control deviation or offset. [14]

An I controller (integral controller) integrates the deviation signal applied to its input over a period of time. The longer there is a deviation on the controller, the larger the manipulating variable of I controller becomes. How quickly the controller builds up its manipulating variable depends firstly on the setting of I component, and secondly on the magnitude of the deviation. The manipulating variable changes as long as there is a deviation. Thus over a period of time, even small deviations can change the manipulating variable to such an extent that the process variable corresponds to the required set-point. In principle, an I controller can fully stabilize after a sufficiently long period of time, i.e. set-point = process variable. The deviation is then zero and there is no further increase in manipulating variable. Unlike the P controller, I controller does not have a permanent control deviation.

Therefore this mode is represented by the following integral equation

$$y(t) = K_I * \int_0^t e \, dt + y\,(0) \tag{2.4}$$

where $y(t)$: Manipulated Variable, MV.

k_I : Integral Gain i.e. how much controller output is required for time accumulation of error .

y(0) : Controller Output when the control action starts.

The step response of the I controller shows the course of the manipulating variable over time, following a step change in the control difference.

Derivative control action responds to the rate at which the error is changing i.e. the derivative of the error. Appropriately the equation for this mode is given by the expression:

$$y(t) = k_D \frac{de}{dt} \tag{2.5}$$

where y(t) : Manipulated Variable, MV.

k_D : Derivative Gain i.e. how much percent to change the controller output is required for rate of change of error per second.

$\frac{de}{dt}$: Rate of change of error per second. [12]

The derivative controllers are implemented to account for future values, by taking the derivative, and controlling based on where the signal is going to be in the future.

It is common in the complex of practical process to find control requirements that do not fit the application norms of any of the above discussed controller modes. It is both possible and expedient to combine several basic modes, thereby gaining the advantages of each mode. In some cases an added advantage is that the modes tends to eliminate some limitations they individually possesses.

I controller takes a relatively long time before the controller has built up its manipulating variable. Conversely, the P controller responds immediately to control differences by immediately changing its manipulating variable, but is unable to completely remove the control difference. This would seem to suggest combining a P controller with an I controller. The result is a PI controller. Such a combination can combine the advantage of the P controller, the rapid response to a control deviation, with the advantage of the I controller, the exact control at the set-point.

The analytic expression for this control process is found from a series combination of P controller equation and I controller equation as

$$y(t) = k_p e_p + k_p k_I \int_0^t e_p \, dt + y_I(0) \quad (2.6)$$

where $y(t)$: Manipulated Variable, MV.

K_p : Proportional Gain.

K_I : Integral Gain.

e : Control Deviation.

$y_I(0)$: Integral action of controller when the control action starts.

A controller which responds in a similar way to the above operator is the PD controller: it consists of a P component with a known proportional action, and a D component with a derivative action. This D component responds not to the magnitude or duration of the control deviation, but to the rate of change of the process variable. Expression for the manipulating variable of PD controller is

$$y(t) = k_p e + k_p k_D \frac{de}{dt} + y_t(0) \quad (2.7)$$

If a new setpoint is applied, the manipulating variable is increased by the P component; this component of the manipulating variable is always proportional to the deviation. The process variable responds to the increased manipulating variable. As soon as the process variable changes, the D component starts to take effect. While the process variable increases, the D component forms a negative manipulating variable, which is subtracted from the manipulating variable of the P component, finally producing the manipulating variable at the controller output. When the process variable is tracking the setpoint, the D component “brakes”, thus preventing the manipulating variable overshooting above the setpoint.

It has been seen earlier that the combination of a D component or an I component with a P controller offered certain advantages in each case. Now it seems logical to combine all three structures, resulting in the PID controller, and the components of a PID controller is shown in Fig. 2.9.

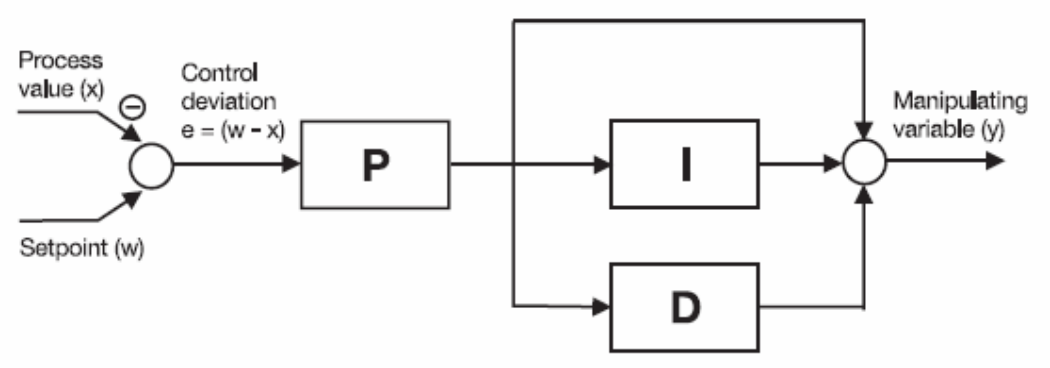


Fig. 2.9: Components of a PID Controller

This is one of the most complex but powerful control mode action. This system can be used virtually any process condition. The analytical expression is

$$y(t) = k_p e + k_p k_I \int_0^t e\,dt + k_p k_D \frac{de}{dt} + y_t(0) \qquad (2.8)$$

With this controller, the K_P, K_I, K_D parameters are adjusted for the P, I and D action. [12-14]

2.3 Tuning of Continuous Controllers

"Tuning" is the engineering work to adjust the parameters of the controller so that the control system exhibits desired property. Usually this is the process of finding the optimum Gain of controllers which have a profound effect on loop performance. There are many methods for determination of the optimum mode gains depending on the nature and complexity of the process. Among them three are most widely used. Two of them are semi-empirical methods in that they depend on measurements made on system to determine factors used in the adjustment formulas. [11]

Two semi-empirical methods are

i. Open-Loop Transient Response Method
ii. Zeigler-Nichols Method.

Open Loop Control Method is developed by Zeigler-Nichols and often referred to as process reaction method. In this method the control system is first considered as open loop. It is implemented by disconnecting the controller output from the final control element. All of the process parameters are held constant at there nominal values. At some times, a small transient disturbance is applied, manual change of the controlling variable using the final

control element. This should be as small as necessary for measurements. The controlled variable is measured versus time at the instant of the introduced disturbance.

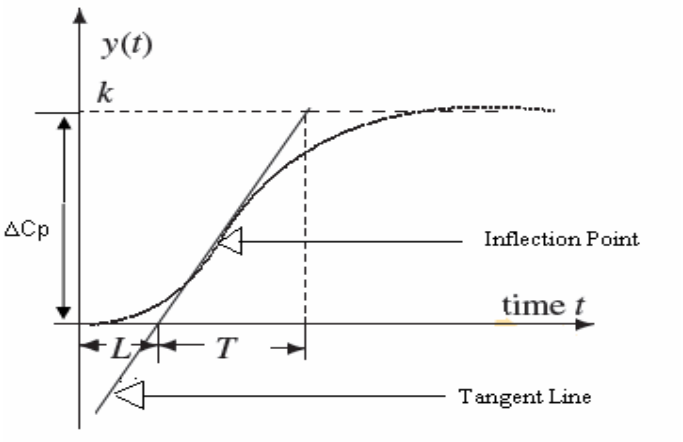


Fig. 2.10 : Open Loop Transient Method

A typical open-loop controller response is shown in Fig. 2.10. Here control deviation is expressed as a percent of range and the final control element disturbance is expressed as percentage change. A tangent line is drawn through the process reaction curve at the inflection point on the curve. Inflection point is defined as that point where the slope stops increasing and begins to decrease. Where the tangent line crosses the origin, there

L = Lag in minutes

as the time from disturbance application to the tangent line intersection. Also from the same graph the process reaction time T is obtained and

$$N = \frac{\Delta Cp}{T}$$

Where N = reaction rate in % / min

ΔCp = variable change in %

T = process reaction time in minutes

Quantities defined by these equations are used with the controlling variable change ΔP to find the controller settings. Expressions for different control mode settings are reported in Table 2.1 [14].

Table 2.1: Tuning of Controller Parameters using Open Loop Transient Method

Parameter	*P*	*PI*	*PID*
Kp	$\frac{\Delta P}{NL}$	$0.9\frac{\Delta P}{NL}$	$1.2\frac{\Delta P}{NL}$
$T_I = \frac{1}{K_I}$		3.33 L	2 L
$T_D = \frac{1}{K_D}$			0.5 L

Second empirical method for controller tuning is also developed by Zeigler and Nichols. This method is alternatively known as Ultimate Cycle Method is based on adjusting a closed loop controller until steady oscillation occurs. Steady oscillation means that the period of each oscillation is same. Steps of tuning controller in this method are listed bellow:

I. Any kind of Integral and Derivative action is to be reduced to their minimum effect.
II. Proportional gain is to be increased until the steady oscillation of process variable occurs.
III. This gain at which the steady oscillation occurs is known as Critical Gain, Kc.
IV. The period at which the process variable oscillates is called Critical Period, Tc.
V. The Critical Period, Tc of this oscillation is measured in minutes.
VI. This values of Critical Gain, Kc and Critical Period, Tc leads the determination of Controller settings using following adjustment formulas for different control modes:

Table 2.2: Tuning of Controller Parameters using Zeigler-Nichols Method [14]

Parameter	*P*	*PI*	*PID*
Kp	0.5 Kc	0.45 Kc	0.6 Kc
$T_I = \frac{1}{K_I}$		$\frac{T_c}{1.2}$	$\frac{T_c}{2}$
$T_D = \frac{1}{K_D}$			$\frac{T_c}{8}$

CHAPTER 3

Experimental Setup and Procedure

Robot drive control test-bench is an experimental setup designed to test a mobile robot drive's response when it is required to run in a predefined speed profile. It is possible to test here whether it can achieve the prescribed speed or getting behind the speed or overshooting the speed. To test a mobile robot in this bed, practical circumstances that a robot faces on it's track is simulated. The setup contains a drive motor which rotates a shaft coupled with a flywheel. The flywheel is used to create inertial load which is present in practical robots. The motor drive is controlled and monitored by two different microcontroller based systems. The test-bench also contains motor drive control electronics, a host of sensors and data-communication hardware and power-supply system as shown in Fig. 3.1.

Fig. 3.1: Photograph of the motor control test-bench.

3.1 Motor Drive Control and Monitory Hardware

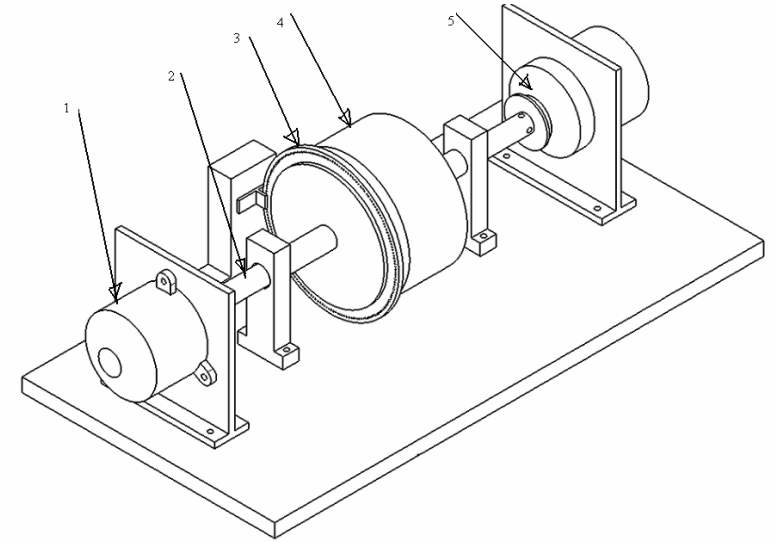


Fig. 3.2: Robot drive control test bench

A 12V DC motor is coupled to the shaft of the test bench as drive motor. It's main purpose is to rotate the shaft at a requisite speed. It is driven with the help of an adjustable power supply. With this power supply it is possible to supply any voltage between 10 to 25V with an accuracy of 0.001 V. It's speed can be varied by varying armature voltage in PWM (Pulse Width Modulation) fashion. Main shaft is rotated by the drive motor and also carries flywheel, encoder disk and disturbance motor on it. Encoder disk is locally made of nylon, circular in size and with equally spaced 180 holes on it's periphery mounted on the shaft. An optical sensor is placed with it whose light emitter is at one side of the encoder disk and the light receiver is at the opposite side of the encoder. Flywheel is made of nylon and attached to the shaft to provide inertia load. Another 12V DC motor is coupled to the shaft of the robot control test bench at the opposite end of the drive motor. This motor is termed in this work as 'Disturbance Motor' and it is not powered by any external power source. It's shaft is rotated at the rpm of drive motor as the both motor is coupled with the same shaft. Following the DC motor principle it produces back emf across it's coil. If this produced back emf is dropped in a resistance, it will cause the braking effect on it's shaft which in turn reduce the speed of whole system and will require control action to boost its power supply to maintain its speed. A rectangular platform of acrylic sheet is used as base which supports all the parts of test bench. All of these components are shown in Fig. 3.2.

Microcontroller based versatile electronic control and monitoring systems are designed for robot control test bench. Three PCBs are designed and populated with electronic devices. One system is to receive input signal and transmit output signal of microcontroller and named as ‘Microcontroller Board’, second board is for controlling the drive motor like changing the direction and speed of motor and to start or stop the motor receiving output signals of microcontroller and named as ‘Motor Control Board’, while the rest is for sending the live parameters of robot control to a laptop is known as ‘DAS (Data Acquisition System) Control Board’.

Microcontroller board is populated with a PIC18F452 microcontroller. In this work it will be termed as “Motor Controller”. It’s a 40 pin dual inline IC package. To receive and transmit signal through these pins, these are connected with connectors. A 4 MHz crystal is connected to it’s assigned pins. Provisions are there for switches and LEDs. This board is designed with a separate power supply of 5V using regulator IC 7805. Microcontroller Board is shown in Fig. 3.3(a) and pinout of PIC18F452 is shown in Fig. 3.3(b).

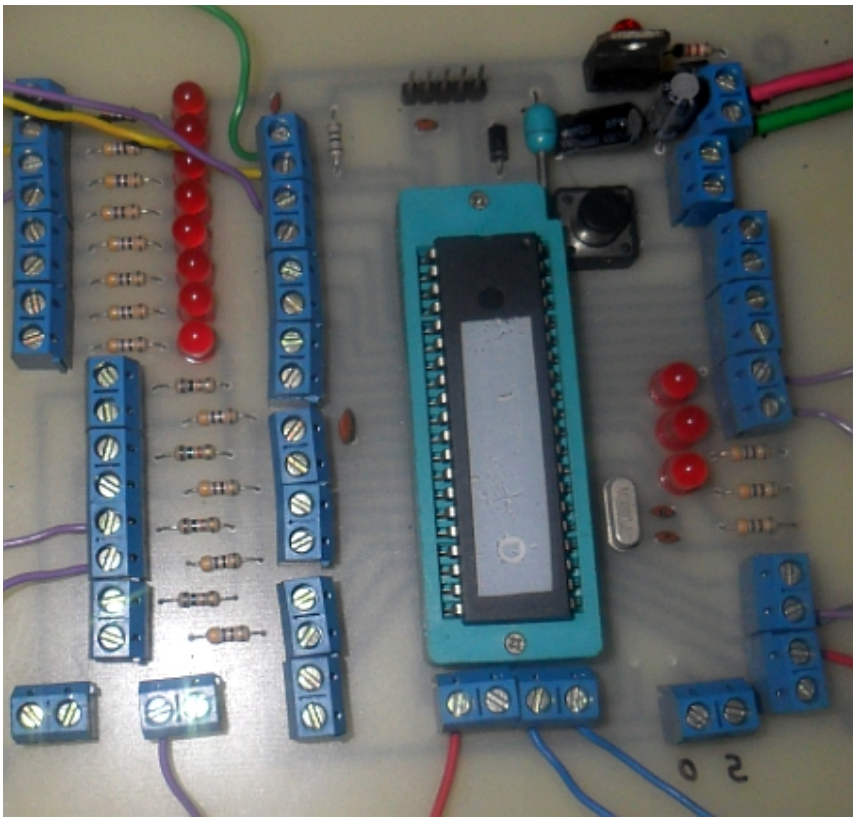


Fig. 3.3(a) : Microcontroller Board

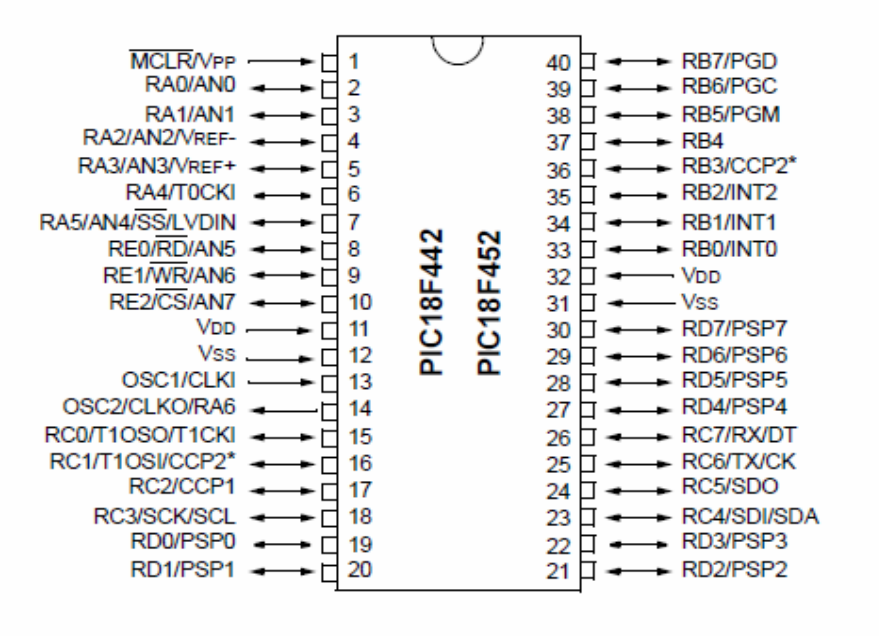


Fig. 3.3(b) : Pinout of PIC18F452

Motor Control Board having relay based H-bridge, control the direction of motor rotation, a MOSFET connected between the ground and H-bridge switched by the PWM signal from microcontroller board supplies variable voltage to the armature coil of motor to control it's speed. This board (shown in Fig. 3.4) includes a signal conditioning circuit for the encoder signal coming from the rotation of shaft. Holes of encoder permit a LED to pass its light which is received by a phototransistor placed at the opposite side of the encoder (shown in Fig. 3.5). When wheel rotates, encoder blocks this light until the next hole on the periphery reached in front of the phototransistor. When this light is blocked, phototransistor provides a signal of 5V but when light is passed through the encoder, it transmits a signal of closely below 5V. This signal can't be used directly to microcontroller as it will receive the signal as high level. To split these signal in two levels in signal conditioning circuit, signals coming from the sensor is used to switch a PNP transistor (BD136). As a result, it puts output signal as 0V when light is blocked and 5V when light passes. In this way signals are conditioned in binary levels. Thus counting this change of state it is possible to calculate the number of rotation of wheel which in turn gives the measure of traveled path.

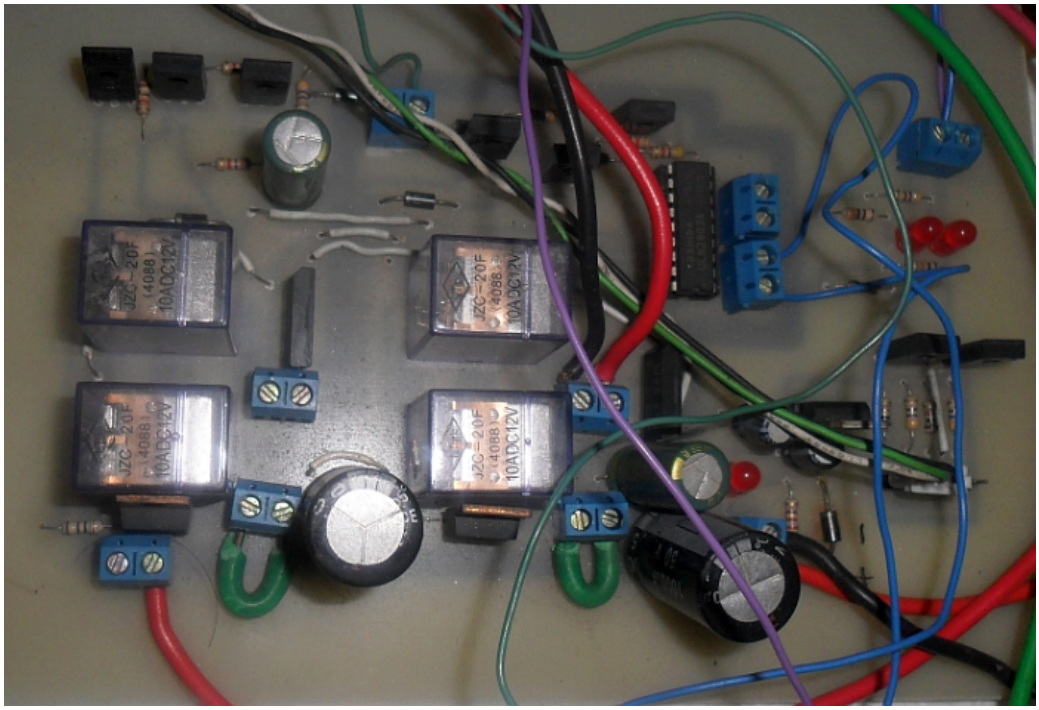


Fig. 3.4: Motor Control Board

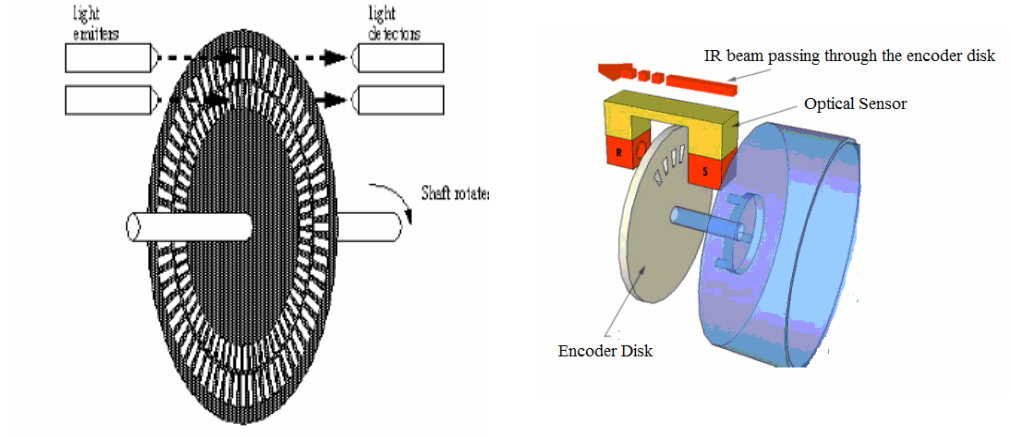


Fig. 3.5: Encoder

DAS control Board (shown in Fig. 3.6) possesses a PIC18F4550 (pinout is shown in Fig. 3.7) which is also a 40 pin DIP IC. In this work it will be termed as “DAS (Data Acquisition System) Controller”. Like the Microcontroller Board it is populated with separate connectors

to receive and transmit signals with its every pin. It possesses a RS232 serial communication system with MAX232 IC. It is also populated with LEDs and switches for further uses. A 4 MHz crystal is used for it. It's main task is to receive signal from Microcontroller Board and to transmit it to a PC.

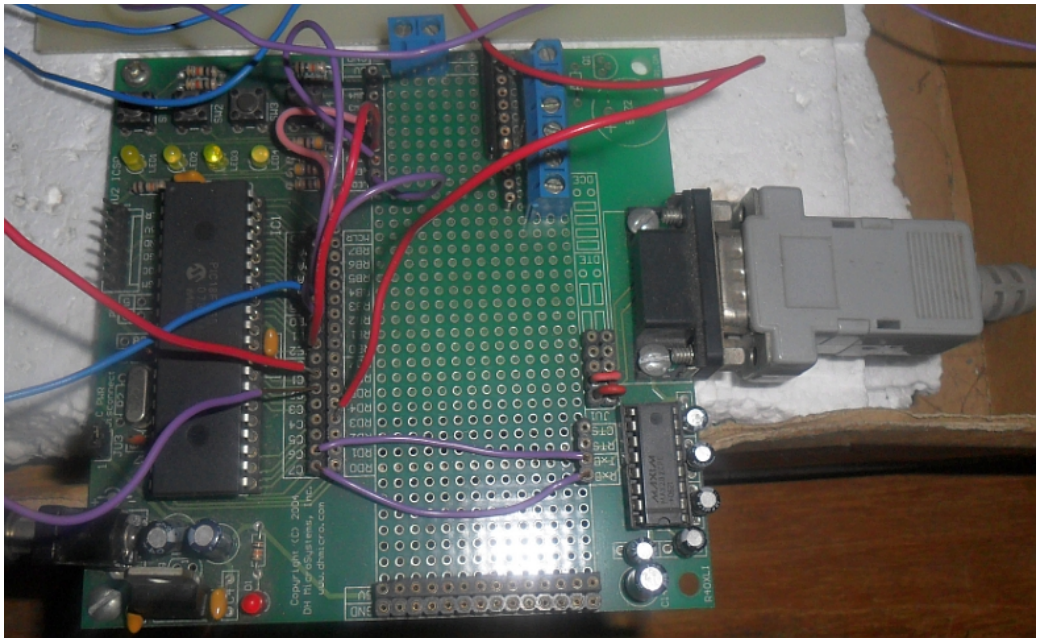


Fig. 3.6: DAS Control Board

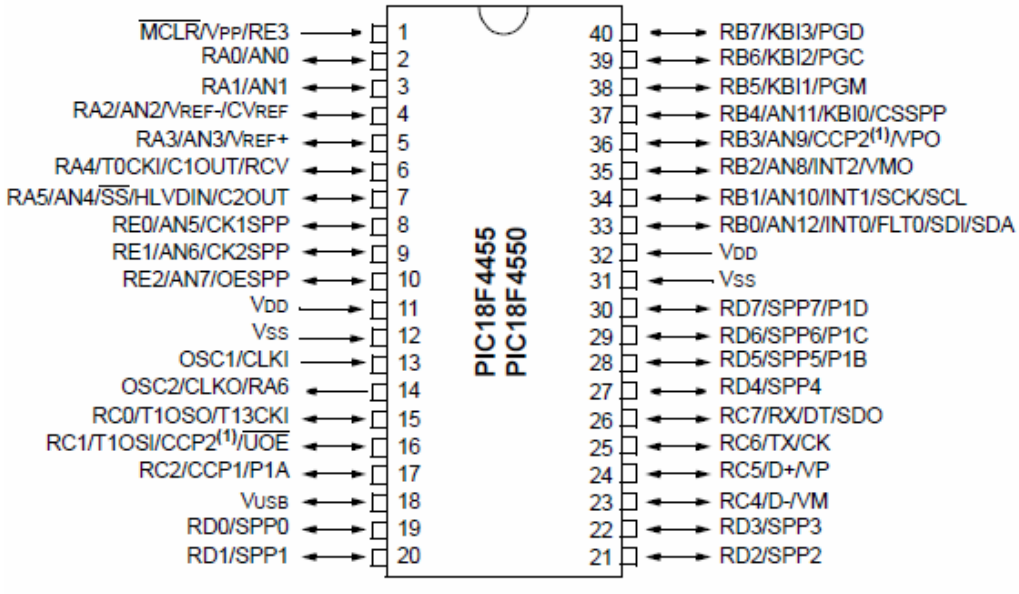


Fig. 3.7: Pinout of PIC18F452

3.2 Experimental Procedure

In the present study, speed of the drive motor is controlled using P, PI and PID controller action and controller performance is studied to optimize it. To control the speed, a feedback loop is created. The block diagram of control system for robot drive control test bench is shown in Fig. 3.8:

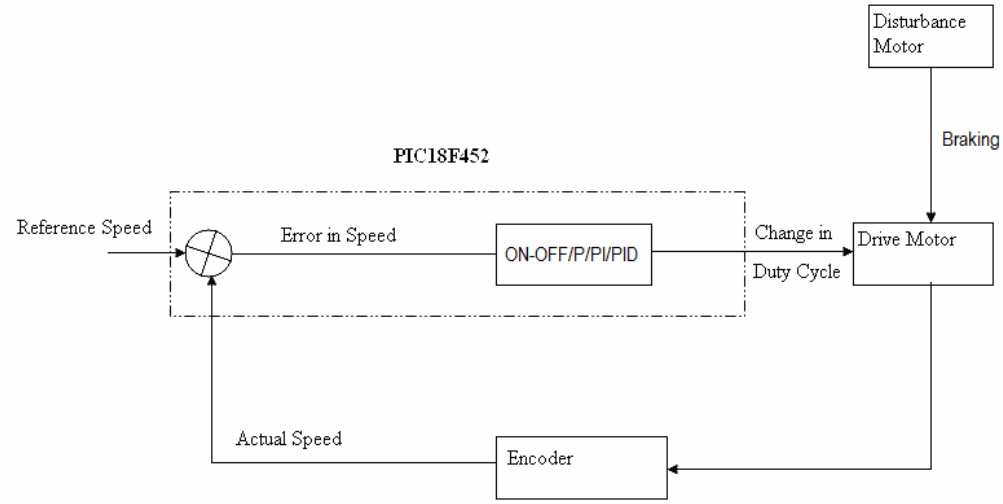


Fig 3.8: Block diagram of robot drive control

Motor controlling in the present study includes the following tasks

a. Reference Speed generation and measurement
b. Actual Speed measurement by DAS and Motor Controller
c. Error in speed calculation.
d. Control Action Timing.
e. Starting or Stopping of the Drive Motor.
f. Development and Tuning of controller.
g. Data Transmission

a. Reference Speed Generation and Measurement

Reference speed is the speed at which the motor is aimed at to rotate. It is to be compared with the actual speed of the motor to control the speed of the motor. It is noticed that the

minimum duty cycle at which the motor can start rotating is 60%, but once started it can run in much less duty cycle. It is observed that the maximum speed of motor at different duty cycle is different. For the convenience in control system it is decided to control the maximum speed of the motor as the maximum speed at 75% duty cycle. So that it can vary it's duty cycle for control action. Speed started increasing from 0(zero) rpm to maximum 1200 rpm at three different profiles after that reference speed becomes fixed at 1200 rpm. Any numerical value can be represented as reference speed. In this work, DAS controller generates a PWM signal of 3 KHz but of varying duty cycle. Duty cycle is varied from 10% to 90%. After reaching 90%, duty cycle stops increasing and becomes fixed till the motor stops. DAS controller generates this duty cycle and motor controller captures the ON period of PWM signal. DAS controller calculates the percentage of duty cycle and represents it into reference speed according to the method shown here and transmits it to PC. PWM generation of PIC18F4450 or PIC18F452 rely on the Timer2 and CCP2 peripheral. Timer2 is used as the basic counter generator for the PWM signal. To count, clock pulse is supplied by selectable prescale clock to Timer2, this prescale circuit will divide the system clock by 1, 4 or 16.

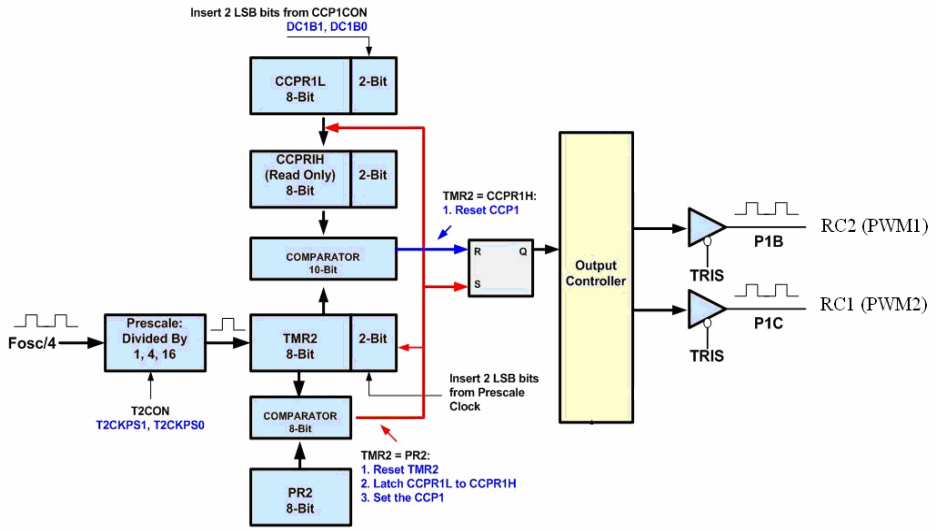


Fig 3.9: PWM Control Circuit of PIC18F452 and PIC18F4450

The TMR2 register value is continuously compared to the PR2 register which determine the TOP value of the TMR2 counter register. When the TMR2 register value reach the PR2 value, then the TMR2 counter register value will be reset to 0.

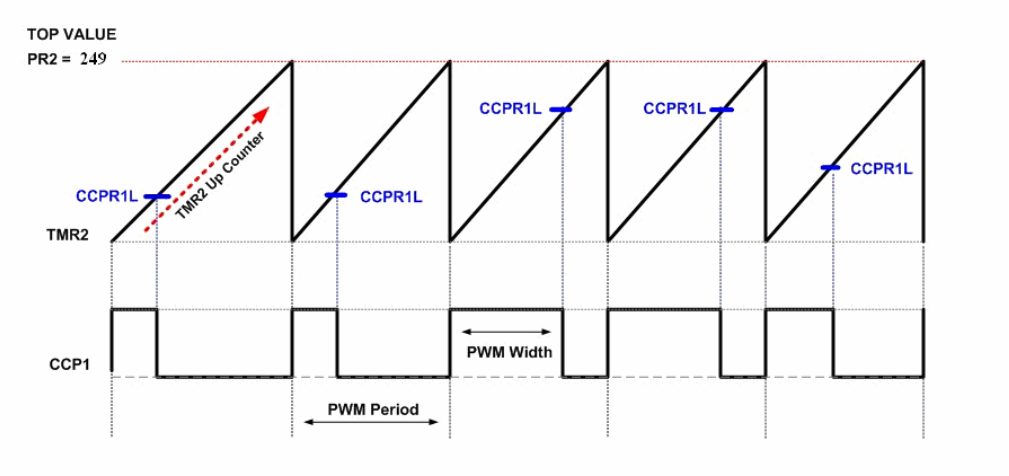


Fig 3.10: PWM frequency and duty cycle generation

At the same time the value of TMR2 counter register is also being compared to the CCPR1L register value (actually with the CCPR1H register value, since the CCPR1H equal to CCPR1L than we could say CCPR1L), when the TMR2 reach the CCPR1L value than the PWM peripheral circuit will reset the CCP1 output (logical "0") and when the TMR2 counter register equal to the PR2 register value then it will set the CCP1 output (logical "1"). Therefore by changing the PR2 value we could change the PWM period and this mean changing the PWM frequency as well. The PWM period could be calculated using this following formula:

$$\text{PWM period} = 4 \times \text{Tosc} \times (\text{PR2} + 1) \times (\text{TMR2 prescale value}) \quad (3.1)$$

Where Tosc is the system clock period in second

PWM frequency = 1 / PWM Period Hz

In this study by assigning the PR2 register value 249 and select the prescale to 16 with an internal system oscillator of 48 MHz and applying all these values to the formula above, PWM period and frequency are 0.000333 second and 3.00 K Hz. Frequency of the duty cycle became fixed but percentage of duty cycle varied to represent reference speed as shown here bellow:

$$DC = \frac{CCPR1L}{4*(PR2+1)} \quad (3.2)$$

PR2 register is loaded with 249, so if CCPR1L = $4*(PR2+1)$ = 4*(249+1) or CCPR1L =1000, percentage of duty cycle is 100. Percentage of duty cycle can be varied from 0 to 100 by varying CCPR1L register value from 0 to 1000. In this work Duty Cycle is varied from 10 to 90% then it became fixed. To do so CCPR1L register is loaded with values varying from 100 to 900 increasing 1 at each 0.01 sec. DAS controller generated varying duty cycle, converts it into reference speed in following way and transmits it to the PC. CCPR1L register value is assigned with a variable named Setup_DC and it is formulated to convert this value into Reference Speed as follows:

$$\text{Reference Speed} = -150 + 1.5* \text{Setup\_DC} \tag{3.3}$$

When Setup_DC is 100 then Reference Speed = 0 rpm and when Setup_DC is 900 then Reference Speed = 1200 rpm. As the value of Setup_DC increased upto 900, it stops increasing. Target is set to rotate the drive motor at following speed profile shown in Fig. 3.11:

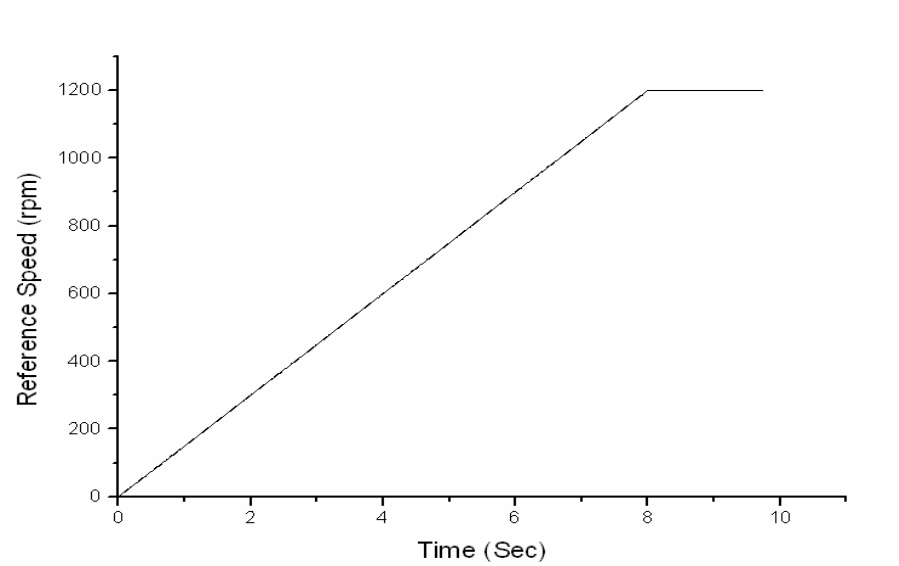


Fig. 3.11: Reference speed profile 2

In this speed profile speed is initially increased at a rate of 2.5 rev/sec^2 until it reaches 1200 rpm. In this work this speed profile is termed as Ref. speed profile – 2. To control motor at different speed profiles, value of Setup_DC is increased by 2 at every 0.01 sec. Then the speed profile becomes as follows:

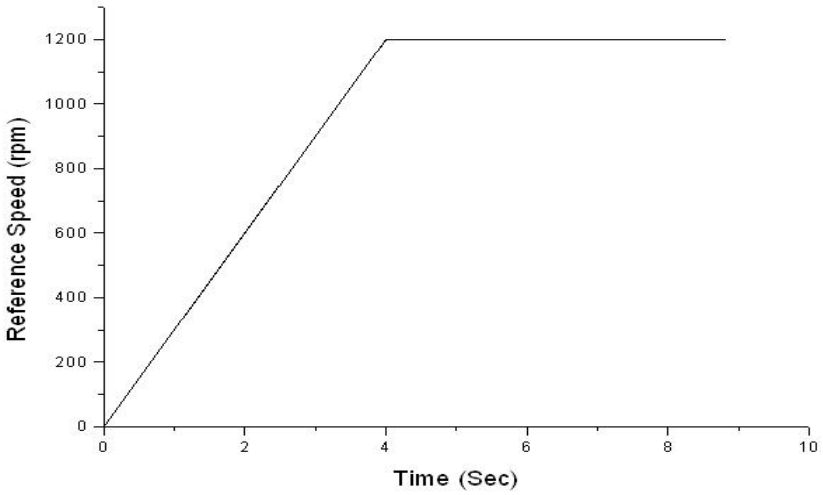


Fig. 3.12: Reference speed profile 3

In this speed profile initial rate of change of speed of the motor was double, 5 rev/sec^2. It is named as Ref. speed profile – 3 in present study shown in Fig. 3.12. For a very slow motor speed, Setup DC is increased by 1 at every 0.03 sec, which generated the following reference speed profile:

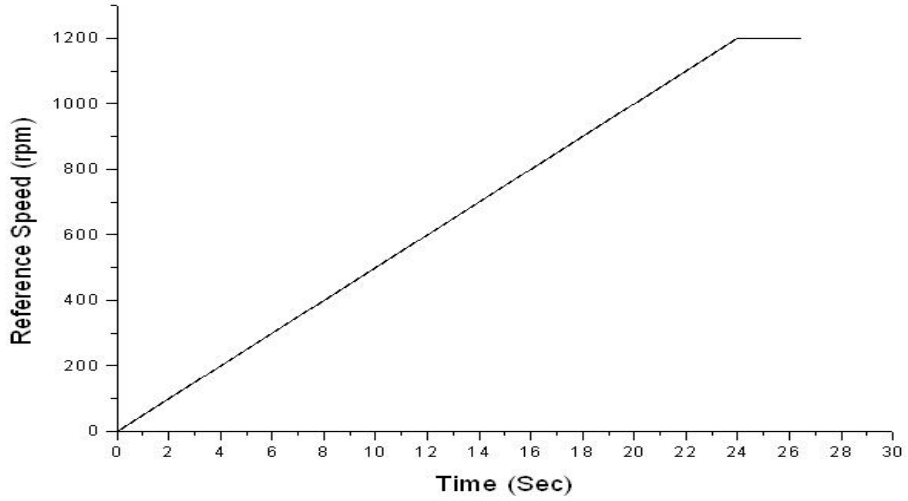


Fig. 3.13: Reference speed profile 1

In this speed profile speed is changed at an acceleration of 0.83 rev/sec^2 until it reaches 1200 rpm. This profile is named as Ref. speed profile – 1 shown in Fig. 3.13. Fig 3.14 compares the reference speed profiles.

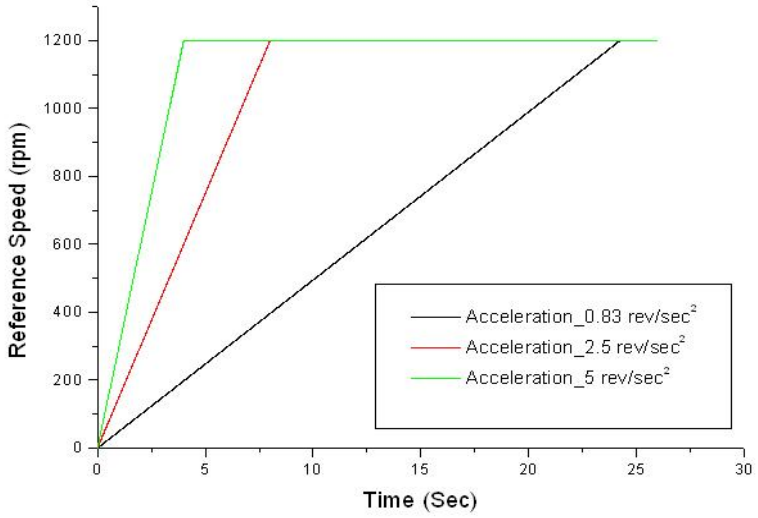


Fig 3.14: Comparison of Reference speed profile 1, 2 and 3.

This varying Duty Cycle generated by DAS Controller is transmitted to the Motor Controller where it is measured. ON periods of Duty cycle is measured capturing at two consecutive rising edges at every 0.01 second. PWM signal coming from RC1 (PWM2) pin of DAS Controller is connected to the CCP1 pin (RC2) of the Motor Controller(PIC18F452) as shown in Fig. 3.16. Motor Controller is so configured that when any rising edge signal is received, an interrupt flag (called CCP1IF) is raised and not cleared until Timer1 is not captured and saved in a variable (Temp_1). When cleared it is waiting for the further flag to be raised. Next flag is not cleared until Timer1 is not captured and saved in a variable (Temp_2). Subtracting these two variables Temp_2 and Temp_1, it is saved in another variable PWM_Capture which gives the measure of ON period of PWM signal. From which it is possible to calculate the Reference Speed follows (as shown in Fig 3.15):

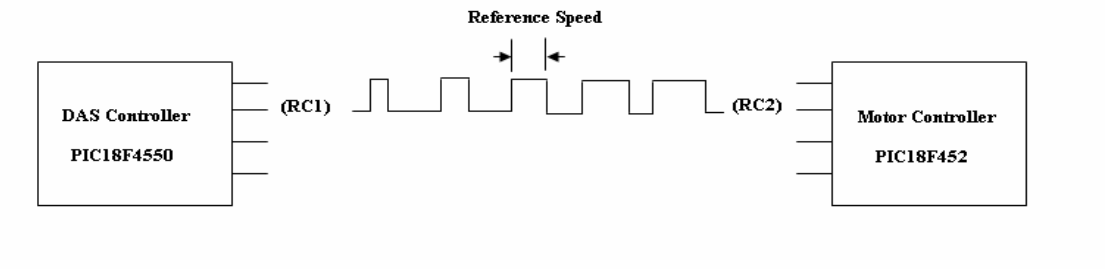


Fig 3.15: Reference Speed Measurement by Motor Controller

Used Peripheral: CCP1, TMR1

Used Pin : RC2

X-Tal : 4 MHz

F_{OSC} : 4 MHz

F_{TMR} : $\frac{F_{OSC}}{4}$ = 1 MHz

T_{TMR} : $\frac{1}{1*10^6}$ = 0.000001 sec

PS : 1

F_{PWM} : 3000 Hz = 3.00 KHz

T_{PWM} : $\frac{1}{3000}$ = 0.000333 sec

It is noted that Duty cycle is varied from 10 to 90% increasing CCPR1L value by 1 or 2 at every 0.01 sec or by at every 0.03 sec.

When Duty Cycle is 10%, ON period = 0.000333*10% = 0.0000333 sec, then captured timer ticks or PWM Capture = $\frac{0.0000333}{0.000001}$ = 33

Again when Duty Cycle is 90%, ON period = 0.000333*90% = 0.0002997 sec, then captured timer ticks or PWM Capture = $\frac{0.0002997}{0.000001}$ = 300

These numbers of timer ticks are so related with reference speed that when timer tick is 33 it represents a reference speed of 0 rpm and when it is 300 it represents a reference speed of 1200 rpm then it is fixed at this rpm. To do so Reference Speed and PWM Capture is related with the following expression:

Reference Speed = -148 + 4.5* PWM Capture (3.4)

Using this expression Reference Speed profile 2 with time is like as the fig 3.17 where CCPR1L value of DAS controller is increased by 1 at every 0.01sec:

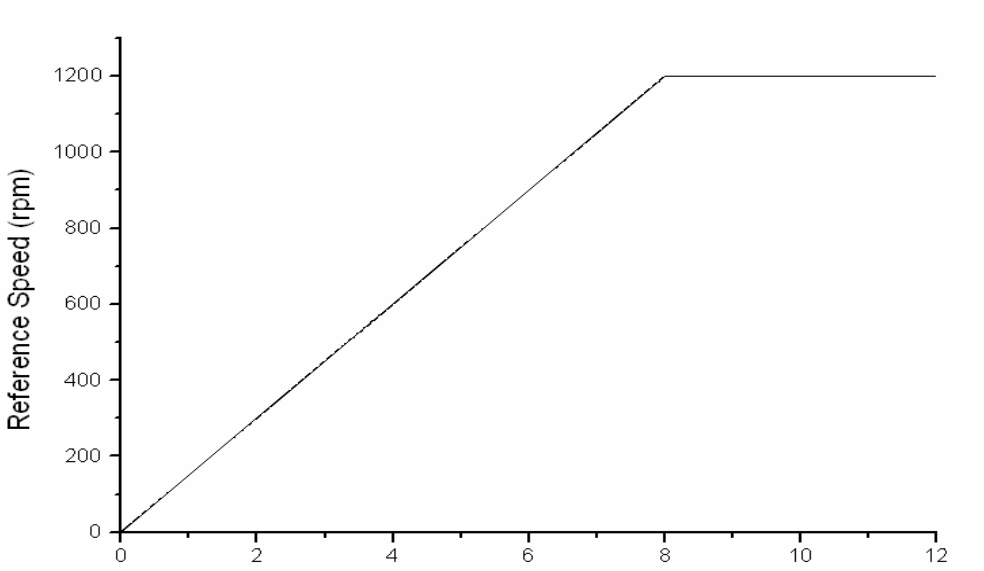


Fig 3.16: Measured reference speed by Motor Controller

Fig 3.16 shows that reference speed is same for DAS controller and Motor Controller at the same time.

Duty Cycle Generation of Motor Controller

To run drive motor PWM signal of 10 KHz frequency is generated from the (PWM2) RC1 pin of Motor Controller. To generate 10 KHz frequency when Motor Controllers F_{OSC} is 4 MHz, PR2 value is determined using the expression:

PWM Frequency = $\frac{F_{OSC}}{4*(PR2+1)*PS}$

⇨ PR2 = 99

So, 100% Duty Cycle obtained loading CCPR1L = 4(PR2+1)

⇨ CCPR1L = 400

To change Duty Cycle CCPR1L value can be varied from 0 to 400.

Duty Cycle measurement by DAS Controller

It has already been discussed that to run motor at different speeds duty cycle can be varied. To check that whether the control action is right or not, Duty cycle generated by the motor controller to be observed. DAS controller measure this Duty Cycle and transmit it to PC for further analysis.PWM signal coming from RC1 (PWM2) pin of motor controller is connected to the CCP1 pin (RC2) of the DAS controller. Controller is so configured that when any rising edge signal received, an interrupt flag (called CCP1IF) is raised and not cleared until the Timer1 is not captured and saved in a variable (Temp_1). When cleared it is waiting for the further flag to be raised. Next flag is raised when any falling edge signal is received and not cleared until Timer1 is not captured and saved in a variable (Temp_2) as shown in Fig.3.18. Subtracting these two variables Temp_2 and Temp_1 saved in another variable DC_Capture which gives the measure of ON period of Duty Cycle. From which it is possible to calculate the Duty Cycle as follows (shown in Fig. 3.17):

Frequency of PWM = 10 KHz

Period of PWM = $\frac{1}{10000}$ Sec = 0.0001 Sec

F_{OSC} for DAS Controller = 48 MHz

Frequency of Timer = 12 MHz

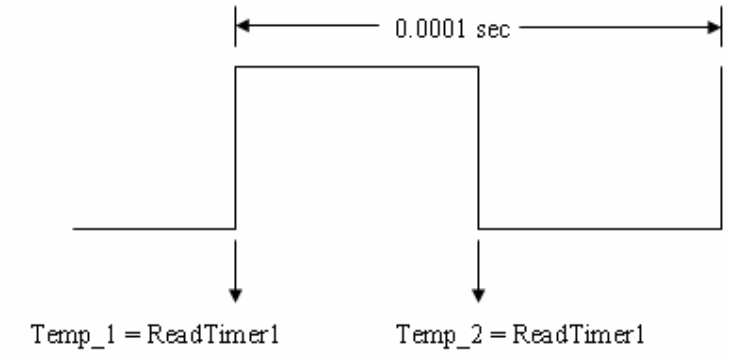


Fig 3.17: Measurement of Duty Cycle by DAS Controller

At every 1 sec number of timer ticks = 12E6

So, for 100% Duty Cycle (at 0.0001 sec) timer ticks = 12E6*0.0001 = 1200

From which Duty Cycle = $\frac{100*(Temp\_2 - Temp\_1)}{1200}$ %

$$\text{Duty Cycle} = \frac{(Temp\_2 - Temp\_1)}{12}\% \quad (3.5)$$

b. Actual Speed Measurement by DAS Controller and Motor Controller

Actual Speed is the speed at which the drive motor is actually rotating. It can be measured in two ways:

a) Measuring the time required to get two consecutive pulses from encoder.
b) Counting the number of encoder pulses received by the controller at a certain period of time.

It is possible to measure time required for two successive encoder pulses using any hardware interrupt and a timer or a Compare-Capture interrupt and a timer of controller. In the second way it can be measured with an external event counter and a timer. In this work all these three methods are implemented using DAS Board and their outcomes are compared to select the most precision method for speed.

i) Actual Speed Measurement by Hardware Interrupt and Timer

Interrupts are a feature of microcontroller. Timers, I/O pins, USARTs, A/Ds and other peripheral can cause interrupts. When an interrupt occurs, the code in the application suspended, an interrupt flag is raised and code in the interrupt subroutine executes until this interrupt flag is not cleared. When the flag is cleared, it executes a return from interrupt instructions and the program returns to where it left off. There are two types of interrupts in PIC Microcontroller according to the way they occur. If it occurs due to the functionality of some peripherals (like Timers, USARTs, A/Ds etc.), it is called peripheral or software interrupt. PIC18F4550 can be interrupted when a low to high or high to low signal is applied to some of it's specific I/0 pins also. These interrupts are called hardware or external interrupts. Named as INT0, INT1 and INT2 and uses RB0, RB1 and RB2 pin respectively. In this present study INT0 is used to measure the speed.

DAS controller (PIC18F4550) has four timers. They are referred to as TMR0, TMR1, TMR2 and TMR3. They can be used either as timers to generate a time delay, to measure time between external events or as counters to count events happening outside the controller. Every timer needs a clock pulse to tick. If internal clock source is used, then $1/4^{th}$ of the

frequency of the crystal oscillator ($F_{OSC}/4$) is fed into the timer. Therefore it is used to generate time delay or to measure time and is known as timer. By choosing the external clock option, pulses are fed through some timer input pins of controller and called as Counter. TMR0, TMR1 and TMR3 have 16 bits wide registers. As this microcontroller has 8 bit architecture, this 16 bit register is accessed as two separate 8 bits registers. Timer2 has 8 bit register. 16 bit timers can be loaded with any value between 0 to 65535. Then the value of timer register will be increased by 1 at a frequency of $F_{OSC}/4$. When this value crosses 65535, timer register resets and loaded with zero again. In the same way Timer2 can be loaded with a value up to 255 from 0. To create a large delay or to measure a long time Timers can be configured with a Prescale option. In this way a timer can be set to tick after getting a certain times more clock pulses. For an example if a 1:8 prescale is used, timer will tick once after getting eight clock pulses. In this measurement Timer3 is used with a 1:8 prescale.

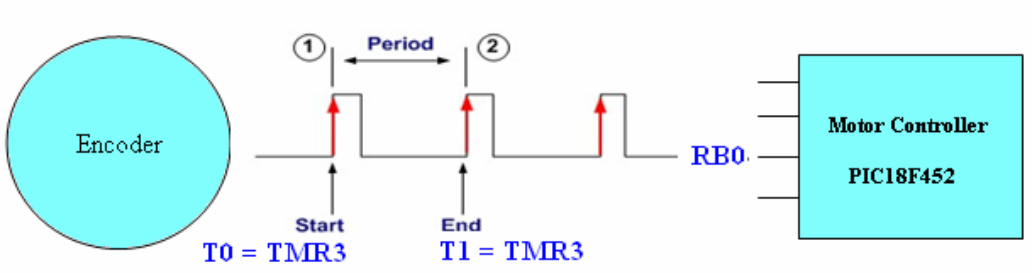


Fig 3.18: Encoder signals

Due to the rotation of drive motor optical sensor sends high signals to controller when IR passes through the holes of encoder. This signal coming out from encoder is connected to the INT0 (RB0) pin of DAS Controller (PIC18F4550) as shown in Fig. 3.19. DAS Controller is configured in such a way that when any rising edge of encoder signal is received at INT0 pin, an interrupt flag is raised. This flag is not cleared until the Timer3 register is not read and saved in a variable T0. After that flag is cleared and controller waits for the next rising edge, when received, it reads Timer3 register value again and saved it in a variable T1. Subtracting these two variables T1 and T0, required time of encoder to rotate between successive two holes is obtained. Using this time actual speed can be calculated as shown here. At every 0.01 sec actual speed is measured once.

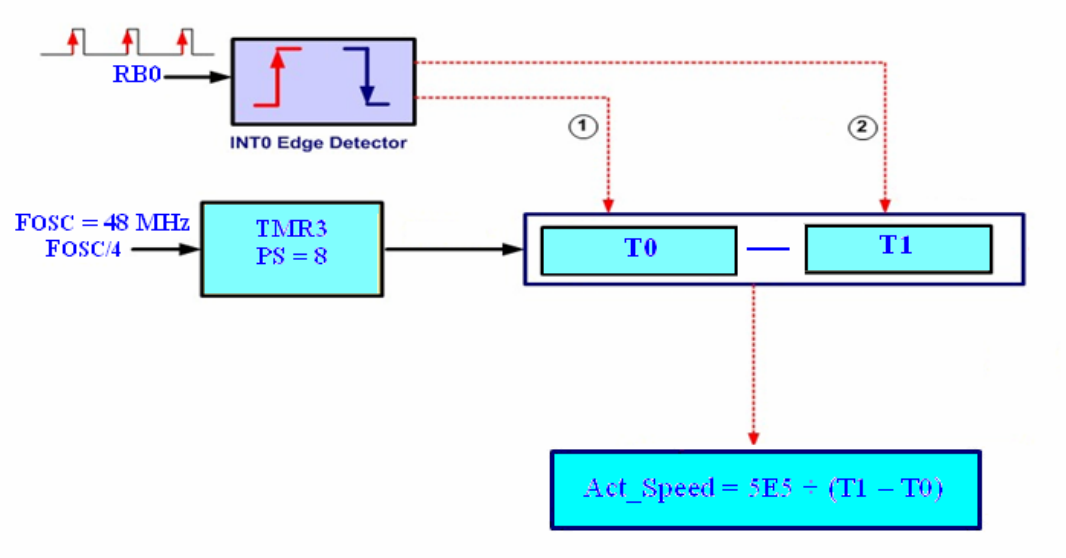


Fig 3.19: Actual Speed Measurement by DAS Controller

Used Peripheral : INT0, Timer3

Used Pin : RB0

X-Tal : 48 MHz

F_{OSC} : 48 MHz

Frequency of Timer, F_{TMR} : $\frac{F_{OSC}}{4}$ = 12 MHz

Period of Timer Tick, T_{TMR} : $\frac{1}{F_{TMR}} = \frac{1}{12*10^6}$ sec

Prescaler, PS : 8

Number of Encoder Holes : 180

Variables used : 1) T0 2) T1 3) Act_speed_INT0

Time required to rotate the encoder from one hole to next = T_{TMR}*PS * (T1 – T0) sec

Time required for one rotation of Shaft = T_{TMR}*PS* (T1 – T0)* 180 sec

So, Act_Speed_INT0 = $\frac{60}{T_{TMR} * PS * (T1 - T0) * 180}$ rpm

⇨ Act_Speed_INT0 = $\frac{12E6 * 60}{8 * (T1 - T0) * 180}$ rpm

$$\text{Act\_Speed\_INT0} = \frac{5E5}{(T1 - T0)} \text{ rpm} \qquad (3.6)$$

Following this method Actual Speed of the motor is measured five times while 75% Duty Cycle at 10 KHz PWM signal is applied to the motor. In fig 3.20 Actual Speed of the motor vs. time is shown:

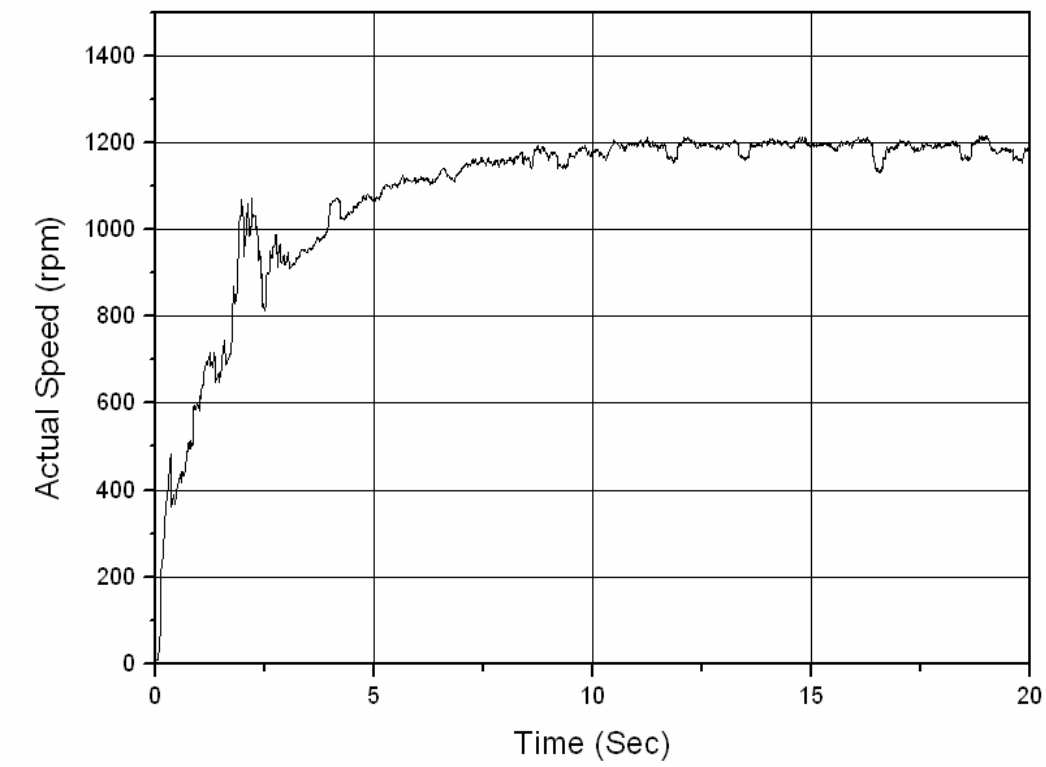


Fig 3.20: Actual Speed Vs. Time Using INT0

ii) Actual Speed Measurement by Compare-Capture Interrupt and Timer

The compare mode of the CCP module of microcontroller can cause an event outside of it. This event can be simply turning on a device connected to the CCP pins (RC2 and RC1) or the start of an ADC conversion as an example. This event is caused when the content of the Timer1 and Timer3 register is equal to register is equal to the 16 bit CCPR1H:CCPR1L register. To use the compare mode of the CCP, we must load the 16-bit Timer1 or Timer3 register with some initial values. As Timer1 or Timer3 counts up it's value is constantly compared with the CCPR1H:CCPR1L register and when a match occurs the CCP pin can perform one of the following in it's interrupt subroutine raising CCP1IF interrupt flag:

a) Drive the CCP pin high.
b) Drive the CCP pin low.

c) Remain Unaffected.
d) Trigger a special event with a hardware interrupt and clear the timer.
e) To use the CCP module in a compare mode CCP pin must be configured as output.

In Capture mode of the CCP module of Microcontroller, an event at the CCP pin will cause the content of the Timer1 or Timer3 to be loaded into the CCP (CCPR1H:CCPR1L) register. The event that causes the Timer1 (or Timer3) to be captured can be a high to low or low to high pulse. DAS controller (PIC 18F4450) has two CCPs named CCP1 and CCP2 with the pins RC2 and RC1 respectively. But for the Capture mode to work CCP pins must be configured as an input pin.

In the same way like the previous method, signal coming from encoder is connected to the CCP1 pin (RC2) of the DAS controller. Controller is so configured that when any rising edge signal received, an interrupt flag (called CCP1IF) is raised and not cleared until the Timer3 is not captured and saved in a variable (Speed_1). When cleared it is waiting for the further flag to be raised. Next flag is not cleared until Timer3 is not captured and saved in a variable (Speed_2). Subtracting these two variables Speed_2 and Speed_1 saved in another variable Speed_Capture which gives the measure of time required to rotate the encoder from one hole to the next. From which it is possible to calculate the Actual Speed of motor using the above method as follows:

$$\text{Act\_speed\_CCP1} = \frac{5E5}{Speed\_2 - Speed\_1} \tag{3.7}$$

At every 0.01 sec Actual speed is measured once.

Actual Speed of the motor is measured five times using CCP1 and Timer3 while 75% Duty Cycle at 10 KHz PWM signal is applied to the motor. In fig 3.21 Actual Speed of the motor vs. time is shown:

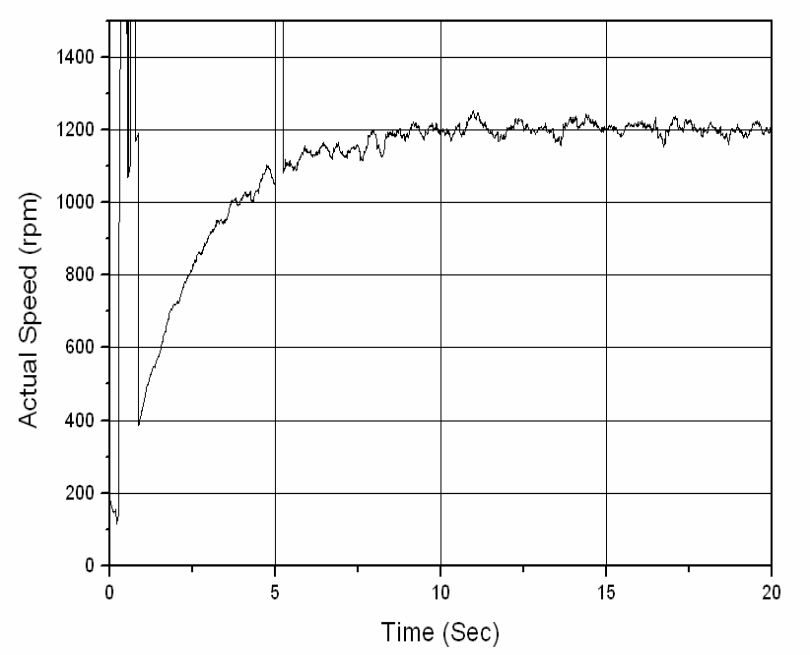


Fig 3.21 Actual Speed Vs. Time Using CCP1

iii) Actual Speed Measurement by an external event counter and a timer

Already discussed in method i) that by choosing the external clock option, pulses are fed through some timer input pins of controller and called as Counter. Using the counter it is possible to count how many signals come from encoder in certain period of interval with the aid of a Timer. In this method Timer0 is configured as counter. Encoder signal is fed into Timer0 input pin (RA4). Timer3 is used as a timer to count encoder signals at every 0.01 sec. Initially Timer0 is read into a variable R1 and after an interval of 0.01 sec Timer0 is read again into another variable R2. Subtracting R2 and R1, number of encoder signals per 0.01 sec is loaded into a variable Encoder_count. From which actual speed or rotation per minute is calculated. R2 is then loaded into R1. Again after 0.01 sec Timer0 is read and saved into R2. In the same way actual speed is measured as follows:

Used Peripheral : Timer0, Timer3
Used Pin : RA4
Number of Encoder Holes : 180

Variables used : 1) R2 2) R1 3) Encoder_count 4) Act_speed_TMR0
Encoder_count = R2 – R1

Number of encoder signal per 0.01 sec is Encoder_count

Number of rotation per 1 minute is $\frac{Encoder\_count * 60}{0.01 * 180}$

$$Act\_speed\_TMR0 = 33.33 * Encoder\_Count \quad (3.8)$$

Actual Speed of the motor is measured five times using Timer0 and Timer3 while 75% Duty Cycle at 10 KHz PWM signal is applied to the motor. In fig 3.22 Actual Speed of the motor vs. time is shown:

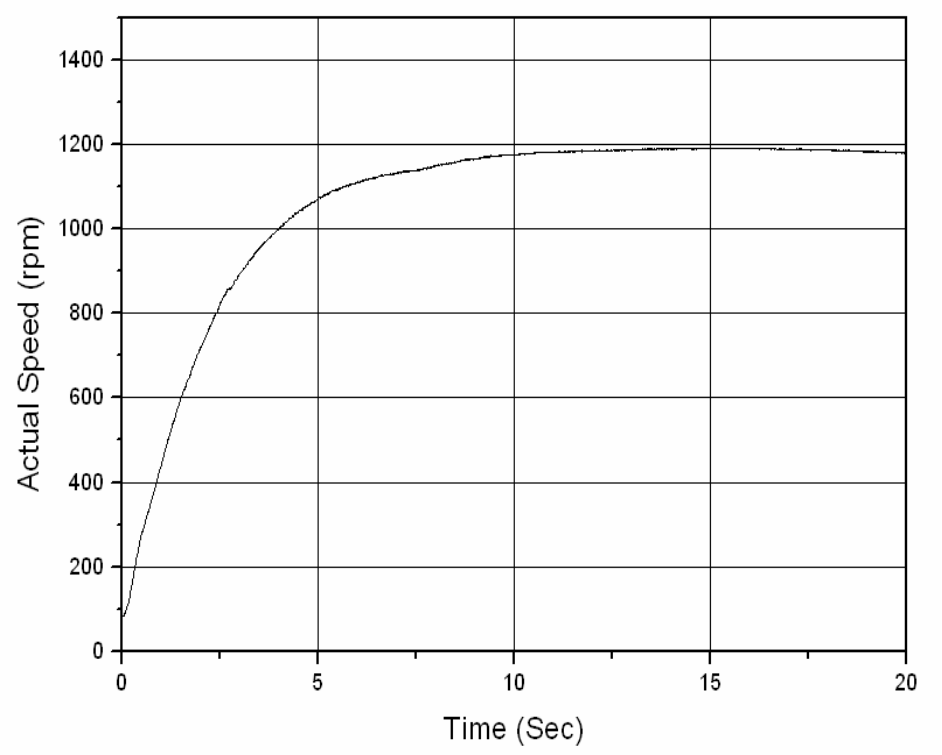


Fig 3.22: Actual Speed Vs. Time Using TMR0

Comparison:

In each of this method Actual Speed at 75% Duty Cycle is measured five times. Actual speeds of these measurements in each method are averaged and plotted against time to compare these methods.

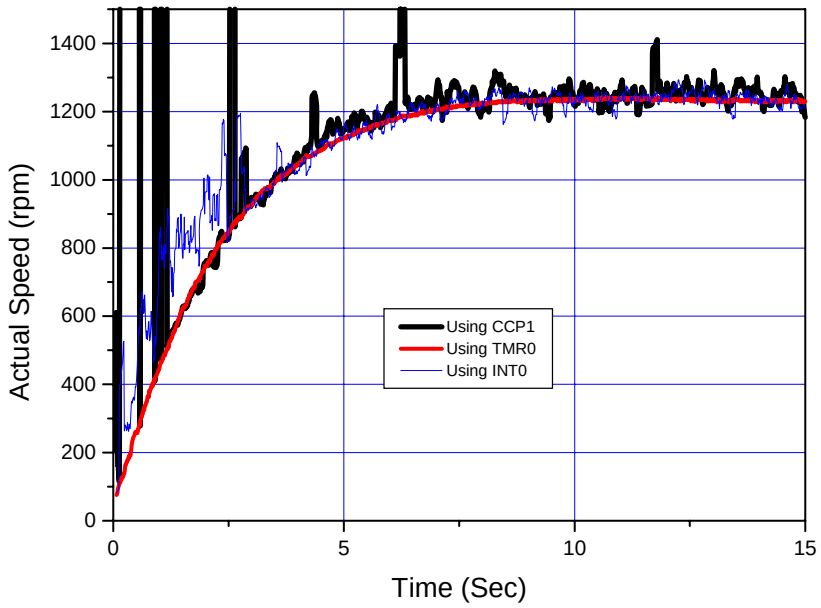


Fig 3.23: Comparison of Actual Speed Vs. Time

From the fig 3.23 it can be easily noticed that the trend of data in each method is almost same. Speed became fixed at 1200 rpm in each method. But data obtained using Timer0 and Timer3 (method-c) is more consistent. So for measuring the process reaction and for control action actual speed is measured in this way.

Actual Speed measurement by Motor Controller

Fig 3.23 shows that speed measured using Timer0 and Timer3 is more reliable. So Motor Controller measures actual speed at every 0.025 sec in this method as follows:

Used Peripheral: Timer0, Timer3

Used Pin : RA4

Number of Encoder Holes : 180

Variables used : 1) R2 2) R1 3) Encoder_count 4) Act_speed_TMR0

Encoder_count = R2 – R1

Number of encoder signal per 0.025 sec is Encoder_count

Number of encoder signal per 60 sec is $\frac{Encoder\_count * 60}{0.025}$

Number of rotation per 1 minute is $\frac{Encoder\_count * 60}{0.025 * 180}$

$$\text{Act\_speed\_TMR0} = 13.33 * \text{Encoder\_Count} \quad (3.9)$$

c. Error in Speed

Error in speed is the deviation of speed from reference speed to actual speed. Control action is generated based on this error. Error in speed can be either positive or negative. If actual speed is less than reference speed, error becomes positive. On the other hand when actual speed is greater than reference speed, error becomes negative. Error in speed is simply calculated using the following expression:

$$\text{Error} = \text{Reference Speed} - \text{Actual Speed} \quad (3.10)$$

d. Control Action and Data Transmission Timing

Control action time is the time interval between two successive error checking or control action generations. In this study it was an important task to time the control action properly. A very slow control action may cause very large error in the system which in turn may produce very large control action, resulting an unacceptable overshoot of actual speed. In a contrary, if control action is very fast, controller may not have adequate time to measure the actual speed, reference speed and duty cycle which may cause inappropriate control action. Data transmission time is the time at which data is transmitted once to a PC. It is now clear that 0.025 sec is enough for Motor Controller and 0.01 sec is enough for DAS Controller to measure and send or control the parameters.

To take control action at every 0.025 sec, Timer3 of motor controller (PIC18F452) is loaded initially with a value so that it raises an interrupt flag after 0.025 sec. This value which resets Timer3 at 0.025 sec is calculated as follows:

Timer3 Register : 16 bit

Timer3 resets at : 65536

Timer3 Prescaler : 1

At 0.01 sec, Number of Timer Ticks = $\frac{0.025}{T_{TMR} * \text{Pr}escaler} = \frac{0.025}{0.000001 * 1} = 25000$

So, Timer3 register was initially to be loaded with the value = 65536 – 25000 = 40536

To transmit data at every 0.01 sec, Timer3 of DAS controller (PIC18F452) is loaded initially with another value calculated as:

Timer3 Register : 16 bit

Timer3 resets at : 65536

Timer3 Prescaler : 8

At 0.01 sec, Number of Timer Ticks = $\frac{0.025}{T_{TMR} * \Pr escaler} = \frac{0.01}{0.0000000833 * 8} = 15000$

So, Timer3 register was initially to be loaded with the value = 65536 – 15000 = 50536

e. Starting or Stopping of the Motor

Starting or stopping of the drive motor is an important task in robot control. Motor is started initially with supplying a duty cycle and energizing specific relays of H-bridge. Once started signal coming from encoder is counted by Motor Controller. To do this motor controller engages its external event counter Timer0. Timer0 can be configured as a 16 bit counter. So it can count upto 65535 encoder signals. If the drive motor is supposed to rotate a certain rotation, total number of encoder signals for that rotation is calculated first. Timer0 is then initially loaded with such a value that it resets after counting required signals. For an example to count 30000 encoder signal Timer0 is initially loaded with a value (65535 – 30000) 35535. So after counting 30000 signals Timer0 register reaches its maximum value 65535 and raises an interrupt flag. In this interrupt subroutine PWM signal supplied to DC motor stopped and after that relays power are cut to stop the motor.

f. Development and Tuning of Controller

Development of a controller is to establish a relation between input and output of controller. For different control modes this relationship also varies. In this work based on the error in speed duty cycle of motor controller will be varied which in turn will change the actual motor speed.Tuning is a process of optimizing the control action. Referring chapter-4 it can be said that changing the controller parameters Kp, Ki and Kd controller performance is varied. Using Zeigler-Nickols second rule or Zeigler-Nickols method for controller tuning controller performance is optimized in this study.

Development of ON-OFF Control System

An ON-OFF controller has only two outputs. In this mode controller output is either 100% or 0% based on error. Here,

Duty Cycle = 0% when $e \leq 0$, or

Duty Cycle = 100% when $e > 0$.

As DAS controller can't measure 100% and 0% Duty Cycle, to run Drive motor, PWM signal of 10 KHz frequency with 90% or 10% Duty Cycle is generated from the RC1 (PWM2) pin of the motor controller. Motor is initially started with 90% Duty Cycle for 0.025 sec then started controlling the motor speed. Reference Speed was measured first, and then the actual speed at which the motor is rotating. At every 0.025 sec it is checked that whether Reference Speed is greater than Actual Speed or not. If it is found that Reference Speed is greater than Actual Speed, motor controller raise it's Duty cycle to 90% otherwise it keeps it's duty cycle at 0%. Algorithm of ON-OFF controller is shown in Fig. 3.24.

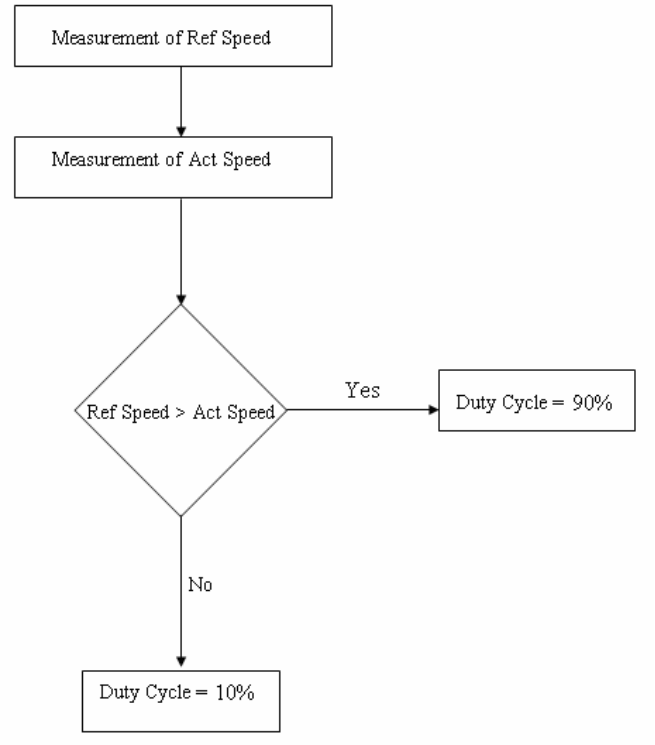


Fig 3.24: Algorithm of ON-OFF Control

ON-OFF control is used to control motor speed at mentioned three different speed profiles.

Development and Tuning of Proportional (P) Control

During Proportional Control, controller takes action proportionally to the error. Initially motor is started with a duty cycle of 75%. Once started at every 0.025 sec motor controller measures reference speed, actual speed and calculate error (e). Depending on this error controller decides to change its duty cycle multiplying error with a proportional gain but initial duty cycle is always added with it. So when error becomes zero this controller delivers initial duty cycle.

In this chapter duty cycle is already discussed in detail that Motor Controller supplies 100% duty cycle when its register CCPR1L is loaded with 400 if its PR2 register is already loaded with 99. y of the above expression is assigned as the value of CCPR1L register in motor controlling program. So the value of manipulated variable y can be varied from 0 to 400 to supply 0 to 100% duty cycle to the motor. But for ease in measurement and control duty cycle varied within a range from 50 to 90%. To do this value of y is varied from 200 to 360.

To tune the proportional controller value of y is started increasing from a very small Kp. In this study Kp is started increasing from 1/16 and increased upto 200. While tuning the controller it is to check that at a certain Kp actual speed will most closely follow the reference speed. Algorithm of P controller is shown in Fig. 3.25.

Development and Tuning of Proportional Integral Controller

During Proportional Integral (PI) Control, I control is added with P control. P Controller of PI controller is developed in the same way as P Controller. I controller is developed considering the accumulation of error. Initial error is taken as Error_0 and initialize with a value zero. Motor started with the initial duty cycle 75%. After rotating 0.025 sec error is calculated again and named as Error_1. To find the Integral error, Error_0 and Error_1 are added and multiplied by the control action time. Integral action is generated by multiplying the integral error by an integral gain and Error_1 is saved as Error_0. This duty cycle is supplied to the motor. After 0.025 sec error is calculated again and saved as Error_1. So in the same way a new duty cycle is generated. And the same cycle is repeated until the drive motor reaches the destination.

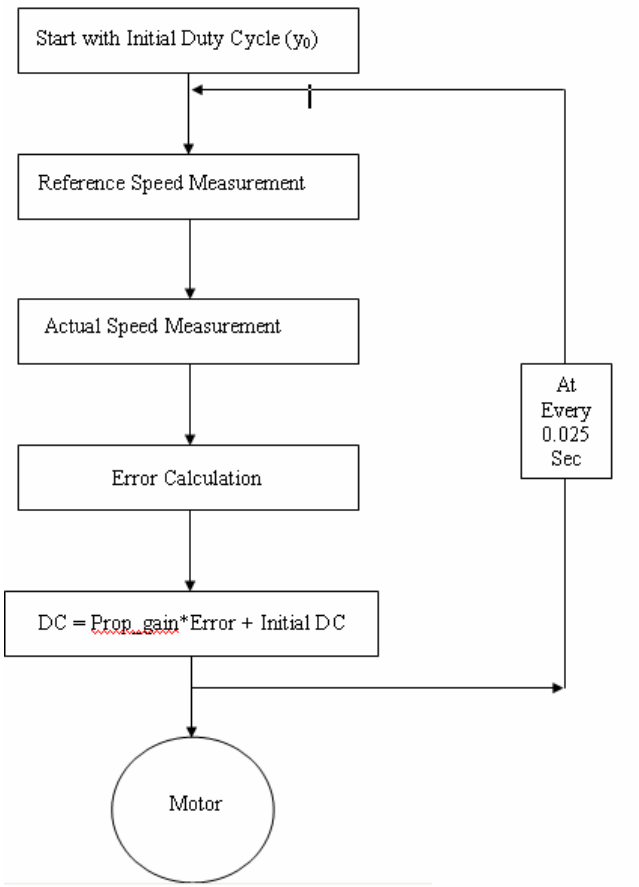


Fig. 3.25: Algorithm of P controller

PI controller is tuned by using Zeigler-Nichols rule. According to this method

Kp = 0.5 Kc

⇨ Kc = 2 Kp

During the tuning of proportional controller an optimum value of Kp is obtained. Using this Kp, Kc can be find out by just doubling this Kp. Once Kc is known proportional gain for PI controller is determined by the following relation:

⇨ Kp = 0.45 Kc

Again Tc = $\frac{1}{Kc}$, as per Zeigler – Nichols second rule

⇨ $T_I = \frac{T_c}{1.2}$

⇨ $K_I = \frac{1}{T_I}$

So using the value of Kp from proportional control method required Kp and K_I for PI control method can be obtained. Fig. 3.26 shows the PI Control Algorithm.

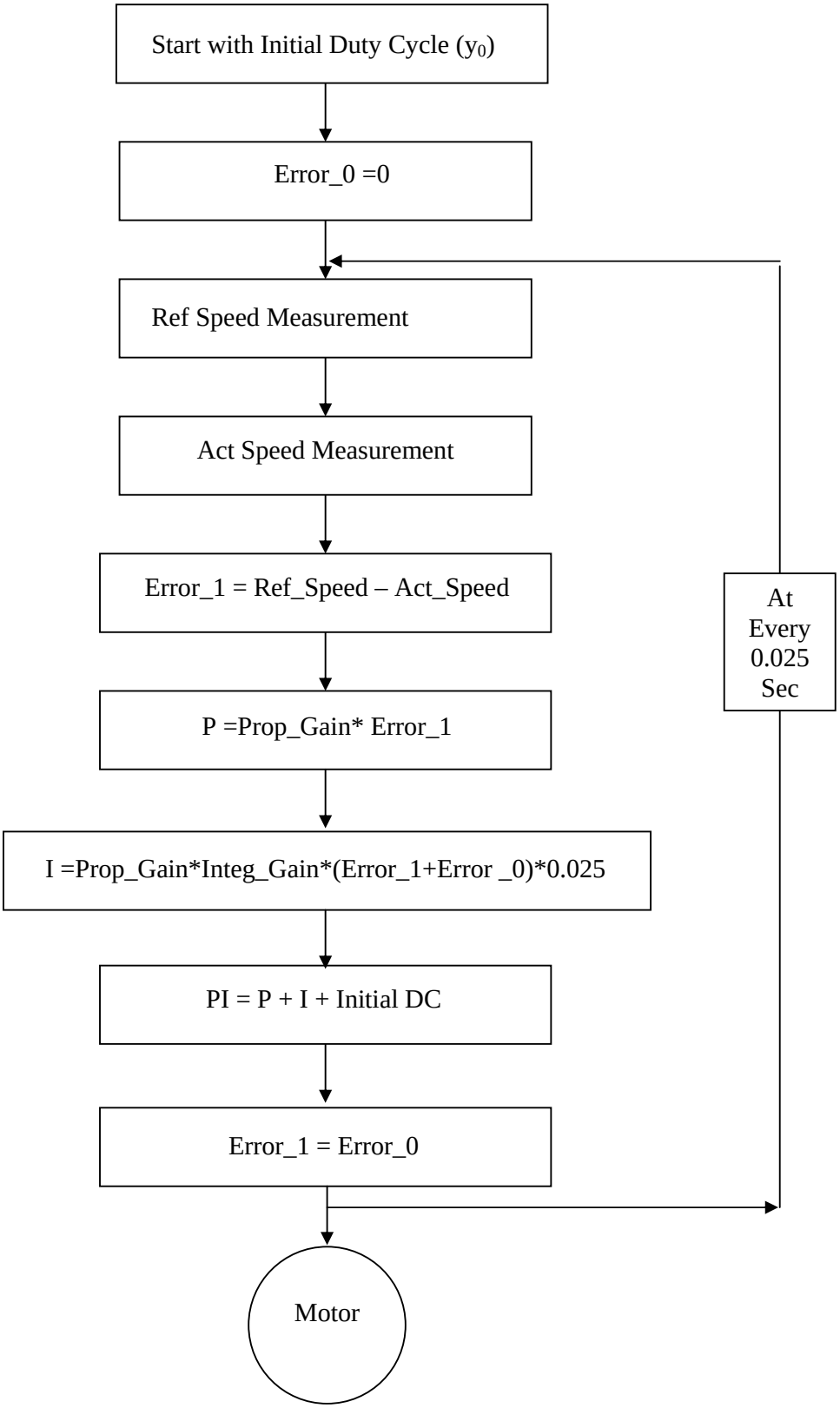


Fig. 3.26: Algorithm of PI Control System

Development and Tuning of Proportional Integral Derivative (PID) Controller

During this control mode D (Derivative) control is added with previous two control mode P (Proportional) and I (Integral) control mode. D controller is developed based on the rate of change of error. For the increasing rate of change of positive error duty cycle is decreased and for negative error change rate it increases the duty cycle.

Tuning of PID controller is a task to find the optimum values of Kp, K_I and K_D. It is noticed that the value of K_D is to be chosen very carefully. It is to be chosen as a very small number. Otherwise in case of any change in speed can cause a big action which may produce instability or sudden rising or lowering of Duty Cycle. In this study considering a very small value of K_I and K_D, Kp is increased to find the optimum control. After that values of K_I and are K_D are changed a little bit to check their effect on drive. All of the above control modes are used to control three different speed profiles. Fig. 3.27 shows the algorithm of PID control system.

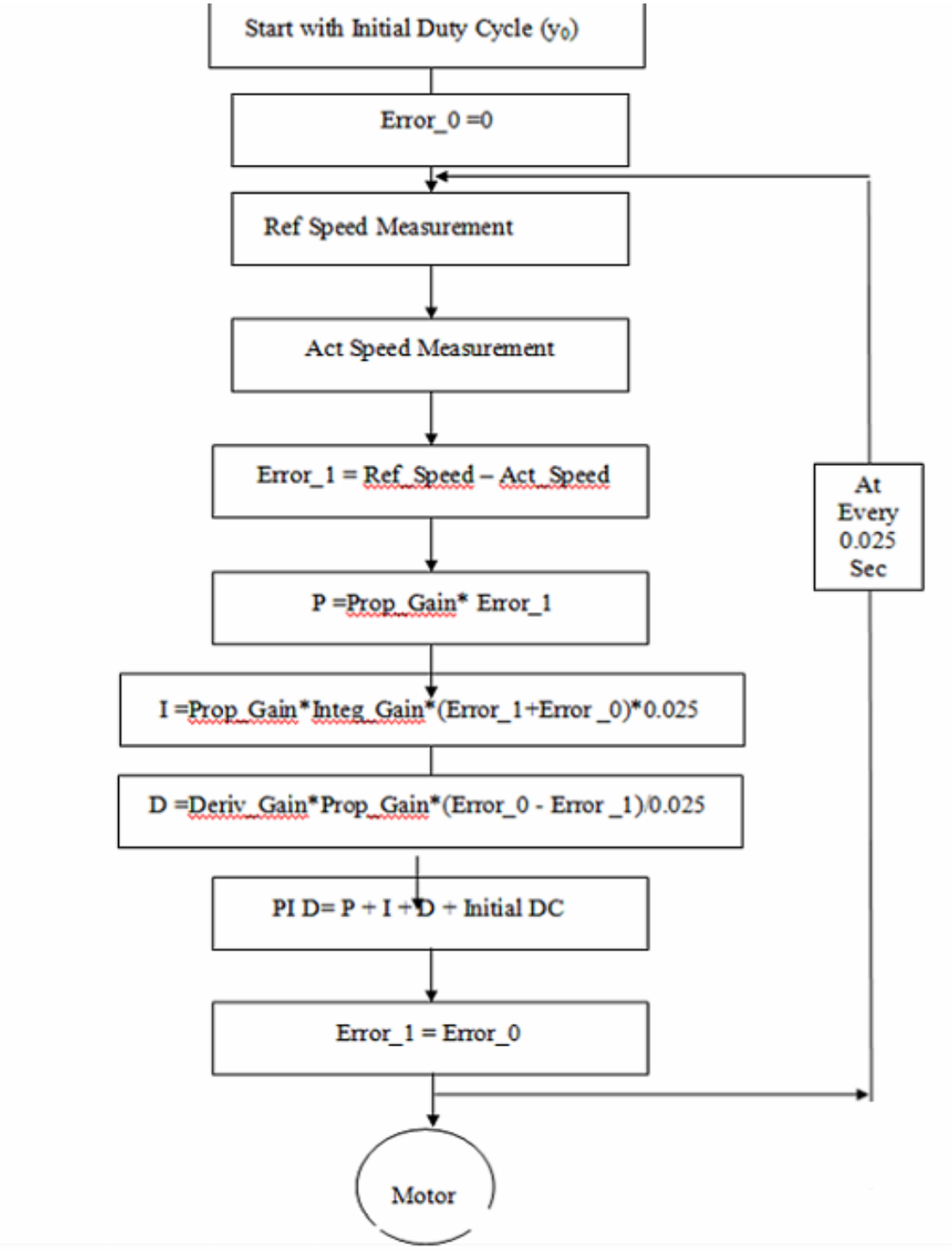


Fig. : 3.27: PID Control Algoithm

h) Data Transmission

In this study reference and actual speed of drive motor are measured form which error in speed is calculated. Depending on this error duty cycle is supplied to the motor. For further analysis and to check controller performance all these data (reference speed, actual speed and duty cycle supplied to the motor) are transmitted to a computer serially following RS232 standard.

Most digital messages are vastly longer than just a few bits. Because it is neither practical nor economic to transfer all bits of a long message simultaneously, the message is broken into smaller parts and transmitted sequentially. Bit-serial transmission conveys a message one bit at a time through a channel. Each bit represents a part of the message. The individual bits are then reassembled at the destination to compose the message. In general, one channel will pass only one bit at a time. Thus, bit-serial transmission is necessary in data communications if only a single channel is available. Bit-serial transmission is normally just called serial transmission.

For the most common serial protocol, data is sent in small packets of 10 or 11 bits, eight of which constitute message information. When the channel is idle, the signal voltage corresponds to a continuous logic '1'. A data packet always begins with a logic '0' (the start bit) to signal the receiver that a transmission is starting. The start bit triggers an internal timer in the receiver that generates the needed clock pulses. Following the start bit, eight bits of message data are sent bit by bit at the agreed upon baud rate. The packet is concluded with a parity bit and stop bit. One complete packet in Fig. 3.28.

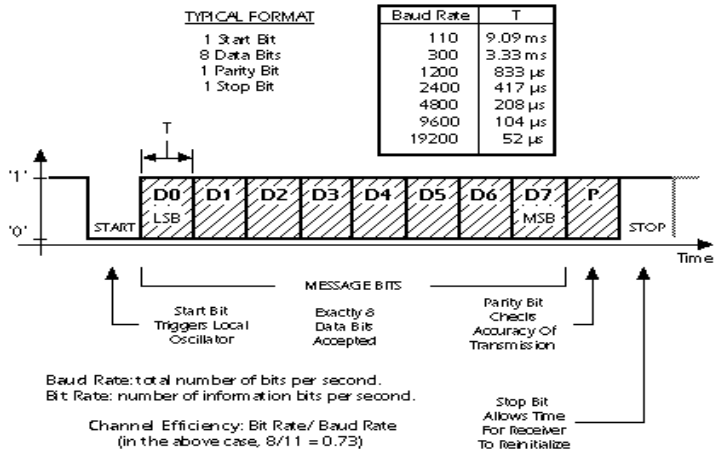


Fig 3.28: Data Packet

The baud rate refers to the signaling rate at which data is sent through a channel and is measured in electrical transitions per second. In the RS232 serial interface standard, one signal transition, at most, occurs per bit, and the baud rate and bit rate are identical. In this study data are sent at a baud rate of 115200. The baud rate in DAS Controller is programmable. This is done with the help of an 8-bit register called SPBRG. For a given crystal frequency, the value loaded into the SPBRG is dictated by the following formula:

F_{OSC} = 48 MHz

For high speed data acquisition, desired Baud Rate = $\frac{F_{OSC}}{16(SPBRG+1)}$

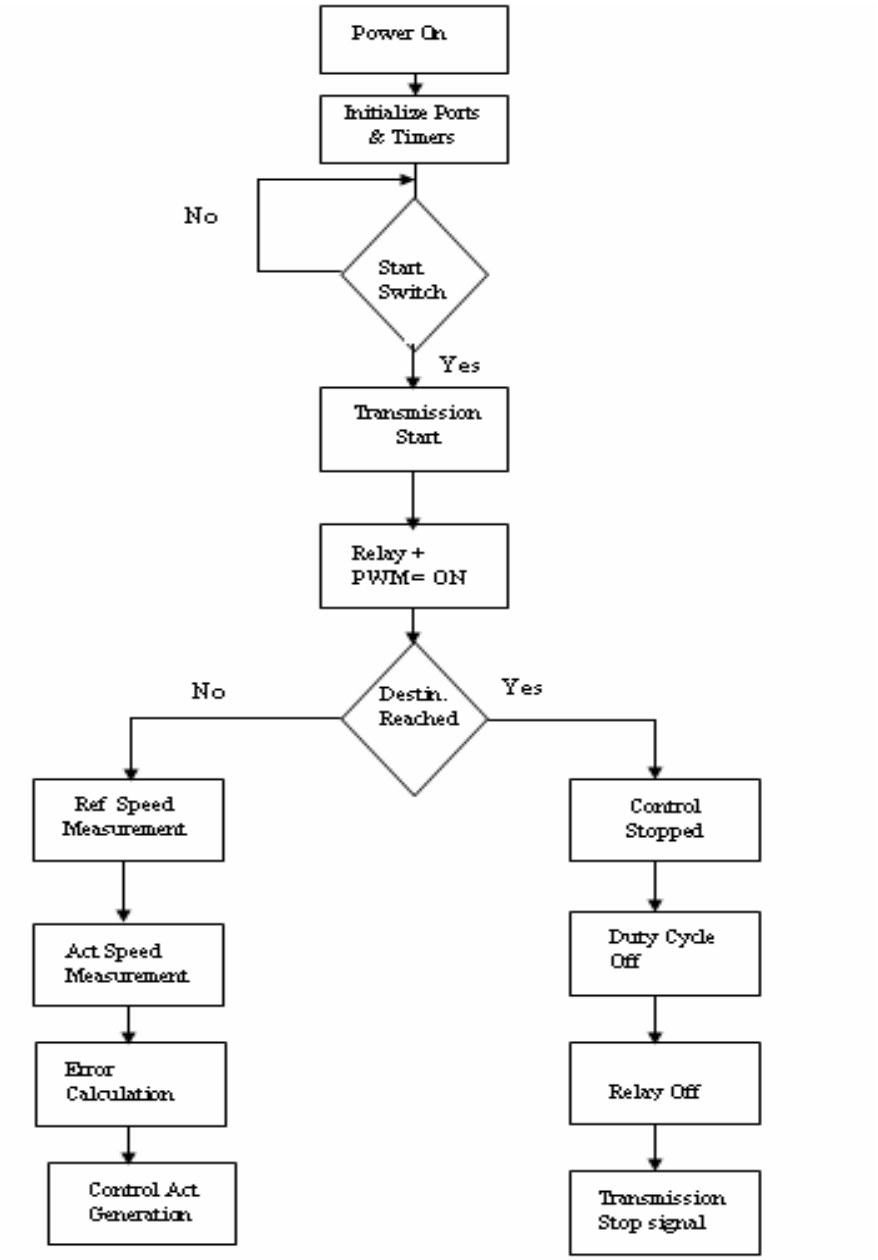
For a baud rate of 115200, X is calculated as:

- ⇨ SPBRG = $\frac{48000000}{115200*16}$ -1
- ⇨ SPBRG = 25

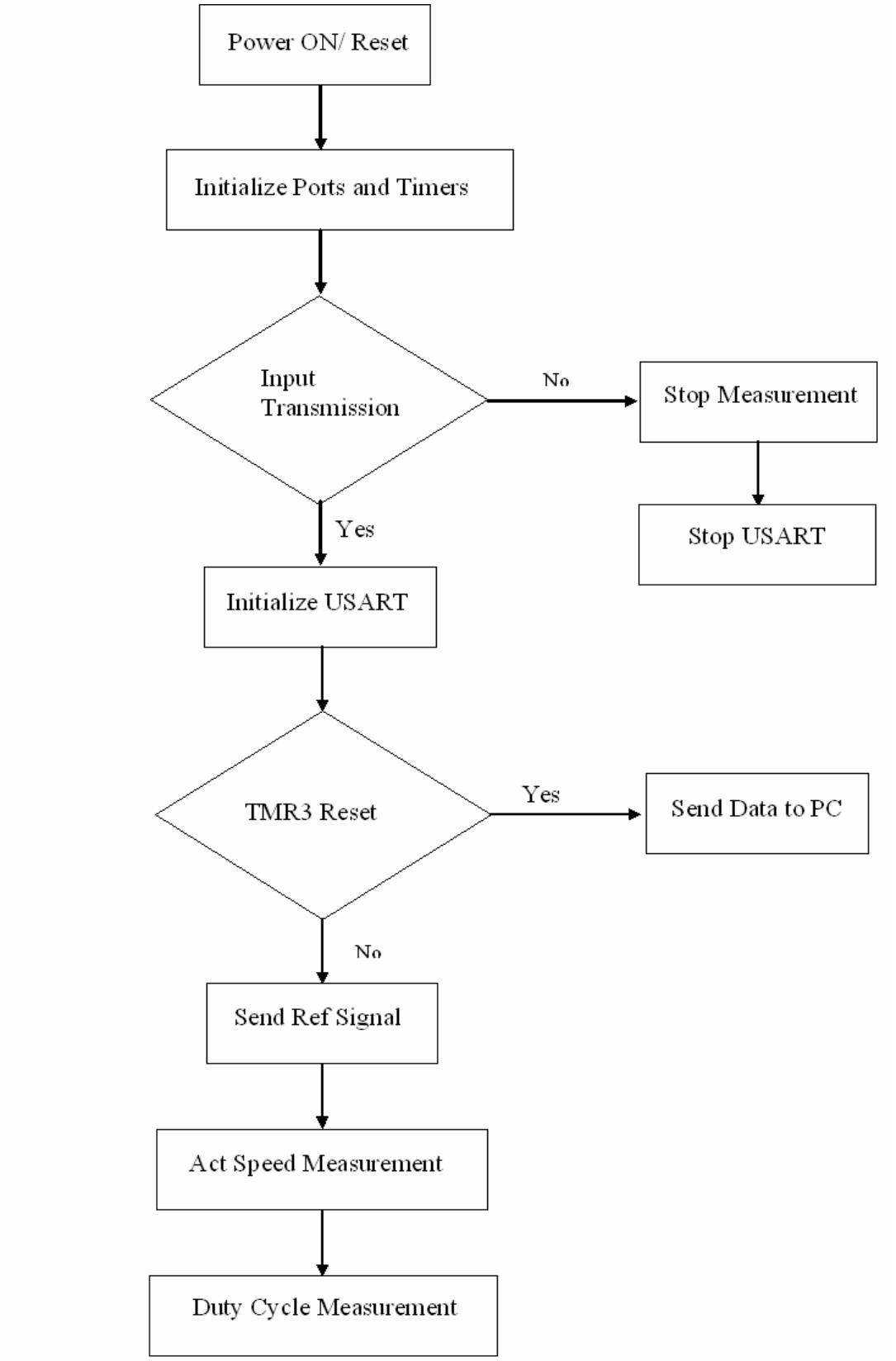
3.3 Control Algorithms

In this study robot drive control includes two tasks, one is to control the motor and the other is to monitor this control action sending live data to a computer. Motor Controller (PIC18F452) is programmed to control motor according to the motor control algorithm and DAS controller is programmed to transmit data according to DAS Control algorithm.

a) Motor Control Algorithm

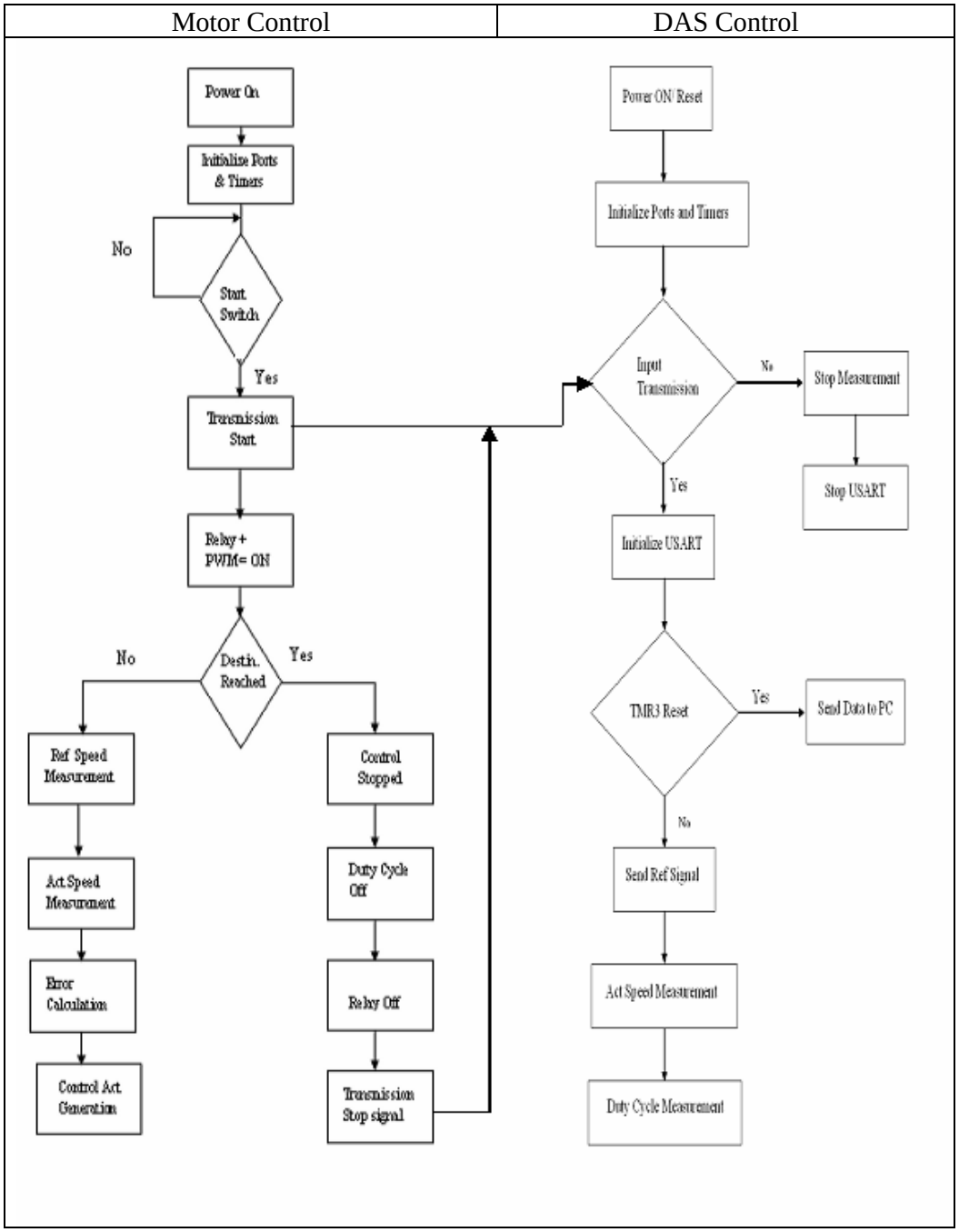


(b) DAS Control Algorithm



(c) Synchronization of Motor Control and DAS Programming:

To acquire the real control data, it is important to transmit data at the time when it is generated by Motor Controller. So Motor control program and DAS control program must be synchronized. DAS controller does not start sending data until it receives input transmission signal from motor controller and stops sending data when receives transmission stop signal from motor controller.



CHAPTER 4

Results and Discussions

In the present study, a drive motor is expected to rotate at three different speed profiles, shown in Fig. 4.1. Experiments are carried out with ON/OFF, P, PI and PID control mode for all these three speed profiles. Parameters of these controllers are tuned to find their optimum values. Motor controller measures reference and actual speed to calculate error and generate control action at every 0.025 sec. DAS controller generates reference speed signal, measure actual speed and duty cycle and transmit it to PC at every 0.01 sec.

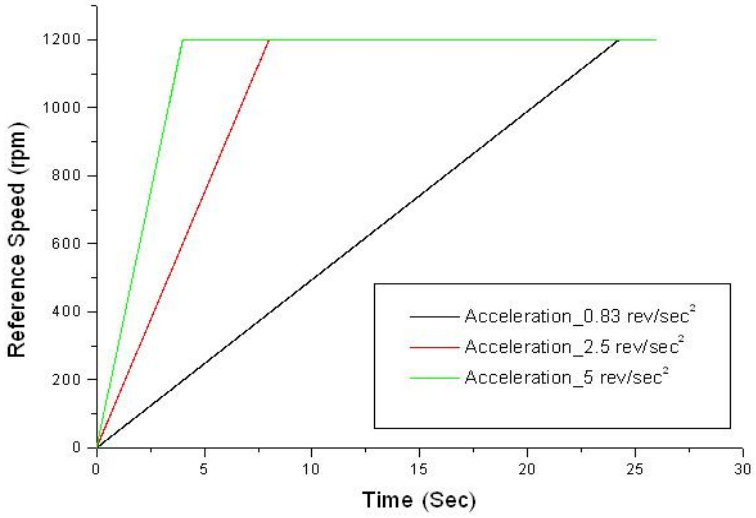


Fig. 4.1 Reference speed profiles for motor-drive system.

4.1 Results and Discussions

In Fig 4.2 actual speed, reference speed and duty cycle are plotted as a function of time when the motor is expected to rotate at reference speed profile 1 using ON-OFF control. At the very beginning motor was failing to rotate at reference speed when it was very low. After 2.46 sec when actual speed was increased to 1000 rpm, control system responded and started decreasing its speed to follow the requisite speed. At 5.53 sec actual speed dropped bellow the reference speed then it started following the desired speed closely.

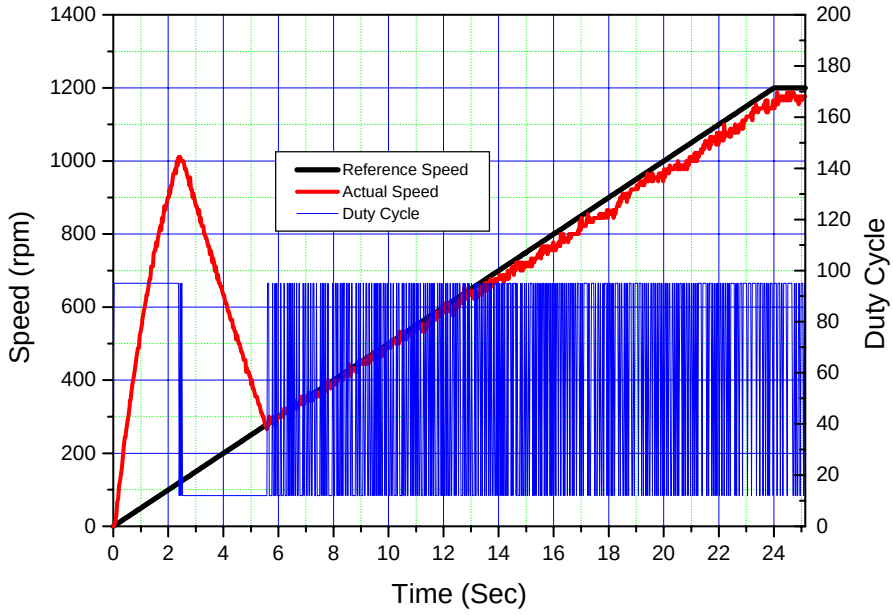


Fig 4.2: ON – OFF control for Reference Speed Profile 1

Fig. 4.3 shows the drive motor's response while motor was rotating to follow the reference speed profile 2 using ON-OFF control. Initial uncontrolled period was reduced to 0.92 sec. Maximum uncontrolled speed was also reduced here to 500 rpm then control system responded and started decreasing its speed to follow the requisite speed. At 1.87 sec actual speed dropped bellow the reference speed then it started following the desired speed.

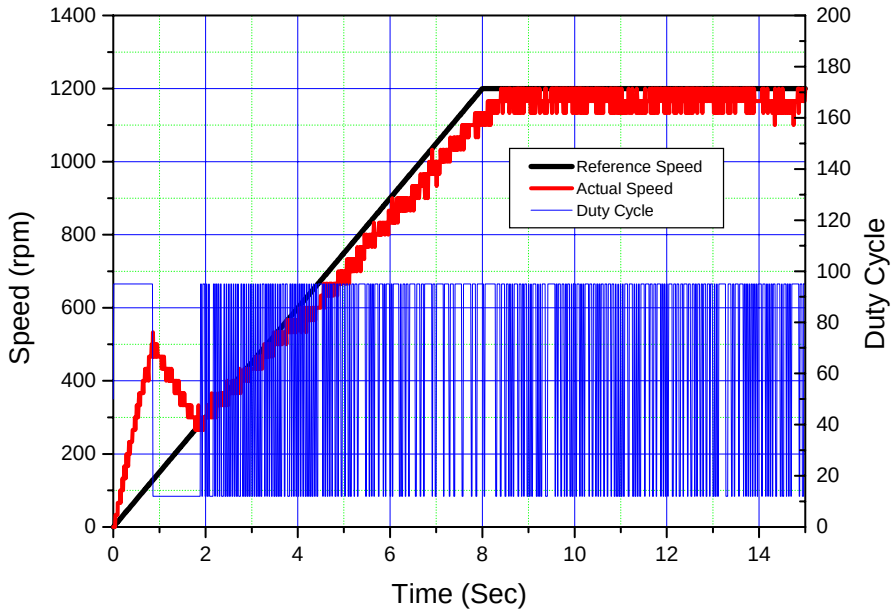


Fig 4.3: ON – OFF control for Ref Speed Profile 2

From Fig. 4.4 it is seen that when the acceleration is high (in reference speed profile 3) control action is better at lower speeds than the lower acceleration speed profiles. But in this profile error is larger at higher speed than the other speed profiles. At this speed profile, initial uncontrolled period reduced to only 0.51 sec and maximum uncontrolled speed was 255 rpm. At 0.6 actual speed started following the reference speed.

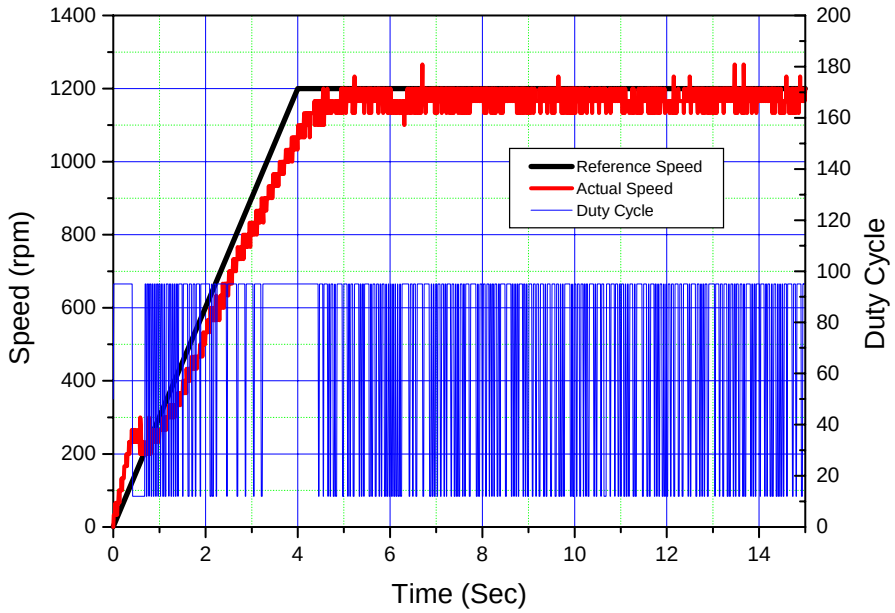


Fig. 4.4: ON – OFF control for Ref Speed Profile 3

From Fig. 4.2, 4.3 and 4.4 it is clear that ON-OFF control is not capable of controlling the full speed profiles. For a better control P controller is approached. P controller was tuned to obtain the optimum control action varying the value of Kp from a lower value to higher. Fig. 4.4 shows the proportional control action for reference speed profile 1 with Kp = 1/200. It shows that as the proportional gain is very low, with the increase of error duty cycle did not change that much to follow the reference speed. Duty cycle changed very slowly whose effect is clearly seen in the motor speed. Speed was changed at the pattern same as the duty cycle. Actual speed's trend was as the reference speed's trend but could not reach it finally.

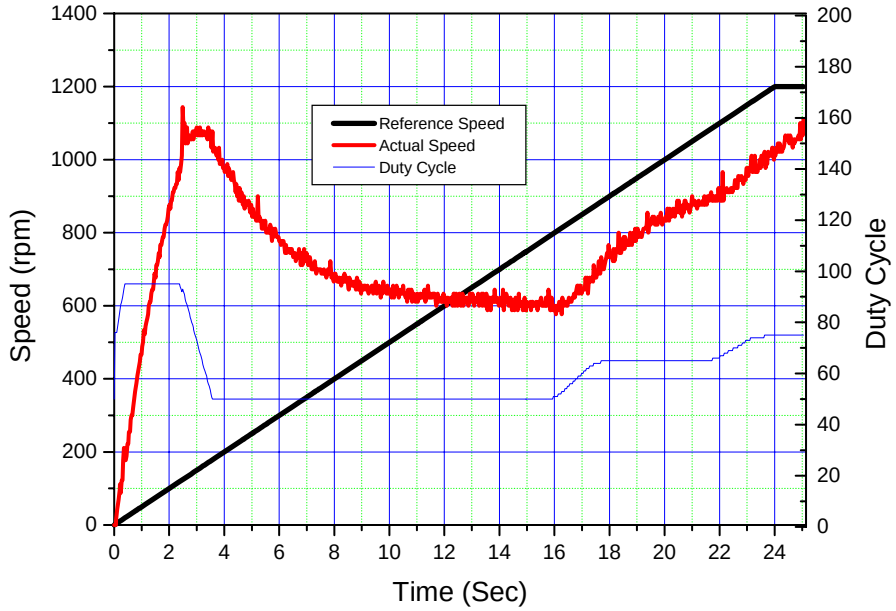


Fig. 4.5: Proportional control for Ref Speed Profile 1 with Kp = 1/200

From Fig. 4.5 it was clear that 1/200 is a much lower value of Kp to control it properly. So it was increased to higher values to observe how the controller performance improved. It was raised to a value of 1/64 and observed that it was not yet enough to follow the reference speed. Duty cycle is changed here faster than the previous and actual speed was closely following the reference speed profile after 11.94 sec. Here the initial uncontrolled period is 2.30 sec and maximum uncontrolled speed is 1100 rpm. After that controller started decreasing its speed and it reached the reference speed at 12.18 sec. Fig. 4.6 shows the proportional control action for same reference speed profile but with a higher Kp = 1/32. It shows that after the initial uncontrolled period while actual speed was oscillating over the reference speed, the period of oscillation is reduced here and following the reference speed more closely here.

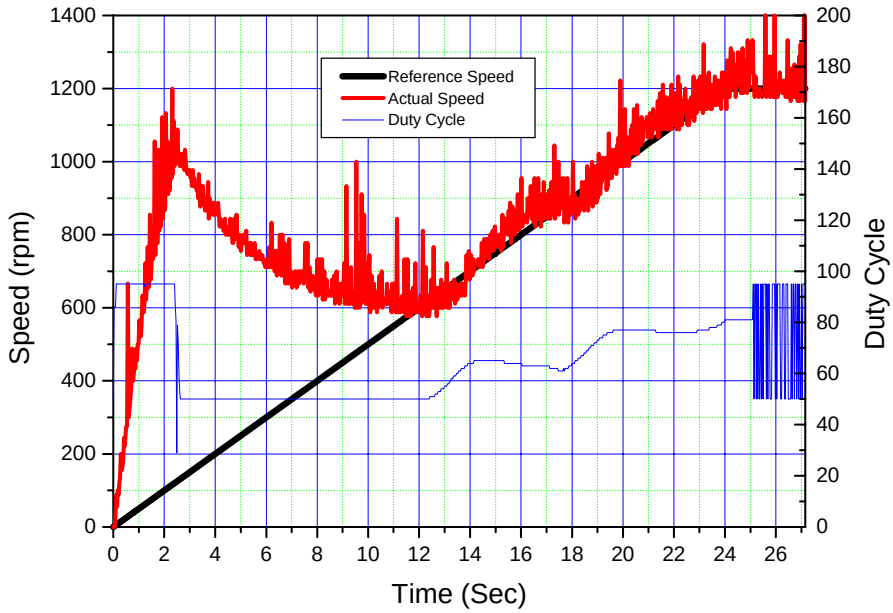


Fig. 4.6: Proportional control for Ref Speed Profile 1 with Kp = 1/64

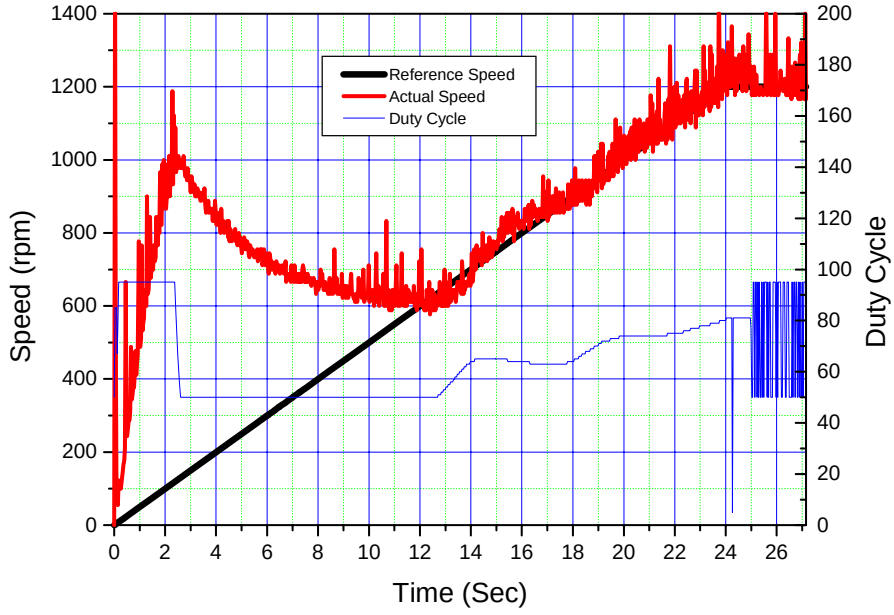


Fig. 4.7: Proportional control for Ref Speed Profile 1 with Kp = 1/32

Controller was tuned to obtain the optimum control increasing the Kp. Fig. 4.8, 4.9,4.10, 4.11, 4.12 and 4.13 are showing the proportional control for reference speed profile 1 with Kp = 1/16, 1/8, 1/4, 1/2, 1 and 2 respectively. This figures show that with the increase in Kp, period of oscillation of actual speed over the reference speed is reduced here, hence the error is reduced to obtain a more accurate control.

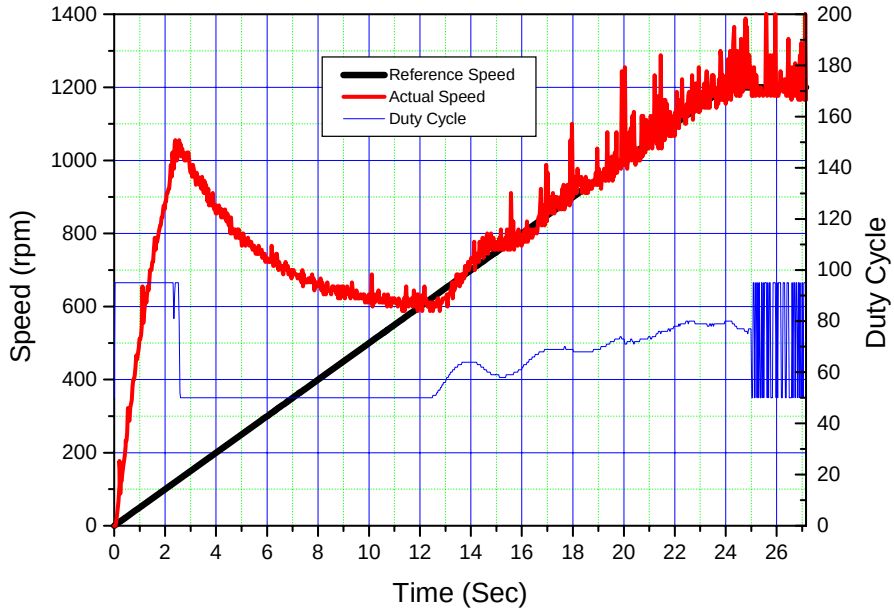


Fig. 4.8: Proportional control for Ref Speed Profile 1 with Kp = 1/16

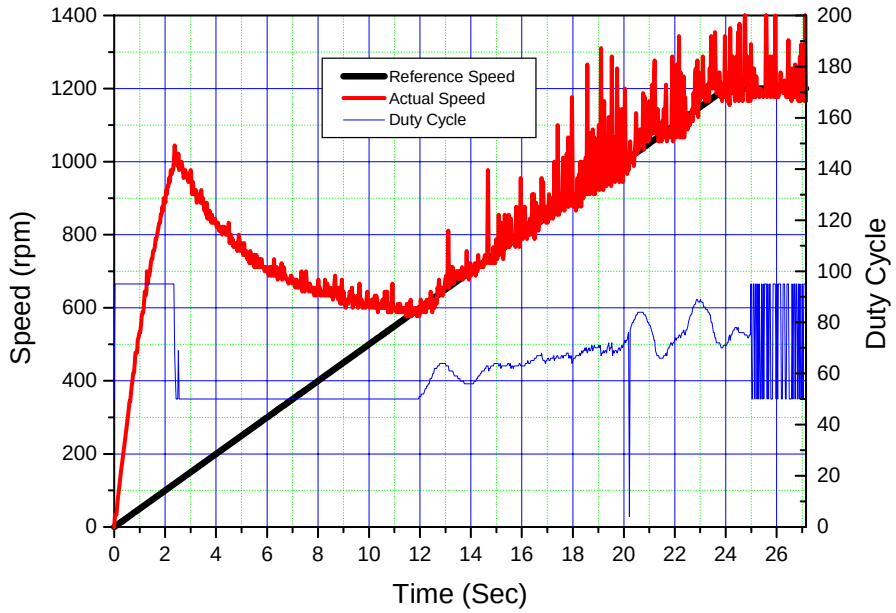


Fig. 4.9: Proportional control for Ref Speed Profile 1 with Kp = 1/8

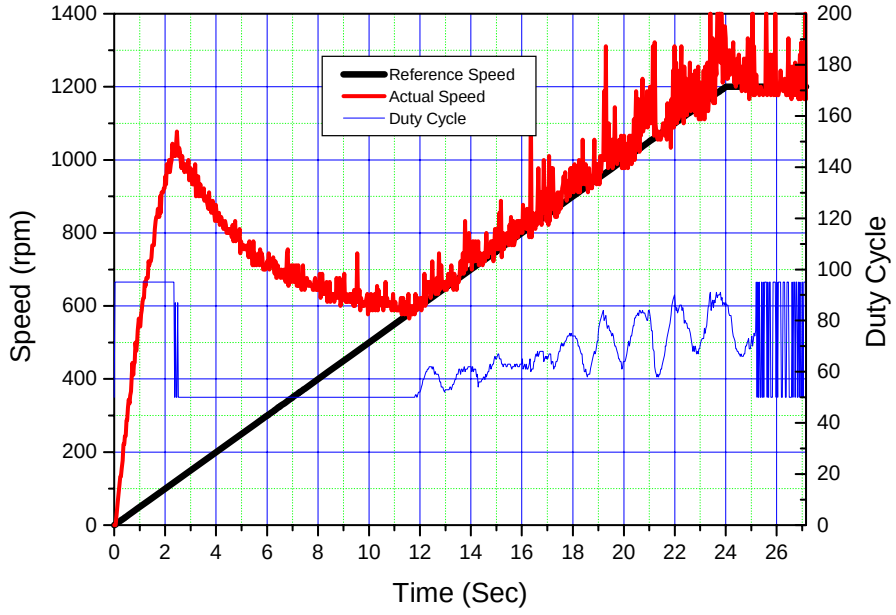


Fig. 4.10: Proportional control for Ref Speed Profile 1 with Kp = 1/4

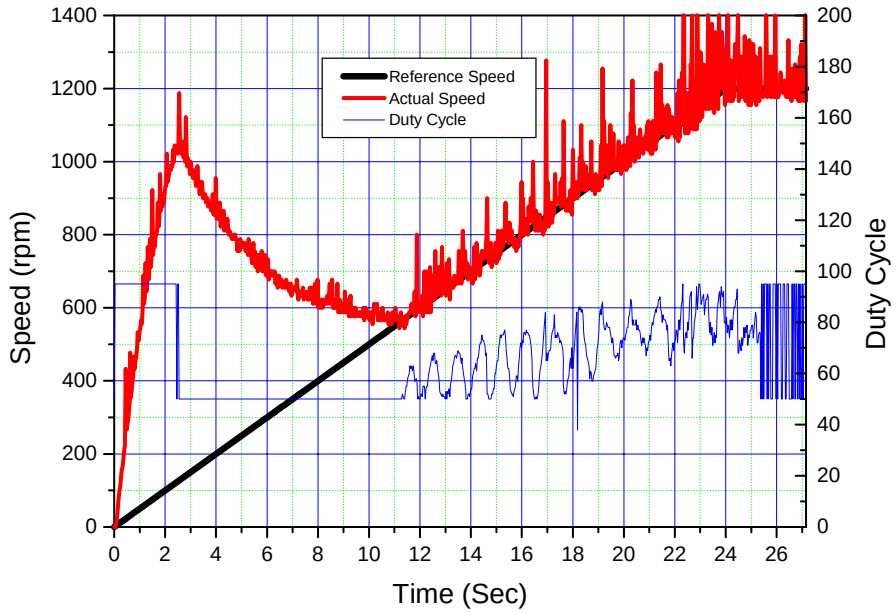


Fig. 4.11: Proportional control for Ref Speed Profile 1 with Kp = 1/2

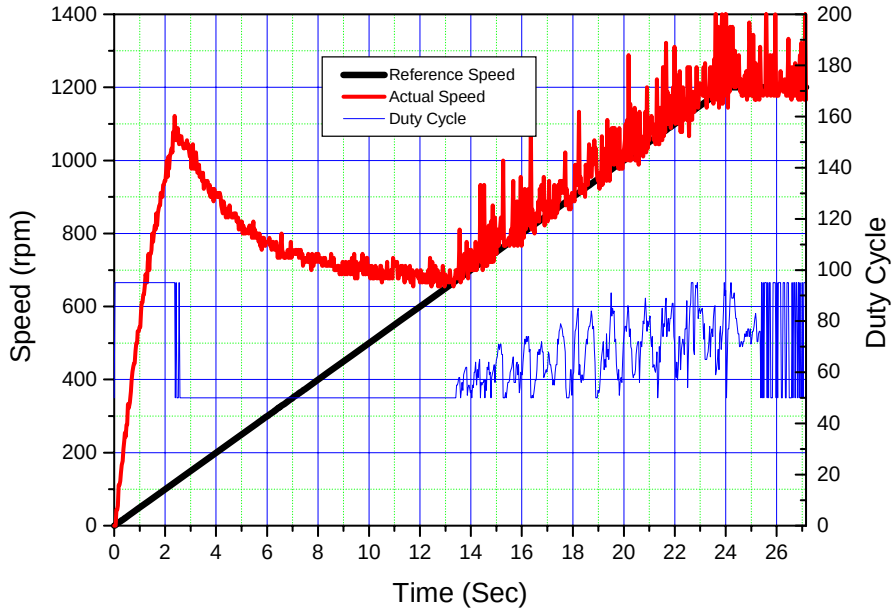


Fig. 4.12: Proportional control for Ref Speed Profile 1 with Kp = 1

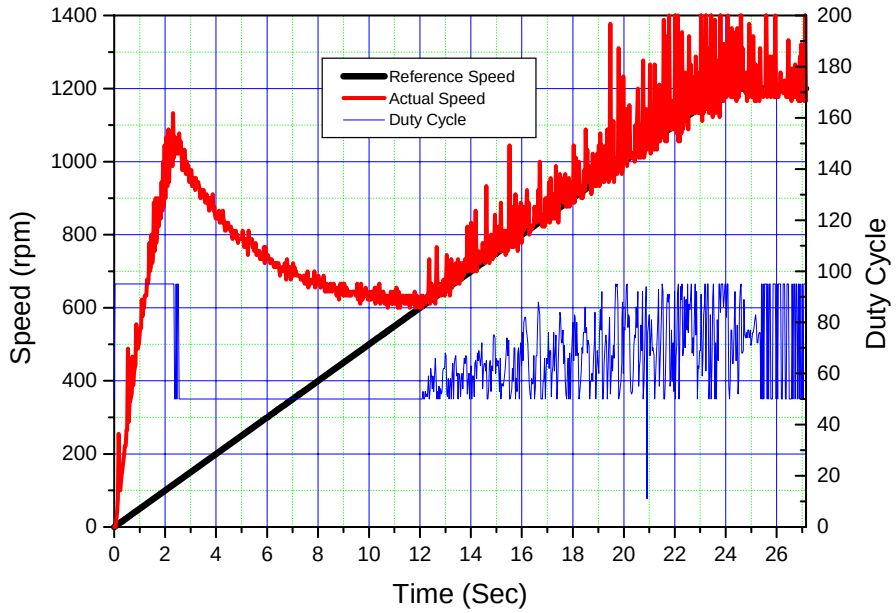


Fig. 4.13: Proportional control for Ref Speed Profile 1 with Kp = 2

Fig. 4.14, 4.15 and 4.16 show the control action when Kp = 4, 8 and 16 respectively. These are all clear evidence of improvement of control with the increase of Kp. Although the initial uncontrolled period is still present, error is significantly reduced at Kp = 16 shown in Fig. 4.16. An average line of duty cycle is drawn in these figures, which make this clear that trend of speed change is similar to the trend of duty cycle change.

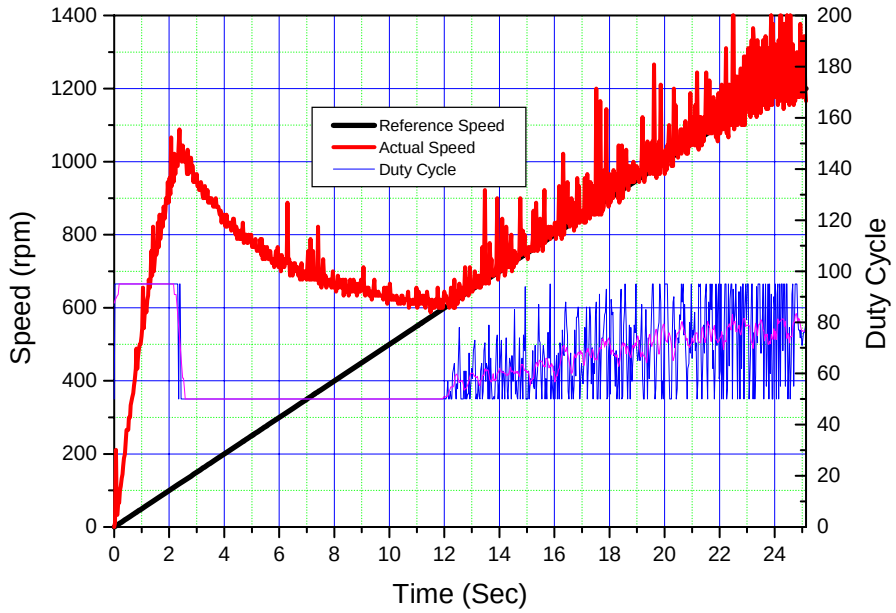


Fig. 4.14: Proportional control for Ref Speed Profile 1 with Kp = 4

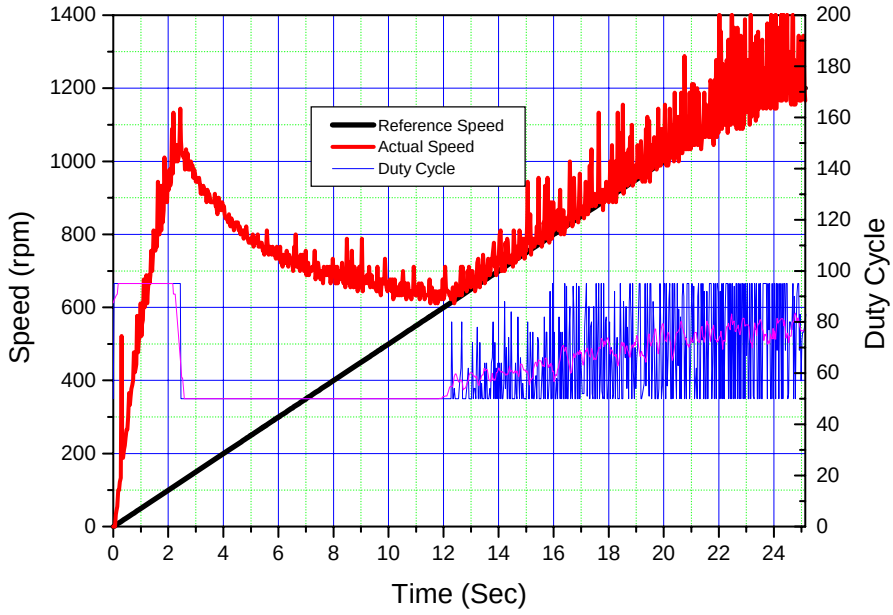


Fig. 4.15: Proportional control for Ref Speed Profile 1 with Kp = 8

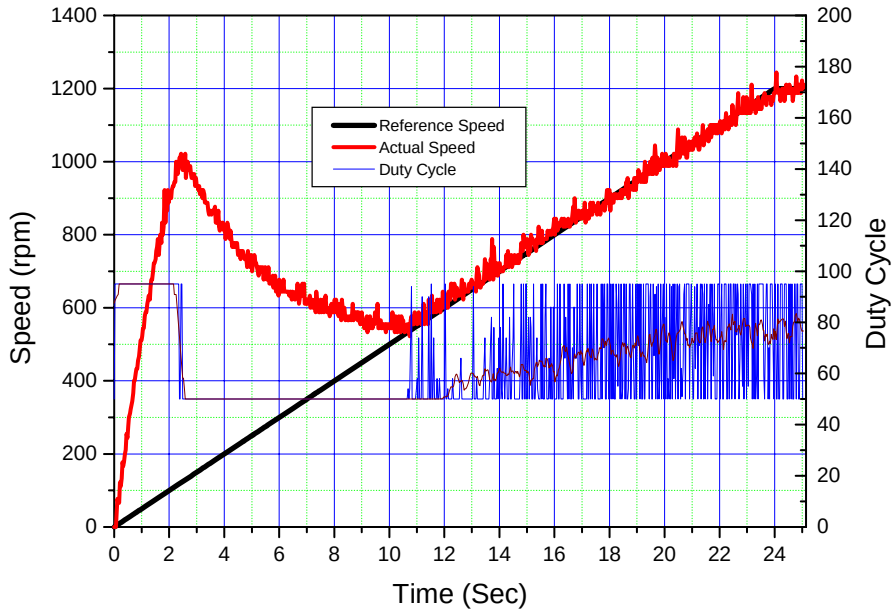


Fig. 4.16: Proportional control for Ref Speed Profile 1 with Kp = 16

Fig. 4.17 shows the best control for the reference speed profile 1. With Kp = 32 initial controlled period was minimum (2.14 sec) and the maximum uncontrolled speed was also minimum (322 rpm) for this reference speed. Further increase of Kp makes the control worse. Fig. 4.18 shows that when Kp = 64, again the initial uncontrolled period increased and when it is further increase to a very larger value 200 in Fig. 4.19 control system became unstable. However these discussions show that for reference speed profile 1, proportional control failed to control it completely but it was better controlled compared to the ON-OFF control. So optimum control was obtained with Kp = 32 for reference speed profile 1.

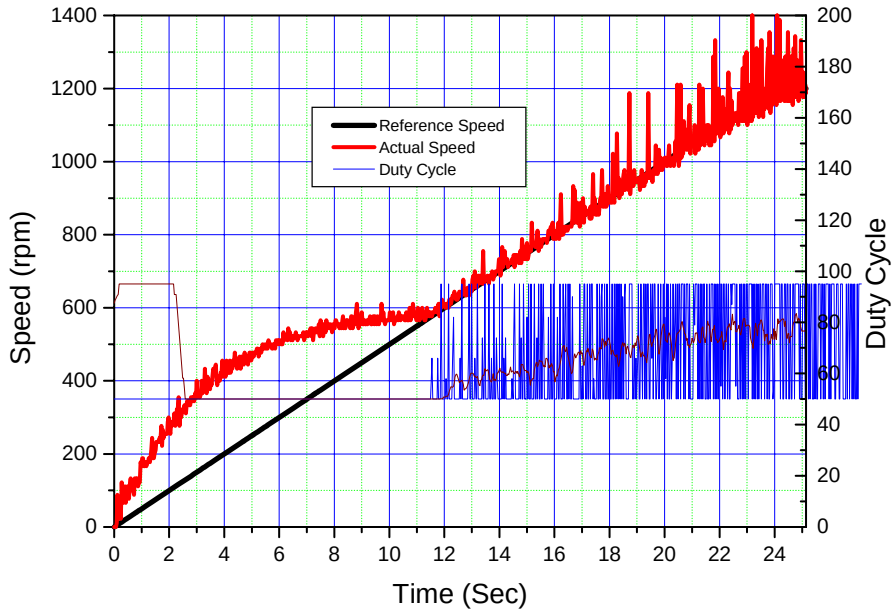


Fig. 4.17: Proportional control for Ref Speed Profile 1 with Kp = 32

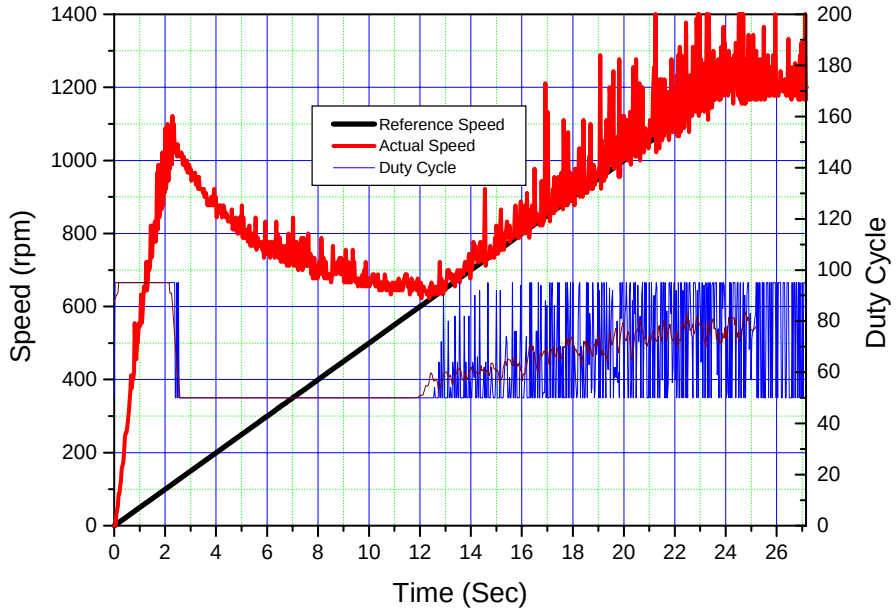


Fig. 4.18: Proportional control for Ref Speed Profile 1 with Kp = 64

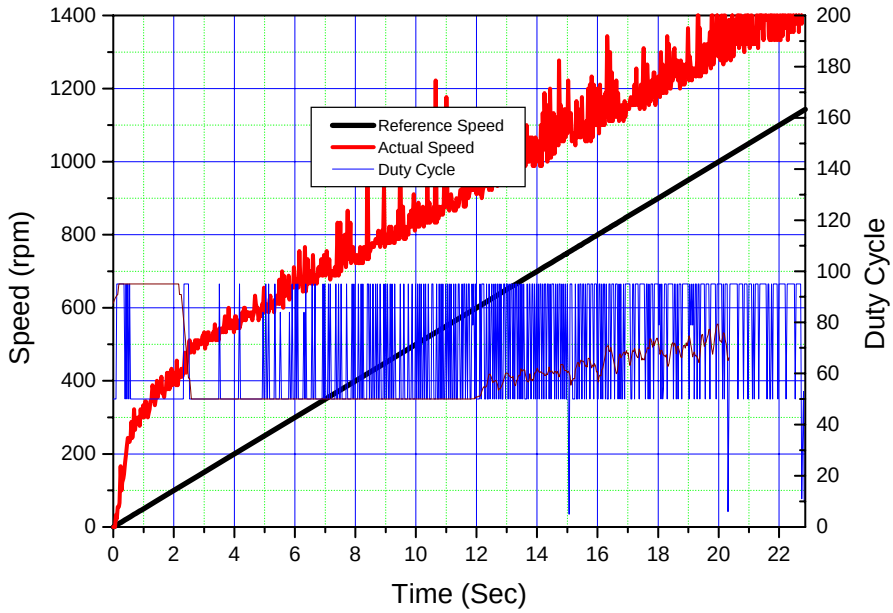


Fig. 4.19: Proportional control for Ref Speed Profile 1 with Kp = 200

Fig. 4.20 to Fig. 4.36 show the proportional control action for reference speed profile 2. Fig. 4.20 shows the control action when Kp = 1/200. At this Kp, actual speed was not even close to the desired speed. So for the improved control Kp was further increased, whose effect is shown here in the next figures. Like the control action for previous reference speed profile, form Fig 4.21 to 4.32 it is observed that when the value of Kp is increased, period of oscillation of actual speed over the reference speed is reduced that means the error is also diminished with the increase in Kp. Fig. 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.30 and 4.31 is showing the control with Kp = 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8 and 16. In Fig 4.32 when Kp = 32, actual speed followed the reference speed from the beginning so the problem with initial control is removed here but at the maximum speed when it is fixed at 1200 rpm actual speed is still oscillating. To tune the controller this value is further increased to 40 which is shown here with the Fig. 4.32. It shows that the actual speed is following the reference speed through the whole profile as expected. Kp was further increased to find optimum value. When Kp was 45, shown in Fig. 4.34, actual speed is again deviated from the reference speed at the beginning. Fig. 4.35 and 4.36 shows that how system became instable with the further increase in Kp. However, these figures find it out that to control the reference speed profile 2 with proportional controller, optimum value of Kp is 40.

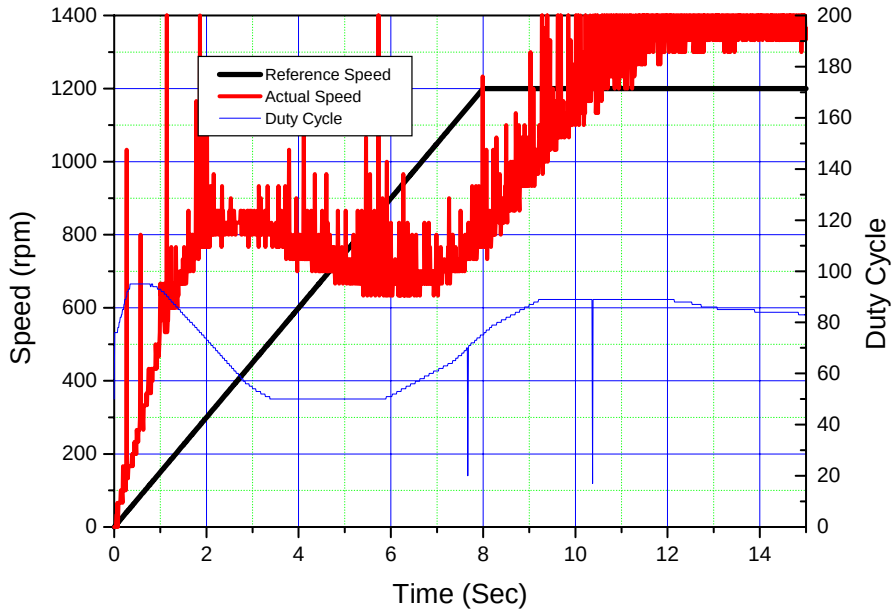


Fig. 4.20: Proportional control for Ref Speed Profile 2 with Kp =1/200

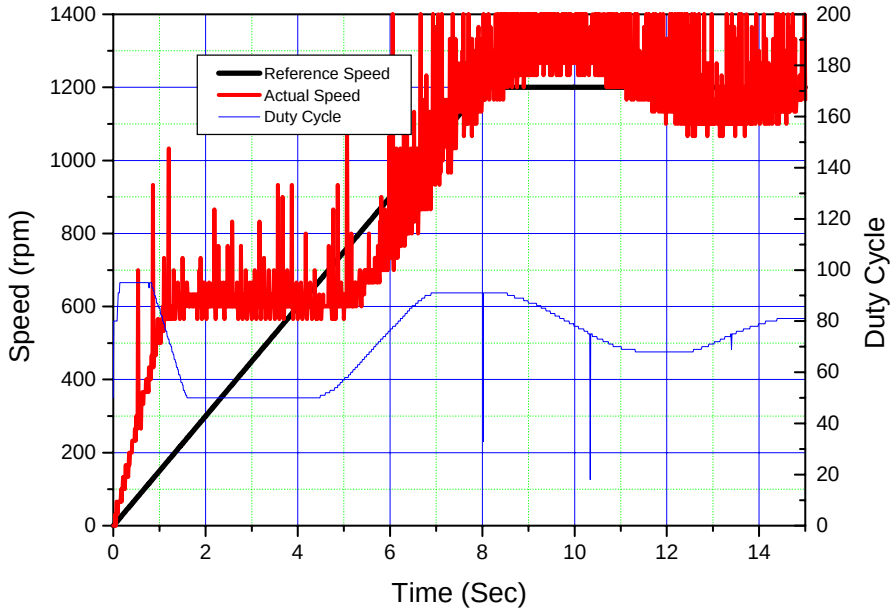


Fig. 4.21: Proportional control for Ref Speed Profile 2 with Kp =1/64

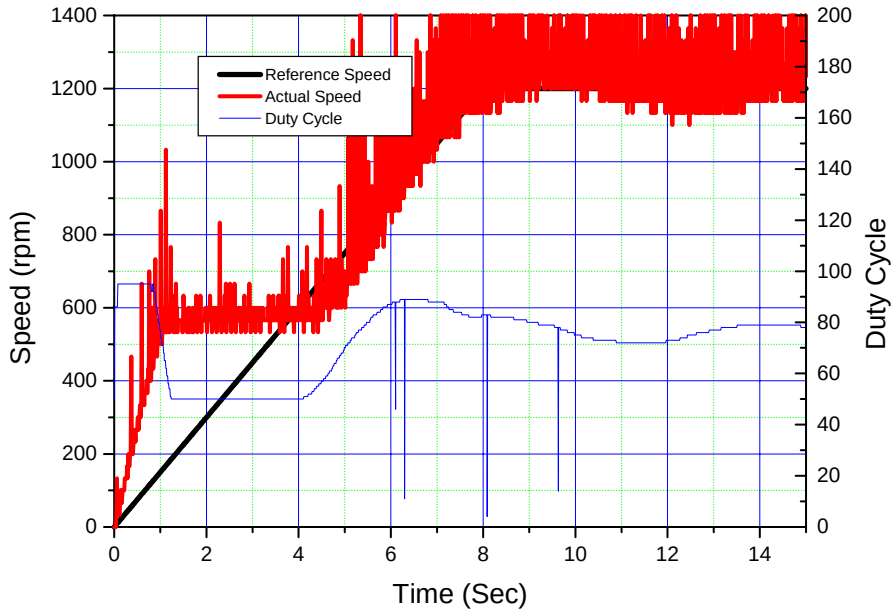


Fig. 4.22: Proportional control for Ref Speed Profile 2 with Kp =1/32

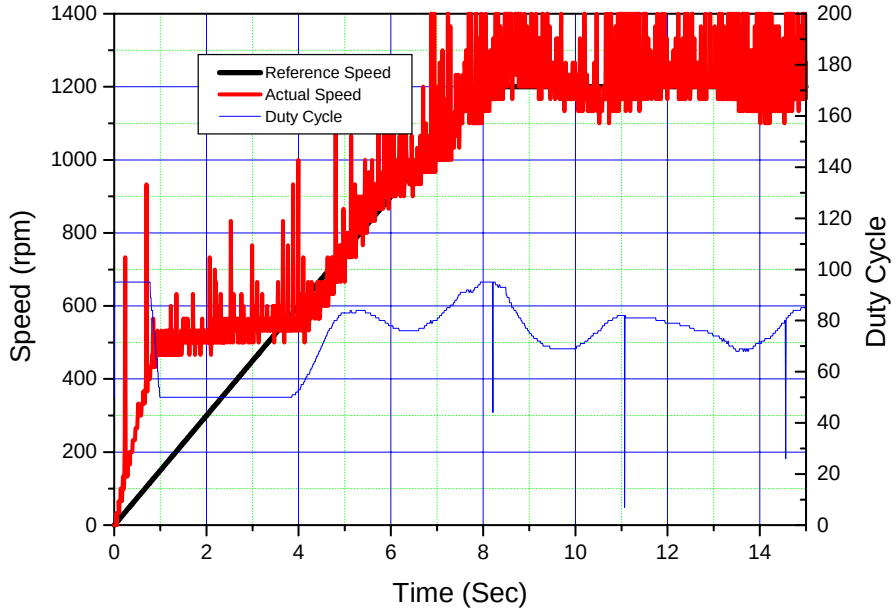


Fig. 4.23: Proportional control for Ref Speed Profile 2 with Kp =1/16

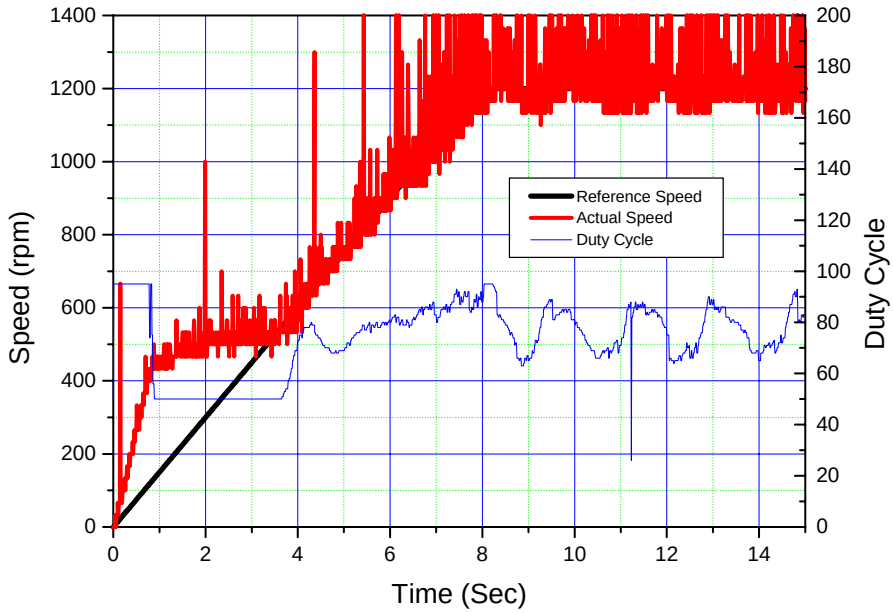


Fig. 4.24: Proportional control for Ref Speed Profile 2 with Kp 1/8

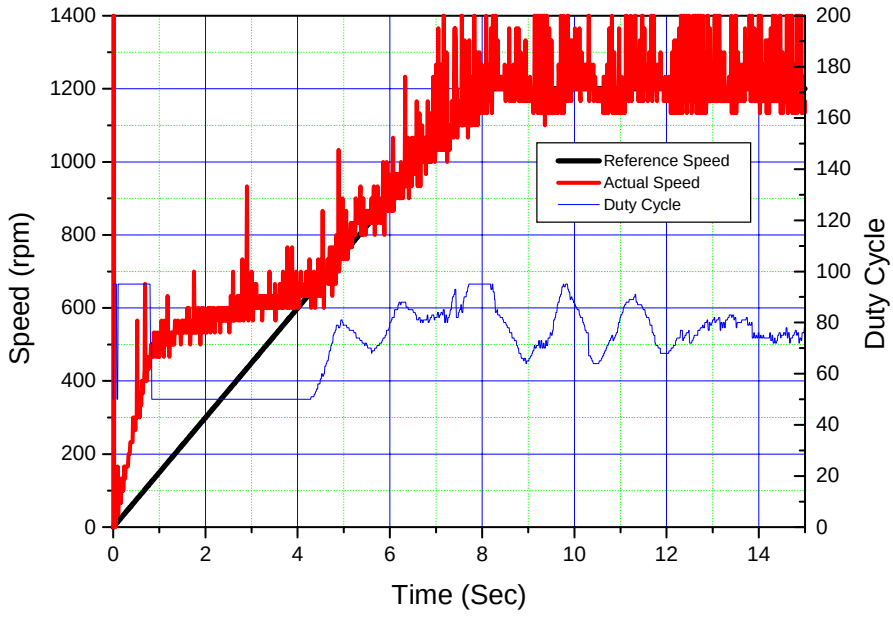


Fig. 4.25: Proportional control for Ref Speed Profile 2 with Kp =1/4

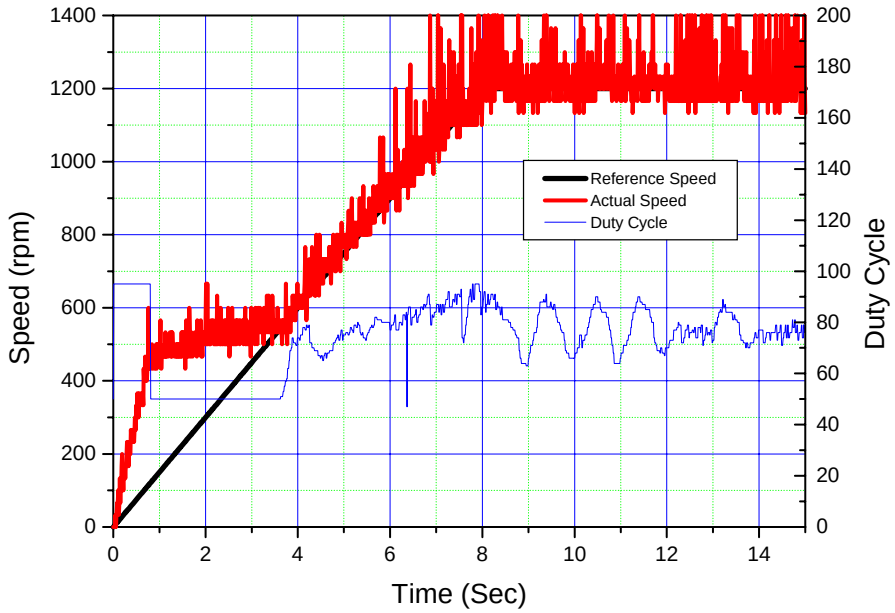


Fig. 4.26: Proportional control for Ref Speed Profile 2 with Kp =1/2

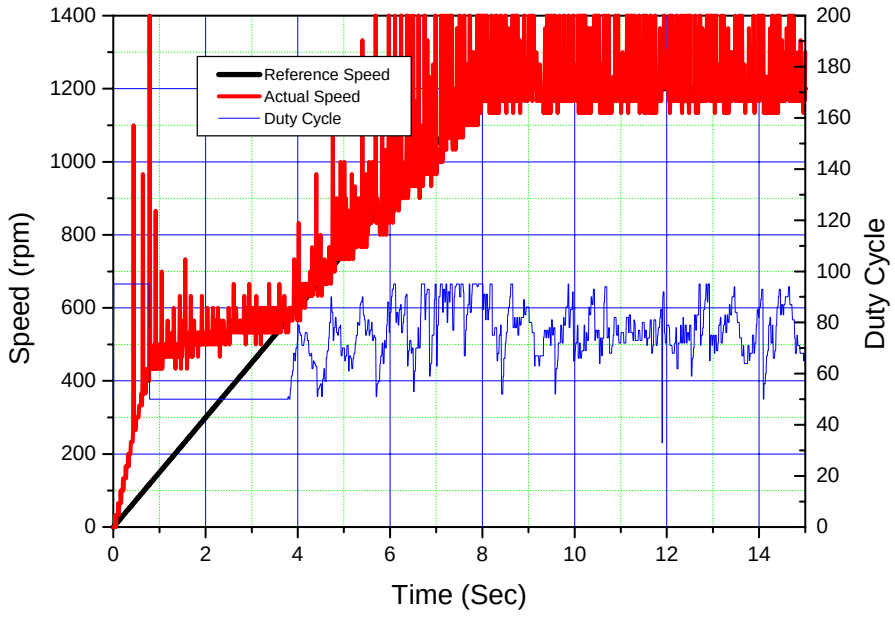


Fig. 4.27: Proportional control for Ref Speed Profile 2 with Kp = 1

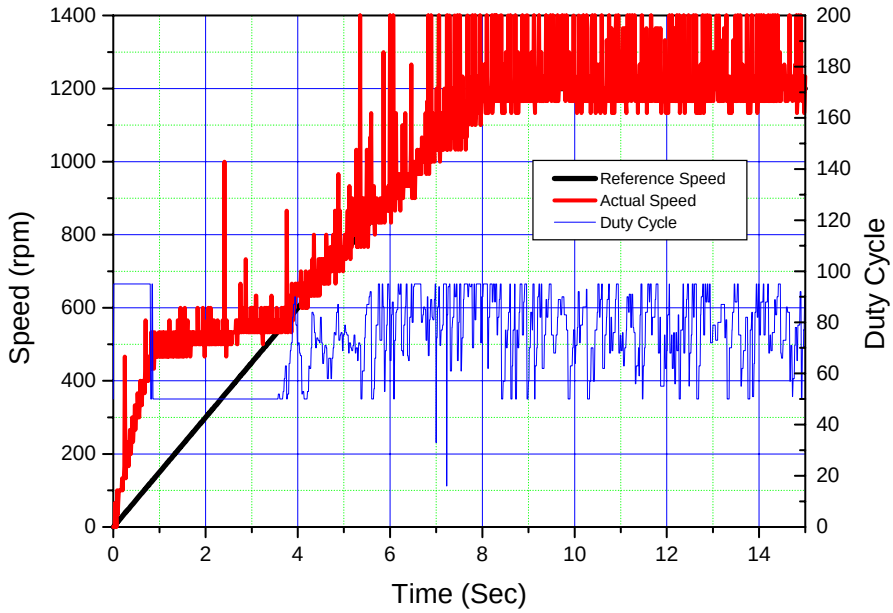


Fig. 4.28: Proportional control for Ref Speed Profile 2 with Kp = 2

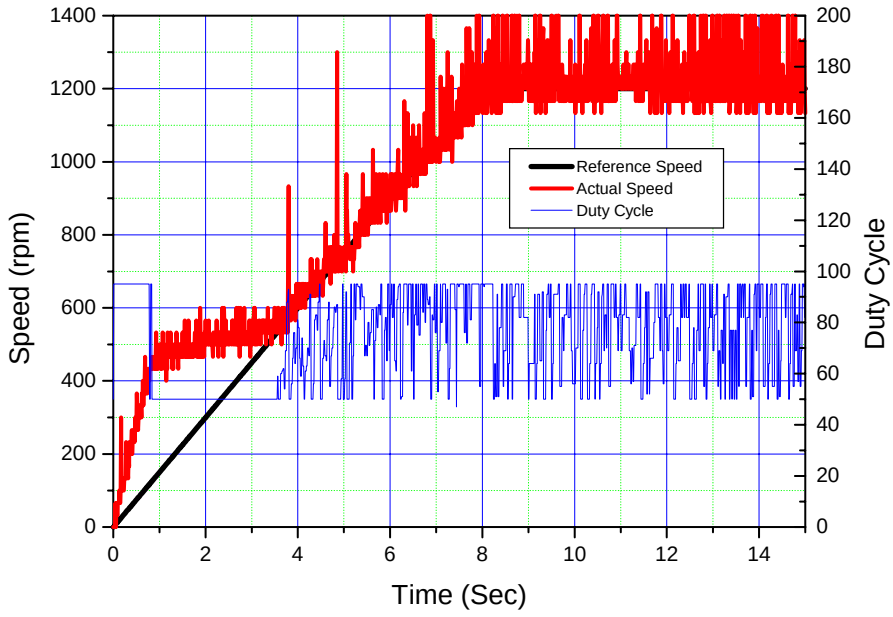


Fig. 4.29: Proportional control for Ref Speed Profile 2 with Kp = 4

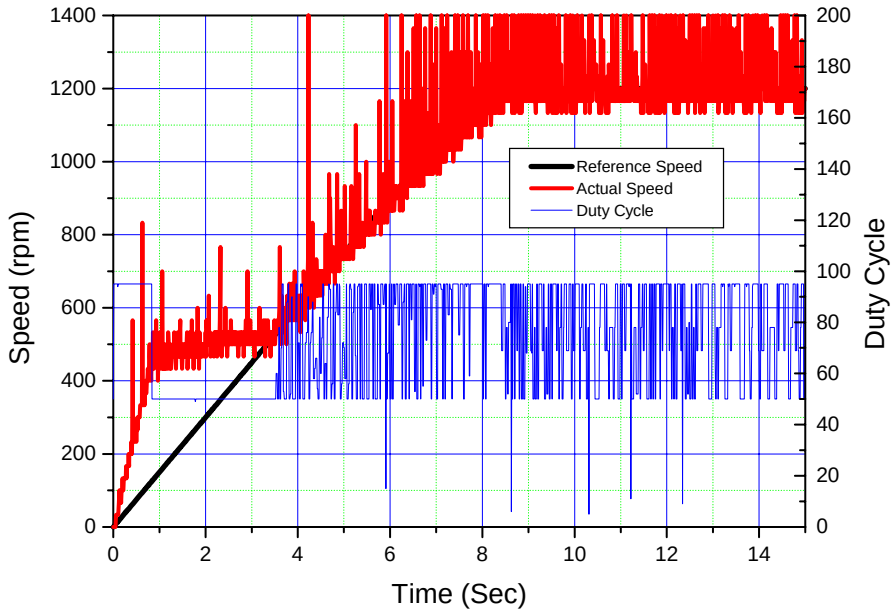


Fig. 4.30: Proportional control for Ref Speed Profile 2 with Kp = 8

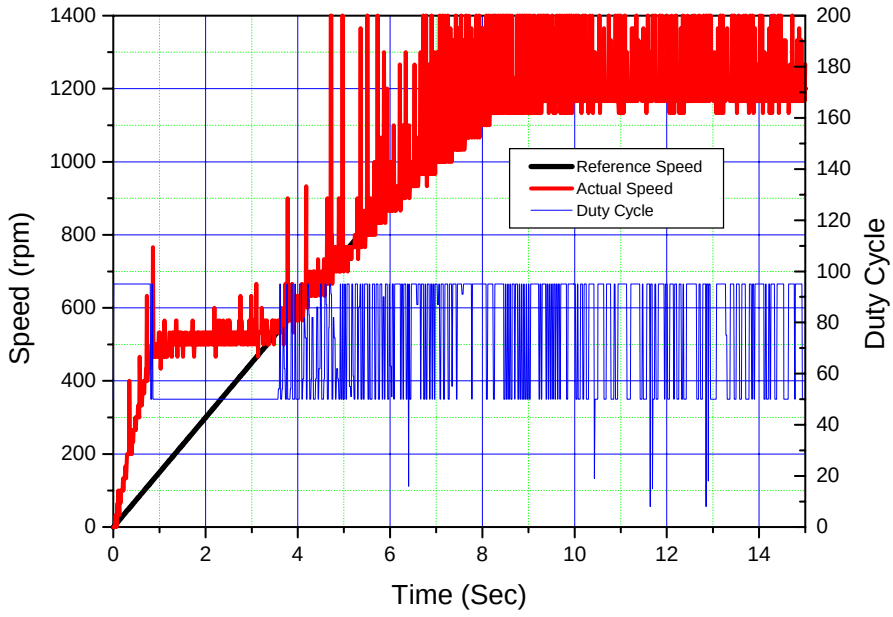


Fig. 4.31: Proportional control for Ref Speed Profile 2 with Kp = 16

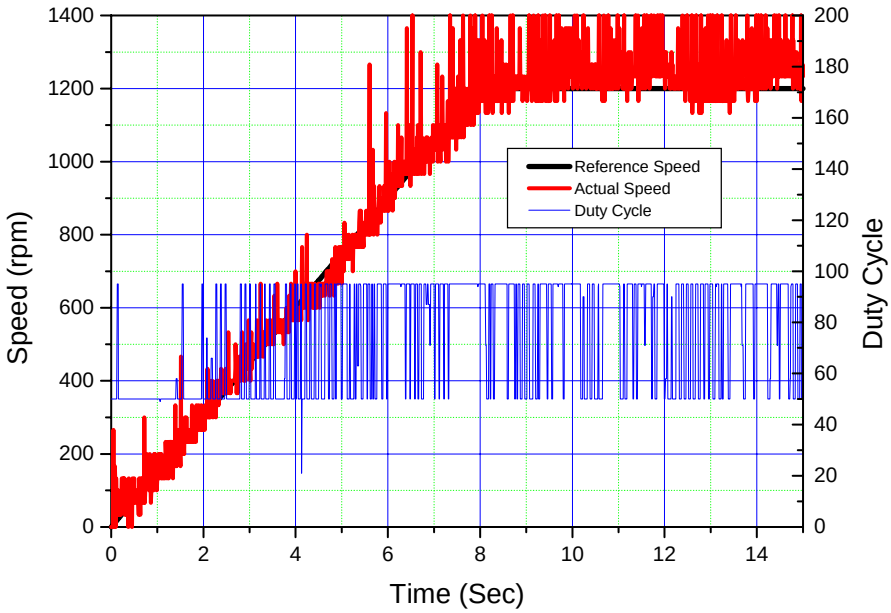


Fig. 4.32: Proportional control for Ref Speed Profile 2 with Kp = 32

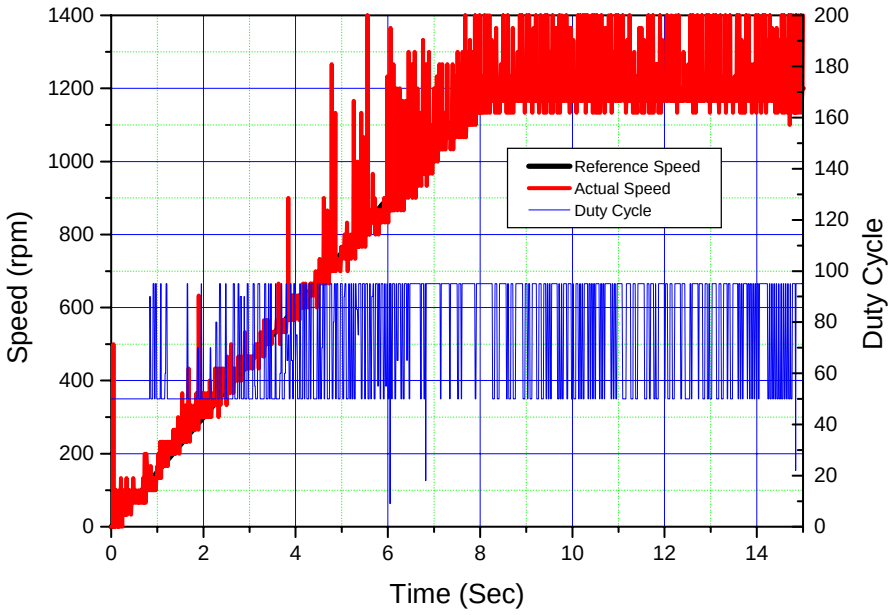


Fig. 4.33: Proportional control for Ref Speed Profile 2 with Kp = 40

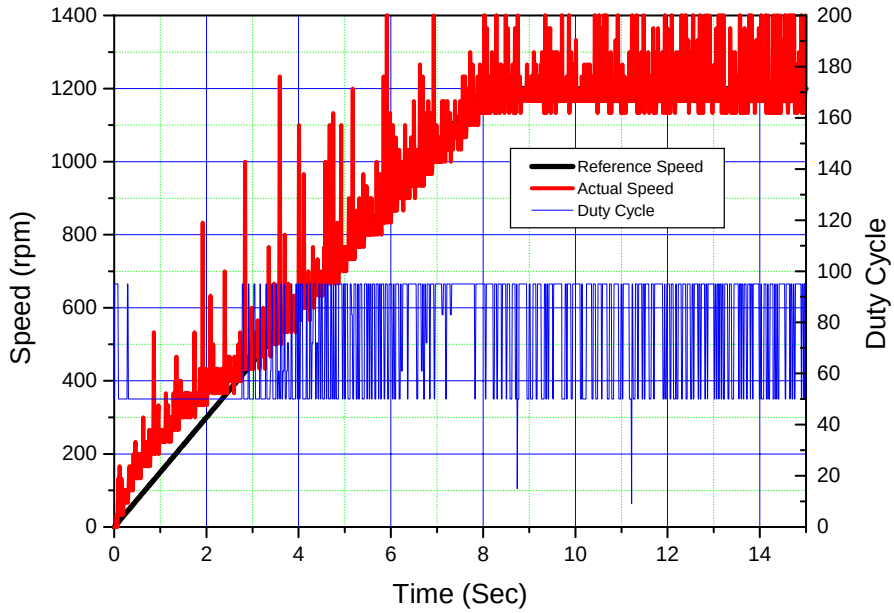


Fig. 4.34: Proportional control for Ref Speed Profile 2 with Kp = 45

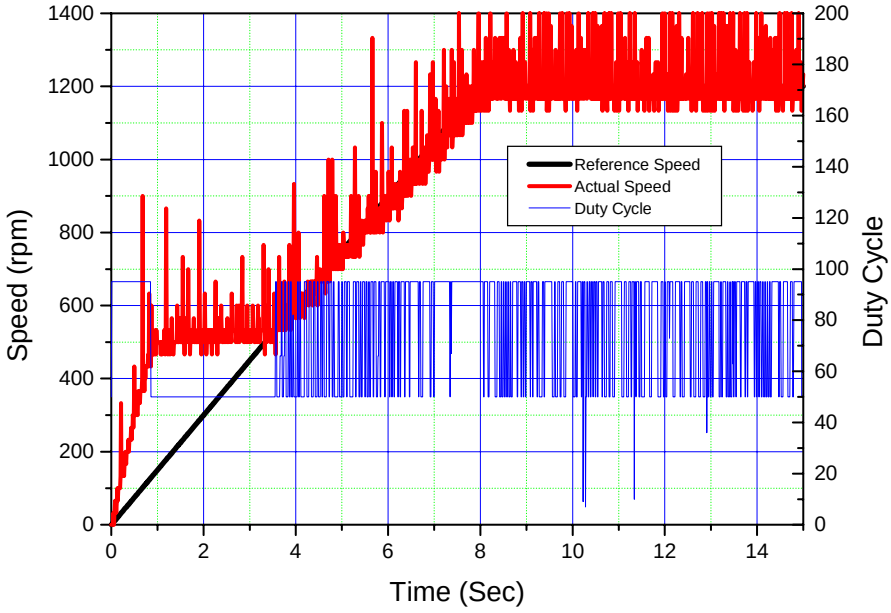


Fig. 4.35: Proportional control for Ref Speed Profile 2 with Kp = 64

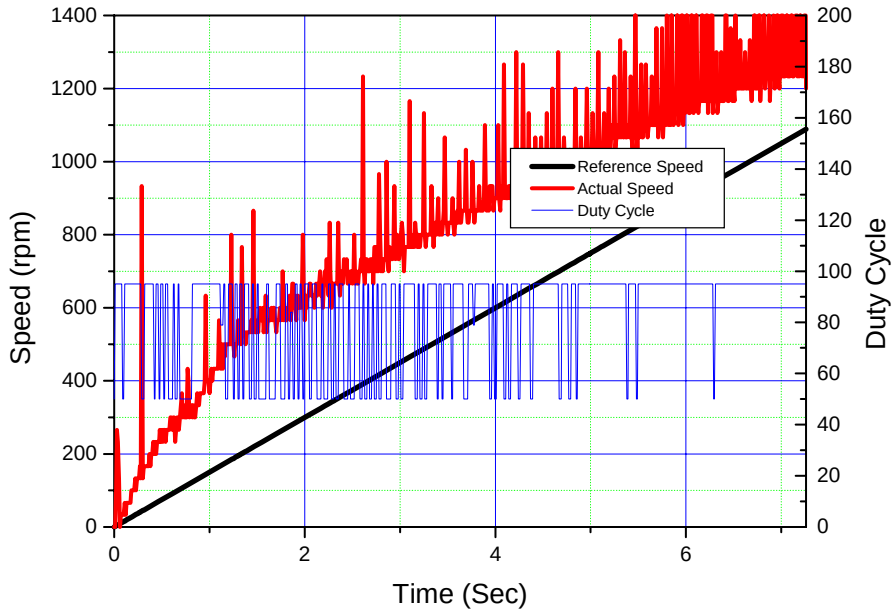


Fig. 4.36: Proportional control for Ref Speed Profile 2 with Kp = 200

Fig. 4.37 to Fig. 4.51 shows the proportional control action for reference speed profile 3. Fig. 4.37, 4.38, 4.39, 4.40, 4.41, 4.42, 4.43, 4.44, 4.45, 4.46, 4.47 and 4.48 is showing the control with Kp = 1/200, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8 and 16. In Fig 4.49 when Kp = 32, actual speed followed the reference speed. To tune the controller this value is further increased to 40 which is shown here with the Fig. 4.50. It shows that the actual speed is still following the reference speed through the whole profile as expected. Kp was further increased. Fig 4.51 and 4.52 when Kp = 64 and 200, shows that the system forwarded to the instability with the further increase in Kp. However, from these experiments decision can be made that to control this reference speed profile of this setup using proportional controller Kp is possible to be kept within 32 to 40.

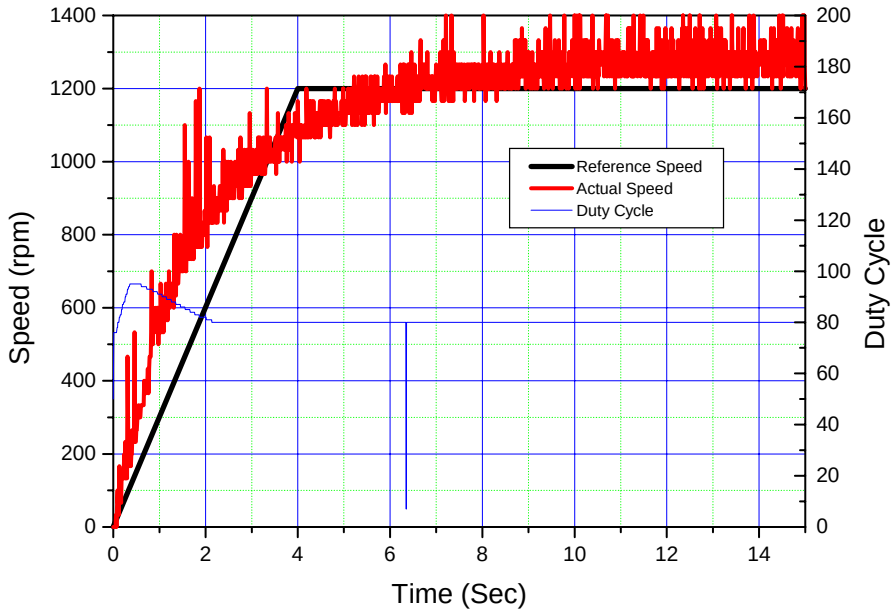


Fig. 4.37: Proportional control for Ref Speed Profile 3 with Kp = 1/200

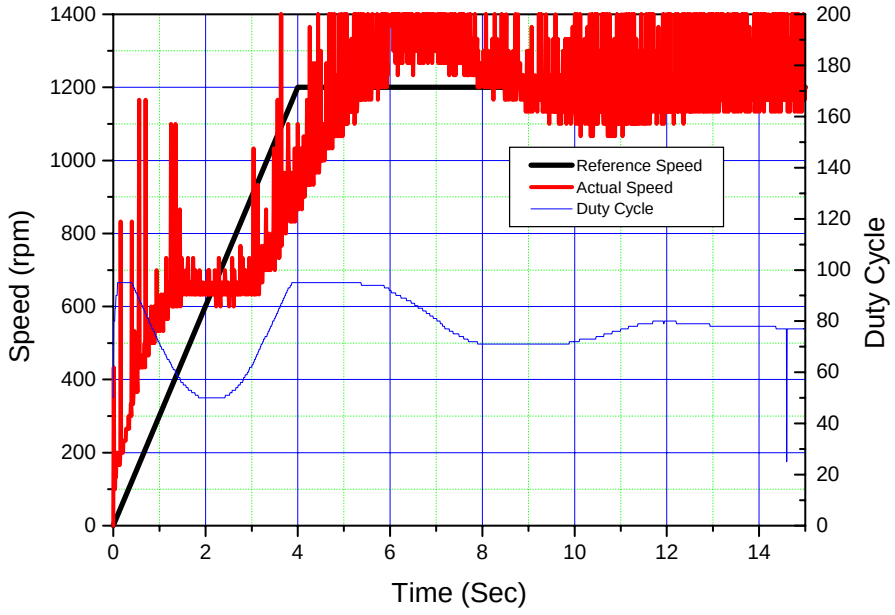


Fig. 4.38: Proportional control for Ref Speed Profile 3 with Kp = 1/64

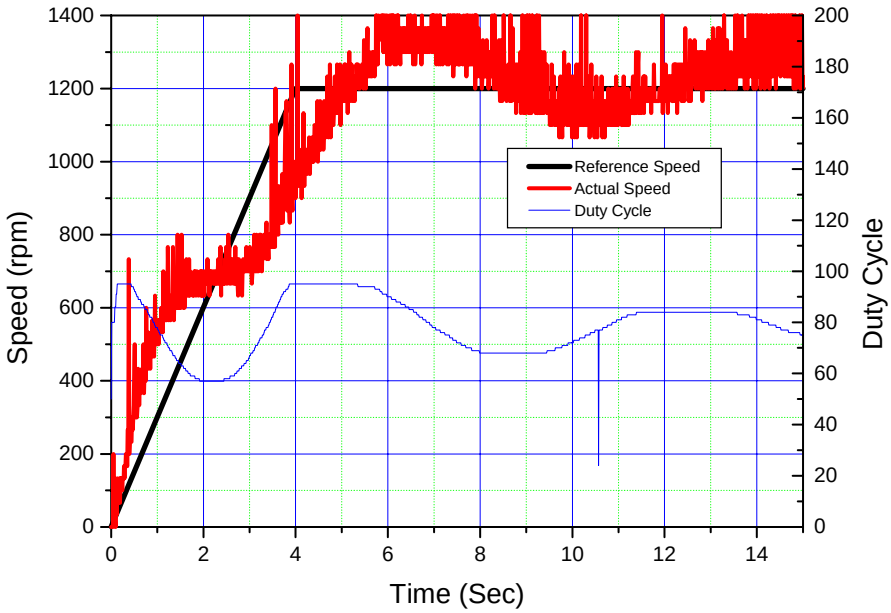


Fig. 4.39: Proportional control for Ref Speed Profile 3 with Kp = 1/32

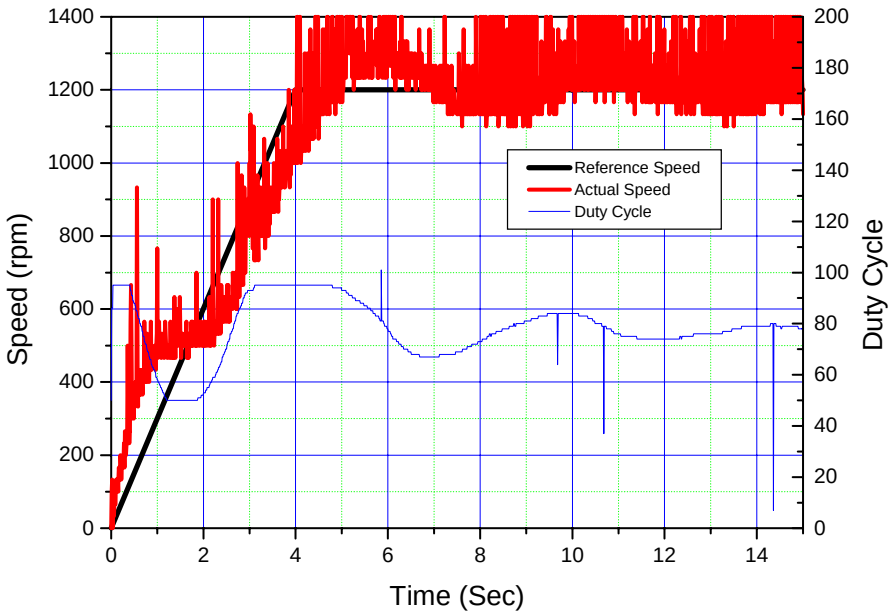


Fig. 4.40: Proportional control for Ref Speed Profile 3 with Kp = 1/16

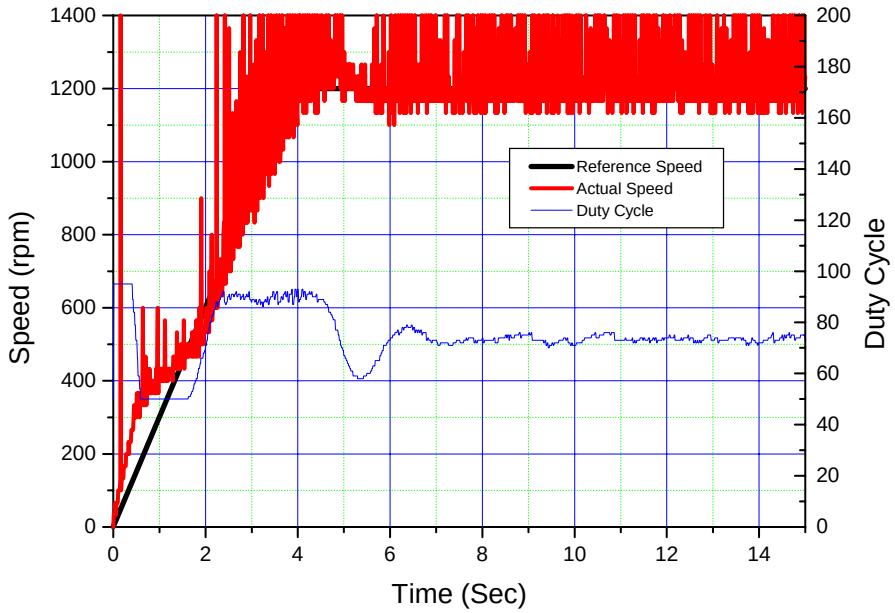


Fig. 4.41: Proportional control for Ref Speed Profile 3 with Kp = 1/8

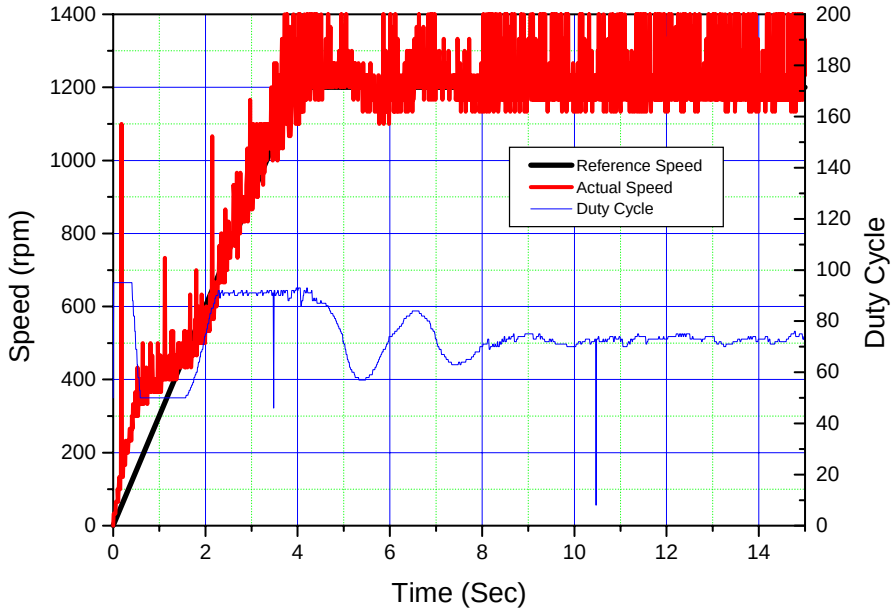


Fig. 4.42: Proportional control for Ref Speed Profile 3 with Kp = 1/4

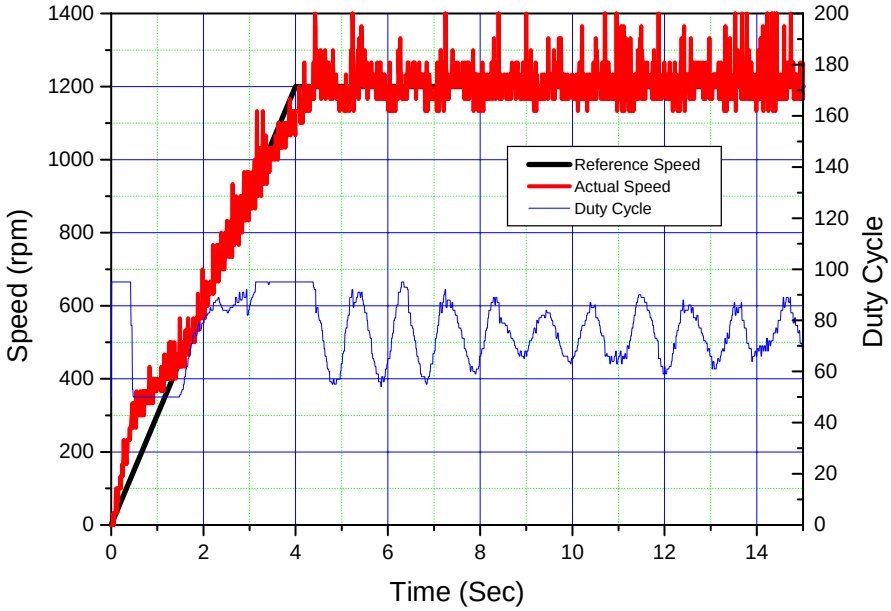


Fig. 4.43: Proportional control for Ref Speed Profile 3 with Kp = 1/2

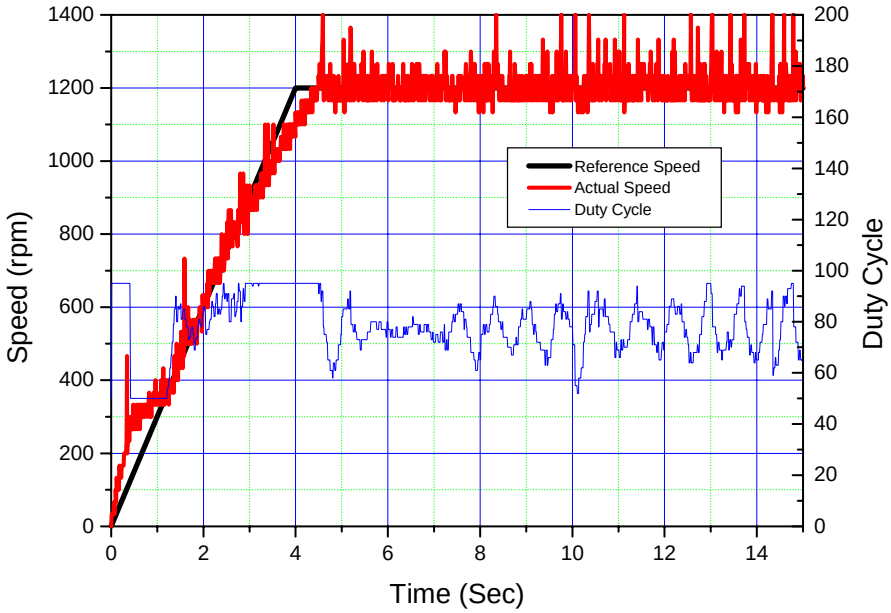


Fig. 4.44: Proportional control for Ref Speed Profile 3 with Kp = 1

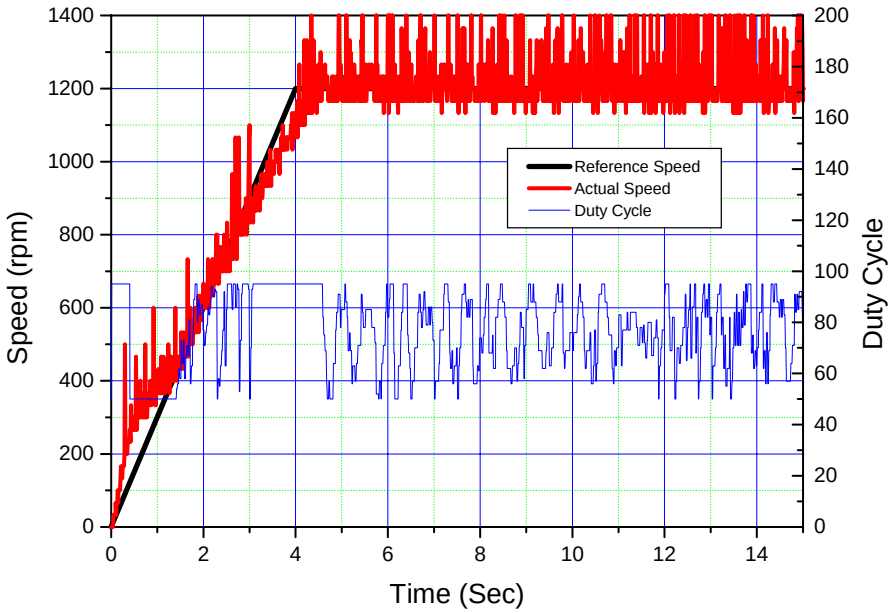


Fig. 4.45: Proportional control for Ref Speed Profile 3 with Kp = 2

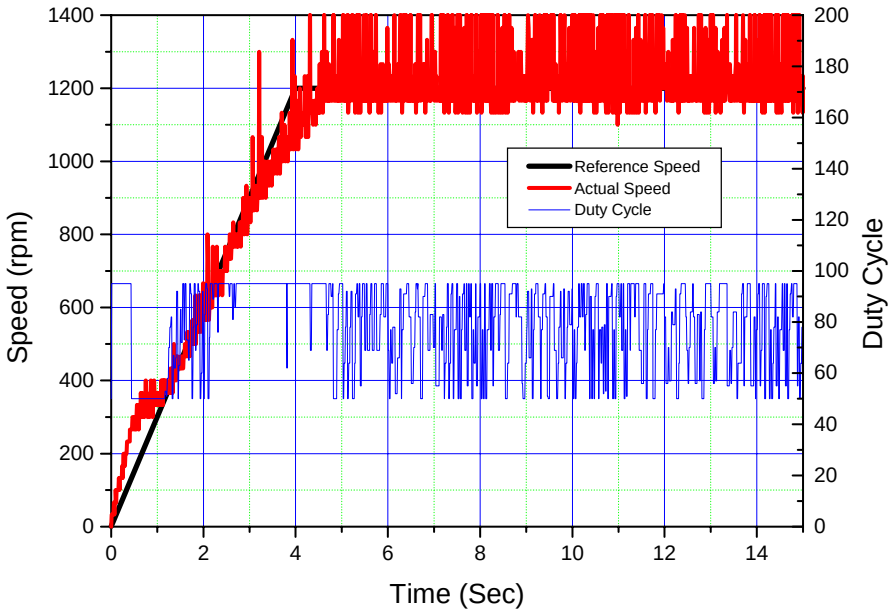


Fig. 4.46: Proportional control for Ref Speed Profile 3 with Kp = 4

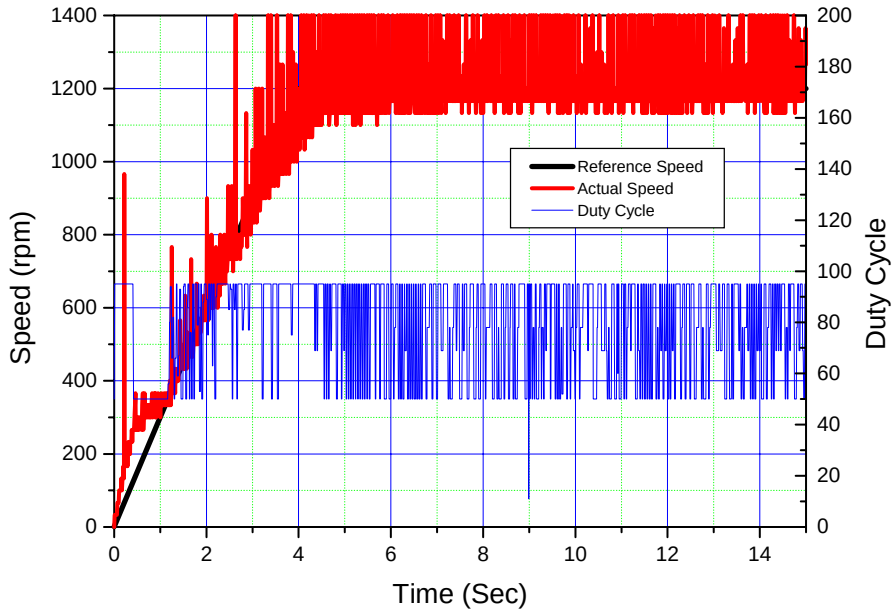


Fig. 4.47: Proportional control for Ref Speed Profile 3 with Kp = 8

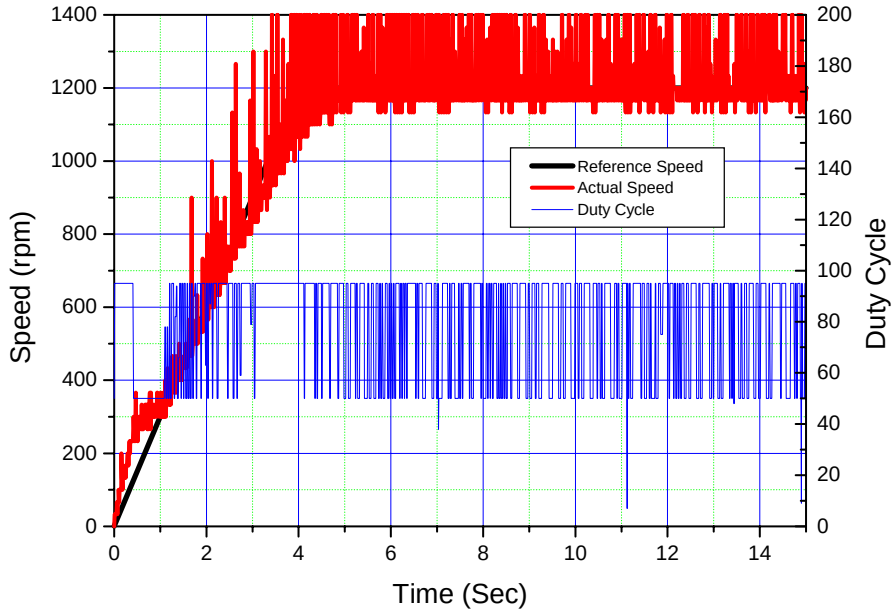


Fig. 4.48: Proportional control for Ref Speed Profile 3 with Kp = 16

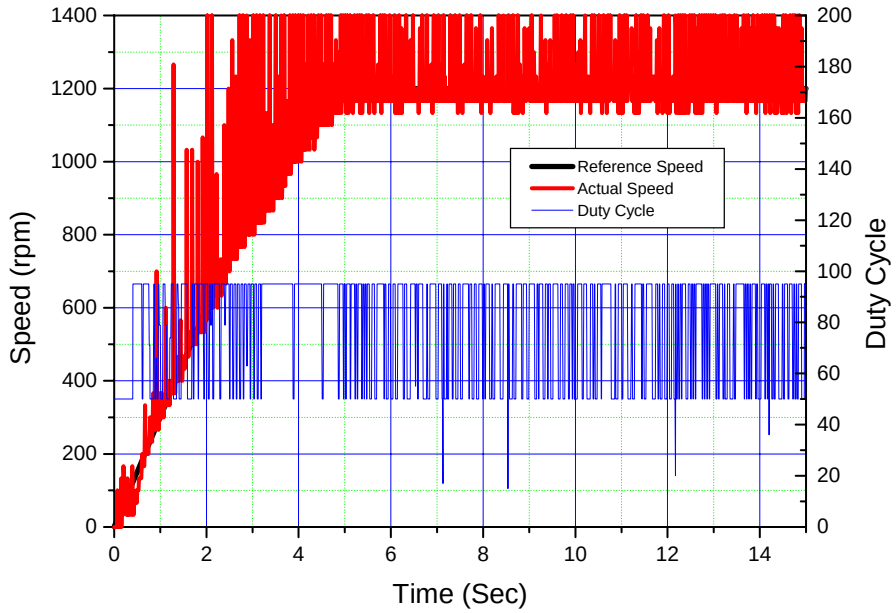


Fig. 4.49: Proportional control for Ref Speed Profile 3 with Kp = 32

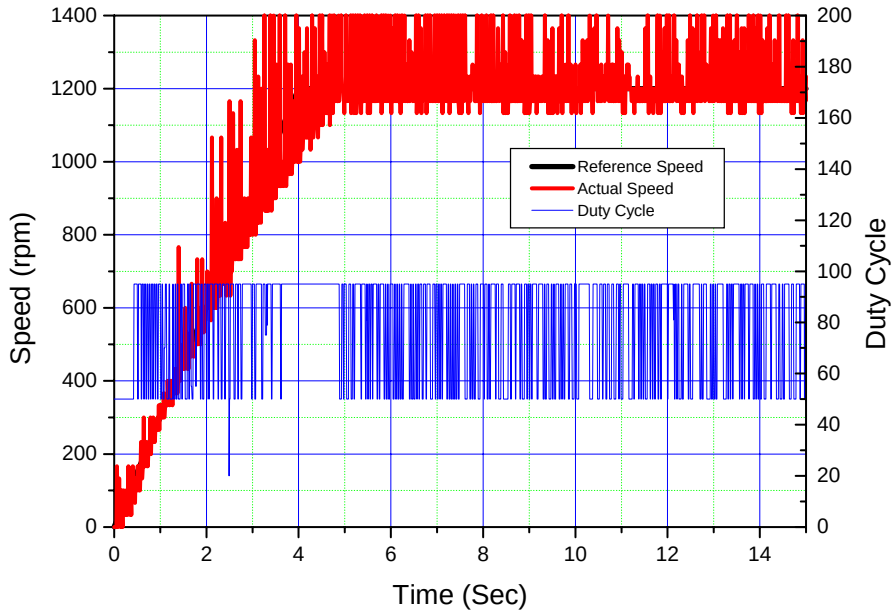


Fig. 4.50: Proportional control for Ref Speed Profile 3 with Kp = 40

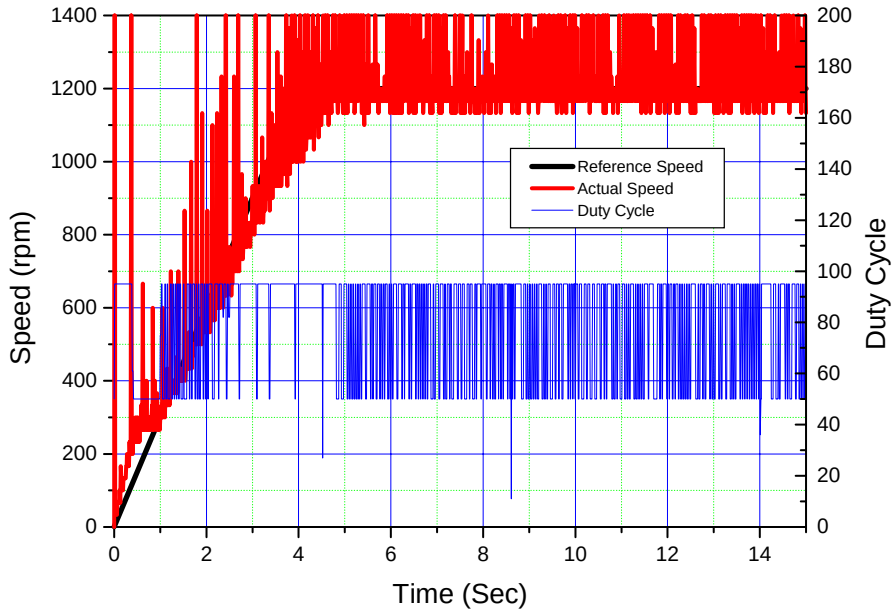


Fig. 4.51: Proportional control for Ref Speed Profile 3 with Kp = 64

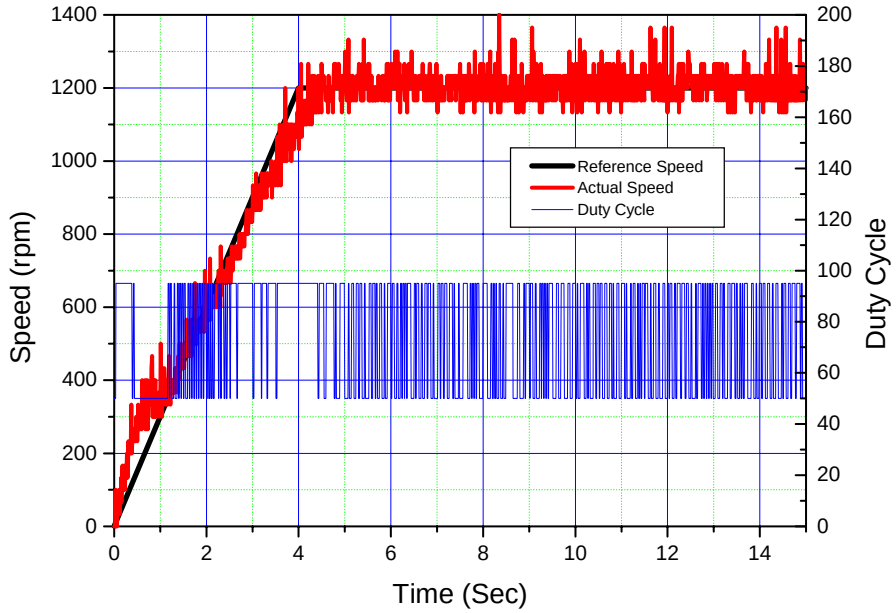


Fig. 4.52: Proportional control for Ref Speed Profile 3 with Kp = 200

These experiments make it clear that proportional control is better than ON-OFF control to control any speed profile. Although proportional controller failed to control the reference speed profile 1 initially but the initial uncontrolled period and uncontrolled maximum speed are both reduced in proportional control. ON-OFF controller failed to control reference speed profile 2 and 3, whereas proportional controller successfully controlled these. For further improvement PI controller is approached adding I component with P controller. PI controller was tuned using Zeigler-Nichols method. According to this method for reference speed profile 1, Kc was calculated using the optimum Kp =40 of Proportional controller for this speed profile.

$Kp = 0.5\ Kc$

$\Rightarrow$ $Kc = 2\ Kp$

Now proportional gain for PI controller is determined by the following relation:

$Kp = 0.45\ Kc = 36$

Again, using these relations $Tc = \frac{1}{Kc}$, $T_I = \frac{T_c}{1.2}$ and $K_I = \frac{1}{T_I}$ integral gain $K_I = 96$.

Using this Kp and K_I, PI controller is developed. Fig. 4.53 shows the PI control with Kp and K_I of 36 and 96 for reference speed profile 1. It still shows the uncontrolled period at the beginning but it is reduced a much than the proportional controller. Fig. 4.54, 4.55, 4.56 and 4.57 shows the control with Kp = 14, K_I =38; Kp = 22, K_I = 58; Kp = 29, K_I = 77; Kp = 58, K_I = 154. These figures show that when Kp = 29 and K_I = 77, control is better than the others but still the tuned Kp = 36 and K_I = 96 provides the better result. So to control this profile with PI controller optimum values of Kp and K_I are 36 and 96.

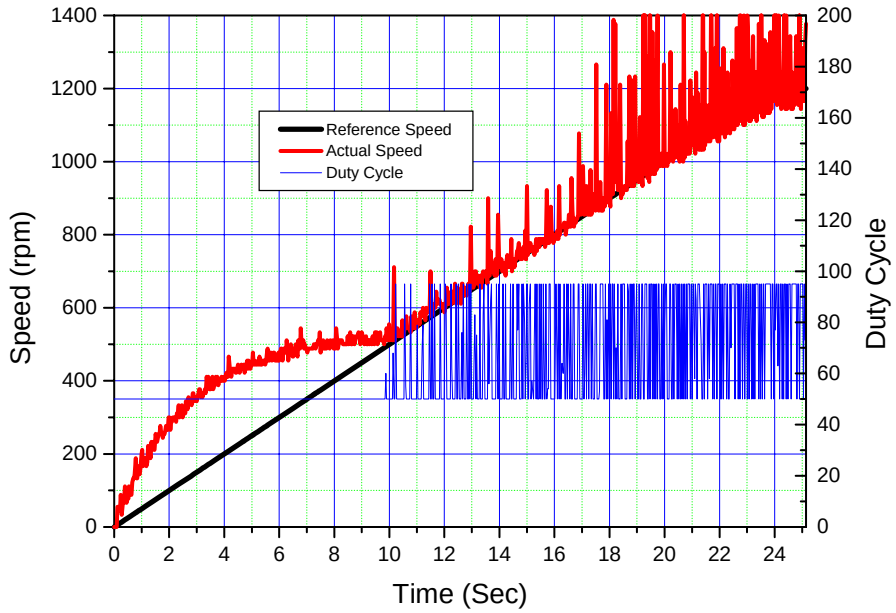


Fig. 4.53: PI Control with Kp = 36 and K_I = 96 for reference speed profile 1

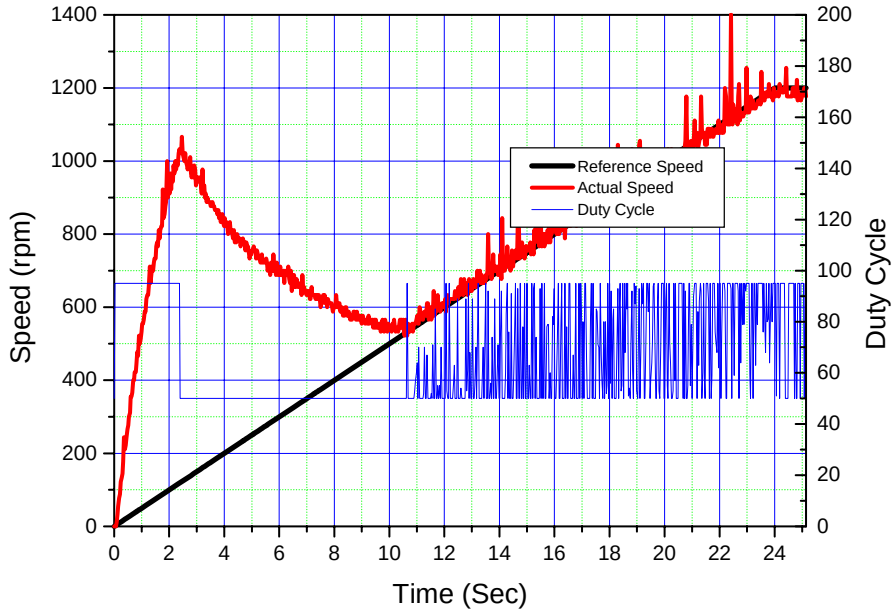


Fig. 4.54: PI Control with Kp = 14 and K_I = 38 for reference speed profile 1

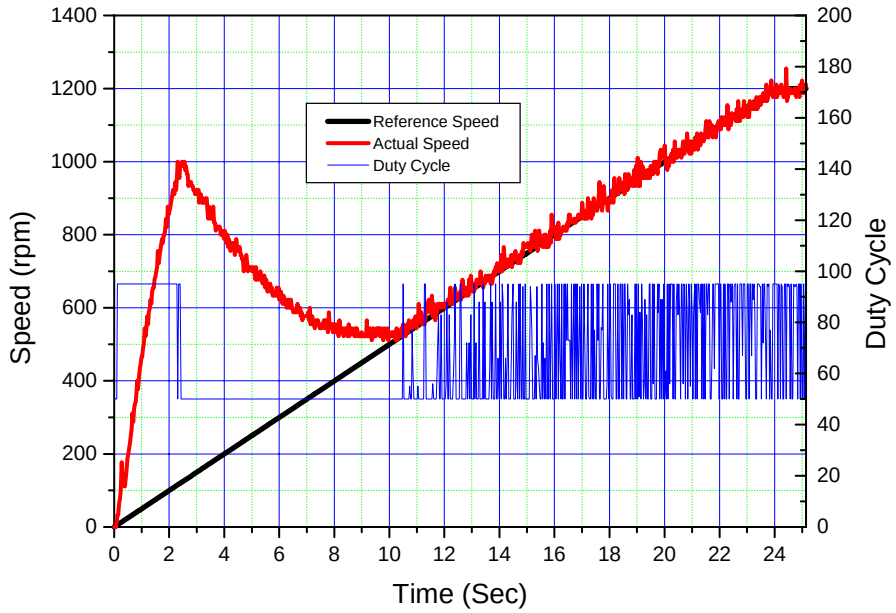


Fig. 4.55: PI Control with Kp = 22 and K_I = 58 for reference speed profile 1

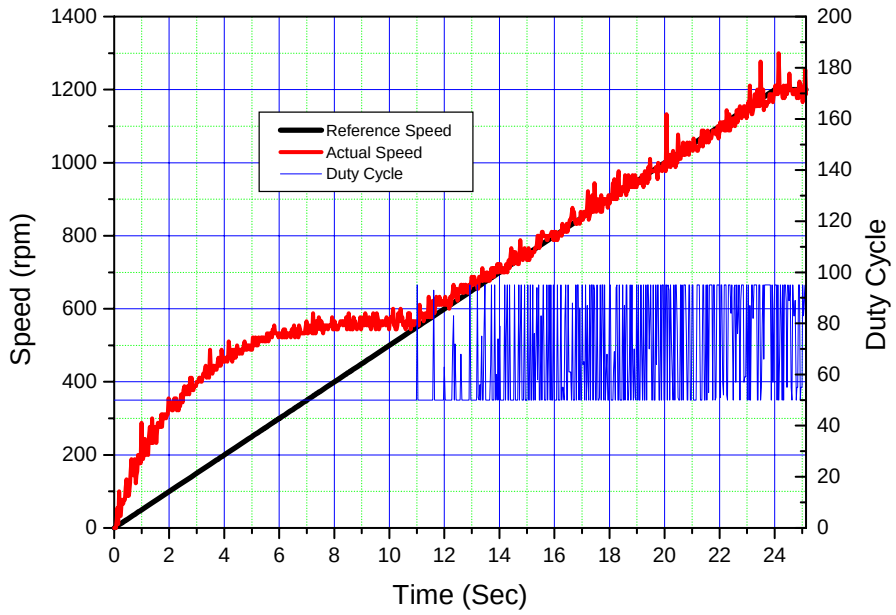


Fig. 4.56: PI Control with Kp = 29 and K_I = 77 for reference speed profile 1

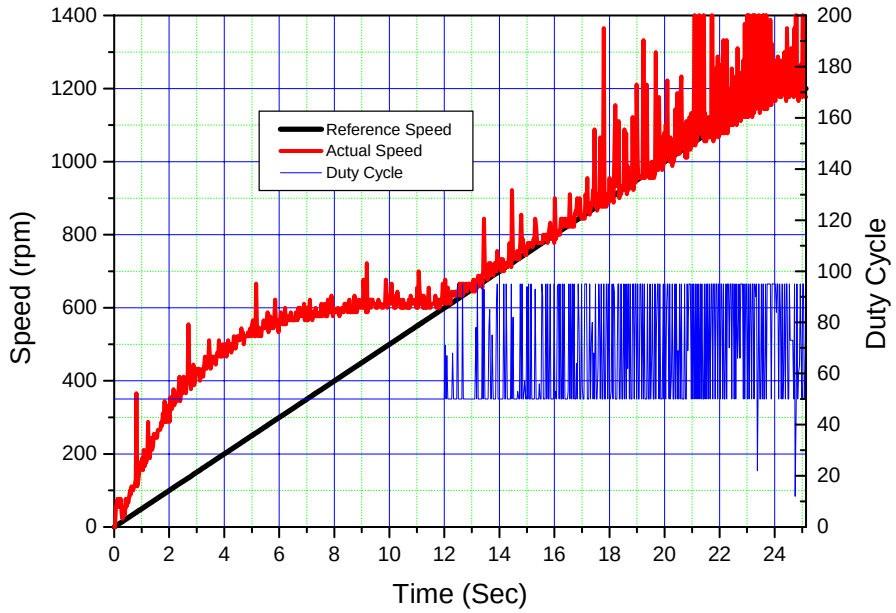


Fig. 4.57: PI Control with Kp = 58 and K_I = 154 for reference speed profile 1

Controller was tuned for this reference speed profile considering it that the optimum value of Kp to control this with proportional controller is from 32 to 40. For those Kp = 29, K_I = 77 and Kp = 36, K_I = 98. Fig. 4.58, 4.59, 4.60, 4.61 and 4.62 show the control with Kp = 29, K_I = 77; Kp = 36, K_I = 96; Kp = 14, K_I =38; Kp = 22, K_I = 58 ; Kp = 58, K_I = 154. These figures show that when Kp = 36 and K_I = 96, control is better than the others. So to control this profile with PI controller optimum values of Kp and K_I are 36 and 96.

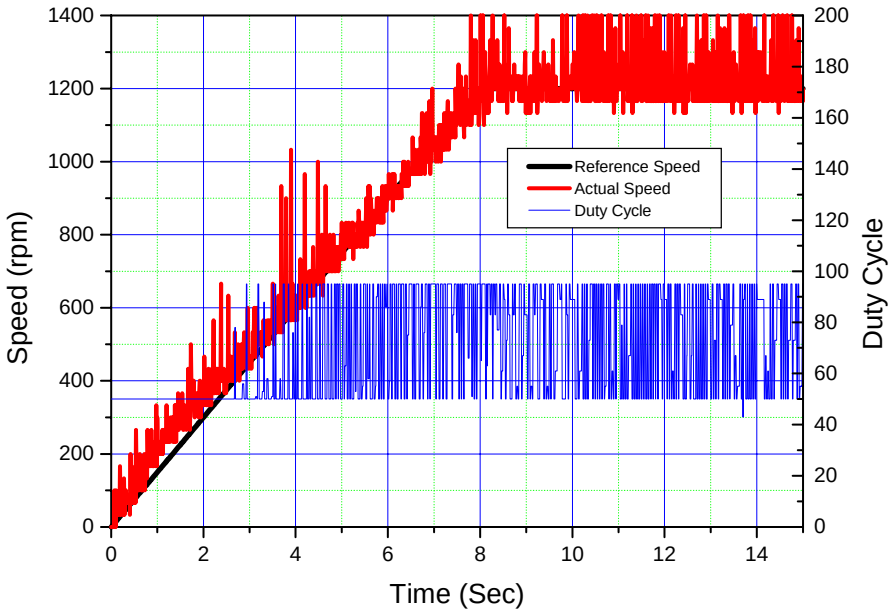


Fig. 4.58: PI Control with Kp = 29 and K_I = 77 for reference speed profile 2

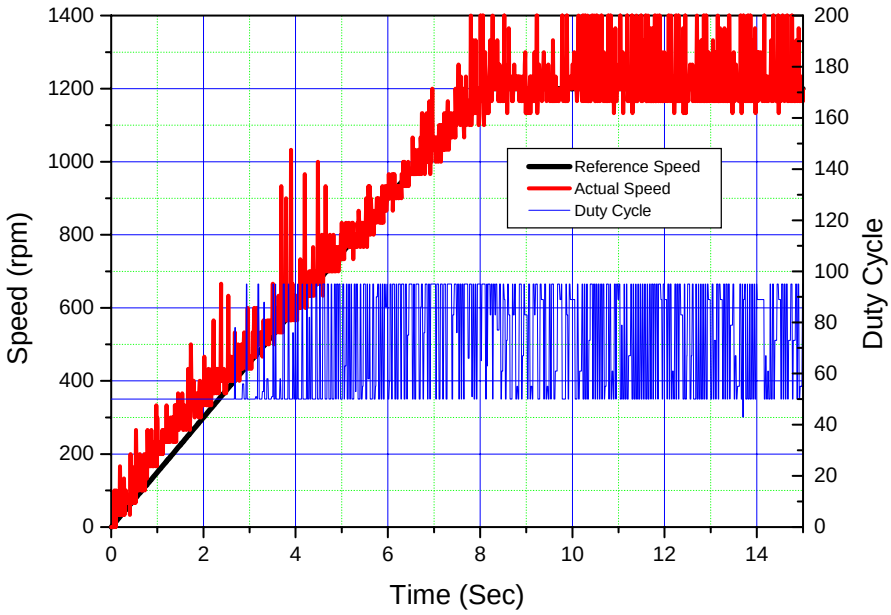


Fig. 4.59: PI Control with Kp = 36 and K_I = 96 for reference speed profile 2

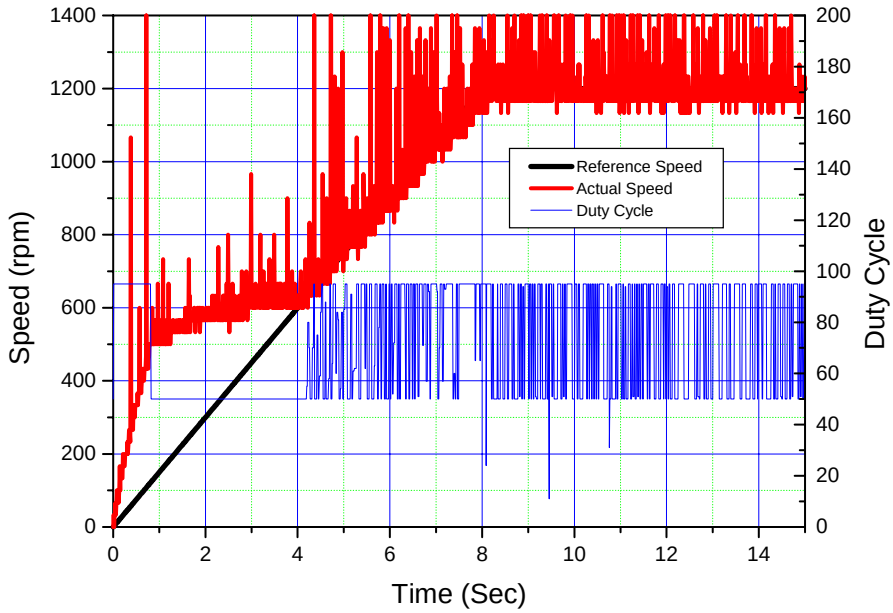


Fig. 4.60: PI Control with Kp = 14 and K_I = 38 for reference speed profile 2

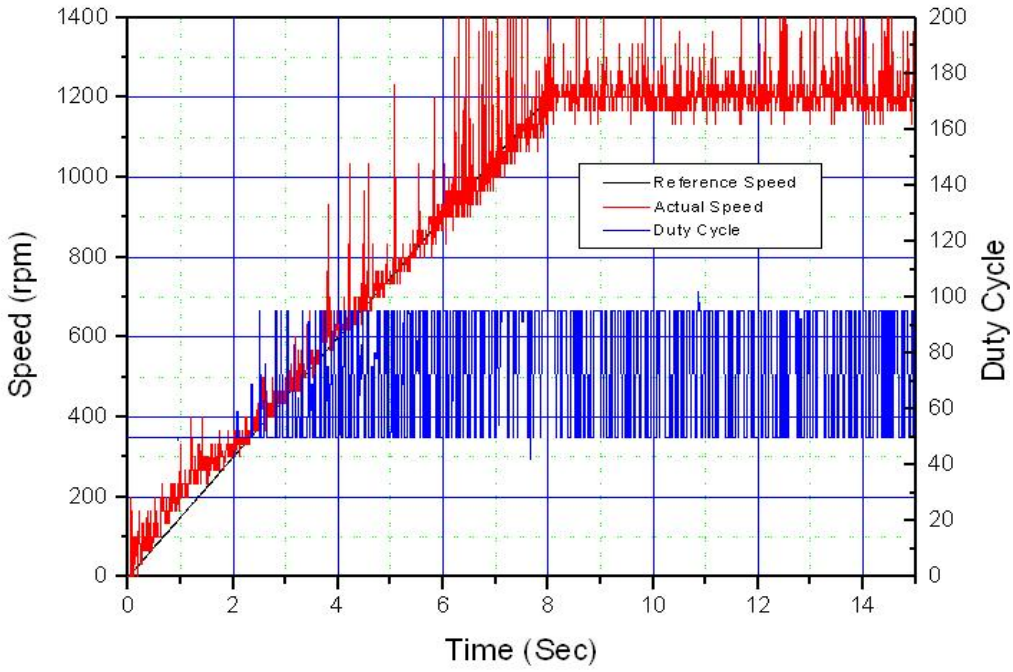


Fig. 4.61: PI Control with Kp = 22 and K_I = 58 for reference speed profile 2

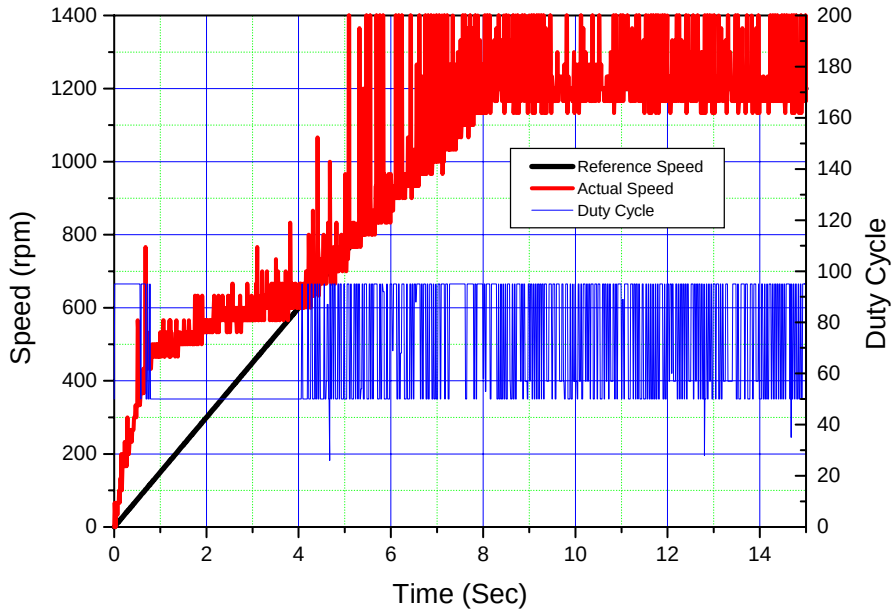


Fig. 4.62: PI Control with Kp = 58 and K_I = 154 for reference speed profile 2

Controller was tuned for this reference speed profile considering it that the optimum value of Kp to control this speed profile with proportional controller is 40. For this Kp PI control parameters are Kp = 36, K_I = 96. Fig. 4.63, 4.64, 4.65, 4.66 and 4.67 show the control with Kp = 36, K_I = 96; Kp = 14, K_I =38; Kp = 22, K_I = 58 ; Kp = 29, K_I = 77 and Kp = 58, K_I = 154. These figures show that when Kp = 36 and K_I = 96 control is better than the others. So to control this profile with PI controller optimum values of Kp and K_I are 36 and 96.

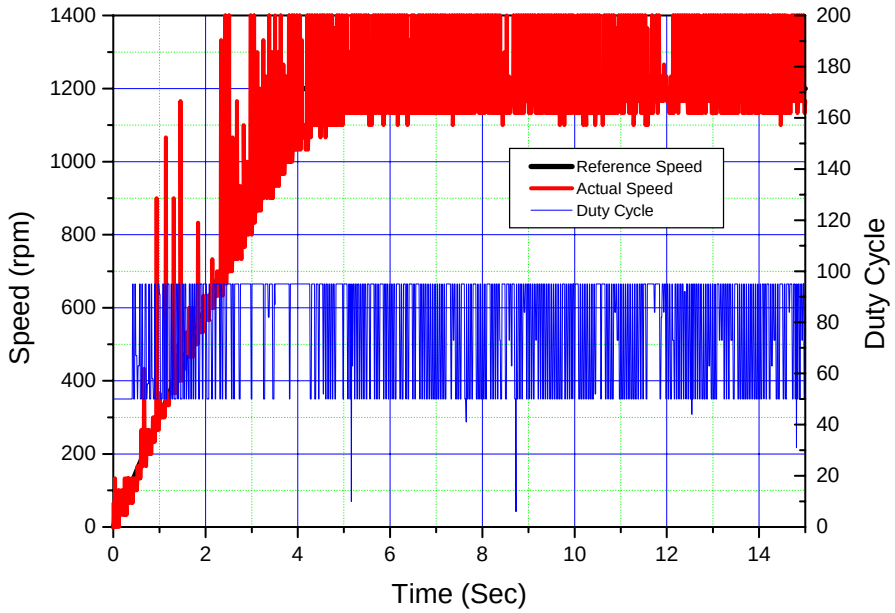


Fig. 4.63: PI Control with Kp = 36 and K_I = 96 for reference speed profile 3

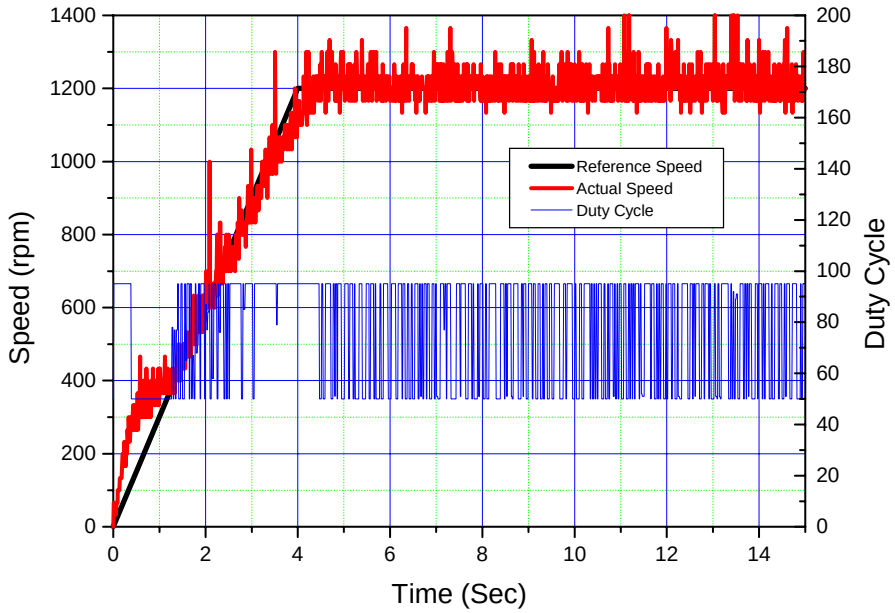


Fig. 4.64: PI Control with Kp = 14 and K_I = 38 for reference speed profile 3

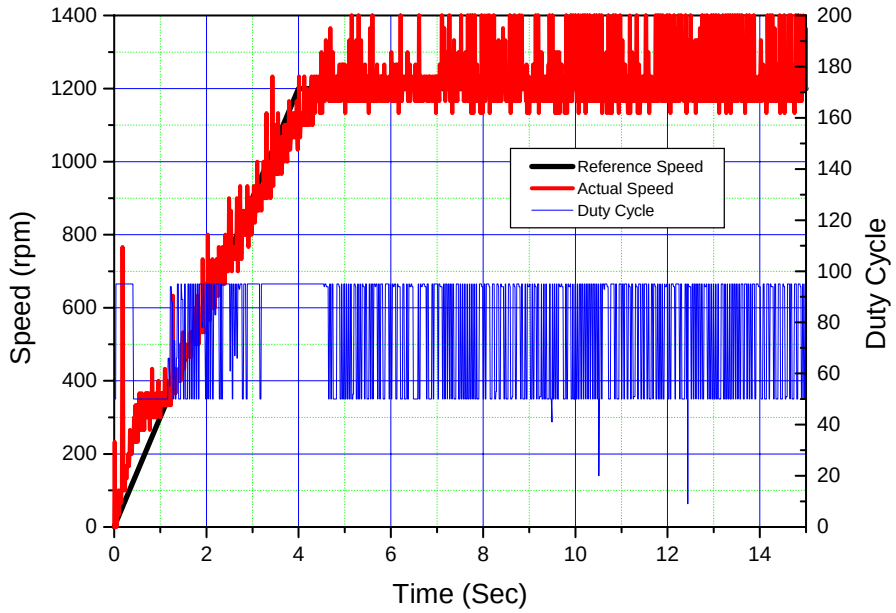


Fig. 4.65: PI Control with Kp = 22 and K_I = 58 for reference speed profile 3

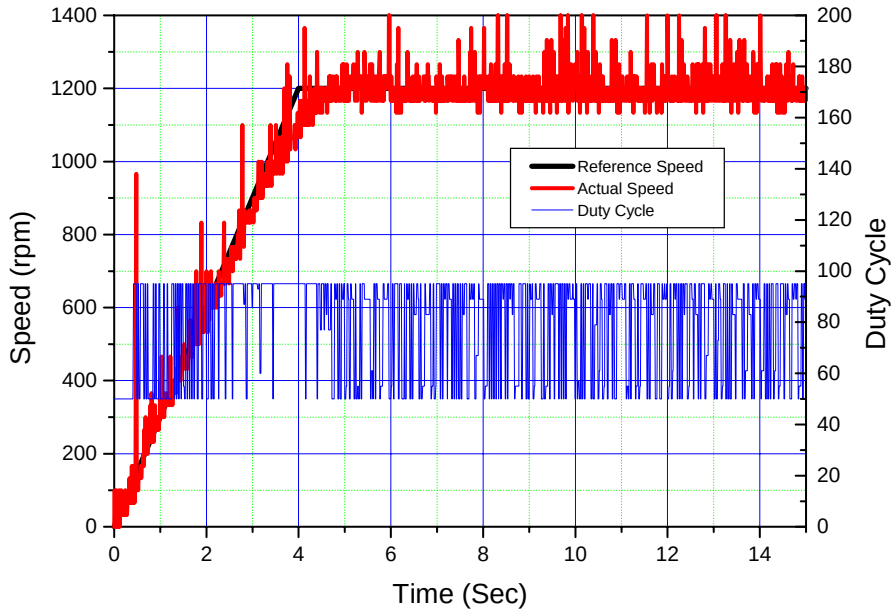


Fig. 4.66: PI Control with Kp = 29 and K_I = 77 for reference speed profile 3

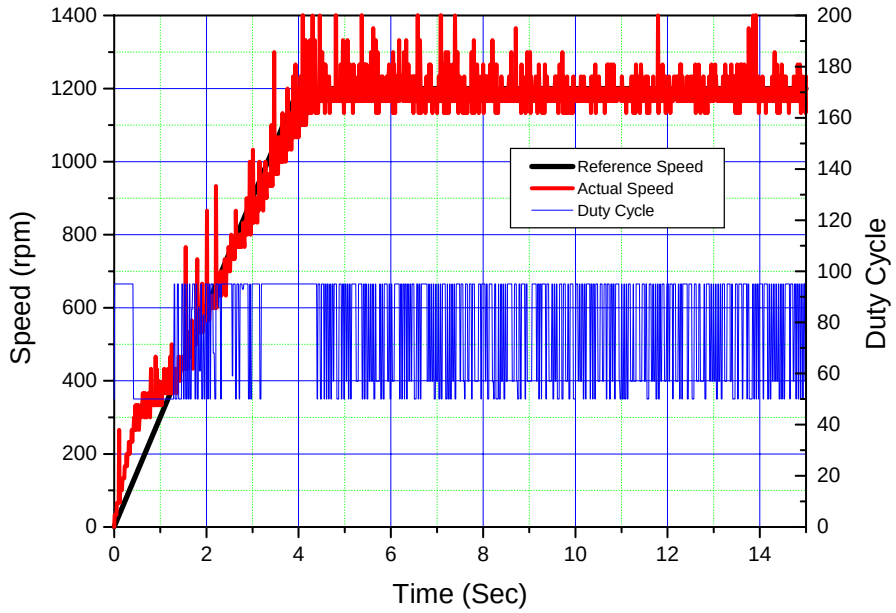


Fig. 4.67: PI Control with Kp = 58 and K_I = 154 for reference speed profile 3

Finally D component was added with PI controller to develop PID controller to control these speed profiles. While adding D component special care was taken as any large value of K_D may cause sudden changes in speed if any change of error is occurred. So to avoid this K_I and K_D component were fixed with a very small value to get their advantage but P component was increased to obtain the optimum control. To control reference speed profile 1 with PID control K_I and K_D component were fixed at 1/2 but K_P was started applying with 20 and increased upto 80. Fig. 4.68, 4.69 and 4.70 show the control with K_P of 20, 40 and 80 while K_I and K_D were fixed at 1/2. From these figures it was seen that error was significantly reduced with the increase in Kp. Fig. 4.70 shows the best control comparing with other controllers.

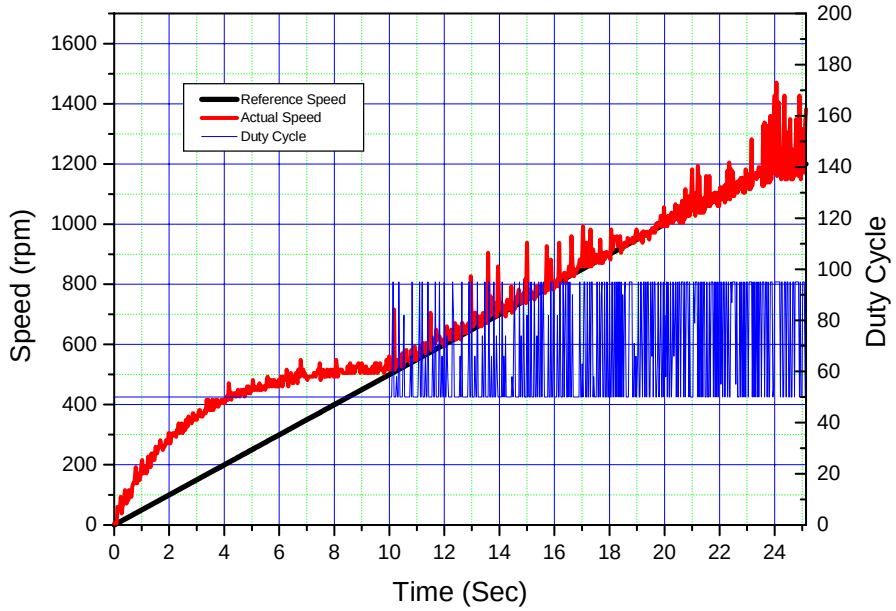


Fig. 4.68: PID Control with Kp = 20, K_I = 1/2 and K_D = 1/2 for reference speed profile 1

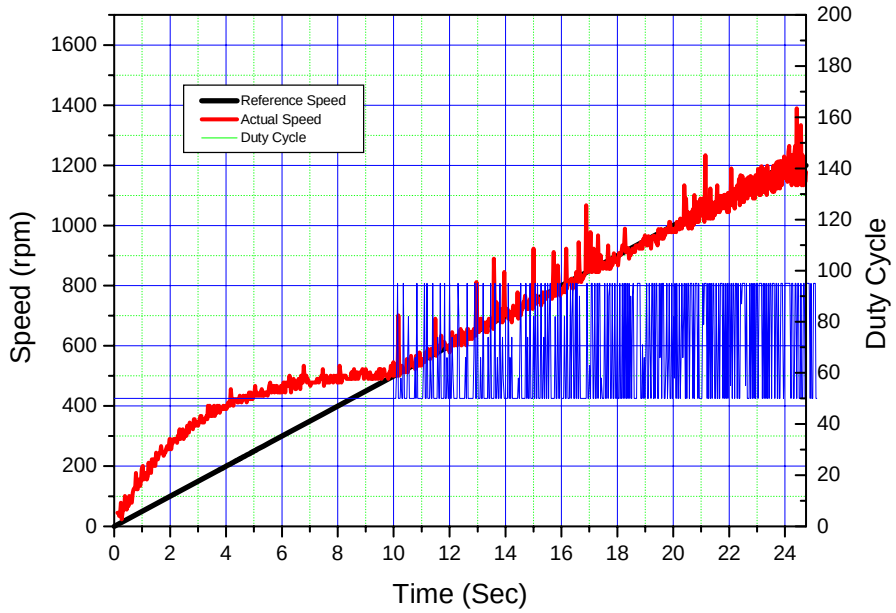


Fig. 4.69: PID Control with Kp = 40, K_I = 1/2 and K_D = 1/2 for reference speed profile 1

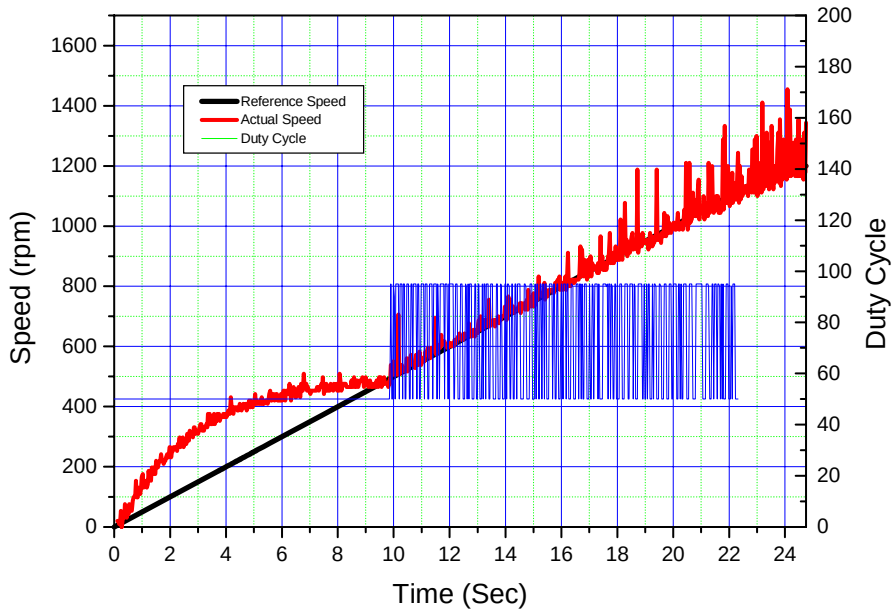


Fig. 4.70: PID Control with Kp = 80, K_I = 1/2 and K_D = 1/2 for reference speed profile 1

To control reference speed profile 2 using PID controller, initially started with a very low value of Kp = 48, K_I = 2 and K_D = 2. Fig. 4.71 shows it. It shows that when rate of change of speed reduced to zero after reaching 1200 rpm, D component suddenly decreased the control action a lot. So actual speed fallen down below the reference speed and could not recover it again. Value of Kp is further increased in Fig. 4.72 and 4.73 by 80 and 90 but due to the derivative action control system failed at the period when rate of change of reference speed became zero. So, K_I and K_D were reduced to 1/2 and were fixed. Then with Kp = 90, significant improvement was noticed in Fig. 4.74. Except the initial uncontrolled period control was excellent over the whole profile. Further increase of Kp with 120 and 125 fixing K_I and K_D at 1/2, makes the control better in Fig. 4.75 and 4.76 but could diminish the uncontrolled period totally. As I controller takes action against the period of error K_I was increased to 20 fixing the other components at their values in Fig. 4.77. It improved the control very clearly. The initial uncontrolled period almost reduced to zero but still not zero. K_I was further increased to 40 in Fig. 4.78 which shows the best control for reference speed profile 2. For any further improvement K_I was increased again to 80 but it shows the breakdown of control at fixed reference speed 1200 rpm. Actual speed was oscillating over the reference speed shown in Fig. 4.79. So the optimum value of Kp, K_I and K_D for reference speed profile 2 is 125, 1/2 and 1/2 respectively.

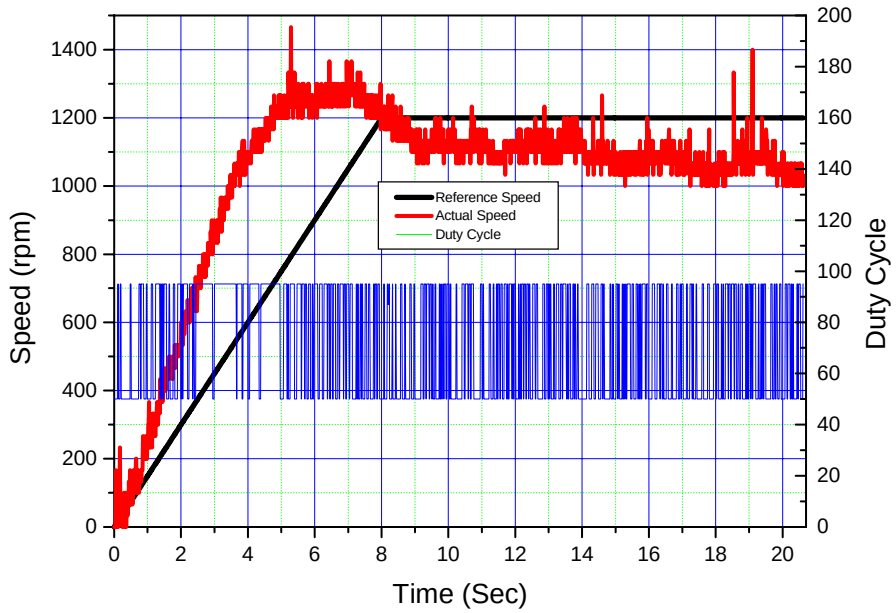


Fig. 4.71: PID Control with Kp = 48, K_I = 2 and K_D = 2 for reference speed profile 2

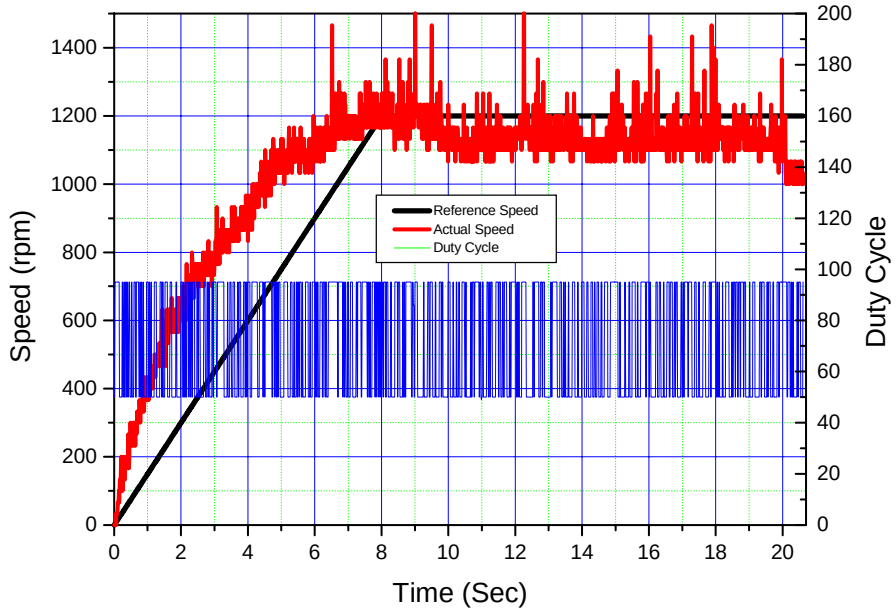


Fig. 4.72: PID Control with Kp = 80, K_I = 2 and K_D = 2 for reference speed profile 2

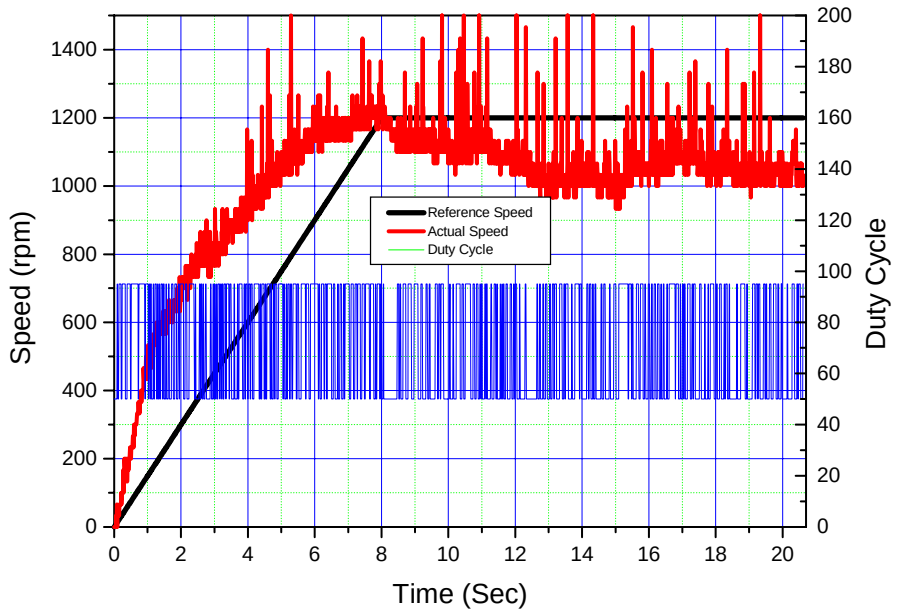


Fig. 4.73: PID Control with Kp = 90, K_I = 2 and K_D = 2 for reference speed profile 2

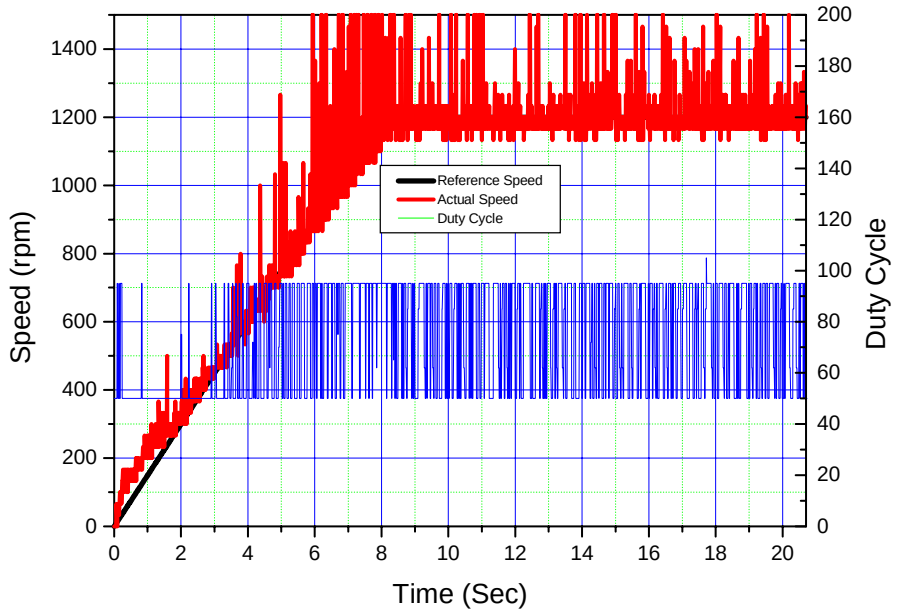


Fig. 4.74: PID Control with Kp = 90, K_I = 1/2 and K_D = 1/2 for reference speed profile 2

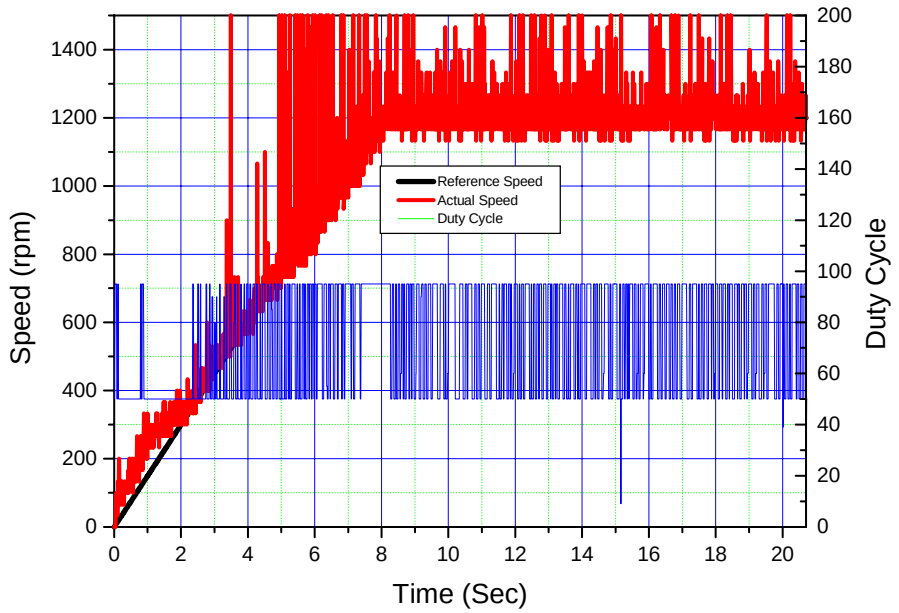


Fig. 4.75: PID Control with Kp = 120, KI = 1/2 and K_D = 1/2 for reference speed profile 2

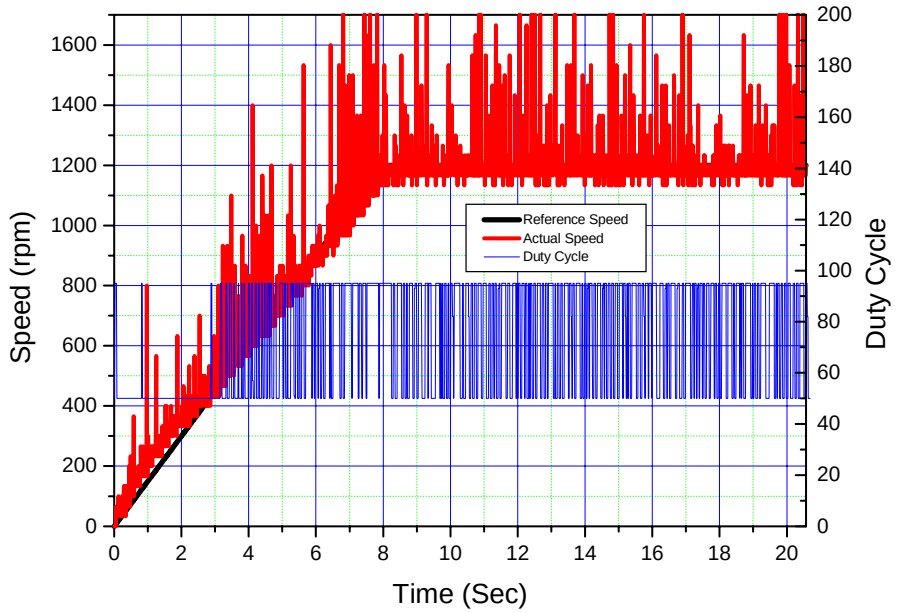


Fig. 4.76: PID Control with Kp = 125, K_I = 1/2 and K_D = 1/2 for reference speed profile 2

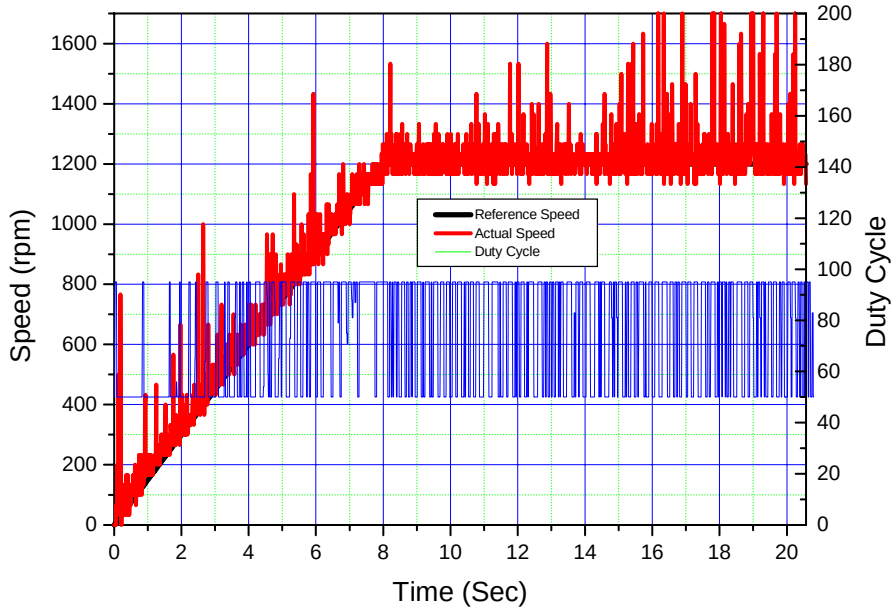


Fig. 4.77: PID Control with Kp = 125, K_I = 20 and K_D = 1/2 for reference speed profile 2

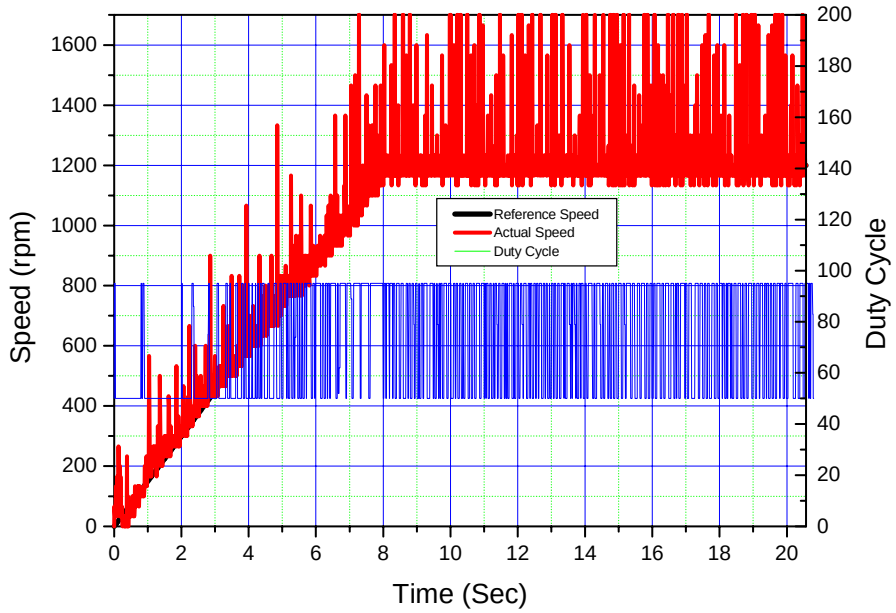


Fig. 4.78: PID Control with Kp = 125, K_I = 40 and K_D = 1/2 for reference speed profile 2

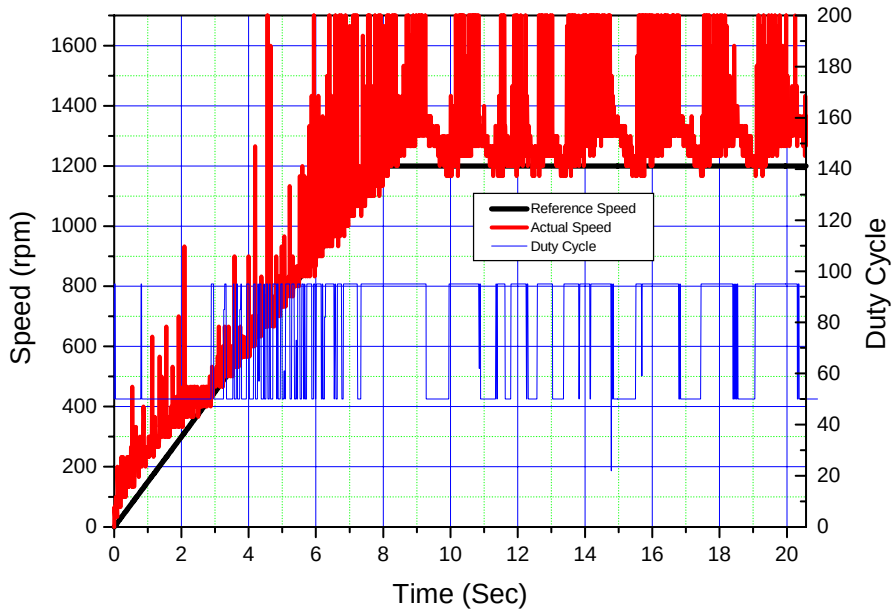


Fig. 4.79: PID Control with Kp = 125, K_I = 80 and K_D = 1/2 for reference speed profile 2

Experience of tuning the controller for reference speed profile 2 made it easier to tune it for reference speed profile 3. K_I and K_D were fixed at a low value 1/2 to get their advantages but Kp was increased in different steps to find the optimum control. Controller was started tuning with a Kp of 40 shown in Fig. 4.80. It shows that with the addition of D component overshoot is significantly reduced. But still there is a problem in control at the higher speed portion of the profile before it is fixed at 1200 rpm. To tune it Kp is further increased with 80 and 160 shown in Fig. 4.81 and 4.82. In Fig. 4.81 it shows that when Kp = 80, this problem is reduced. Further it is increased to Kp = 160, control action did not improve but overshoot is increased shown in Fig. 4.82. From these figures it is noticed that optimum control is obtained at Kp = 80 and $K_I = K_D$ = 1/2.

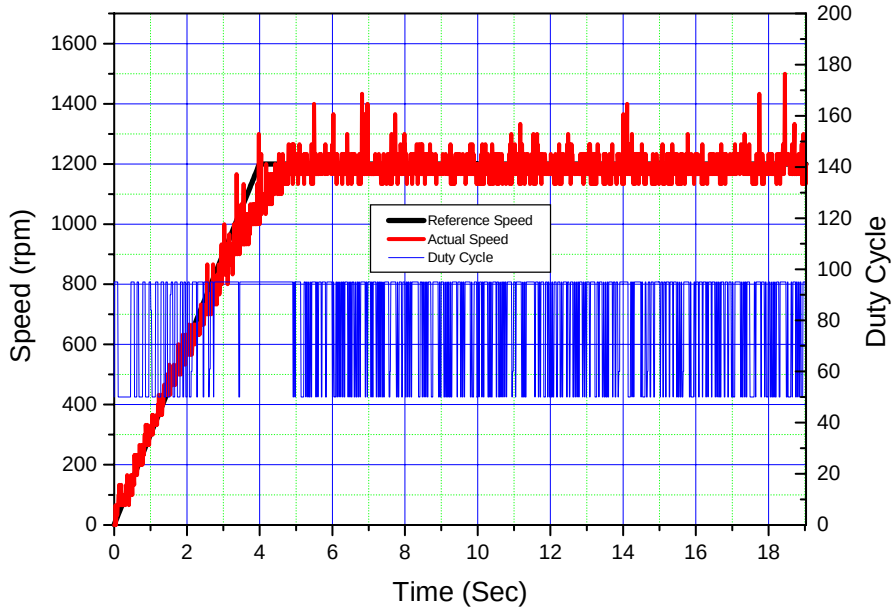


Fig. 4.80: PID Control with Kp = 40, KI = 1/2 and K_D = 1/2 for reference speed profile 3

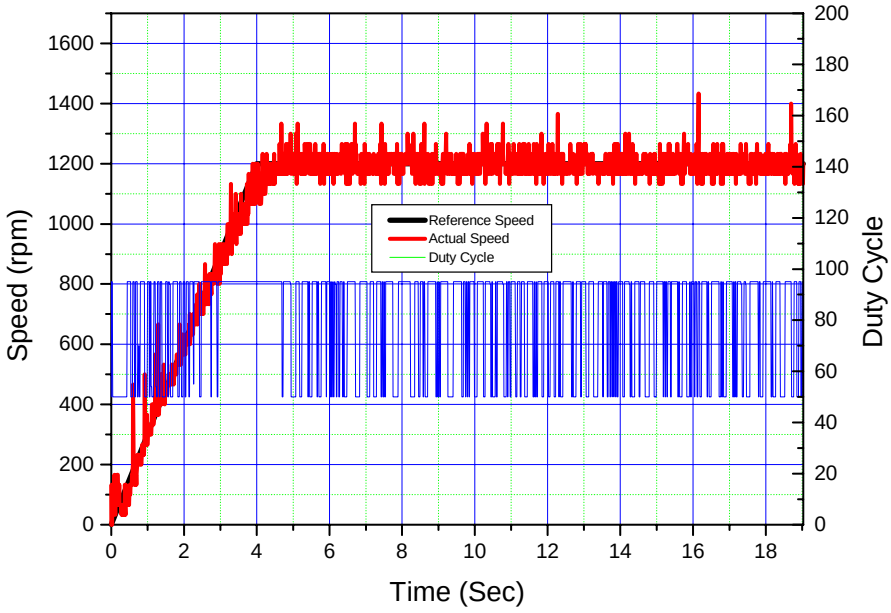


Fig. 4.81: PID Control with Kp = 80, KI = 1/2 and K_D = 1/2 for reference speed profile 3

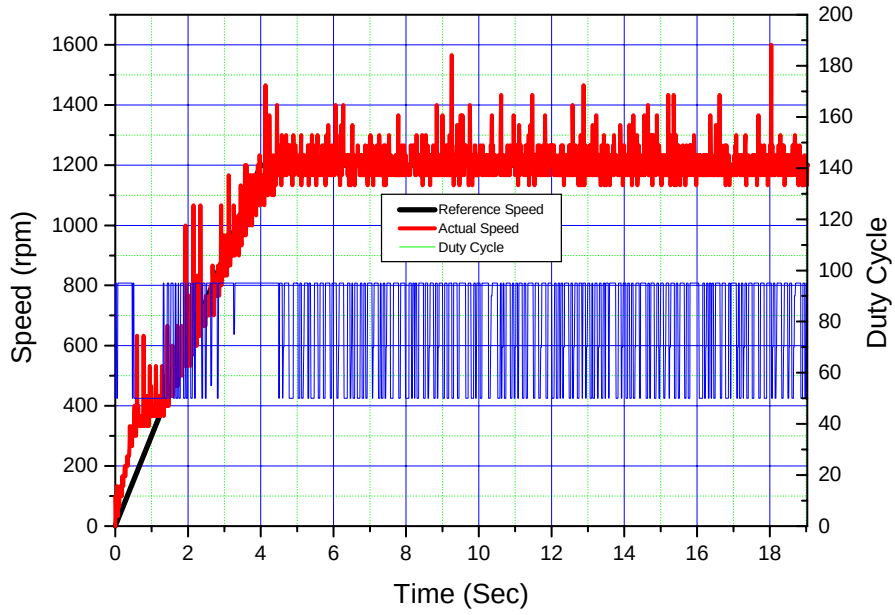


Fig. 4.82: PID Control with Kp = 160, KI = 1/2 and K_D = 1/2 for reference speed profile 3

Controllers are then compared here in Fig. 4.83, 4.84 and 4.85 for the reference speed profiles 1, 2 and 3. From Fig. 4.83 it is clear that PID controller is working the best of all. From 4.82 and 4.83 it is seen that ON-OFF controller is worst of the all but overshoot is low here. P controller improves the control but initial uncontrolled period is not completely diminished with this control. PI controller reduces this problem but overshoot is increased. But after D component is approached, PID control removes both of these two problems decreasing initial uncontrolled period and overshooting.

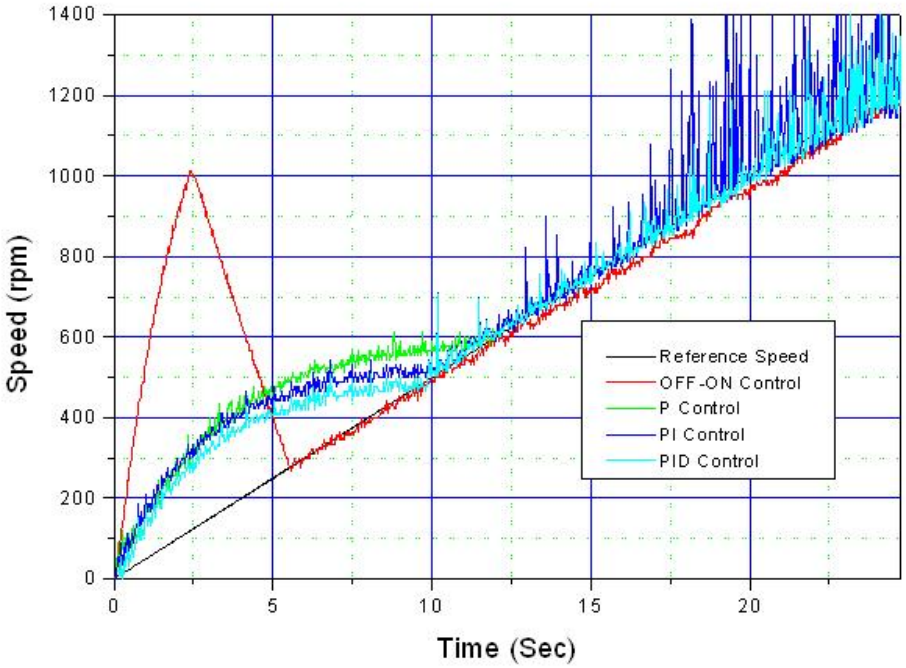


Fig. 4.83: Comparison of Control for Reference Speed profile – 1

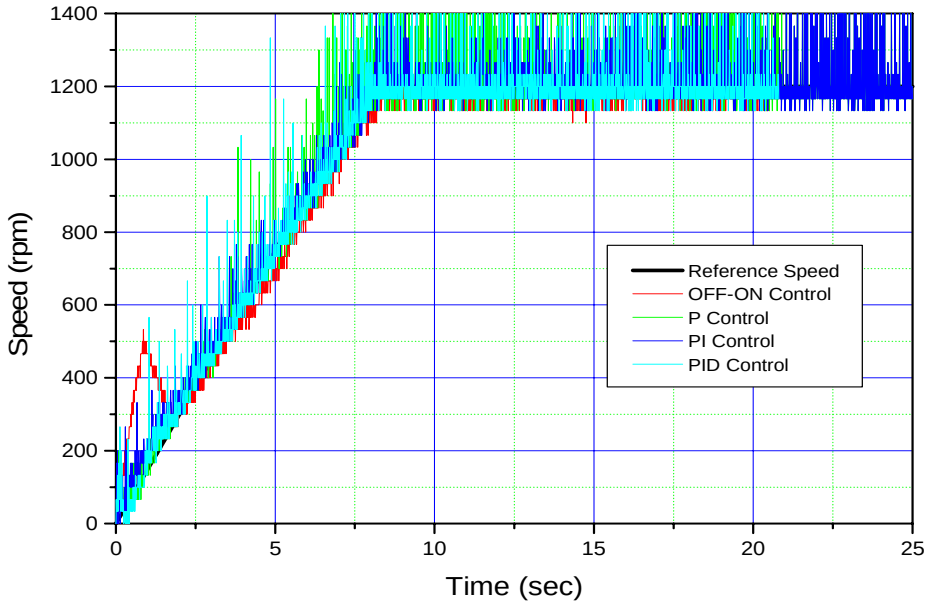


Fig. 4.84: Comparison of Control for Reference Speed profile - 2

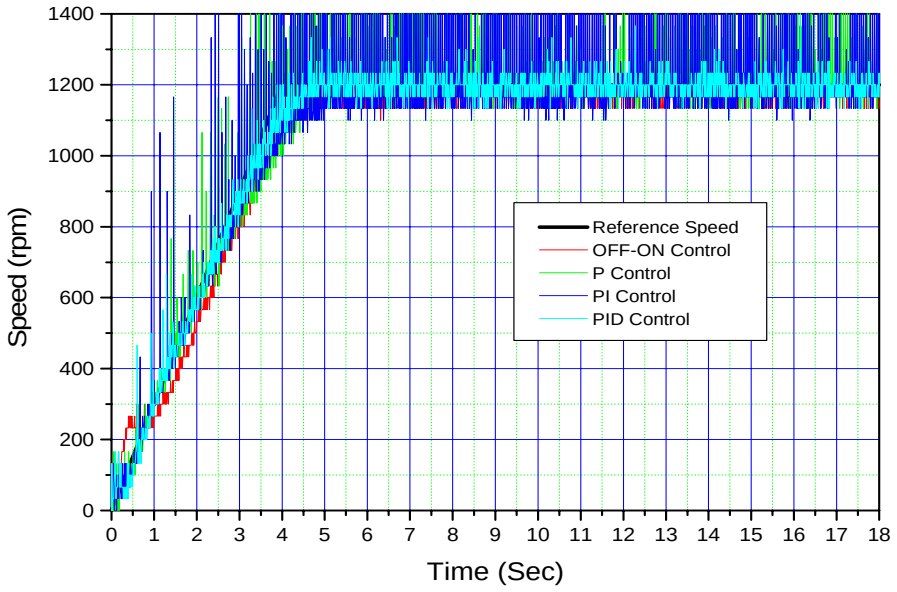


Fig. 4.85: Comparison of Control for Reference Speed profile - 3

4.2 Conclusions

Conclusions of this present study are as follows:

a) PWM (Pulse Width Modulation) is an effective means to control DC motor speed.
b) ON-OFF Controller is not suitable to control speed of a DC motor when it is just started.
c) P Controller can control it better than ON-OFF control but can't diminish the initial problem completely.
d) PI controller can remove the initial problem but creates another problem of overshooting of actual speed.
e) PID controller when tuned carefully can solve both of these two problems, but if larger value of derivative gain is used it makes system instable.

4.3 Scope of Further Works

a) Controlling of complex speed profiles with PID Controller.
b) Controlling of Robot Drive with variable load using PID Controller.
c) Improvement of Controller performance at low speed.
d) Enhancement of controller performance reducing overshoot.

References

[1] Sugisaka, M., Hazry, D. 2007, “Development of a Proportional Control Method for a Mobile Robot”, Applied Mathematics and Computation 186:74 –82.

[2] M. Y. Ali, S. G. M. Hossain, H. Jamil and M. Z. Haq, “Development of Automated Guided Vehicles for Industrial Logistics Applications in Developing Countries Using Appropriate Technology”, International Journal of Mechanical & Mechatronis Engineering IJMME-IJENS Vol:10 No:02.

[3] Kabir, M, E. (2009), Dynamic characteristics of servocontrolled Mobile robot using optimum pulse width modulation (PWM).

[4] Joseph L, J. Seiger, B,A. Flynn, A,M. (1999), Mobile Robots Inspiration to Implimentation, second edition

[5] Klafter, R, D., Thomas A, C. Michael N.(2005), Robotic Engineering An Integrated Approach

[6] Kilian, C, T. (2004), Modern Control Technology: Components and System, 2nd edition

[7] Gordon, M., Predko, M. (2006), Robot Builder’s Bonanza

[8] Alciatore, D, G., Histand, M, B. (2007), Introduction to Mechatronics and Measurement Systems, Third Edition

[9] Bartelt, T.(2009), Industrial Electronics Circuits, Instruments and Control Techniques

[10] Godfrey, C, O. (2008), Mechatronics Priciples and Applications

[11] Kuo B, C. [1995] Automatic Control Sytem, seventh edition

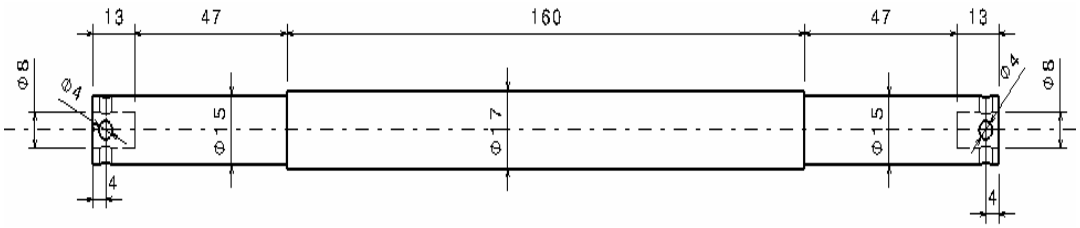
[12] Schleicer, M., Blasiner, F. (2003), Control Engineering a guide for beginners.

[13] Considine, D.M. (1993), Process/Industrial Instruments & Control Handbooks, McGrawHill Co. Ltd.

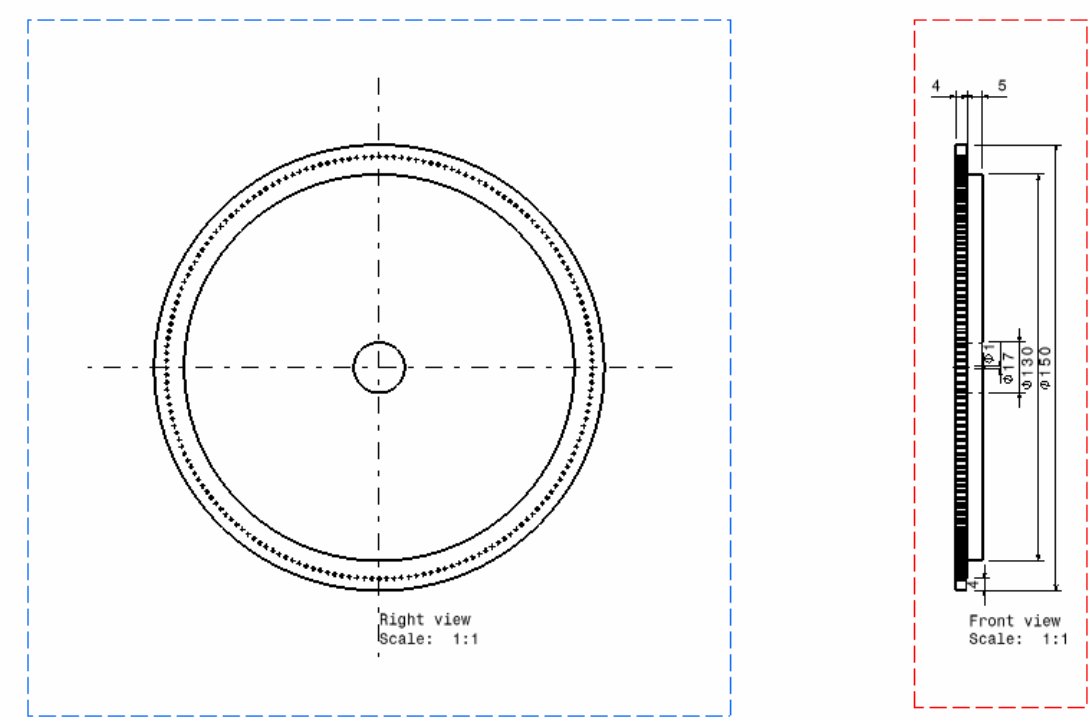
[14] Johnson, C, D. (2007), Process Control Instrumentation Technology, eighth edition

Appendix

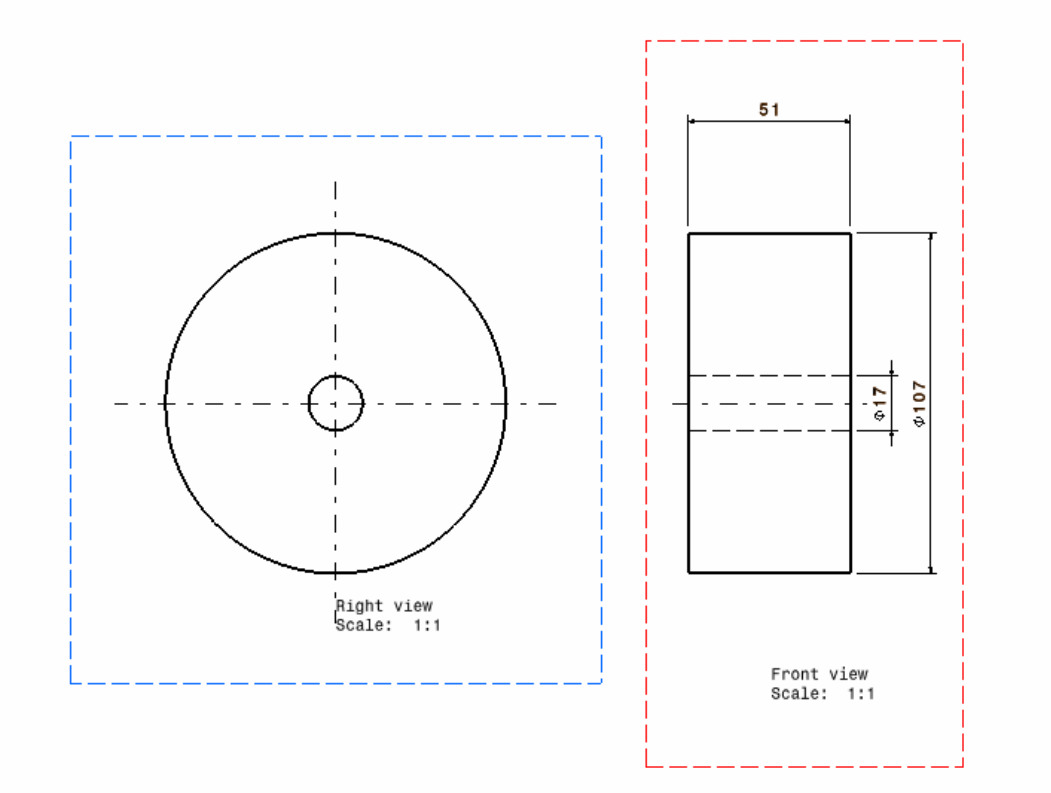
Detail Design of Setup Components with Dimensions



Shaft



Encoder



Flywheel