Optimum PID Control of A Servo-Motor Subjected to Frictional Loads, Inertia and Disturbances

by

Kazi Arafat Rahman

Department of Mechanical Engineering

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

A thesis submitted for the degree of
Master of Science

July 23, 2014

Certificate of Approval

The thesis titled, "**Optimum PID Control of A Servo-Motor Subjected to Frictional Loads, Inertia And Disturbances**", submitted by Kazi Arafat Rahman, Roll No: 0411102078 P, Session: April, 2011, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN MECHANICAL ENGINEERING** on **July 23, 2014**.

Dr. Md. Zahurul Haq Professor & Head Department of Mechanical Engineering BUET, Dhaka, Bangladesh	Chairman (Supervisor & Ex-Officio)
Dr. Mohammad Arif Hasan Mamun Professor Department of Mechanical Engineering BUET, Dhaka, Bangladesh	Member (Internal)
Dr. Mohammad Mamun Professor Department of Mechanical Engineering BUET, Dhaka, Bangladesh	Member (Internal)
Dr. Md. Shafiqul lslam Principal Engineer Nuclear Safety, Security & Safeguards Division BAEC, Dhaka, Bangladesh	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.

____

Kazi Arafat Rahman

Acknowledgements

The author would like to thank Professor Dr. Md. Zahurul Haq for all his help, support, and seemingly endless enthusiasm. Being within his “radius of understanding” allowed the author to gain much insight into the field of control as well as many other fields of research and life. Without him this research would not have been possible.

Author also expresses the deepest gratitude and thanks to everyone at BUET who helped him one way or other in course of writing it.

Abstract

The control of dc servo-motors has been, and still is, the subject of numerous research activities. These motors are used in wide range of industrial and mobile robot applications. Traditionally, dc motors have been controlled by ON-OFF controllers. With such simple controllers it is very difficult to achieve precise motion and path control as required by many robotic and servo applications. The rise in popularity of microcontrollers and microcontroller based precision data acquisition system, in recent years, have opened up new arenas for creating smart controllers for such robotic systems.

Now-a-days, proportional controllers are being investigated for servo-motors. However, proportional controllers have inherent steady-state error and these fail to respond quickly in case of rapid load variations. Therefore, precise trajectories are not achievable in many applications. PID controllers seem attractive for such systems and are now being tried to be implemented in microcontroller. However, PID controllers require an optimum balance of proportional (P), integral (I) and derivative (D) control actions and several procedures are available for disposal. In the present study, Ziegler-Nichols method is used to have optimum control parameters of the PID controller for a dc servo-motor subjected to frictional loads, inertia and disturbances. The control action is implemented in microcontroller based data acquisition and control system - LabJack. Both data acquisition and control actions are implemented in python scripting language. Performance results are obtained using these controlled parameters and results are analyzed to obtain optimum PID control strategy using LabJack for servo-motors.

List of Symbols

D	Derivative
DC, dc	Direct current
e	Control deviation
I	Integral
J	Inertia load
Kcr	Critical gain
K_D	Derivative gain
K_I	Integral gain
K_p	Proportional gain
Ks	Process transfer co-efficient
L	Inductance
MV	Manipulated variable
P	Proportional
PT1	First order system
PT2	Second order system
PV	Process variable
PWM	Pulse Width Modulation
R	Resistance
SP	Setpoint
Pcr	Critical period
$T_{d,}$ T_v	Derivative time
Tg	Delay time
Ti	Integral time
Tn	Reset time
V	Volt
x	Process variable
y	Manipulated variable
y_o , y(0)	Initial value of controller
z	Disturbance

Table of Contents

List of Figures

List of Tables

Chapter 1 : INTRODUCTION

1.1 Background

The control of dc servo-motors has been, and still is, the subject of numerous research activities [1-2].These motors are used in wide range of industrial and mobile robot applications. Traditionally, dc motors have been controlled by ON-OFF controllers. With such simple controllers it is very difficult to achieve precise motion and path control as required by many robotic and servo applications [3-4]. The rise in popularity microcontrollers in recent years have opened up new arenas for creating smart controllers for such robotic systems.

Motor control and then especially motor's motion control constitutes one of the most important key components for the development of industrial robotics. By applying and developing advanced control, it is possible to continuously improve the motor performance, which is necessary in order to increase performance and lower cost of industrial robot automation [5, 6].

1.2 Motivation

Automatic control is becoming more and more important in this age of automation. In manufacturing processes it ensures that certain parameters, such as temperature, pressure, speed or voltage, take up specific constant values recognized as the optimum, or are maintained in a particular relationship to other variables. In other words, the duty of control engineering is to bring these parameters to certain pre-defined values (setpoints), and to maintain them constant against all disturbing influences [7]. However, this apparently simple duty involves a large number of problems which are not obvious at first glance. This is more obvious in the case of robotic vehicle where there are many applications of driving alone same path with high accuracy. Permanent magnet (PM) brushed DC motors and their control drives are used in a wide range of industrial and mobile robotics applications. Tuning the speed and position control parameters of these highly dynamic servo drives is a time-consuming task. 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. The control scheme for motion stabilization and trajectory tracking may 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 parameters and their proper tuning for stabilized motion and tracking [8, 9]. This thesis intends to develop a controller that will be used to control a dc servo-motor while following a specific speed profile.

1.3 Control Theory

There has been much research in the area of control theory and many modern controllers have been developed as a result. The most widely used controller is still the PID (proportional, integral, derivative) controller because of its simplicity and ease of implementation.With optimum feedback control using suitable sensors it is possible to achieve higher accuracy for motor drives in doing some specific task as following a specific velocity profile. Among the classes of controllers that has been implemented in this thesis are: Proportional (P), Integral (I) And Derivative (D) controllers. Other composite control moods such as PI and PID controller were also implemented in this thesis to get precise velocity profile for a dc motor.

1.4 Controller Tuning

A variety of approaches of tuning have been reported in the literature. Specifically, manual tuning, Ziegler–Nichols(ZN) tuning [10], and robust control tuning[11] are widely used in industrial applications. System parameters, such as load inertia and friction, change with operating conditions; for this reason, it is highly desirable to have an automatic tuning method in place. Design of the PID controller based on application of tuning formulae is initiated by Ziegler and Nichols and still predominates over the optimization methods [12-14]. The basic idea of this approach is to find some transformations of a small amount of measured process characteristics into the values of the PID gains.

1.5 Contributions of this Thesis

The following contributions were made during this thesis work:

- A simulation environment is developed to test the various control algorithms used on controlling dc servo-motor.
- To model the motor under fixed and inertial load, a motor control test bench is used.
- Different sensors, measuring instruments are implemented for getting feedback control.
- Different control algorithms have been studied and implemented.

- To program a high speed data acquisition and control systems- Labjack [38] and to log the system response and controller actions under actual operating condition.
- To find the optimum control, systems response (i.e. maximum overshoot, rise time, settle time etc.) is analyzed.
- Finally these control algorithms are implemented on the drive motor of the motor control test bench to find whether it can follow a specific velocity profile.

1.6 Organization of this Thesis

This thesis is organized as follows:

- Chapter 2 describes the literatures available in the field of PID control. It also discusses the advances on the field of getting optimum PID controller parameters.

- Chapter 3 gives the drive control and actual speed measurement techniques for dc servo motor. It describes how actual speed are being measured and used to get error signals. Using this error signal controller decides the control actions.

- Chapter 4 fully describes the simulation environment and gives simulation results. Simulation was done using SimApp. Using SimApp a simulation environment was created which closely approximate the no load condition of our servo motor. Different control parameters were then tested using that model and their influence on the motor was observed.

- Chapter 5 provides complete documentation of the hardware, electronics and methodology implementation for the controlling of dc servo-motor in the motor control test bench. It completely describes the controller used here- LabJack U3-HV. The programming algorithm are also presented here.

- Chapter 6 provides results and discussion of the experiment. The optimum settings were found for PID controller here. This optimum parameters were then tested with different speed profiles.

- Chapter 7 gives conclusions and possibilities for future work.

Chapter 2 : LITERATURE REVIEW

Although the proportional-integral-derivative (PID) controller has only three parameters, it is not easy, without a systematic procedure, to find good values (settings) for them. In fact, a visit to a process plant will usually show that a large number of the PID controllers are poorly tuned [15].

With its three-term functionality offering treatment of both transient and steady-state responses, proportional-integral-derivative (PID) control provides a generic and efficient solution to real world control problems [16]. The three-term functionalities include: The proportional term provides an overall control action proportional to the error signal through the gain factor, The integral term reduces steady-state errors through low-frequency compensation, The derivative term improves transient response through high-frequency compensation. The wide application of PID control has stimulated and sustained research and development to "get the best out of PID'' [17], and "the search is on to find the next key technology or methodology for PID tuning" [18].

Now-a-days, Proportional (P) controllers are being investigated for dc servo-motors. However, proportional controllers have inherent steady-state error and these fail to respond quickly in case of rapid load variations. Proportional control is simple, makes sense, and is the basis of most control systems, but it has one fundamental problem—steady-state error. In practical systems, proportional control cannot drive the controlled variable to zero error because as the load gets close to the desired position, the correcting force drops to near zero. This small force may not be enough to overcome friction, and the load comes to a stop just short of the mark.Therefore, precise trajectories are not achievable in many applications [5, 19-20]. For these reason only P controller is not sufficient for controlling problems and thus the concept of composite controller arises. The first variant of these type of composite controller is the PI (Proportional-Integral) one. The introduction of integral control in a control system can reduce the steady-state error to zero. Integral control creates a restoring force that is proportional to the sum of all past errors multiplied by time. For a constant value of error, the value of $\Sigma(\text{error} * \Delta t)$ will increase with time, causing the restoring force to get larger and larger. Eventually, the restoring force will get large enough to overcome friction and move the controlled variable in a direction to eliminate the error. Although the addition of integral feedback eliminates the steady-state-error problem, it reduces the overall stability of the system. The problem occurs because integral feedback tends to make the system overshoot, which may lead to oscillations. PID controllers seem attractive for such systems and are now being tried to be implemented in microcontroller [21]. One solution to the overshoot problem is to include Derivative (D) control. Derivative control “applies the brakes,” slowing the controlled variable just before it reaches its destination.So, thus came the concept of PID controller. It is interesting to note that more than half of the industrial controllers in use today are PID controllers or modified PID controllers.

A purely mathematical approach to the study of automatic control is certainly the most desirable course from a standpoint of accuracy and brevity. Unfortunately, however, the mathematics of control involves such a bewildering assortment of exponential and trigonometric functions that the average engineer cannot afford the time necessary to plow through them to a solution of his current problem. The usefulness of PID controls lies here, i.e. in their general applicability to most control systems. In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used. A PID controller can be considered as an extreme form of a phase lead-lag compensator with one pole at the origin and the other at infinity. Similarly, its cousins, the PI and the PD controllers, can also be regarded as extreme forms of phase lag and phase-lead compensators, respectively. However, the message that the derivative term improves transient response and stability is often wrongly expounded. Practitioners have found that the derivative term can degrade stability when there exists a transport delay [23].

PID is a generally applicable control technique that derives its success from simple and easy-to-understand operation. However, because of limited information exchange and problem analysis, there remain misunderstandings between academia and industry concerning PID control. These misconceptions may explain why the argument exists that academically proposed PID tuning rules sometimes do not work well on industrial controllers. In practice, therefore, switching between different structures and functional modes is used to optimize transient response and meet multiple objectives [24]. Difficulties in setting optimal derivative action can be eased by complete understanding and careful tuning of the D term.

PID controls prove to be most useful in the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control. So, PID controllers require an optimum balance of proportional (P), integral (I) and derivative (D) control actions and several procedures are available for disposal [22].Because most PID controllers are adjusted on-site, many different types of tuning rules have been proposed in the literature. In the present study, Ziegler-Nichols method is used to have optimum control parameters of the PID controller for a dc servo-motor subjected to frictional loads, inertia and disturbances. The control action is implemented in microcontroller based systems data acquisition system LabJack. Performance results are obtained using these controlled parameters and results are analyzed to obtain optimum PID control strategy using LabJack for dc servo-motors in mobile robotic applications.

Over the past half century, researchers have sought the next key technology for PID tuning and modular realization. Many design methods can be computerized and, with simulation packages widely used, the trend of computerizing simulation- based designs is gaining momentum. Computerizing enables simulations to be carried out automatically, which facilitates the search for the best possible PID settings for the application at hand. A simulation-based approach requires no artificial minimization of the control amplitude and helps improve sluggish transient response without windup.

In tackling PID problems, it is desirable to use standard PID structures for a reasonable range of plant types and operations. Modularization around standard PID structures should also help improve the cost effectiveness of PID control and maintenance. This way, robustly optimal design methods such as PID can be developed. By including system identification techniques, the entire PID design and tuning process can be automated, and modular code blocks can be made available for timely application and real-time adaptation [23].

There has been previous work along these lines, including the classical paper by Ziegler and Nichols [10], the IMC PID-tuning paper by Rivera et al. [25], and the closely related direct synthesis tuning rules in the book by Smith and Corripio [26]. The Ziegler–Nichols settings result in a very good disturbance response for integrating processes, but are otherwise known to result in rather aggressive settings [27, 28], and also give poor performance for processes with a dominant delay. On the other hand, the analytically derived settings are known to result in a poor disturbance response for integrating processes (e.g., [29, 30]), but are robust and generally give very good responses for setpoint changes.

[31] proposed tuning rules for integer-order and fractional-order PID controllers. In particular, the tuning rules allow to minimize the integrated absolute error (for either the set-point following or the load disturbance rejection task) subject to a constraint on the maximum sensitivity. The improvement of the performance that can be achieved by employing a fractional order PID

controller has been specified quantitatively. Indeed, if a PI controller is considered, then the use of a fractional-order integral action does not provide any improvement in the performance, while for PID controllers, the use a fractional-order derivative action is convenient [31].

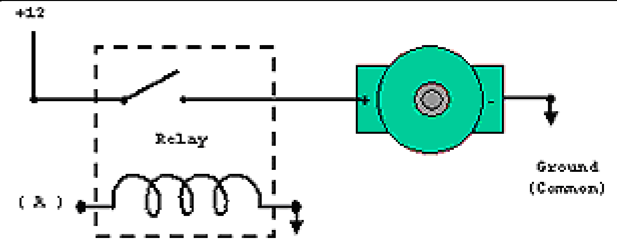
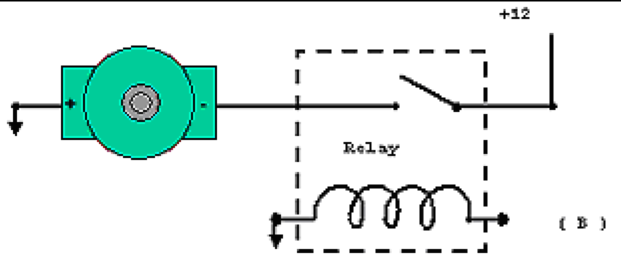
Recently, application of fractional calculus is becoming a hot topic in control area not only for the fractional order controller design of the traditional integer order systems but also for the controller design of the fractional order systems. [32] suggests two fractional order proportional integral controllers for a class of fractional order systems. For fair comparison, the proposed fractional order proportional integral (FOPI), fractional order [proportional integral] (FO[PI]) and the traditional integer order PID (IOPID) controllers are all designed following the same set of the imposed tuning constraints, which can guarantee the robustness of the designed controllers to the loop gain variations. From the results presented in simulation and experiment sections, both of the two designed fractional order controllers work efficiently [32].

Chapter 3 : DC MOTOR MOTION CONTROL

3.1 Direction Control of DC Motor

The simplest way to make a motor turn is to hook the positive side of battery to one side of a DC motor and to connect the negative side of the battery to the other motor lead then the motor spins forward. If the battery leads are swapped the motor spins in reverse.

To control the motor 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. If relay is used as a switch a diode should be connected across the coil of the relay which will keep the spike voltage (back EMF) coming out of the coil of the relay, from getting into the MCU and damaging it. Using the following circuits it is possible to only get the motor to stop or turn in one direction, forward for the first circuit or reverse for the second circuit

Forward	Backward
	
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).	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. It requires an H-Bridge. Relays configured in H"-looking configuration in the following graphic makes an H-Bridge.

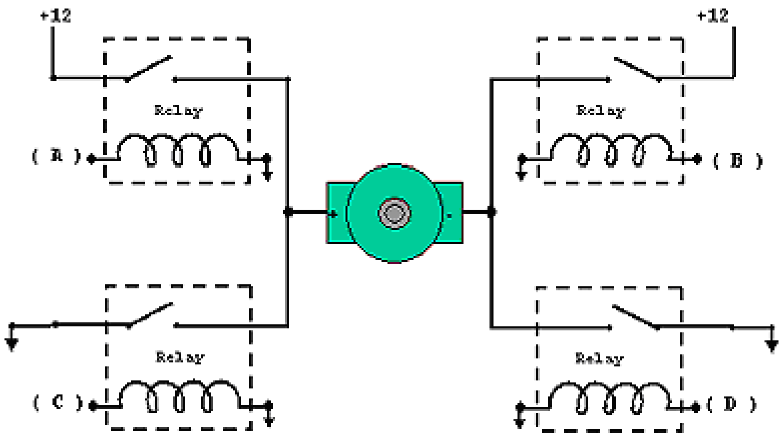
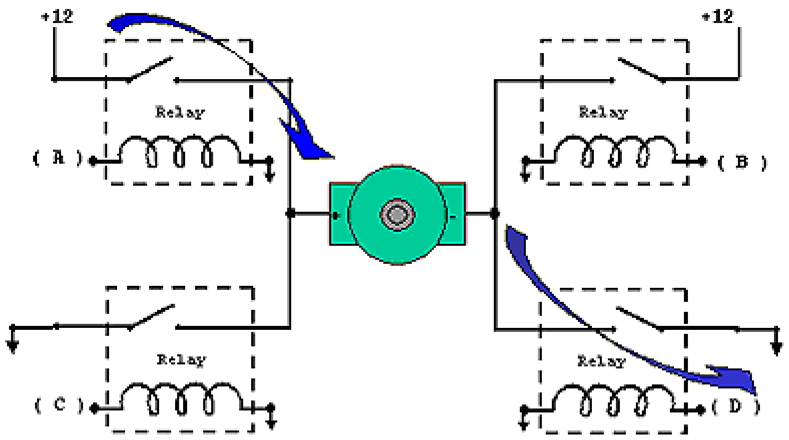


Figure 3-1: H-Bridge Circuit

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.

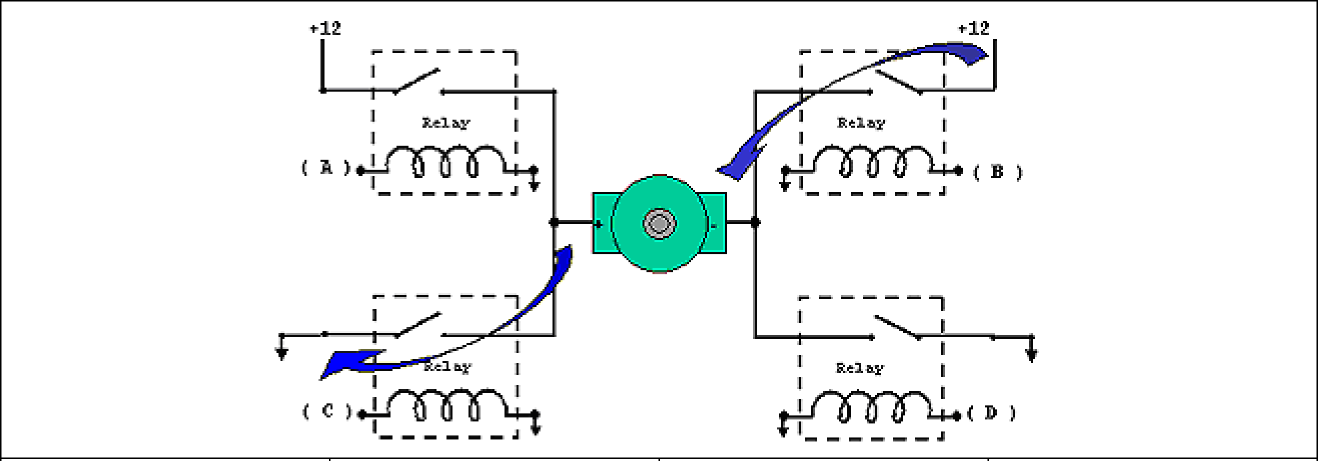
So, turning on the upper left and lower right circuits, flows power through the motor forward to turn motor forward.



A	B	C	D
1	0	0	1

Figure 3-2: Logic for H-Bridge Circuit

Then turning on the upper right and lower left circuits will flow power through the motor in reverse, and make the motor turn reverse.



A	B	C	D
0	1	1	0

Figure 3-3: Logic for H-Bridge Circuit for Reverse Motion

Here relays are shown with a single NO (Normally Open) connection which becomes closed when relays magnetic coil are energized with microcontroller signal. But in practice every relay has atleast 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.

Same control is possible to have using Field Effect Transistors (FETs) or MOSFETs replacing Relays. The L293, L298, LMD 18200 are some ICs which contain two H-Bridges on board and can handle 2 to 3 amps current only.

3.2 Speed Control of DC Motor

Often, 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 MOSFETs. A MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor) is a device that can turn very large currents on and off under the control of a low signal level voltage from MCU (Microcontroller Unit).

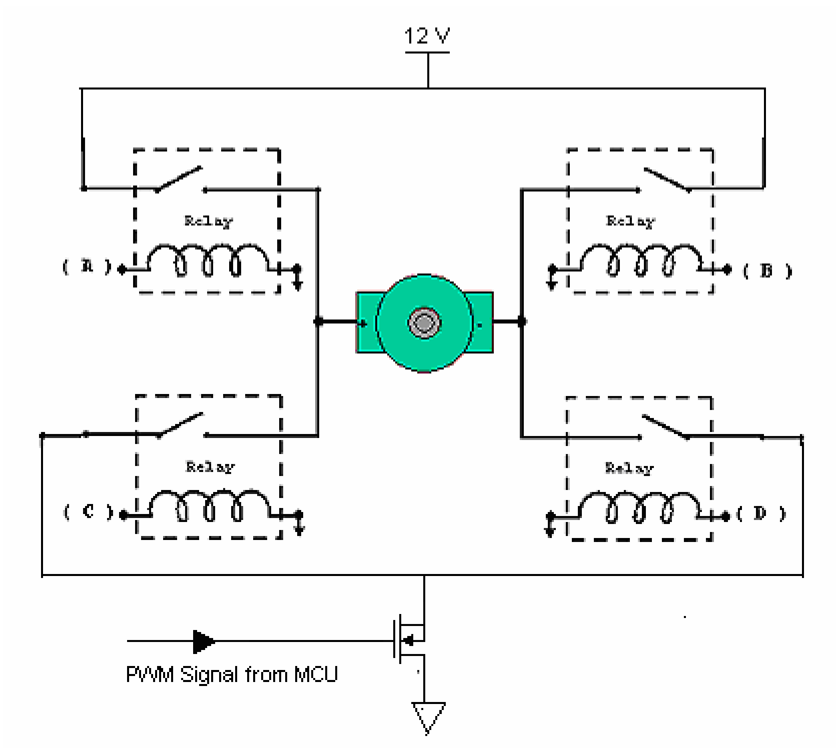


Figure 3-4: Speed Control of a Mobile Robot

In a nutshell, PWM is a way of digitally encoding analog signal levels. 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 is termed as duty cycle.

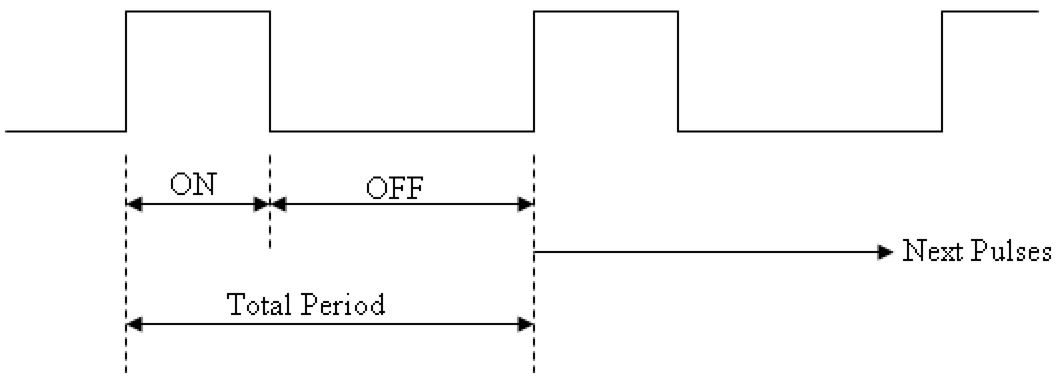


Figure 3-5: PWM Signal

$$\text{Duty Cycle} = \frac{ON\_Time}{Total\_Period} \text{ X } 100\% \qquad \textbf{Equation 3-1}$$

Following figure shows three different PWM signals. These three PWM outputs encode three different analog signal values, at 10%, 50%, and 90% of the full strength. If, for example, the supply is 12 V and the duty cycle is 10%, a 1.2 V analog signal results.

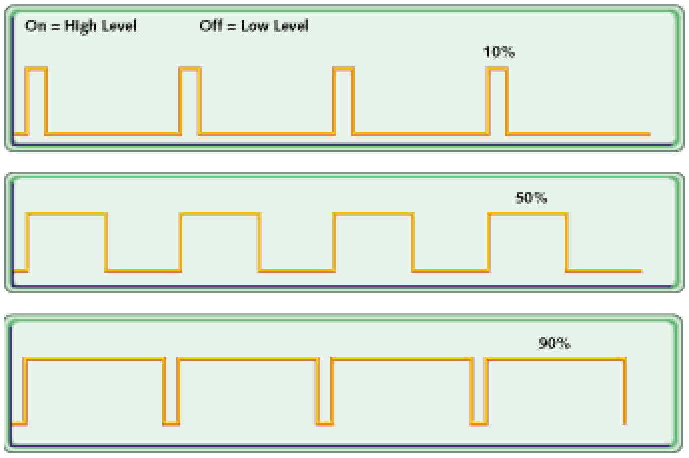


Figure 3-6: Three different Duty Cycle

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.

3.3 Monitoring Speed of MOTOR

To determine the no of rotation of wheel, an Optical Encoder is used. Optical encoder is a sensor, circular in size and with equally spaced holes on its periphery mounted on the shaft of wheel.

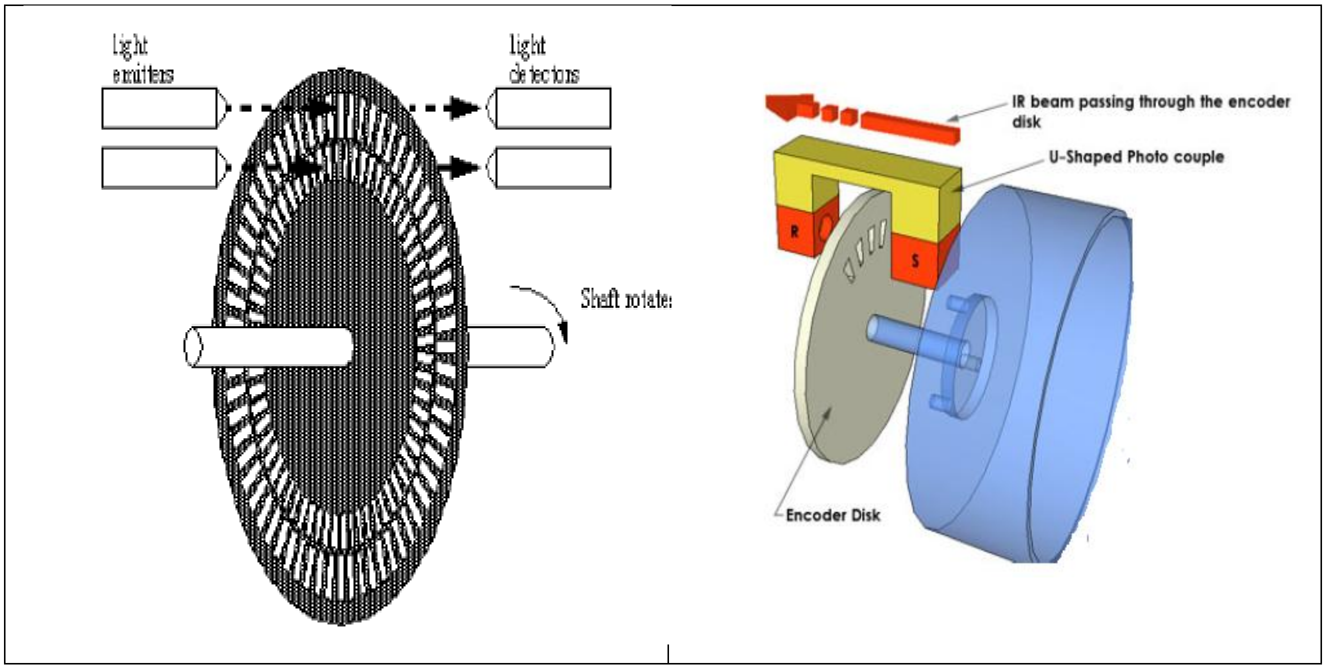


Figure 3-7: Working Principle of Optical Encoder

These holes permit IR (Infrared) Emitter to pass IR which is received by an IR receiver placed at the opposite side of the encoder. When wheel rotates, encoder blocks IR until the next hole on the periphery reached in front of IR Emitter. IR receiver generates a very small voltage across it when it does not receive any IR signal. When IR passes through the holes and received by the receiver it changes its state and voltage fallen down to zero. This change of state can be recognized by a signal conditioning circuit. Thus counting this change of state it is possible to calculate the no. of rotation of wheel which in turn gives the measure of traveled path.

3.4 Braking of DC Motor

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, we must apply a braking torque to ensure a rapid stop. One way to brake the motor is by simple mechanical friction, in the same way we stop a car. A more elegant method consists of supplying a reverse current in the armature, so as to brake the motor electrically. Two methods are employed to create such an electromechanical brake: (1) dynamic braking and (2) plugging.

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.

Chapter 4 : CONTROLLER MODES - DYNAMIC RESPONSE SIMULATION - RESULTS AND DISCUSSION

4.1 Control System

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 [33]. 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 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 beingmade until the measured difference is eliminated and the controlled variable equals the reference variable. Hence, the characteristic feature of closed loop control is a closed action flow. Various terminologies used in close loop control system is shown in Figure 4-1.

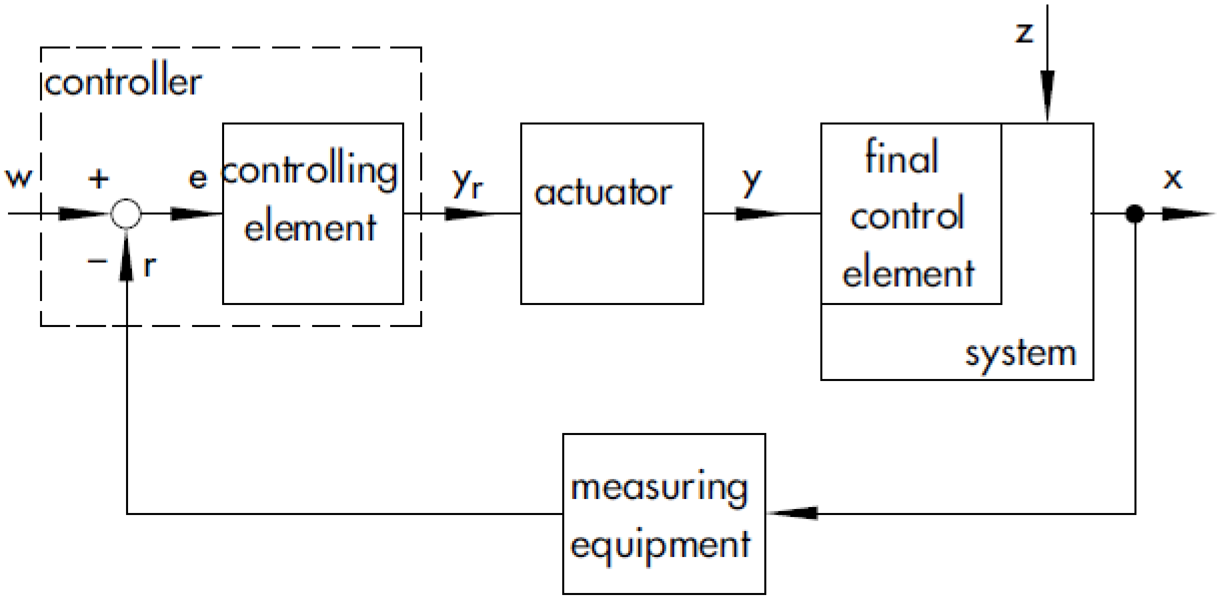


Figure 4-1: Various Terminologies in Close Loop Control System

Variables indicated in Figure. 4-1 are defined as follows [30]:

Table 4-1: Definition of the Variables of a Close Loop Control System

w	Reference Variable
r	Feedback variable
e	Error
y_r	controller output variable
y	Manipulated variable
z	Disturbance variable
x	Actual Value

4.2 Controller Modes

A controller generates a control signal to the final element, based on a measured deviation of the controlled variable from the setpoint. 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 controllers   ii. Continuous controllers

4.3 ON/OFF Controllers

The most elementary control mode is ON/OFF or two-position control mode. This is an example of a discontinuous control mode. The actuator can push the controlled variable with only full force or no force.Equation for the controller output can be generally written as:

$$y = 0\% \quad \text{when } e_p < 0 \text{ , or, } \qquad y = 100\% \quad \text{when } e_p > 0. \qquad \text{..........................4-1}$$

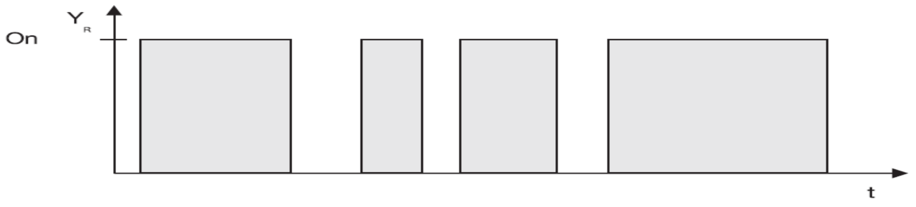


Figure 4-2: Discontinuous (OFF/ON) Controller Mode

A perfect on–off controller is 'on' when the measurement is below the setpoint (SP) and the manipulated variable (MV) is at its maximum value. Above the SP, the controller is 'off' and the MV is a minimum. A common example of a discrete controller is a home hot water heater. When the temperature of the water in the tank falls below setpoint, the burner turns on. When the water in the tank reaches setpoint, the burner turns off. Because the water starts cooling again when the burner turns off, it is only a matter of time before the cycle begins again. For this reason there is usually a dead zone due to mechanical delays in the process. This is often deliberately introduced to reduce the frequency of operation and wear on the components. The end result of this mode of control is that the temperature will oscillate about the required value [34].

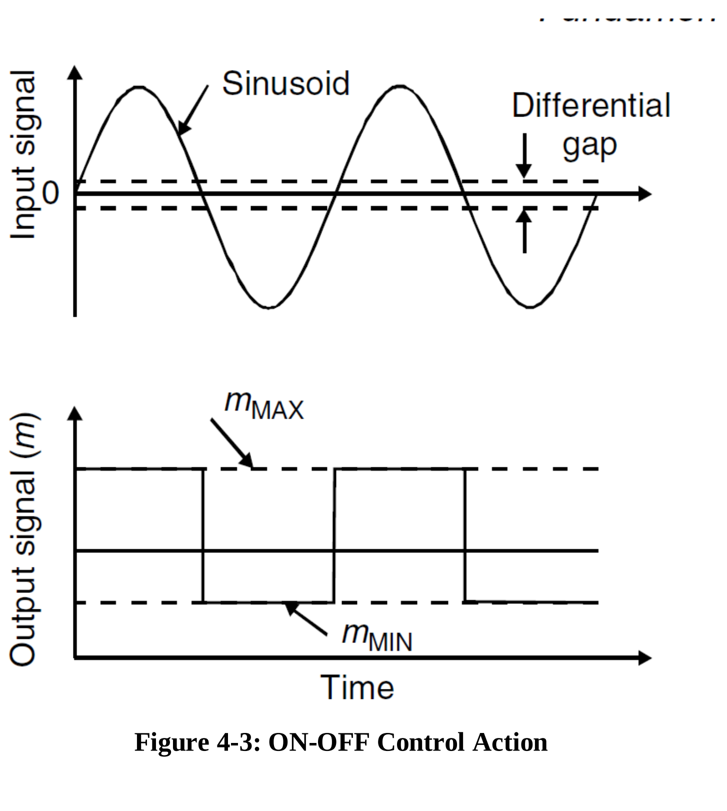


Figure 4-3: ON-OFF Control Action

4.4 P (Proportional) Controller

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

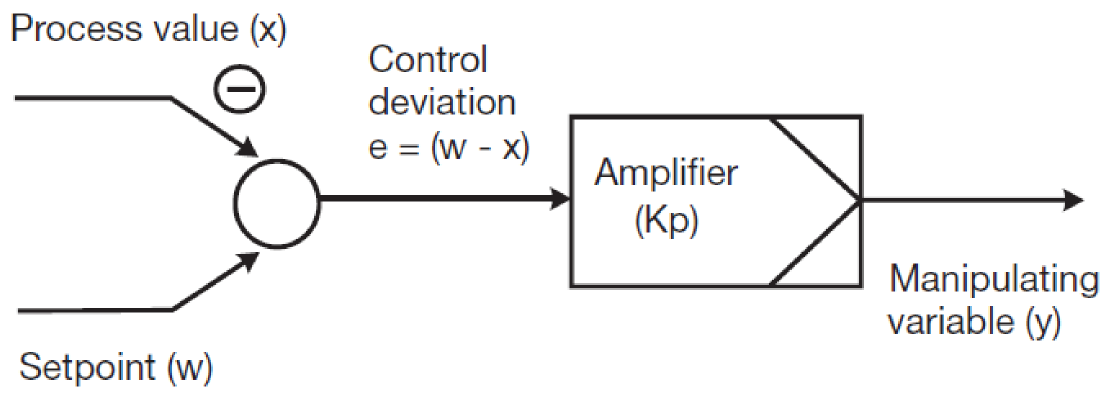


Figure 4-4: Proportional Controller

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,

$$\boldsymbol{y = K_p e_p + y_0} \quad \text{4-2}$$

Where y = Manipulated Variable, MV.

k_p= Proportional gain between error and controller output

e_p = Error

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.

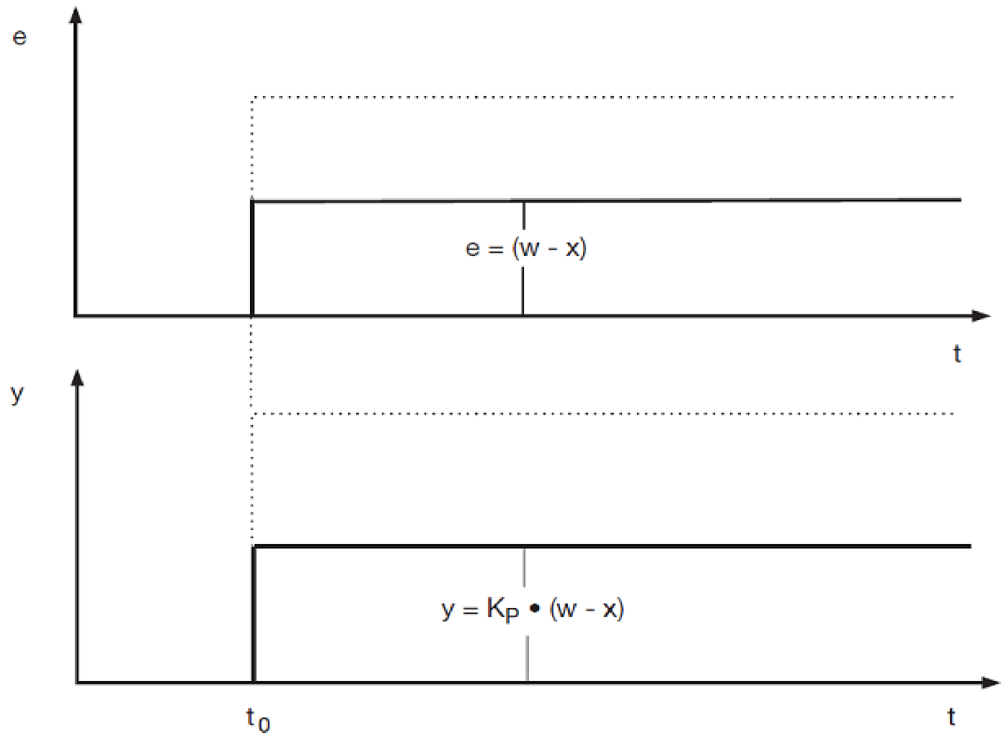


Figure 4-5: Proportional Controller Response

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 setpoint. So a P controller always has a permanent control deviation or offset [35].

Simulation is done for a P controller for controlling a second order system for step and ramp inputs by SimApp [39].

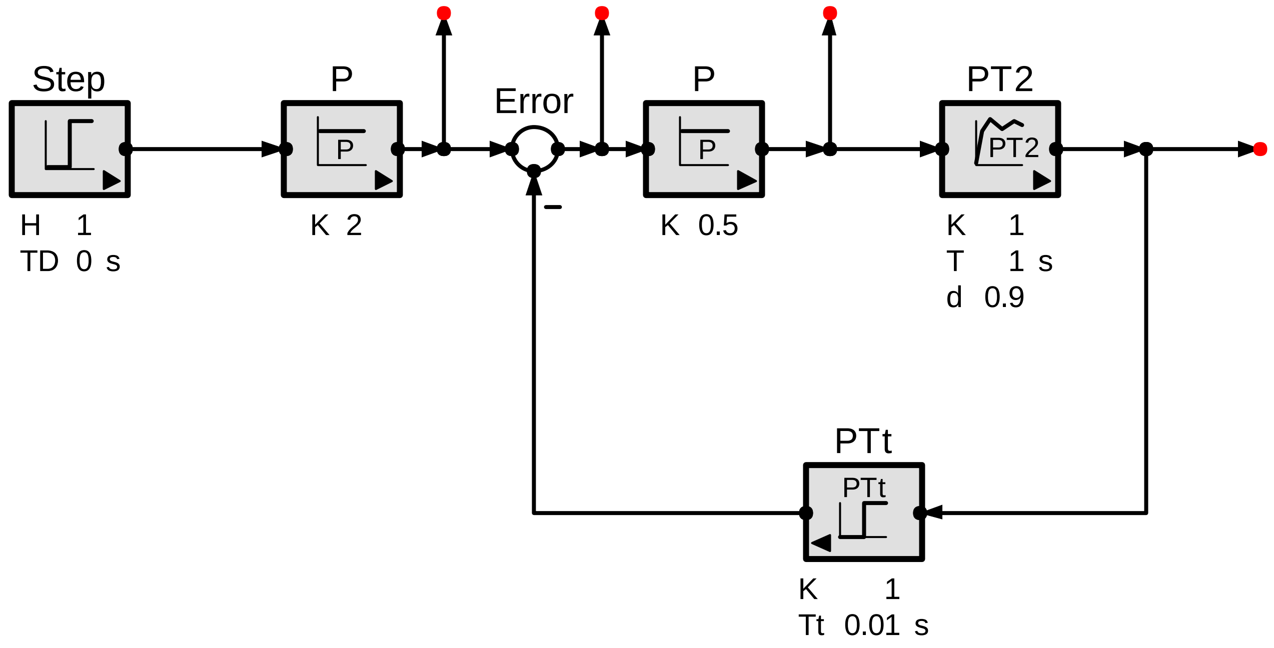


Figure 4-6: Second Order System with Step input and P controller

Simulation is done for the above system by SimApp with gain, K=1, time constant, T= 1s, and damping, d= 0.9 and sampling time T_s= 0.1s for PT2 system. The step input was taken with size, H=1, and no delay, TD= 0s. P controller with gain, K=2 is used as multiplier of the step input. P controller is placed in the system and simulated for K_p= 2, 4, 6 and 8. From Figure 4-7,with Kp=2, controller fails to reach the setpoint and there remains a permanent deviation. With the increase of Kp system responses quickly and reaches its setpoint more rapidly but at higher value of Kpprocess variable overshoots and oscillates. The oscillation dampens quickly for lower Kp. Figure 4-9 shows the response of P controller with ramp input. Similar results are found as with higher value of Kp system becomes oscillatory.

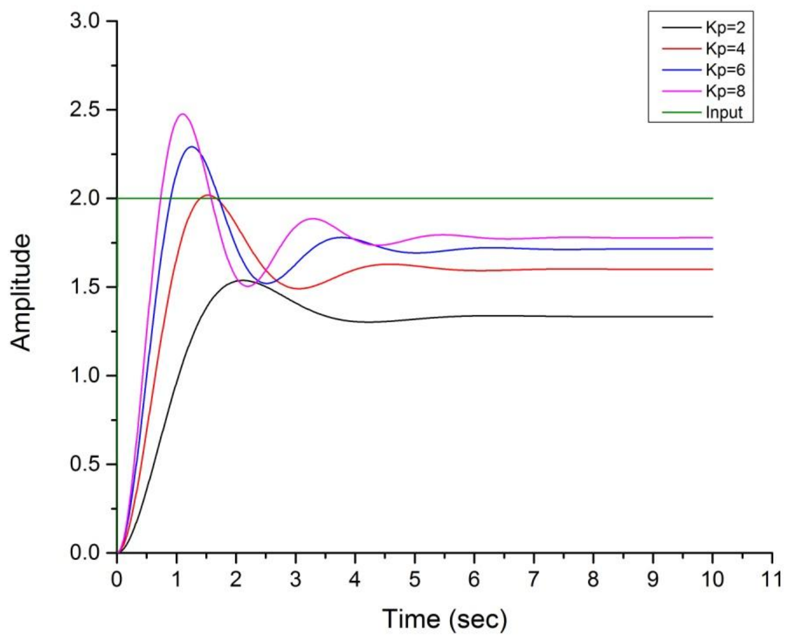


Figure 4-7: Response curve for second order system with Step input and P controller

Similar study is done for ramp input and P controller:

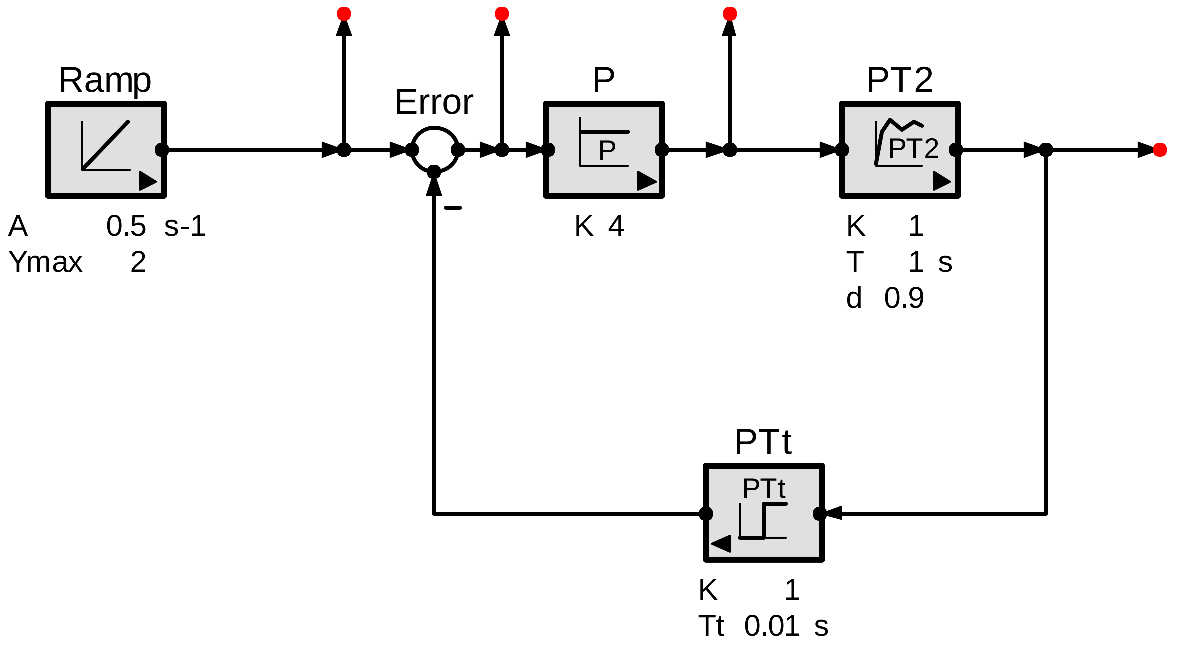


Figure 4-8: Second Order System with Ramp input and P controller

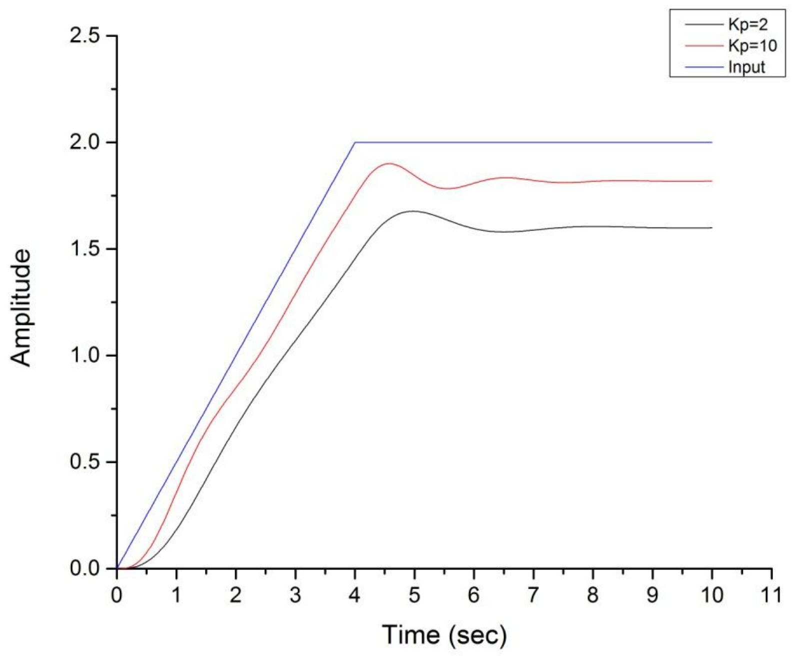


Figure 4-9: Response curve for second order system with Ramp input and P controller

4.5 I (Integral) Controller

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 theI controller becomes. How quickly the controller builds up its manipulating variable depends firstly on the setting of the 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 setpoint. In principle, an I controller can fully stabilize after a sufficiently long period of time, i.e. setpoint = process variable. The deviation is then zero and there is no further increase in manipulating variable. Unlike the P controller, the I controller does not have a permanent control deviation. Therefore this mode is represented by the following integral equation:

$$\boldsymbol{y(t) = K_I \times \int_0^t e\, dt + y(0)} \quad \text{...4-3}$$

Where,

$y(t)$ = Manipulated Variable, MV.

k_I= Integral Gain

e=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.

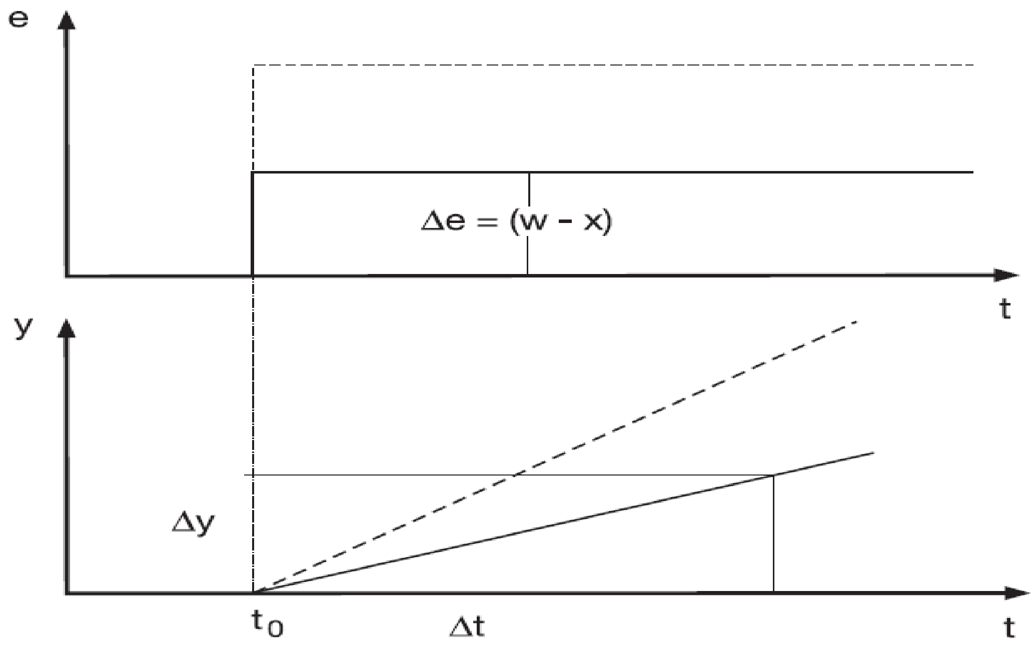


Figure 4-10: Step Response of an I Controller

4.5.1 PI (Proportional-Integral) Controller

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 setpoint.

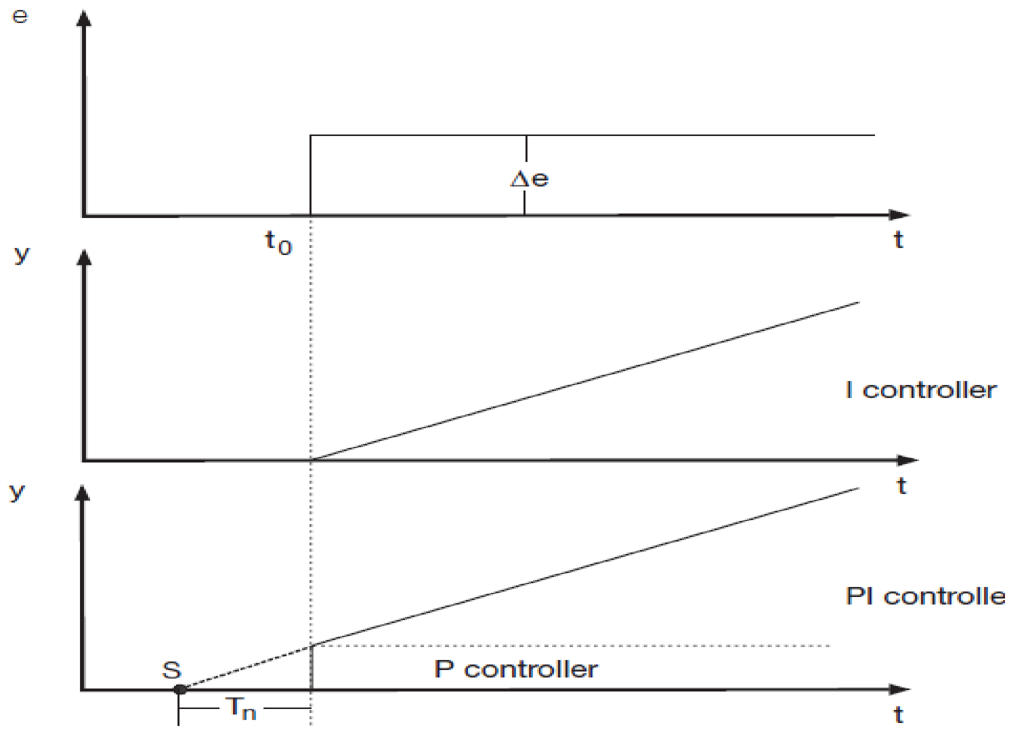


Figure 4-11: PI Controller Action

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

$$\boldsymbol{y(t) = K_p e_p + K_I \times \int_0^t e\,dt + y(0)} \quad \text{..} \mathbf{4\text{-}4}$$

Where,

y(t) = Manipulated Variable, MV.

K_p= Proportional Gain

K_I = Integral Gain

e = Control Deviation

y(0) = Action of controller when the control action starts.

The integrated error is thus inversely proportional to the integral gain ki . The integral gain is thus a measure of the effectiveness of disturbance attenuation. A large gain ki attenuates disturbances effectively, but too large a gain gives oscillatory behavior, poor robustness and possibly instability [36].

Simulation is done for a PI controller controlling a second order system for step and ramp inputs by SimApp.

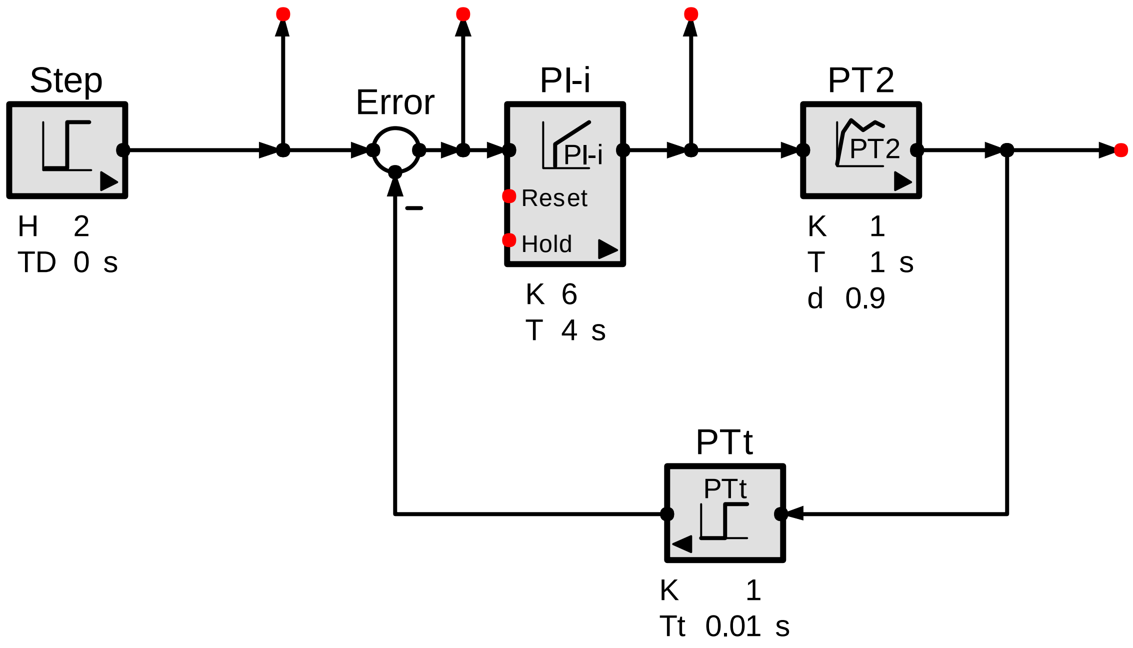


Figure 4-12: Second Order System with Step input and PI controller

Simulation is done for the above system by SimApp with gain, K=1, time constant, T= 1s, and damping, d= 0.9 and sampling time T_s= 0.1s for PT2 system. The step input was taken with size, H=1, and no delay, TD= 0s. PI controller simulated withK_p= 6 ,integral time, Ti= 4, 12 and 20. Figure 4-13 shows higher overshoots for lower value of Ti/T i.e greater action from I component of PI controller. Though PI controller shows system approaches the setpoit steady thus reducing the steady state error. Figure 4-15 depicts the response of PI controller for ramp input. Simulation is done to observe the oscillation problem inherent to I controller. With an I controller, oscillations aboutthe setpoint can occur quite frequently. This is especially the case if the I component is too strong,i.e. when the selected integral time Ti is too short. Similar result is obtained from the simulation. As the Ti is decreased system becomes more stable but with higher Ti is oscillates around the set point.

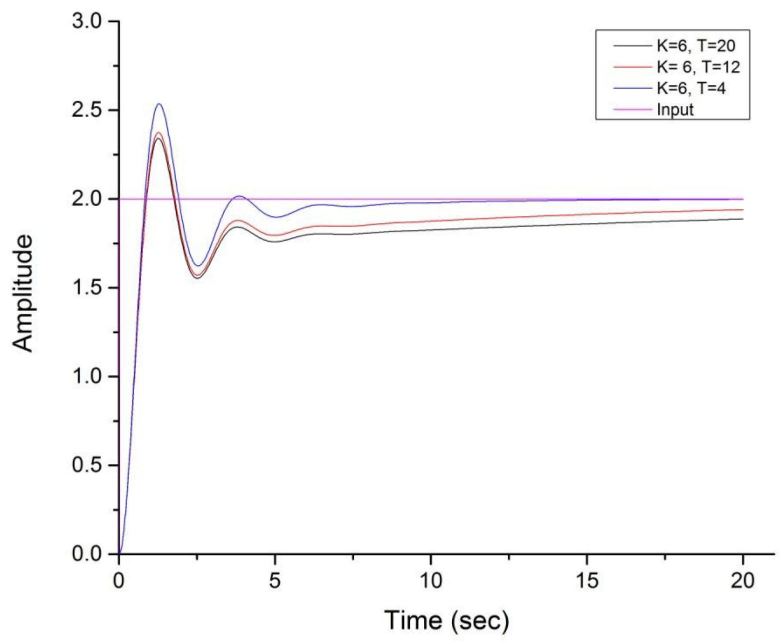


Figure 4-13: Response curve for second order system with Step input and PI controller

Similar simulation is done for ramp input and P controller:

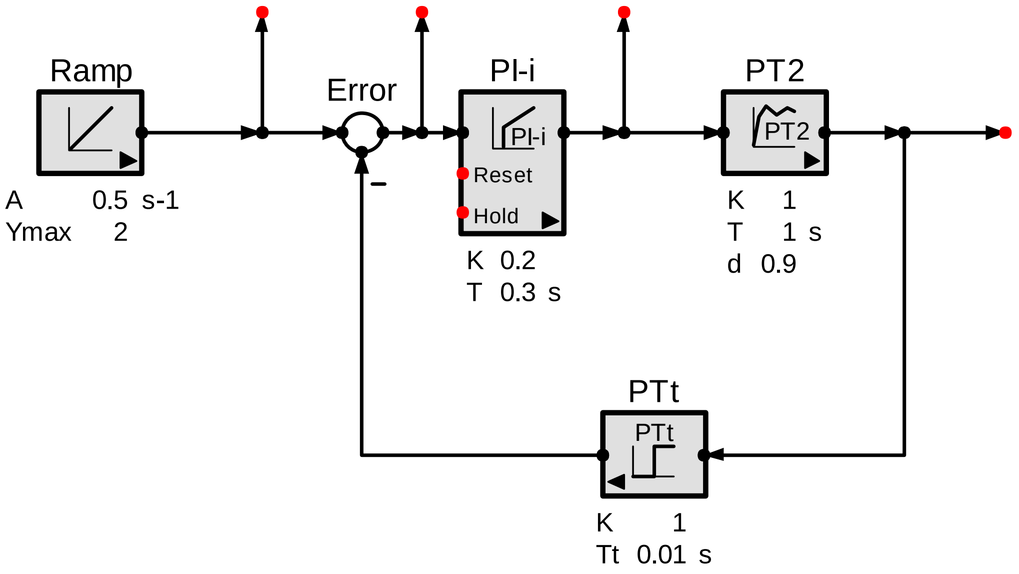


Figure 4-14: Second Order System with Ramp input and PI controller

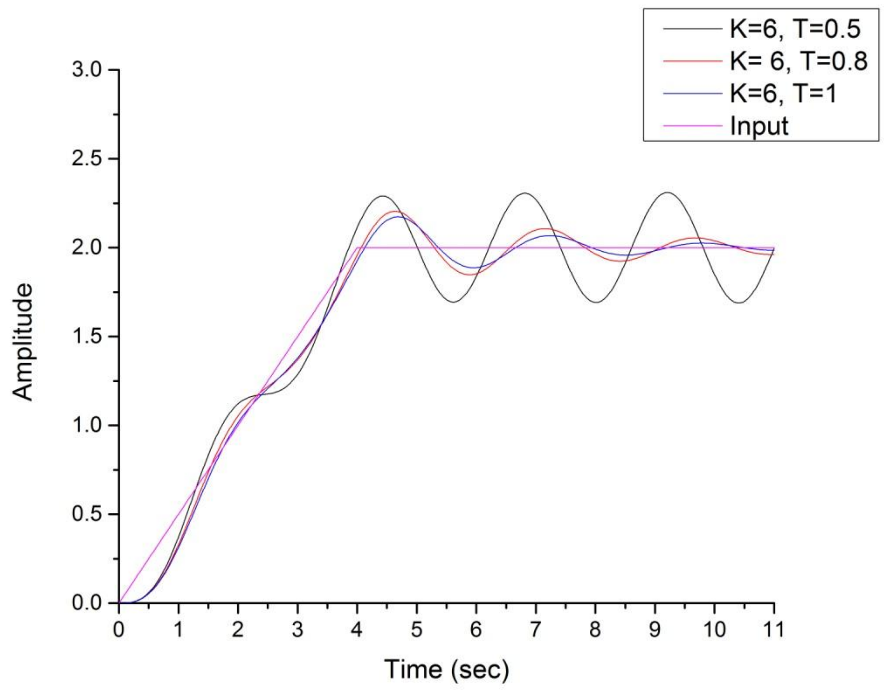


Figure 4-15: Response curve for second order system with Ramp input and PI controller

4.6 D (Derivative) Controller

When the error is zero, it is seemed that there should be no control action. But if the process shows the variation of error in time and looks like the following figure:

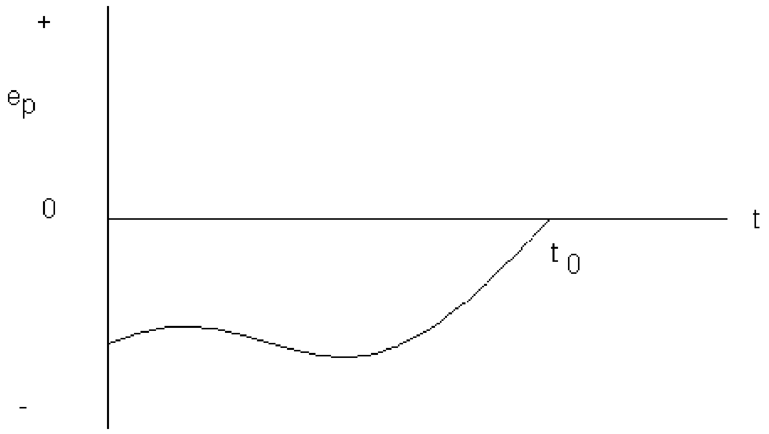


Figure 4-16: Necessity of Derivative Control Action

It is clear that there is no error at time $t_{0,}$ but it is changing in time and will certainly not be zero in the following time. Therefore some actions should be taken even though the error is zero. This scenario describes the nature and need of derivative control action.

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:

$$\boldsymbol{y(t) = K_p K_D \times \frac{de}{dt}} \quad \textbf{………………………………..……………..4-5}$$

Where,

y(t) = Manipulated Variable, MV.

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

$\frac{de}{dt}$ = Percent per second rate of change of error

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.

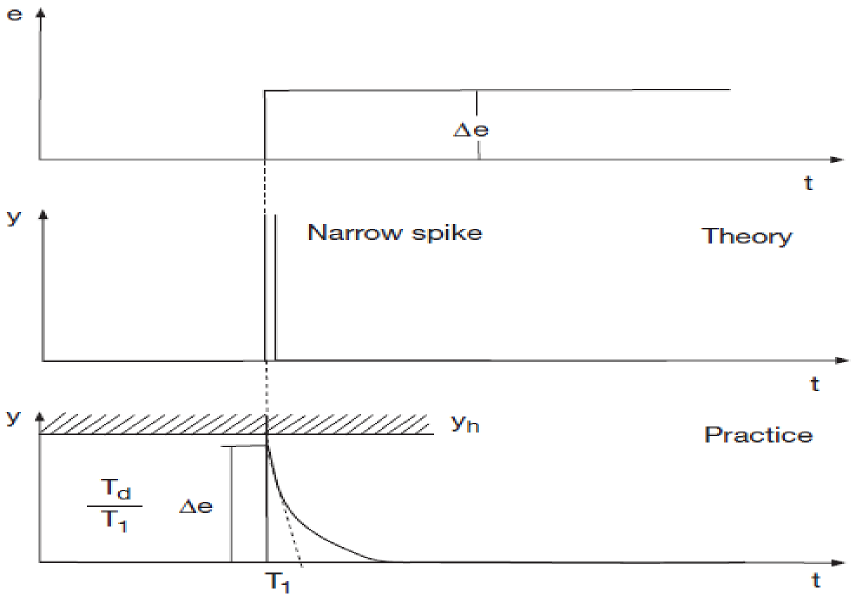


Figure 4-17: Derivative Control Action

Derivative controllers should be used with care, because even small amount of high-frequency noise can cause very large derivatives, which appear like amplified noise. Also, Derivative controllers are difficult to implement perfectly in hardware or software, so frequently solutions involving only integral controllers or proportional controllers are preferred over using derivative controllers. The system is oscillatory when no derivative action is used, and it becomes more damped as the derivative gain is increased. Performance deteriorates if the derivative gain is too high. When the input is a step, the controller output generated by the derivative term will be an impulse [36].

4.7 PID Controller

The control action is thus a sum of three terms: the past as represented by the integral of the error, the present as represented by the proportional term and the future as represented by a linear extrapolation of the error (the derivative term). This form of feedback is called a proportional-integral-derivative (PID) controller [36].

Block diagrams of closed loop systems with PID controllers are shown in Figure 4-18. The control signal u for the system in Figure is formed entirely from the error e. The command signal r is called the reference signal in regulation problems, or the setpoint in the literature of PID control. The input/output relation for an ideal PID controller with error feedback is:

$$\boldsymbol{y(t) = K_p e_p + K_I \times \int_0^t e\,dt + K_p K_D \times \frac{de}{dt} + y(0)} \qquad \text{.......................4-6}$$

The control action is thus the sum of three terms: proportional feedback, the integral term and derivative action. For this reason PID controllers were originally called three-term controllers. The controller parameters are the proportional gain k_p, the integral gain ki and the derivative gain k_d . The time constants Ti and Td, called integral time (constant) and derivative time (constant), are sometimes used instead of the integral and derivative gains.

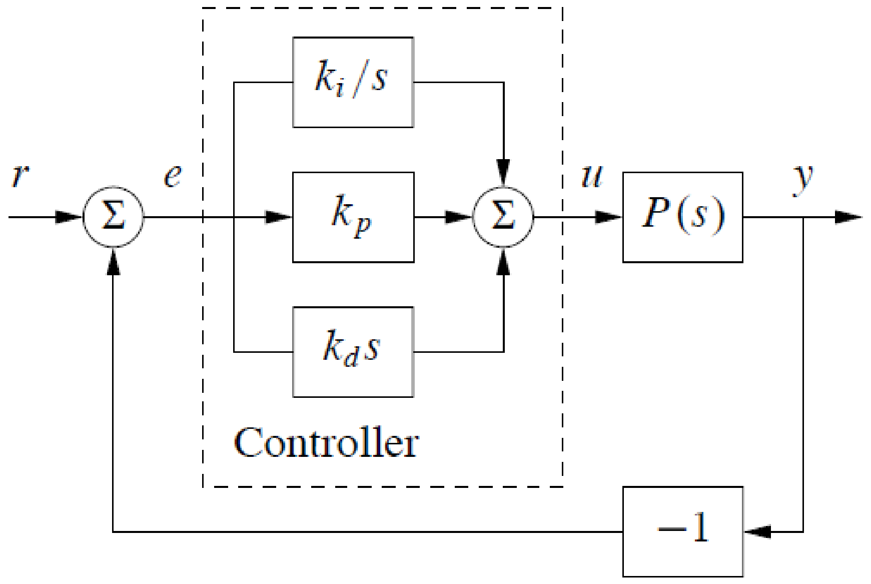


Figure 4-18: Block Diagram of PID Controller

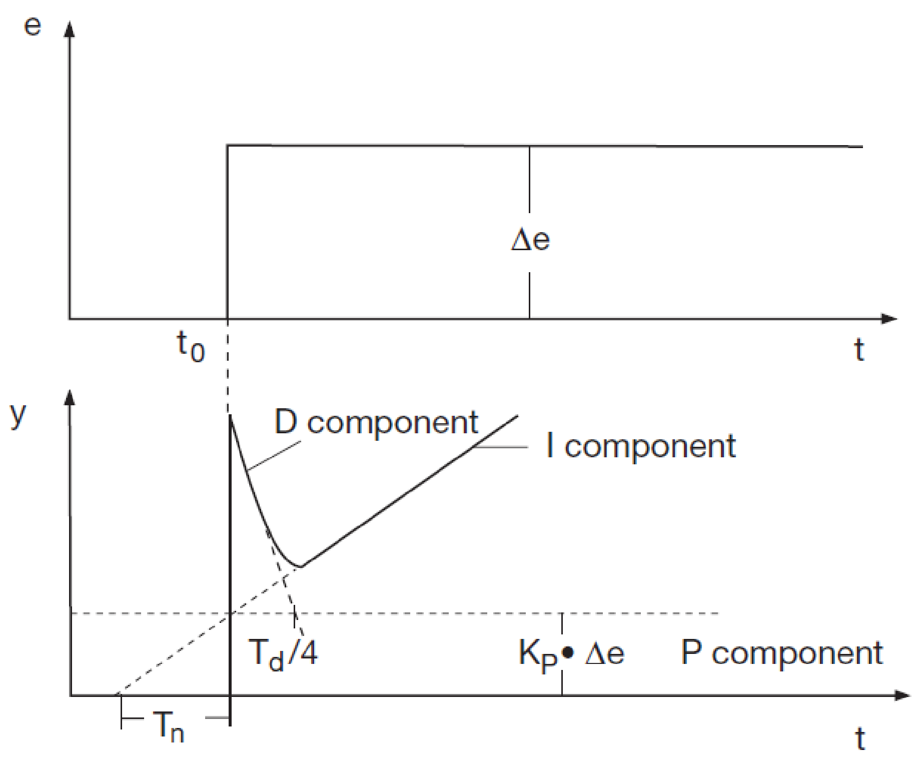


Figure 4-19: PID Control Action

Simulation is done for a PID controller controlling a second order system for step and ramp inputs by SimApp.

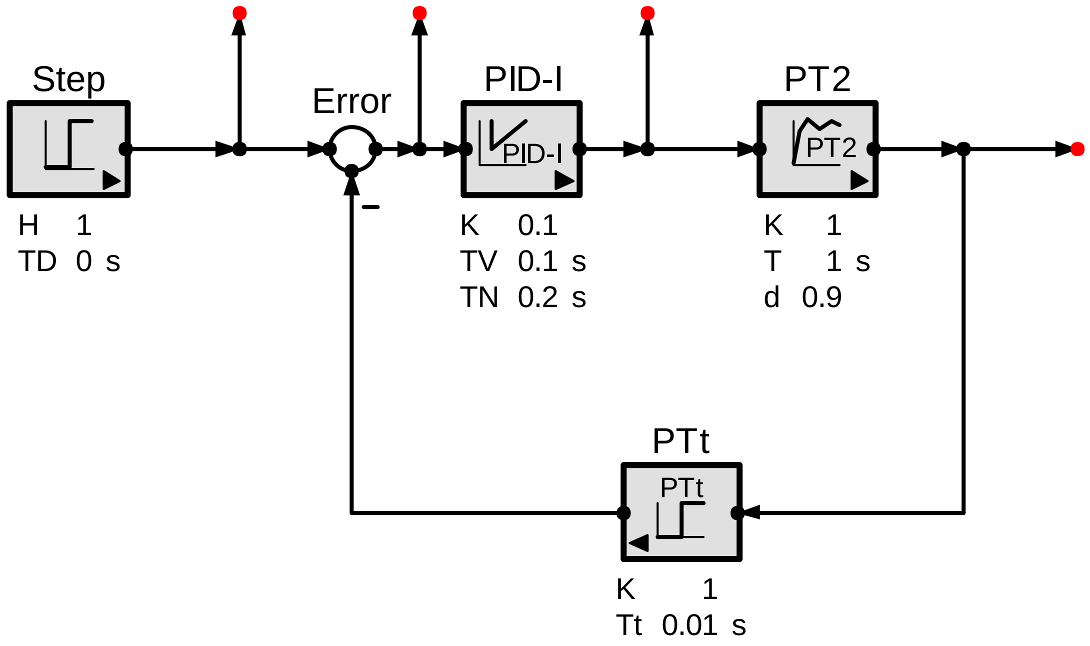


Figure 4-20: Second Order System with Step input and PID controller

Simulation is done for the above system by SimApp with gain, K=1, time constant, T= 1s, and damping, d= 0.9 and sampling time T_s= 0.1s for PT2 system. The step input was taken with size, H=2, and no delay, TD= 0s. PID controller simulated with optimum value of Kp and Ti which is found from previous simulation of P and PI controller and varying Td to properly realize the action of D component. It is found that higher value of Td reduces overshoots as D component applies brakes when the process variable approaches the set point thus preventing the manipulating variable overshooting above the setpoint. If the process variable has reached its maximum value after an overshoot above the setpoint, and is now reducing, the D component gives out a positive manipulating variable. In this case, the D component counteracts the change in process variable.

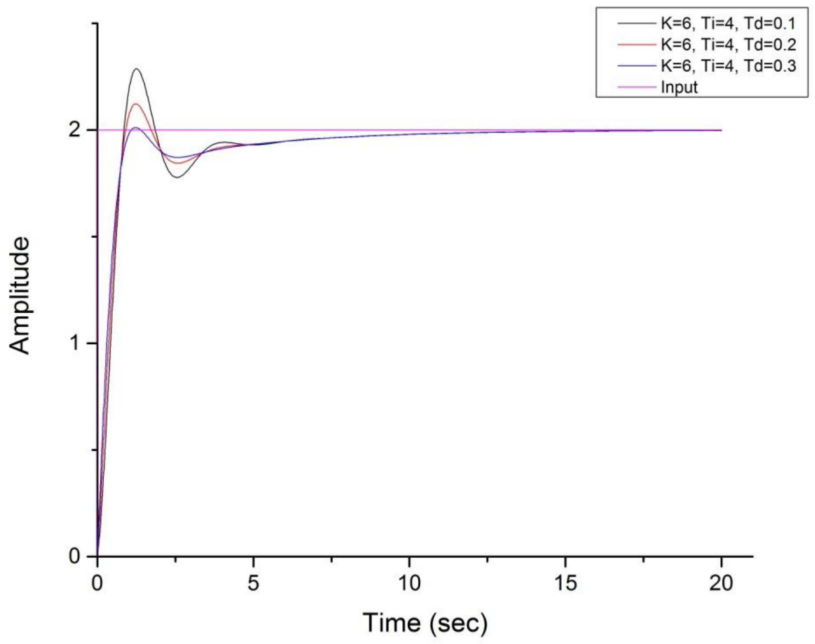


Figure 4-21: Response curve for second order system with Step input and PID controller

Similar simulation is done for ramp input and PID controller. In Figure 4-23, the interesting observation is that here with lower value of Td overshoots occurs as for this case Ti is increased i.e. the effect of I component is reduced.

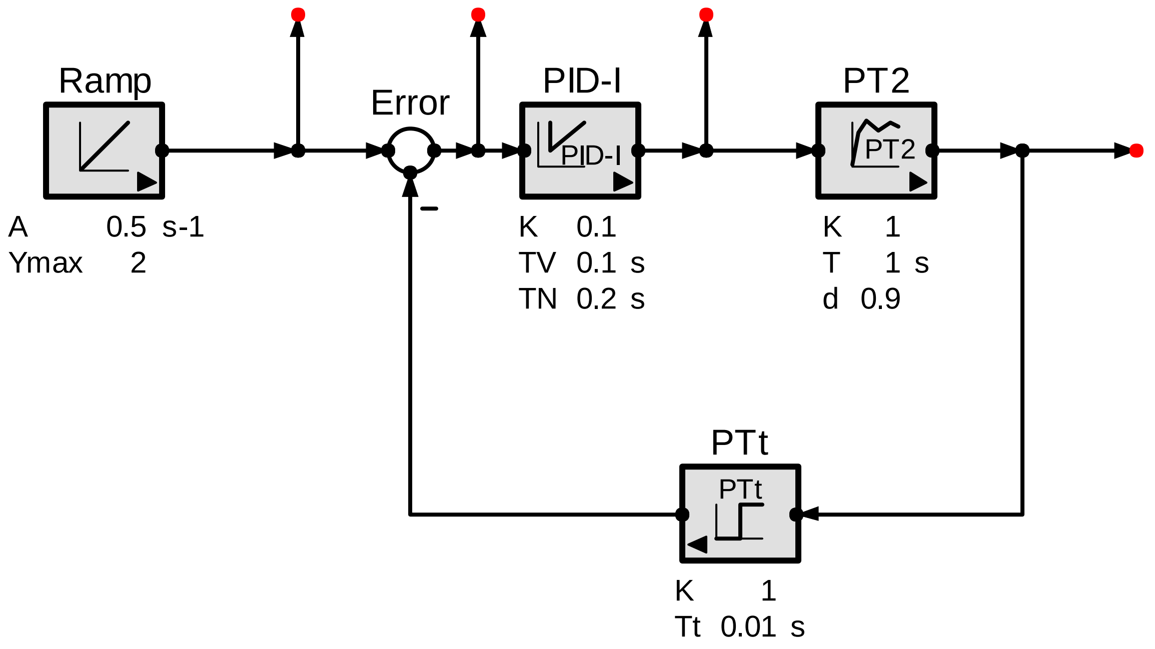


Figure 4-22: Second Order System with Ramp input and PID controller

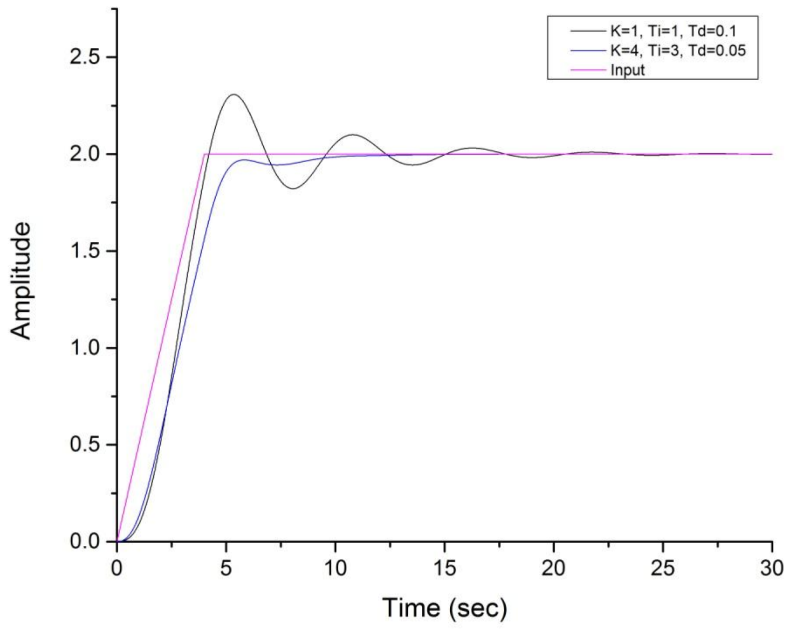


Figure 4-23: Response curve for second order system with Ramp input and PID controller

4.8 Controller Tuning

It is interesting to note that more than half of the industrial controllers in used today are PID controllers or modified PID controllers. Because most PID controllers are adjusted on-site, many different types of tuning rules have been proposed in the literature. Using these tuning rules, delicate and fine tuning of PID controllers can be made on-site. Also, automatic tuning methods have been developed and some of the PID controllers may possess on-line automatic tuning capabilities. Modified forms of PID control, such as I-PD control and multi-degrees-offreedom PID control, are currently in use in industry. The usefulness of PID controls lies in their general applicability to most control systems. In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used, PID controls prove to be most useful. In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control [37]. In this chapter we first present the design of a PID controlled system using Ziegler and Nichols tuning rules.

Figure 4-24 shows a PID control of a plant. If a mathematical model of the plant can be derived, then it is possible to apply various design techniques for determining parameters of the controller that will meet the transient and steady-state specifications of the closed-loop system. However, if the plant is so complicated that its mathematical model cannot be easily obtained, then an analytical or computational approach to the design of a PID controller is not possible. Then we must resort to experimental approaches to the tuning of PID controllers. The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. Ziegler and Nichols suggested rules for tuning PID controllers (meaning to set values Kp , Ti , Td) based on experimental step responses or based on the value of that results in marginal stability when only proportional control action is used. Ziegler–Nichols rules, which are briefly presented in the following, are useful when mathematical models of plants are not known. (These rules can, of course, be applied to the design of systems with known mathematical models.) Such rules suggest a set of values of Kp , Ti,and Td that will give a stable operation of the system. However, the resulting system may exhibit a large maximum overshoot in the step response, which is unacceptable. In such a case we need series of fine tunings until an acceptable result is obtained. In fact, the Ziegler–Nichols tuning rules give an educated guess for the parameter values and provide a starting point for fine tuning, rather than giving the final settings for Kp , Ti, and Td in a single shot[37].

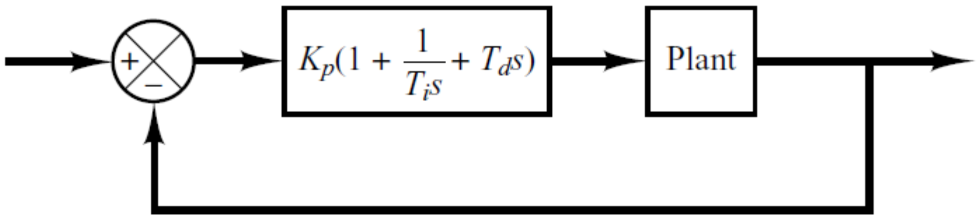


Figure 4-24: PID Controller

Ziegler–Nichols Rules for Tuning PID Controllers

Ziegler and Nichols proposed rules for determining values of the proportional gain$k_{p,}$ integral time $T_{i,}$and derivative time T_d based on the transient response characteristics of a given plant. Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on-site by experiments on the plant. There are two methods called Ziegler–Nichols tuning rules: the first method and the second method.

First Method: Open-Loop Transient Response Method

This 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 their 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.

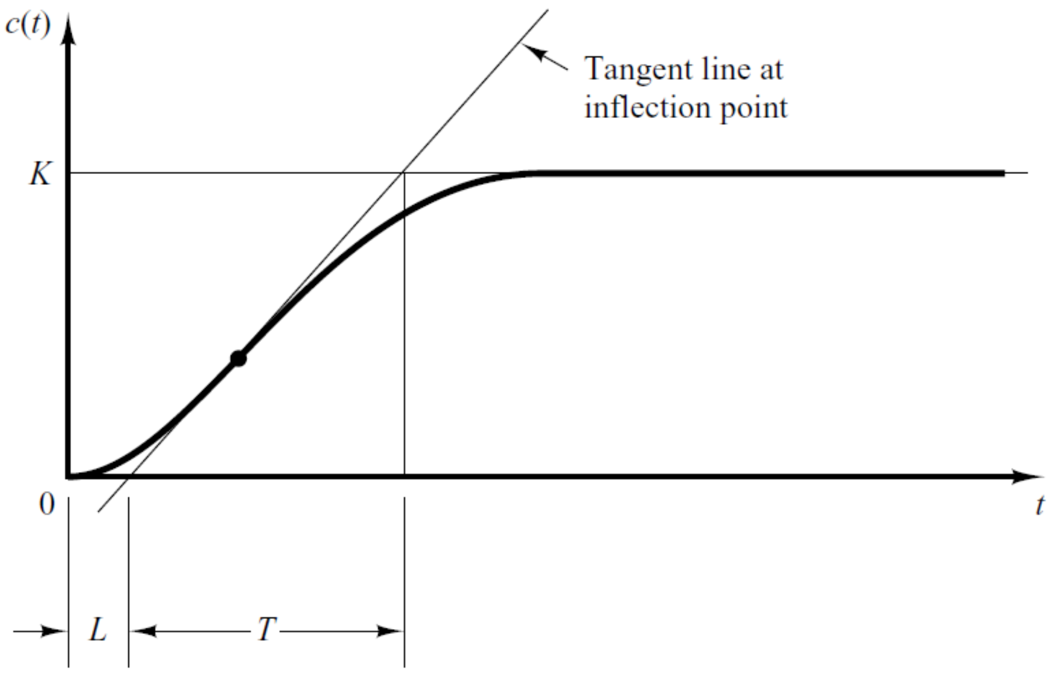


Figure 4-25: Open Loop Transient Method

A typical open-loop controller response is shown in Figure 4-25. The S-shaped curve may be characterized by two constants, delay time L and time constant T. The delay time and time constant are determined by drawing a tangent line at the inflection point of the S-shaped curve and determining the intersections of the tangent line with the time axis and line c(t)=K, as shown in Figure 4-25.

Here,
L = Lag in minutes
T = process reaction time in minutes

Ziegler and Nichols suggested to set the values ofkp , Ti, and Td and according to the formula shown in Table [37]

Table 4-2: Zeigler-Nichols tuning rule based on Step Response of the Plant (First Method)

Type of Controller	kp	Ti	Td
P	T/L	∞	0
PI	0.9 T/L	L/0.3	0
PID	1.2 T/L	2L	0.5L

Second Method: Sustained Oscillation Method

Second empirical method for controller tuning is also developed by Zeigler and Nichols [37]. 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 below:

1. Any kind of Integral and Derivative action is to be reduced to their minimum effect.
2. Proportional gain is to be increased until the steady oscillation of process variable occurs.
3. This gain at which the steady oscillation occurs is known as Critical Gain, Kcr.
4. The period at which the process variable oscillates is called Critical Period, Pcr.
5. The Critical Period, Pc of this oscillation is measured in minutes.
6. This values of Critical Gain, Kcr and Critical Period, Pcr leads the determination of Controller settings using following adjustment formulas for different control modes:

Table 4-3: Ziegler–Nichols Tuning Rule Based on Critical Gain kcr and Critical Period Pcr (Second Method)

Type of Controller	kp	Ti	Td
P	0.5kcr	∞	0
PI	0.45kcr	1/1.2 Pcr	0
PID	0.6kcr	0.5 Pcr	0.125 Pcr

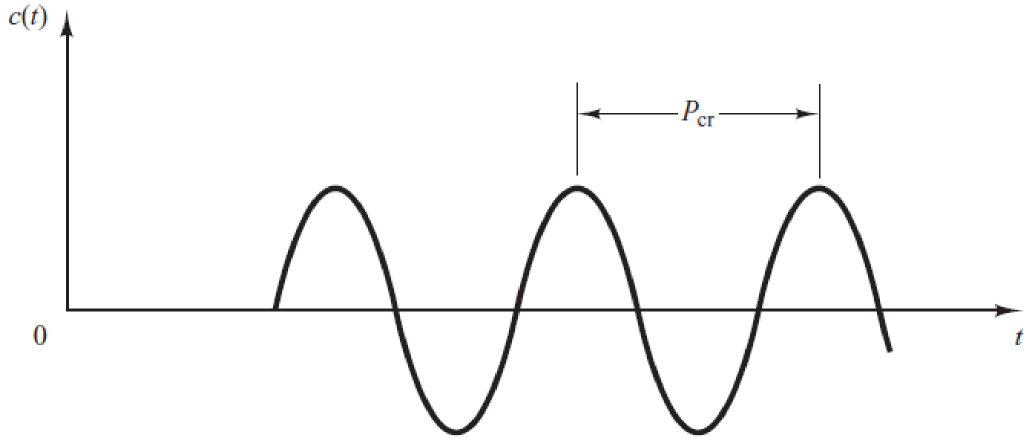


Figure 4-26: Sustained oscillation with period Pcr(Pcr is measured in sec.)

4.9 Simulation of a DC motor model

A model of DC motor is made which closely resembles the motor that is used in our experimental setup with armature inductance La= 0.000165 H, armature resistance, Ra= 0.576 ohm, Back-emf constant, Kf= 0.075 Nm/A, and Inertia of motor and load, J= 0.01 kg-m^2. This model is fed with 12 V voltage and simulation shows similar speed profile as our drive motor of experimrnt.

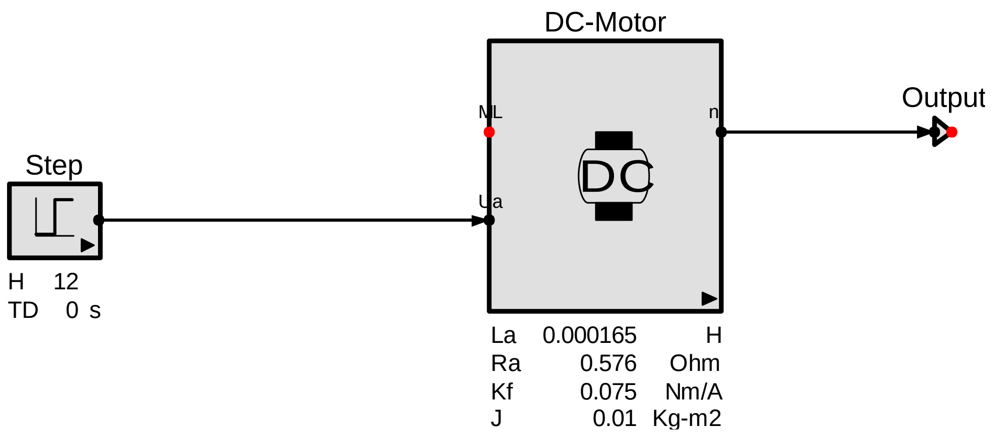


Figure 4-27: DC Motor Model

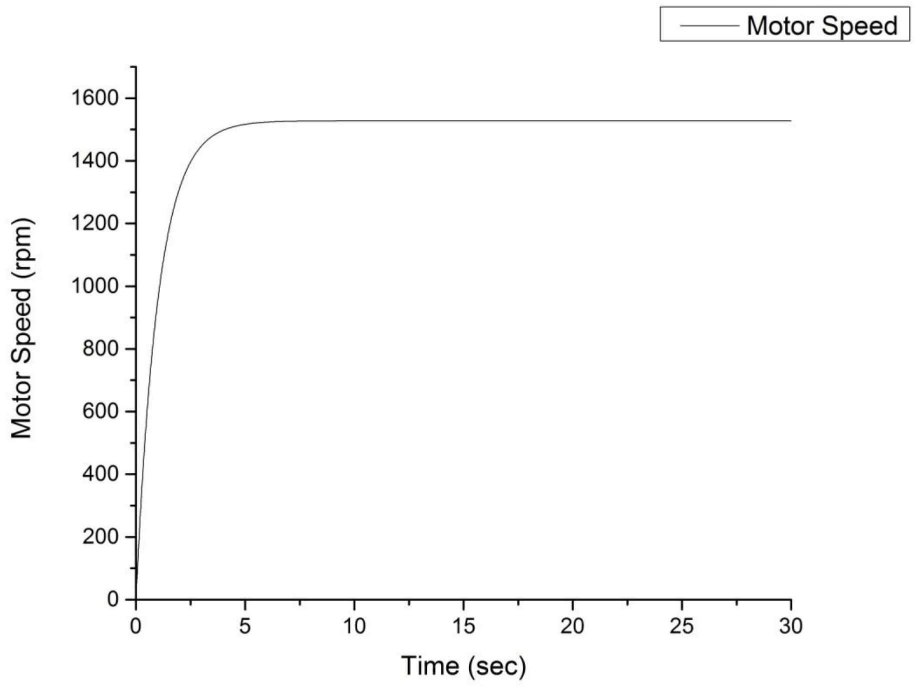


Figure 4-28: Speed profile of model motor at 100% duty cycle

The simulation is done to replicate our actual motor that is used in the experiment. To do so a reference speed profile is generated which is a step function. Reference speed rises instantly from 0 to 1200 rpm. Another reference speed profile is generated as a ramp reaching 1200 rpm in 4 sec. Feedback from the motor is used to calculate the error then the error is converted to voltage using the actual motors speed profile [Figure 4-28] Then the response of this motor is simulated for different controllers with step and ramp input.

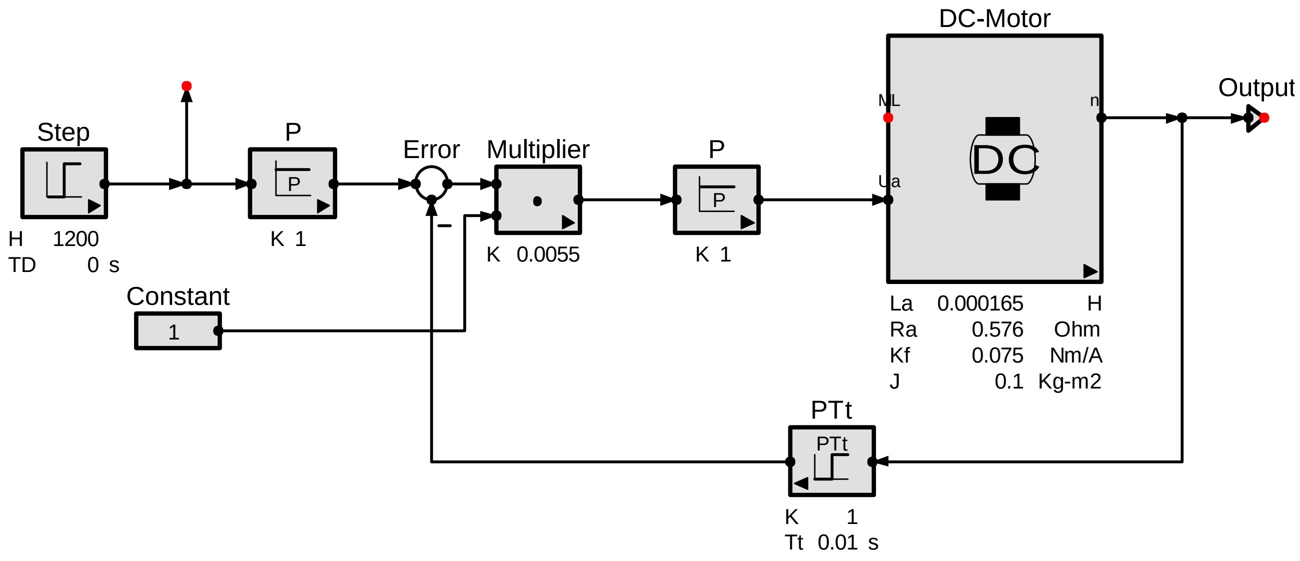


Figure 4-29: Block diagram of DC motor model with step input and P controller

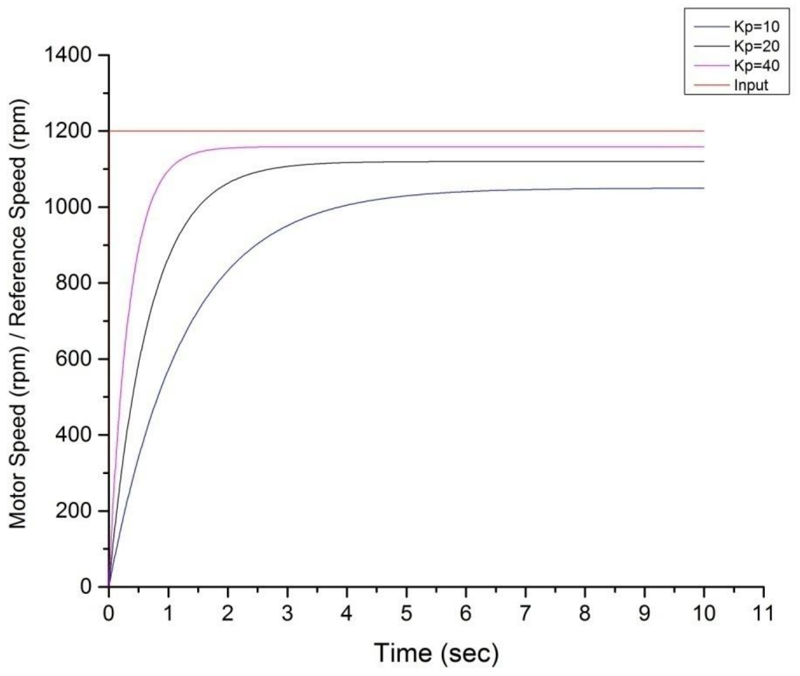


Figure 4-30: Response curve of DC motor model with step input and P controller

From the Figure 4-30, it is found that motor lagged behind the reference speed for lower values of Kp. As Kp is increased it approaches the reference with lesser time. Although a steady state error is there for P controller.

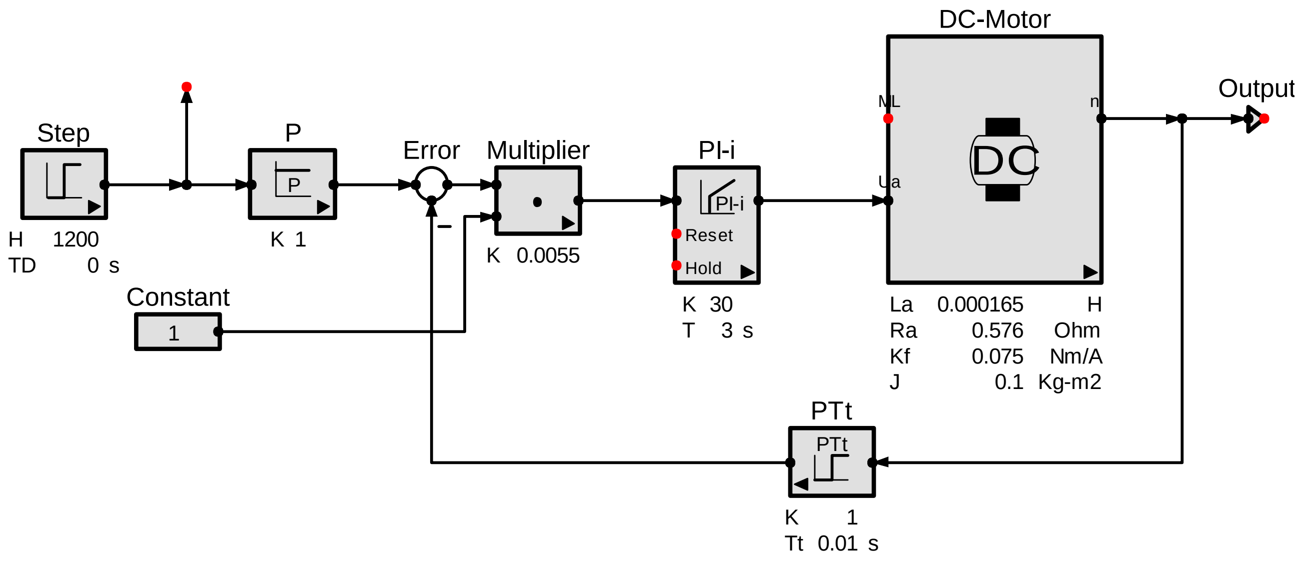


Figure 4-31: Block diagram of DC motor model with step input and PI controller with tuned parameters

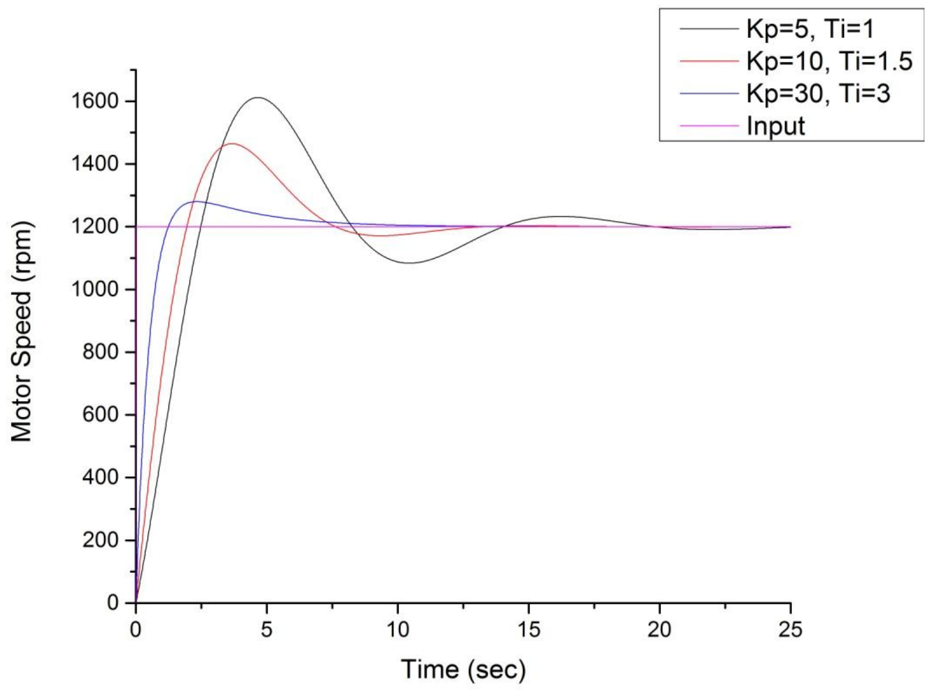


Figure 4-32: Response curve of DC motor model with step input and PI controller with tuned parameters

For simulating PI controller Kp value found from the simulation of P controller is used with slightly reducing it. For Kp=5 and Ti= 1 the system oscillates around the set point but reducing the integral action diminishes the oscillation. Increasing the Kp value makes the system response faster but higher Kp value should not be used with greater I action i.e. lower Ti which will make the system unstable again.

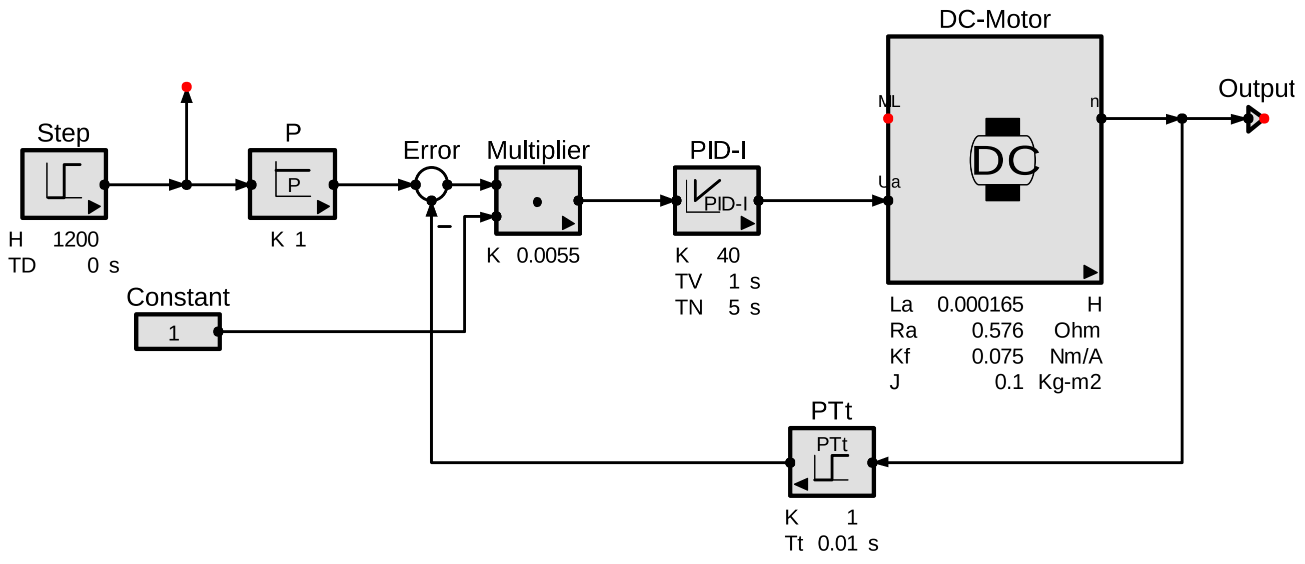


Figure 4-33: Block diagram of a DC motor model with step input and PID controller

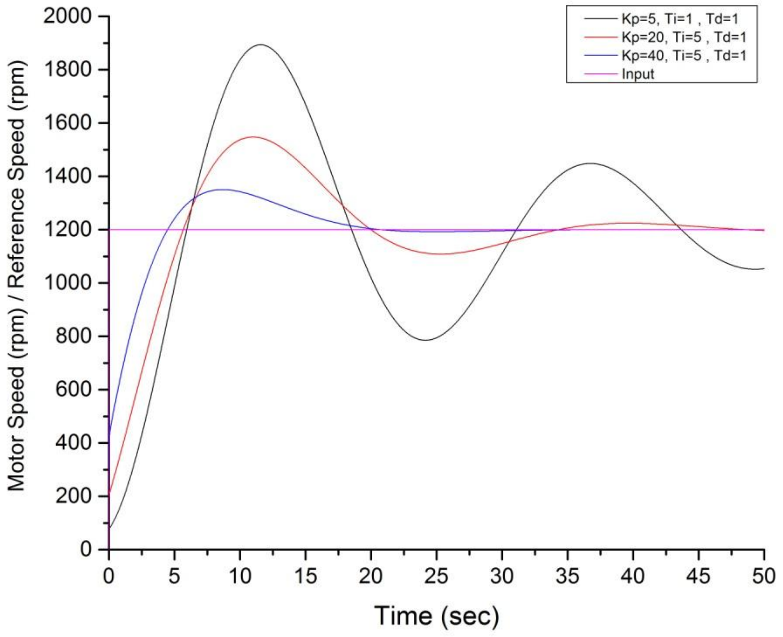


Figure 4-34: Response curve for the DC motor model with step input and PID controller

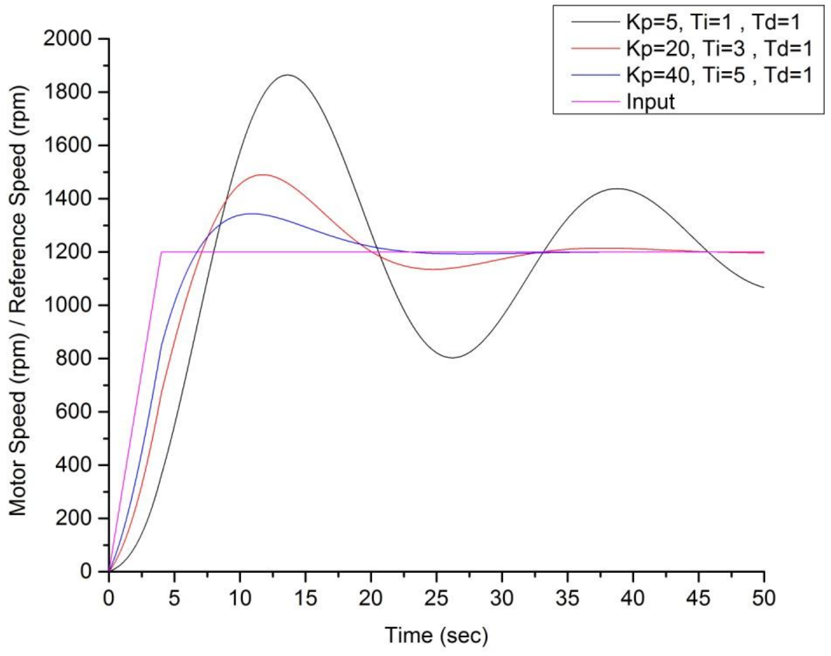


Figure 4-35: Response curve for the DC motor model with ramp input and PID controller

Figure 4-34 and 4-35 show response of the motor for step and ramp input using PID controller. From Figure 4-34, it is found that for the first case Kp=5, Ti=1 and Td=1 and the system overshoots with large amplitude. Then Kp value is increased that means increasing P and D action and Ti also increased which means decreasing I action. This makes the system to respond faster with lower overshoot. Further increasing the Kp without changing Ti and Td makes the system stable at setpoint.

Figure 4-35 depicts similar result for ramp input though for same set of Kp, Td and Ti it reaches the set point later than the step input simulation.

Chapter 5 : EXPERIMENTAL SETUP AND METHODOLOGY

5.1 Experimental Setup

5.1.1 Test Bench

Motor control test bench is setup designed to test a mobile dc servo-motor's response when it is required to run at a predefined speed profile [19]. It is possible to test here whether it can follow the required speed or getting behind the speed or overshooting the speed. To test a motor in this bed, practical circumstances that a motor faces on its track is simulated. The setup contains a drive motor which rotates a shaft coupled with a flywheel. The flywheel simulates the inertial load which is present in practical case. This drive motor is controlled and monitored by a microcontroller based data acquisition system called LabJack.

The test bench also contains motor drive control electronics which is discussed in chapter 3, a host of sensors and power supply system as shown in Figure 5-1:

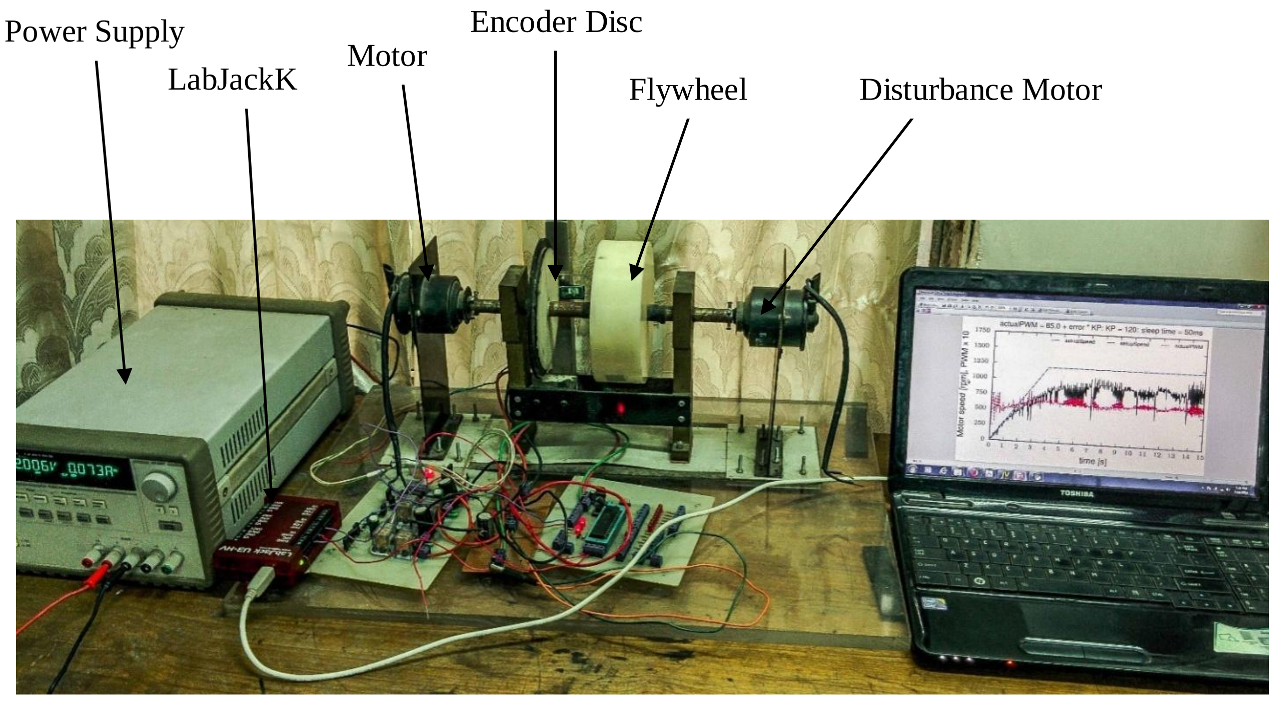


Figure 5-1: Motor Control Test Bench

A 12V DC motor is coupled to the shaft of the test bench as drive motor. Its main purpose is to rotate the shaft at a predefined speed. It's driven with the help of an adjustable power supply. With this power supply it is possible to supply any voltage between 10 and 25 with an accuracy of 0.001 voltage. Motor'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 disc [chapter 3] and disturbance motor on it. Encoder disc is locally made of nylon, circular in size and with equally spaced 180 holes in its periphery mounted on the shaft. An optical sensor is placed across the encoder disc [chapter 3]. Flywheel is made of nylon and provides inertia load. Another 12 V DC motor is coupled to the shaft of the robot control test bench at the opposite side of the drive motor. This motor is termed as 'Disturbance Motor'. This motor is not powered by any external power source. Its 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 of its coil. If this produced back emf is dropped in a resistance it will cause the braking effect on its shaft which in turn reduce the speed of whole system and will require control action to boost its power supply to maintain speed. A rectangular platform of acrylic sheet is used as base which supports all the parts of test bench.

5.1.2 Motor Drive System

Motor drive system [chapter 3] 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 its speed. This board includes a signal conditioning circuit for the encoder signal coming from the rotation of shaft. Normally when signal is blocked, this sensor provides a signal of 5V but when light is passed through the encoder, optical sensor attached with encoder transmits signal of voltage 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). In result it puts output signal as 0V when IR is blocked and 5V when IR passes. In this way signals are conditioned in binary levels.

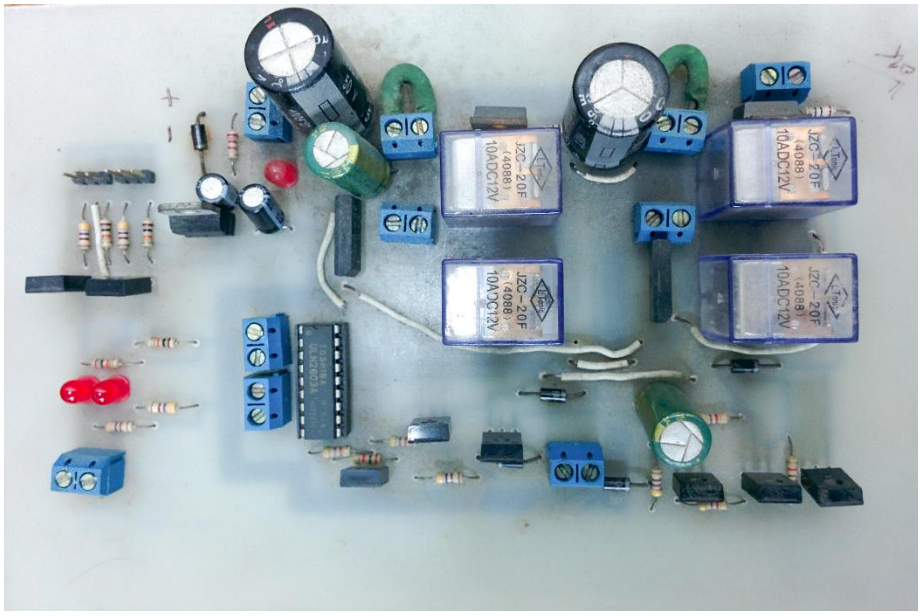


Figure 5-2: Motor Drive System

5.1.3 Data Acquisition and Control System: LabJack U3-HV

Labjack U3-HV is a data acquisition system which provides discrete digital I/O, analog I/O, and counter/timer I/O. It uses a USB interface for all command and data response transactions. In this thesis we used to monitor and generate control signal for controlling the drive motor. The U3 is powered from the USB interface. There is no provision for an external power supply, which is common for devices of this type. So even though there is +5 V DC available at some of the terminal positions, care should be taken not to draw too much current and cause the USB channel to go into shutdown.

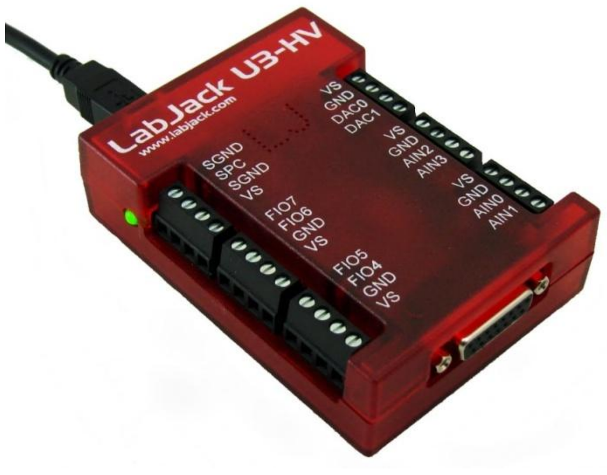


Figure 5-3: LabJack U3-HV

The LabJack U3 provides eight FIO (Flexible I/O) ports on the terminal blocks, and eight EIO (Extended I/O) ports and four CIO (Control I/O) ports on the DB-15 connector. In addition, there are two DAC outputs, six VS (+5 V DC) terminals, five common ground terminals, and two protected ground terminals. The terminals labeled AIN0 through AIN3 are actually FIO ports FIO0 through FIO3. These are configured as analog inputs by default, but one can change their function by modifying the LabJack's configuration. The two DAC outputs operate in either 8-bit or 16-bit mode with a range of between about 0.04 and 4.95 volts. Using the 8-bit mode may result in a more stable output with less noise. The U3 contains two timers and two clocks. Whenever a timer or clock is enabled, it will take over an FIO channel. In the U3-HV FIO4 is first, followed by FIO5, and so on. The standard U3 starts the counter/timer assignment at FIO0. It has also 12-bit ADC facility.

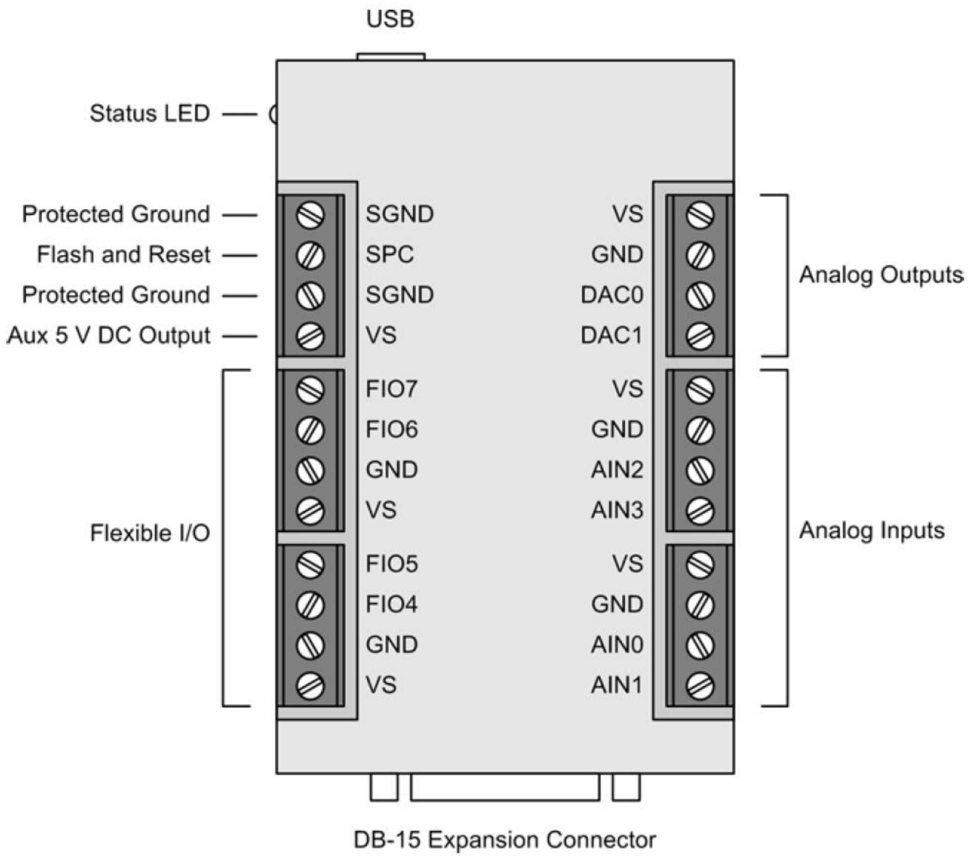


Figure 5-4: LabJack U3-HV connections

The Python interface for LabJack devices, called LabJackPython, is a wrapper for the low-level interfaces. LabJack programmed by Python programming language works as the Data Acquisition system and control system in this thesis.

Testing the response of Data Acquisition system

To test LabJack's capability of representing various signals, a 1kHz square wave signal is generated by a function generator (TG215 Function Generator). Then signal is sampled with 5kHz, 10 kHz and 20 kHz sample frequency. This sampled values are continuously streamed to the PC and plotted to find the response of the acquisition system. It is found that slew rate increases with increasing sampling frequency. Though the input signal is represented properly in each of the three sampling frequency as they are all greater than the Nyquist frequency, the higher sampling rate produces more accurate representation of the input signal.

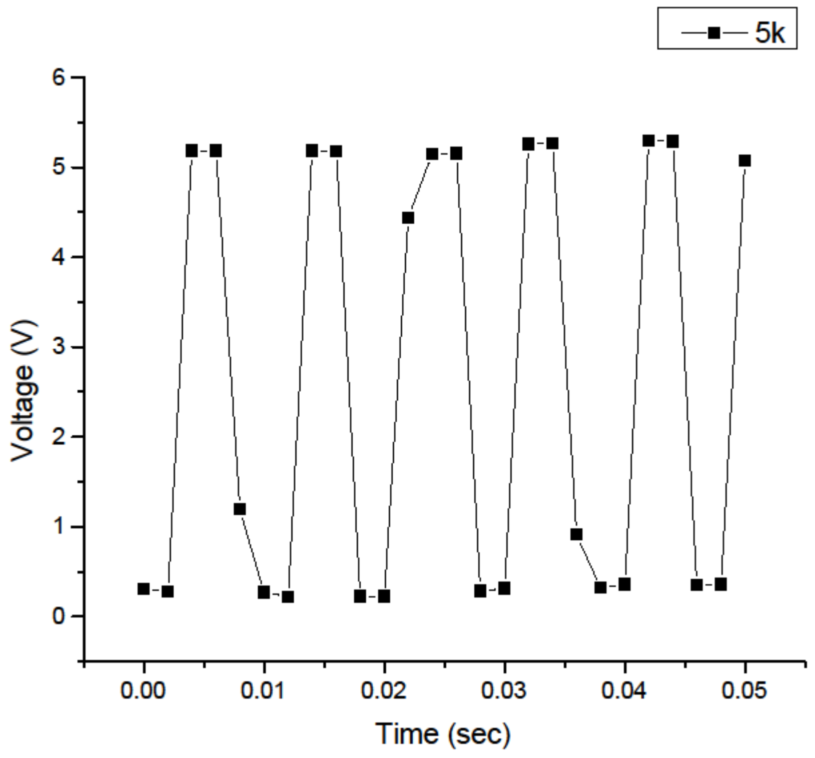


Figure 5-5: 1 kHz square wave sampled at 5 kHz

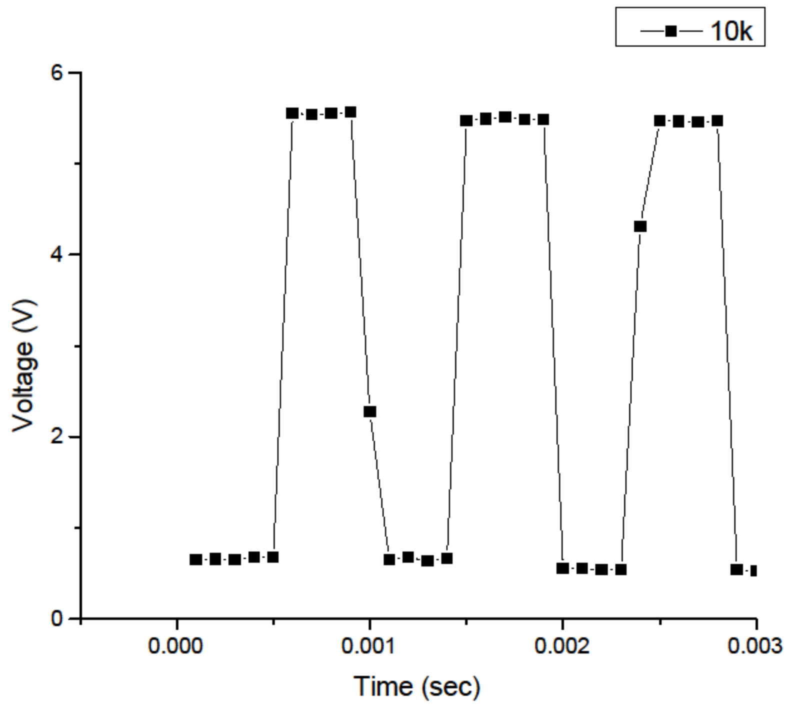


Figure 5-6: 1 kHz square wave sampled at 10 kHz

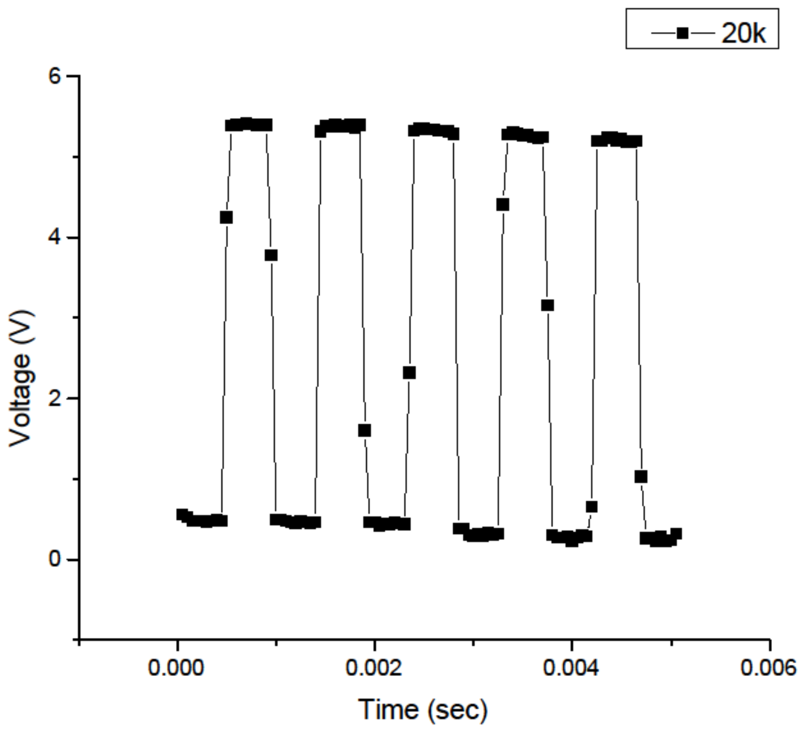


Figure 5-7: 1 kHz square wave sampled at 20 kHz

5.2 Methodology

In the present study, speed of the drive motor is controlled using P, PI and PID controller and controller action and performance is studied to optimize it. To control the speed, a feedback loop is created.

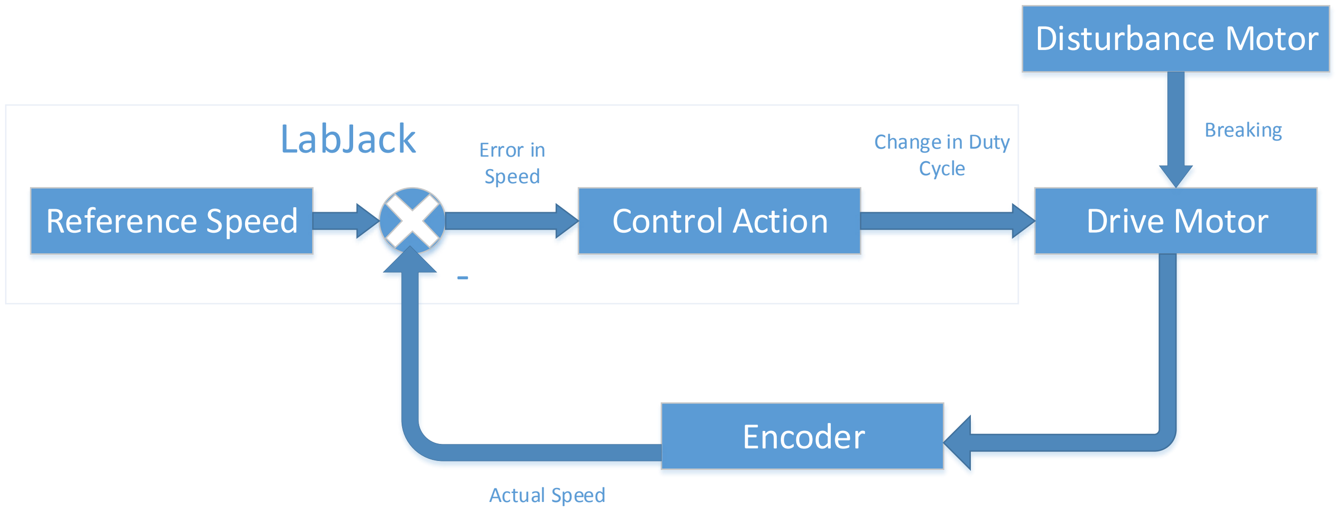


Figure 5-8: The Block Diagram of the Control System

Motor controlling in the present study includes the following tasks:

1. Reference Speed generation and Measurement
2. Actual Speed measurement
3. Error in speed calculation
4. Control Action
5. Development and Tuning of controller
6. Starting or Stopping the Drive Motor
7. Monitoring the controlled action and different parameters

5.2.1 Reference Speed

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. It is found that minimum duty cycle at which the motor starts in the setup is around 50% but once started it can run with much less duty cycle. For the convenience of the control system it is decided to limit the maximum speed of the required velocity profile at 1200 rpm which is around 80% duty cycle. Two reference velocity profile is created by the LabJack. The first one as a step function staring from 0 rpm (the minimum speed of the required velocity profile) jumping instantly to 1200 rpm. The second one as a ramp starting from 0 rpm to 1200 rpm in 4 seconds. As minimum duty cycle at which motor starts is around 50%, so at first we will start the motor with a higher duty cycle. So, the controller will detect negative error instantly and will reduce the speed to match the reference speed. Reference speed is saved in a array named *setupSpeed*and continuously checked with the actual speed.

Timer0 configuration settings in LabJack

TimerCounterPinOffset = 5

Timer Mode = 0 (16 Bit PWM output)

Timer Clock Base = 2 (732 Hz PWM signal)

5.2.2 Actual Speed Measurement

The LabJack U3-HV comprises of 2 counters (Counter0-Counter1). Each counter (Counter0 or Counter1) consists of a 32-bit register that accumulates the number of falling edges detected on the external pin. Here number of pulses from the encoder is counted by Counter0 (FIO7). The time required to accumulate these pulses is calculated by Time() function. So, the actual speed can be determined from the following equations:

$$Rate\ of\ encoder\ pulse = \frac{Difference\ \ in\ Successive\ \ counts}{Time\ \ elapsed\ \ between\ \ Successive\ \ count}\ \text{Hz} \qquad \text{…………..5-1}$$

$$rpm = {Rate\ of\ encoder\ pulse \times 60}\Big/{No.of\ encoder\ holes} \qquad \text{………..5-2}$$

The measurement of actual speed is very critical for this study. Uncertainty associated with can lead to system instability. For this reason the actual speed was measured at different time interval and their effect was studied. Time interval used for actual speed measurement was 20 ms, 30 ms, 50 ms. After measuring the actual speed it was being checked with reference speed and error was calculated.

5.2.3 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. Percentage of Error in speed can be calculated using the following expression:

$$error = {}^{setupSpeed - actualSpeed}\!/_{setupSpeed} \times 100\ \% \qquad \text{……………..5-3}$$

5.2.4 Control Action and Data Transmission

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. In our present study the control action is generated instantly after the error checking i.e. in two successive programming cycle. Control algorithm varies with the type of controller i.e. whether P/PI/PID are being used. The control action is a PWM signal saved in an array named *actualPWM*. This is the duty cycle that the motor receives.

Configuring LabJack for generating control action

Variable used :actualPWM

Timer0 (Pin FIO5) of LabJack is used for generating this PWM signal.

Configuring the Tiome0

The U3 has 2 timers (Timer0-Timer1) and 2 counters (Counter0-Counter1). When any of these timers or counters are enabled, they take over an FIO/EIO line in sequence (Timer0, Timer1, Counter0, then Counter1), starting with FIO0+TimerCounterPinOffset. Here TimerCounterPinOffset= 5. So, FIO5 pin is configured as control action generating pin.

Timer0 Outputs a pulse width modulated (PWM) rectangular wave output. Value passed to Timer0 should be 0-65535, and determines what portion of the total time is spent low (out of 65536 total increments). That means the duty cycle can be varied from 100% (0 out of 65536 are low) to 0.0015% (65535 out of 65536 are low). In this work percentage of the duty cycle is varied from 43% to 80% then it becomes fixed. To do so Duty Cycle is varied from 43% to 80% increasing at 0.025 sec by changing the passed value to Timer0. Relation between passed Value and actualPWM becomes:

$$Value = 65535 - 655.35 * actualPWM \qquad \textbf{.....................5-4}$$

Timer0 configuration settings in LabJack

TimerCounterPinOffset = 5

Timer Mode = 0 (16 Bit PWM output)

Timer Clock Base = 2 (732 Hz PWM signal)

Data transmission is done by command response mood by LabJack. All the variable are stored in their respective arrays and transmitted to the computer by a single command in one instruction cycle.

5.3 Programming Algorithm

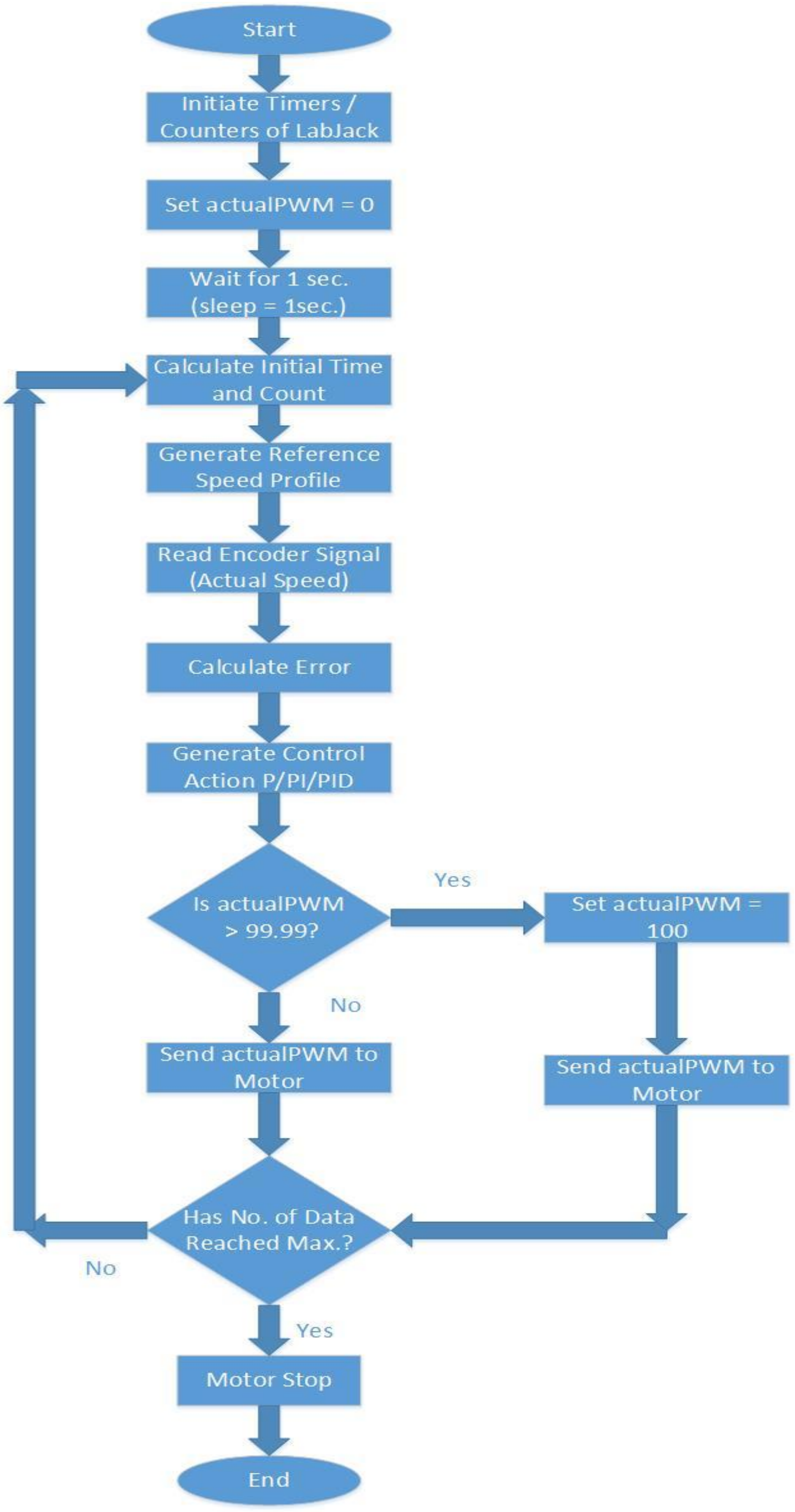

Start
Initiate Timers / Counters of LabJack
Set actualPWM = 0
Wait for 1 sec. (sleep = 1sec.)
Calculate Initial Time and Count
Generate Reference Speed Profile
Read Encoder Signal (Actual Speed)
Calculate Error
Generate Control Action P/PI/PID
Is actualPWM > 99.99?
Yes
Set actualPWM = 100
No
Send actualPWM to Motor
Send actualPWM to Motor
Has No. of Data Reached Max.?
No
Yes
Motor Stop
End

Figure 5-9: Flow Chart of the Algorithm

Chapter 6 : EXPERIMENTAL RESULT AND DISCUSSION

A predefined velocity profile was being generated and the motor's response was observed that how it responds in following this setup speed. Different control mood had been implemented to get required response from the motor. The control parameters were being varied to achieve optimum control action. The first predefined speed profile(setup speed profile-1) which the motor is going to follow is a ramp of 1200 rpm in 4 seconds.

At first the drive motor was run with various duty cycles without any control action and its speed profile are found as follows:

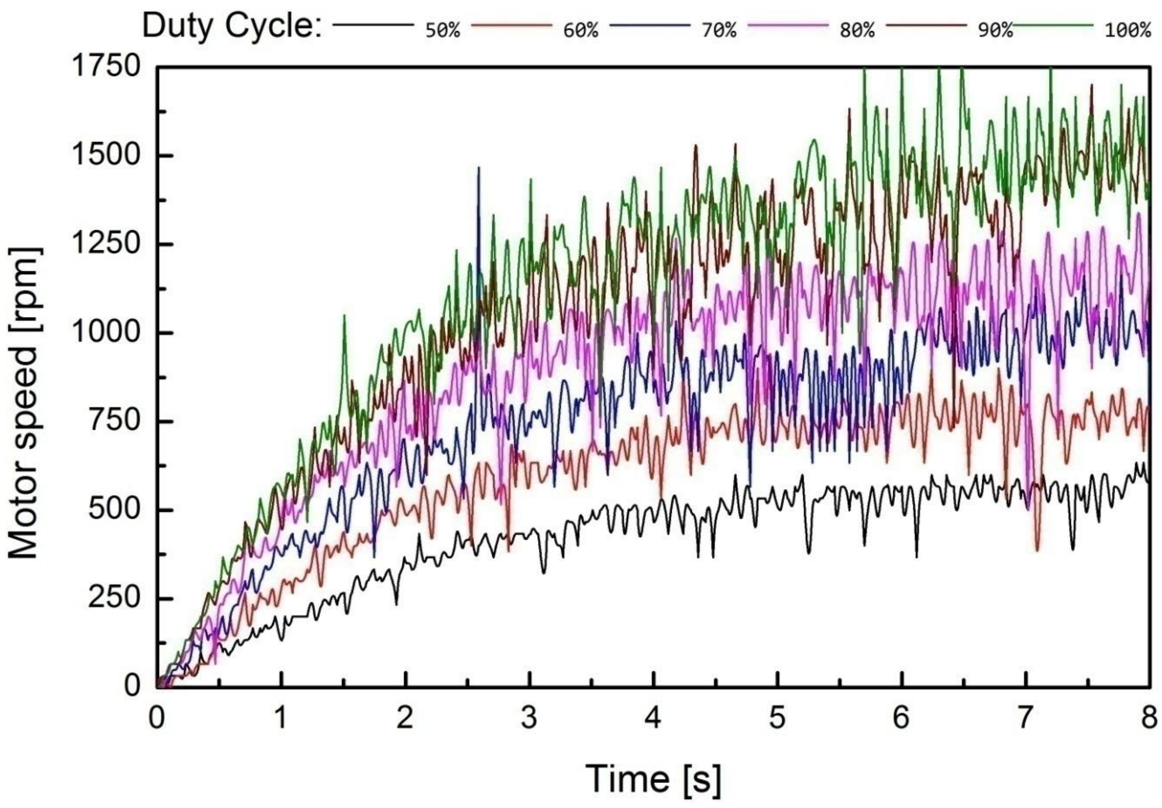


Figure 6-1: Motor's Speed at Different Duty Cycle

Figure 6-1 shows at different duty cycle motor's speed reaches its maximum value after around 7 sec i.e. steady state occurs at this time. It is also found from the Figure 6-1 that maximum speeds at 50%, 60%, 70%, 80%, 90% and 100% duty cycle are approximately 550, 750, 950, 1125, 1375, 1500 rpm respectively. So, the speed of the motor varies linearly with duty cycle up to 80%. Figure 6-2 was obtained by using polynomial curve fitting of order 4 with Figure 6-1 data.

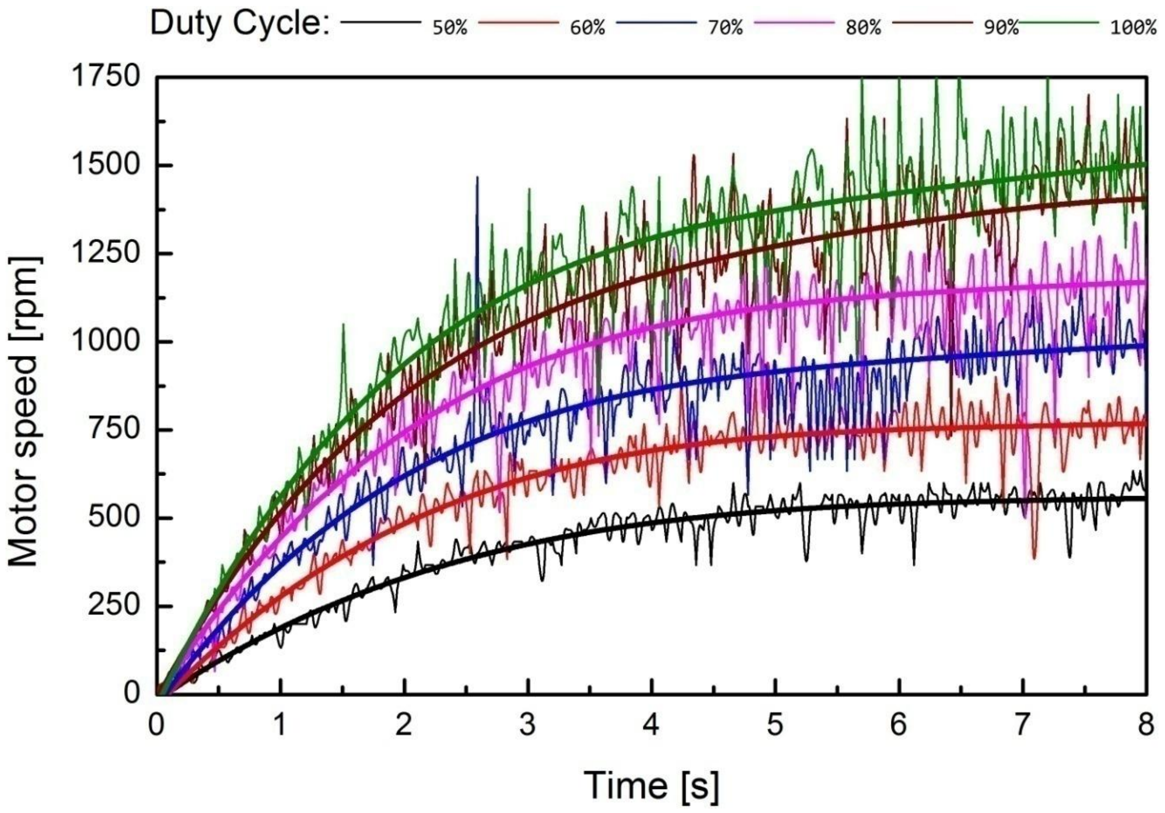


Figure 6-2: Polynomial Curve of Motor's Speed at Different Duty Cycle

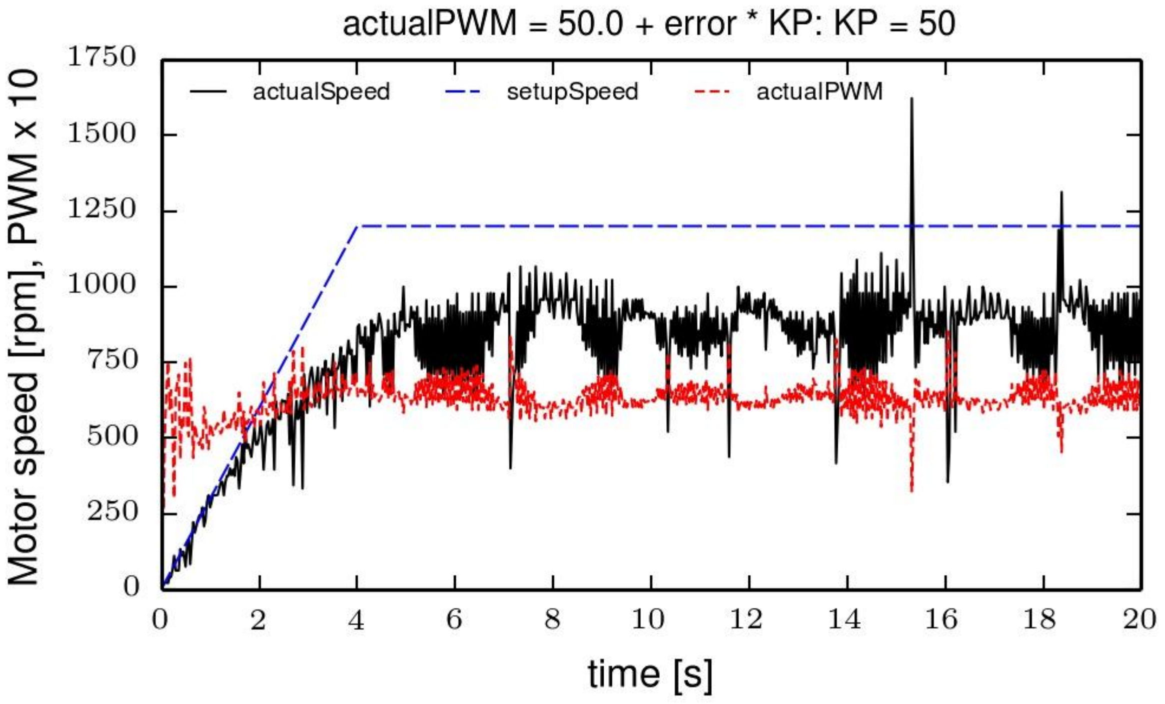


Figure 6-3: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 50% duty cycle, Kp = 50)

Then, a reference speed was generated and the motor was controlled by some control action. For the first case P controller is used. As the motor would not start at lower duty cycle than 50%, initial value of the controller's output which is a PWM signal that corresponds to actual PWM, is set at 50% and the value of proportionality constant is increased starting from 50. Figure 6-3 shows the reference speed (setupspeed), actual speed and control action generated by LabJack (actualPWM). It shows at Kp=50 motor was unable to reach the reference speed. It follows the reference speed for 1.5 seconds but then lagged behind. As reference speed reaches its maximum value, error becomes constant and at this fixed value of Kp, controller could not produce higher control action. For this reason actualPWM becomes approximately fixed after 2.5 sec. The motor reaches its stedy state at around 12 seconds though slight oscillation is found which results from action of the controller.

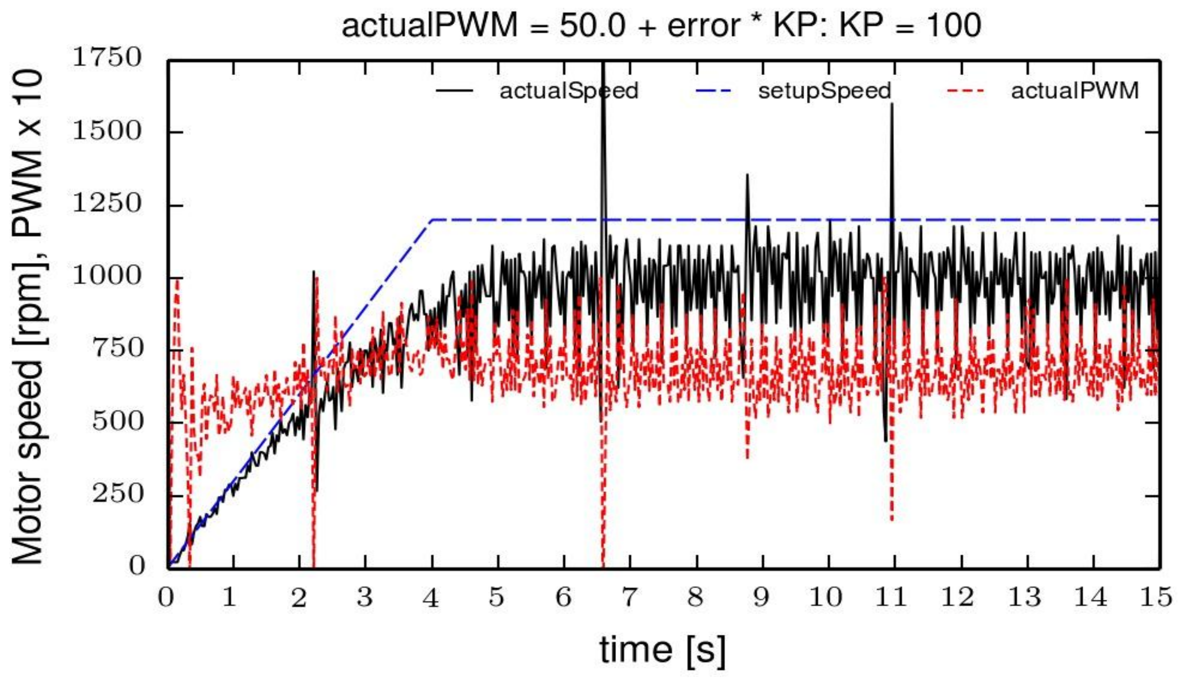


Figure 6-4: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 50% duty cycle, Kp = 100)

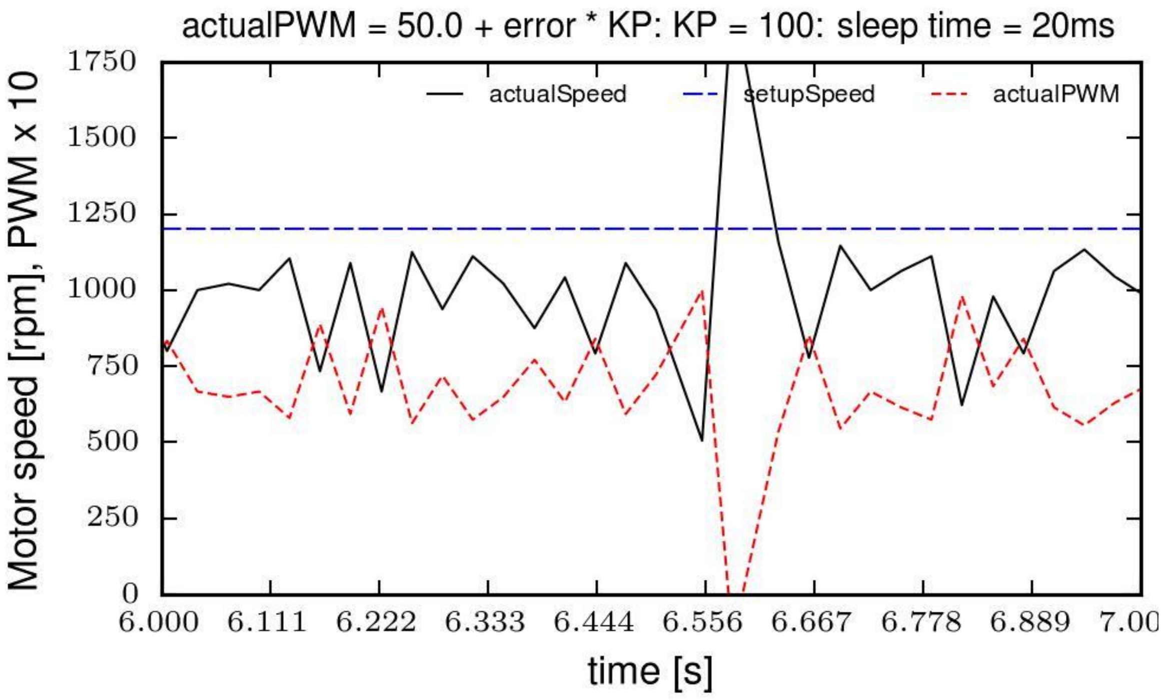


Figure 6-5: A closer look at the control action

Then the value of Kp was increased to 100 and it is found from the Figure 6-4 that actual speed follows the reference speed more time than the above case. Though it could not reach the final speed but it tries to keep up with the reference speed up to 5 sec as the actualPWM showing increasing trend up to this moment. Interesting observation was that the controller was working with extreme precision which can be viewed from Figure 6-5. After 6th second there is sudden rise of actual speed and at that specific time controller produced an opposite surge in actualPWM to mitigate the effect of increased actual speed.

The jagged pattern of the actualPWM i.e. control action curve and actual speed curve is originated from the uncertainty associated from the measurement of actual speed by optical encoder. As this value is repeatedly measured at a fixed interval, there arises the uncertainty from bias error and precision error. These errors can be greatly reduced by increasing sampling time in expense of delaying the control action. So, the sampling time of encoder signal were increased from 20 ms to 30 ms and 40 ms. Another measure was taken with the actual speed value. A linear optimized curve function was declared and optimized actual speed was measured by taking feedback from previous two readings of optical encoder.

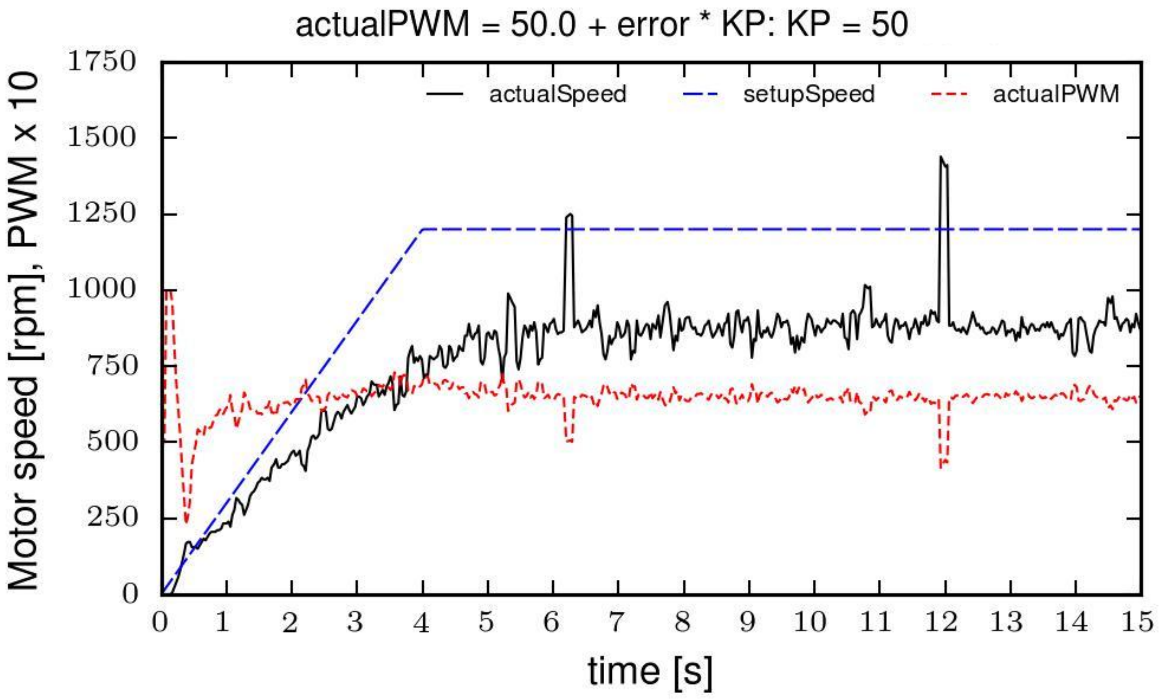


Figure 6-6: Response of the motor after linearization and 20 ms sampling time

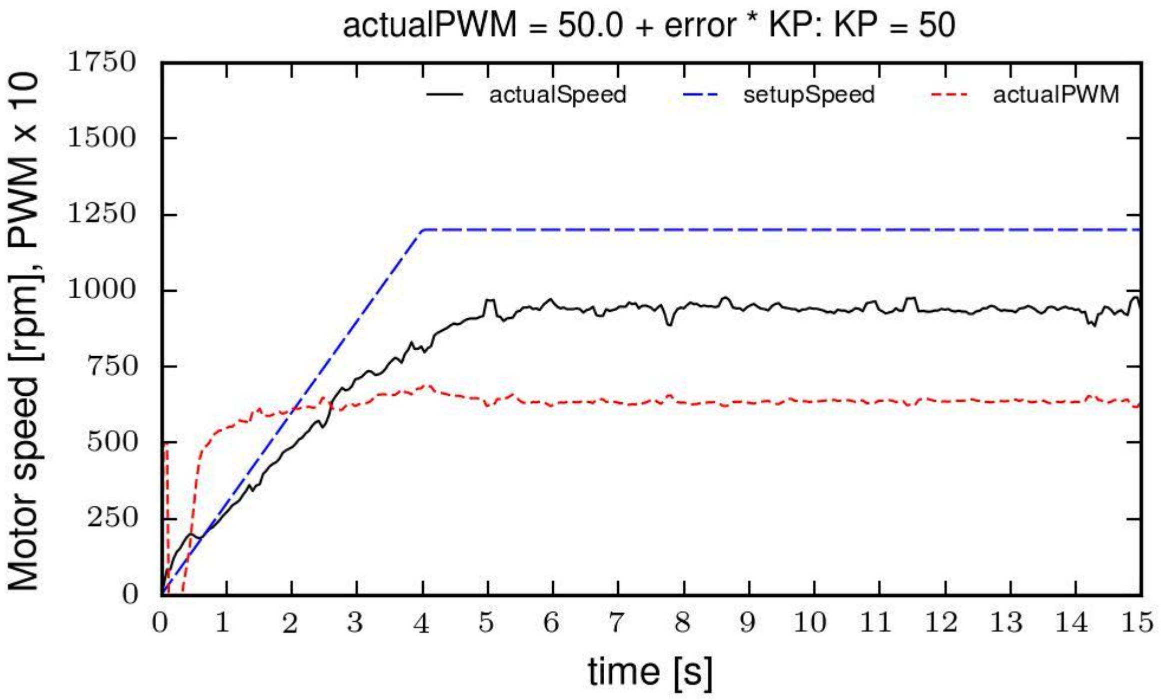


Figure 6-7: Response of the motor after linearization and 30 ms sampling time

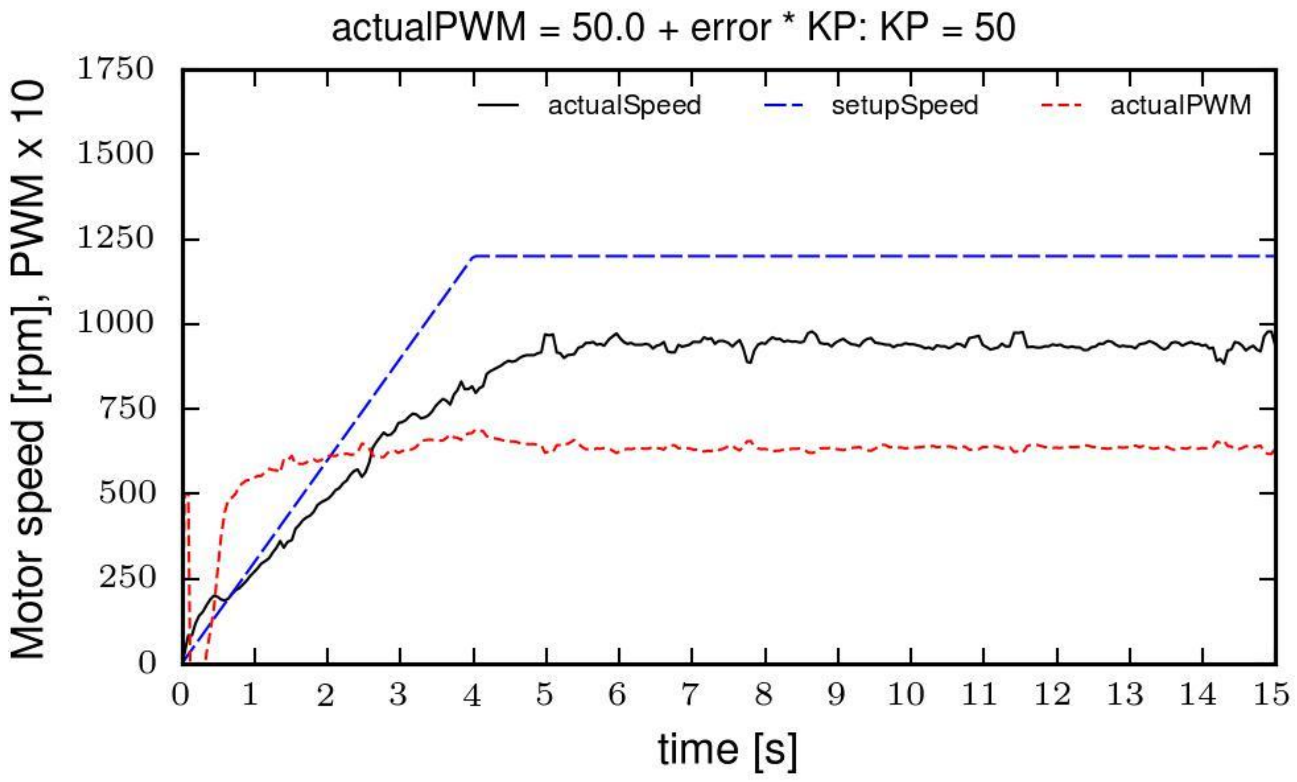


Figure 6-8: Response of the motor after linearization and 40 ms sampling time

Results from Figure 6-6 - Figure 6-8 shows that the jagged pattern of the controllers output and actual speed has been greatly reduced by linearization of the encoder signal. It is also evident that increasing the sampling time also reduces oscillation greatly. So, throughout the experiment the sampling time of 30 ms was used as optimum value.

From Figure 6-9 and Figure 6-10 it is observed that at Kp=200 and Kp=250 actual speed almost reached its desired value with Kp=250 providing better response. Though it is found that with the increase of Kp actualPWM becomes more noisy, and so as the actual speed. This is due to higher value of control parameter which produces greater control action at lesser error value. All these figures show similar starting trend that the motor at beginning follows the reference speed but after some time lagged behind. It is also found that with the increase of Kp motor reaches its steady state more rapidly.

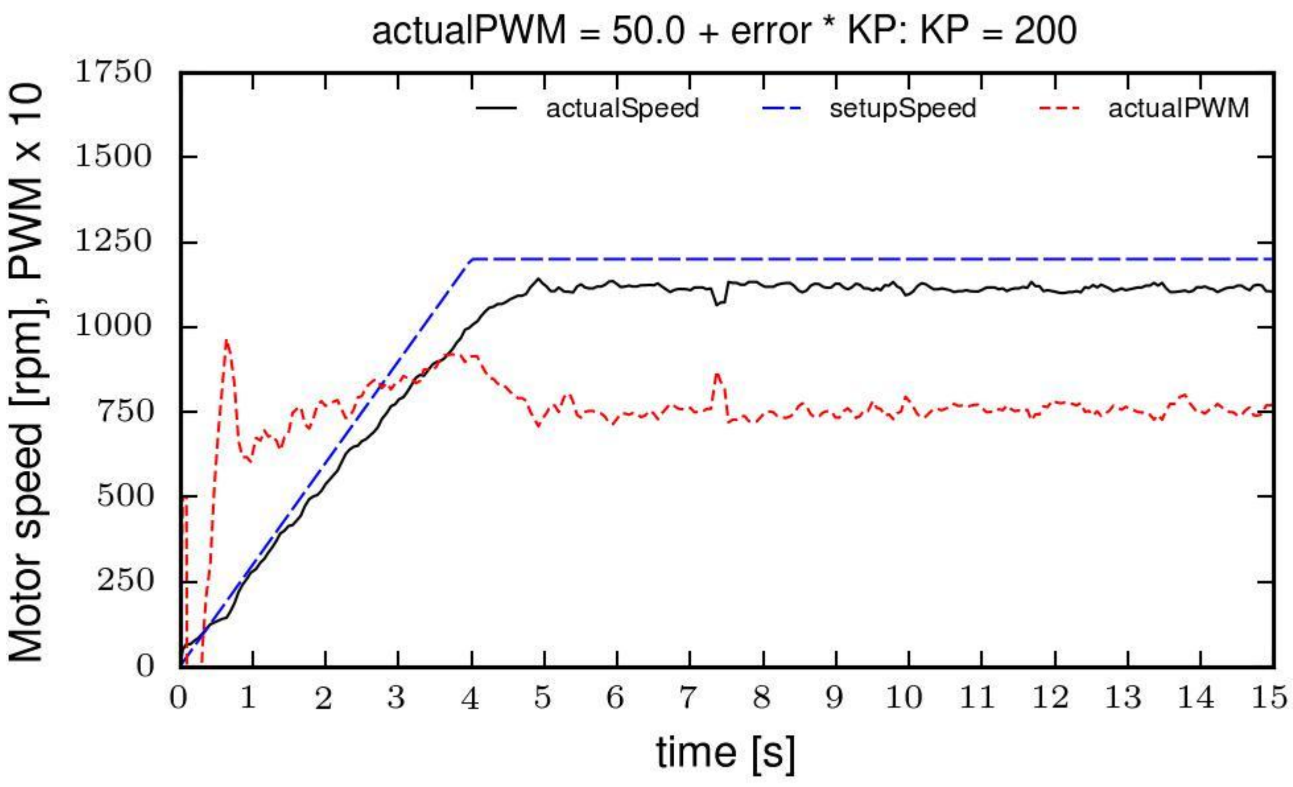


Figure 6-9: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 50% duty cycle, Kp = 200)

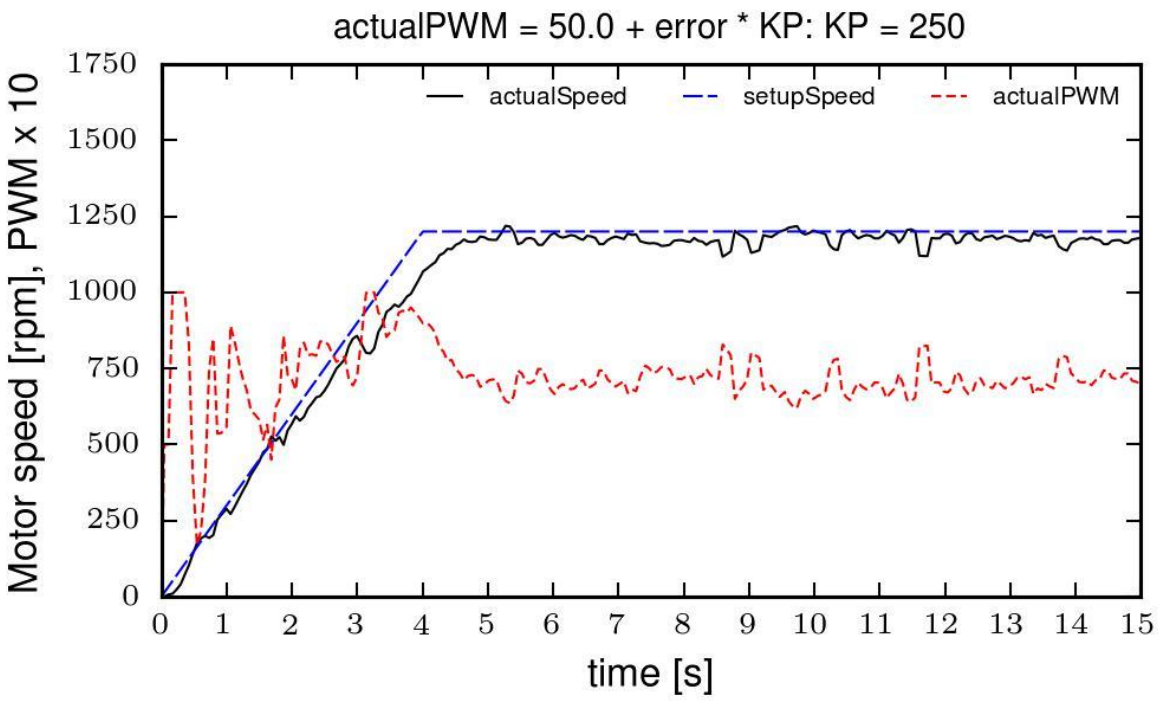


Figure 6-10: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 50% duty cycle, Kp = 250)

Error! Reference source not found. show the response of the motor Kp=300. At Kp=300 actual peed follows the reference speed up to 3.5 sec. In this case motor reaches its desired speed.

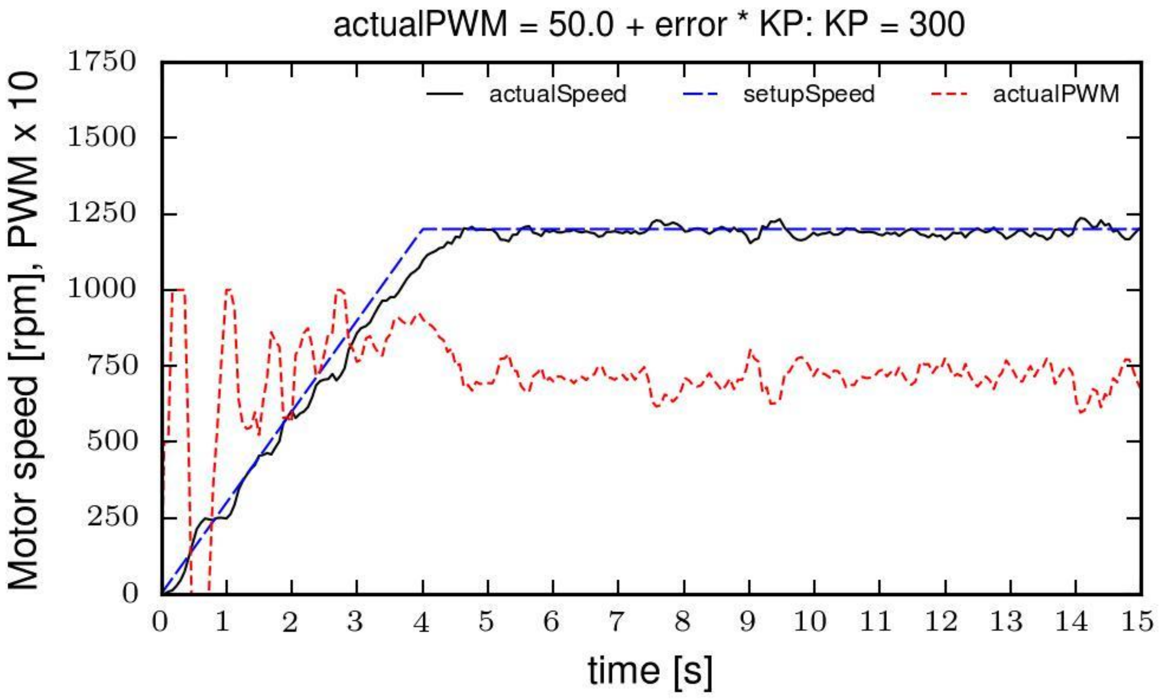


Figure 6-11: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 50% duty cycle, Kp = 300)

Comparing the graphs in Figure 6-3 to Figure 6-11, it can be concluded that increasing the value of Kp boosts control action, thus helps motor reach its desired speed more rapidly. But it makes control action instable hence the system response becomes more jagged.

Now, the value of Kp is kept at 300 and initial value is being varied to improve control action. At first, initial value is increased from previous value of 50 to 60. Figure 6-12 shows the response of drive motor with initial duty cycle 60% and Kp=300. It is found that up to 3.5 seconds actual speed almost follows the reference speed and in the vicinity of 3.5 seconds controller action becomes almost regular hence the system response. But once the drive motor reaches its desired speed and slightly overshoots, controller starts producing negative control action to mitigate the overshooting and thus the jagged pattern returns.

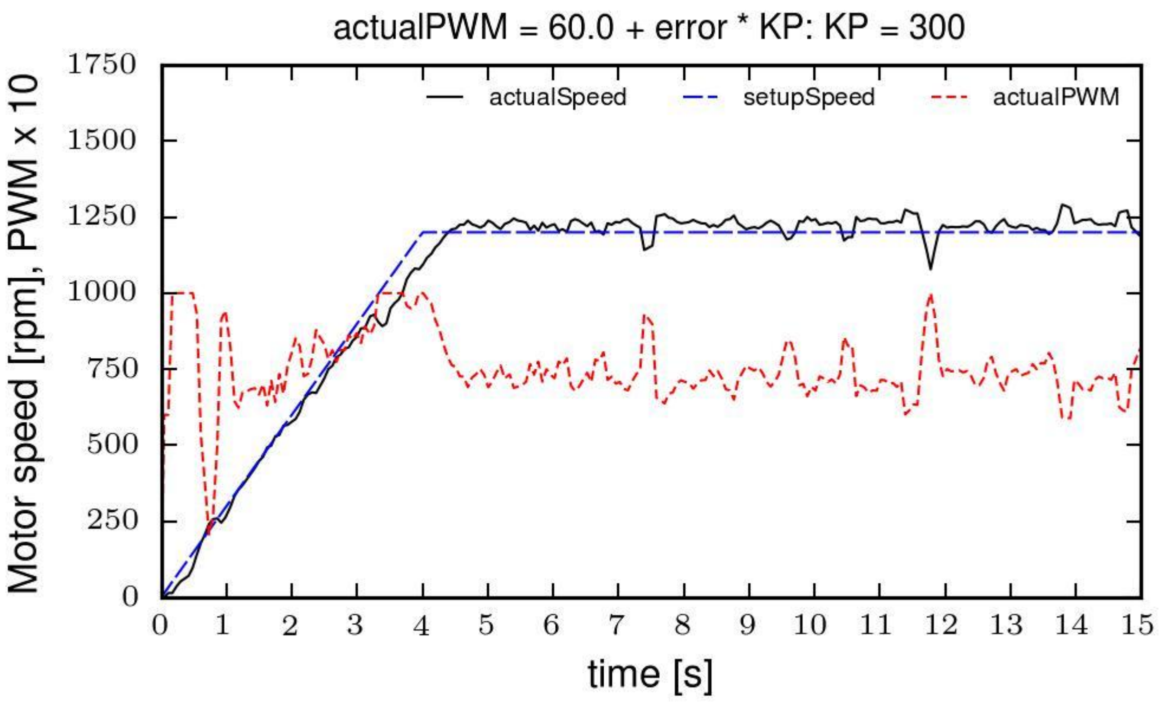


Figure 6-12: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 60% duty cycle, Kp = 300)

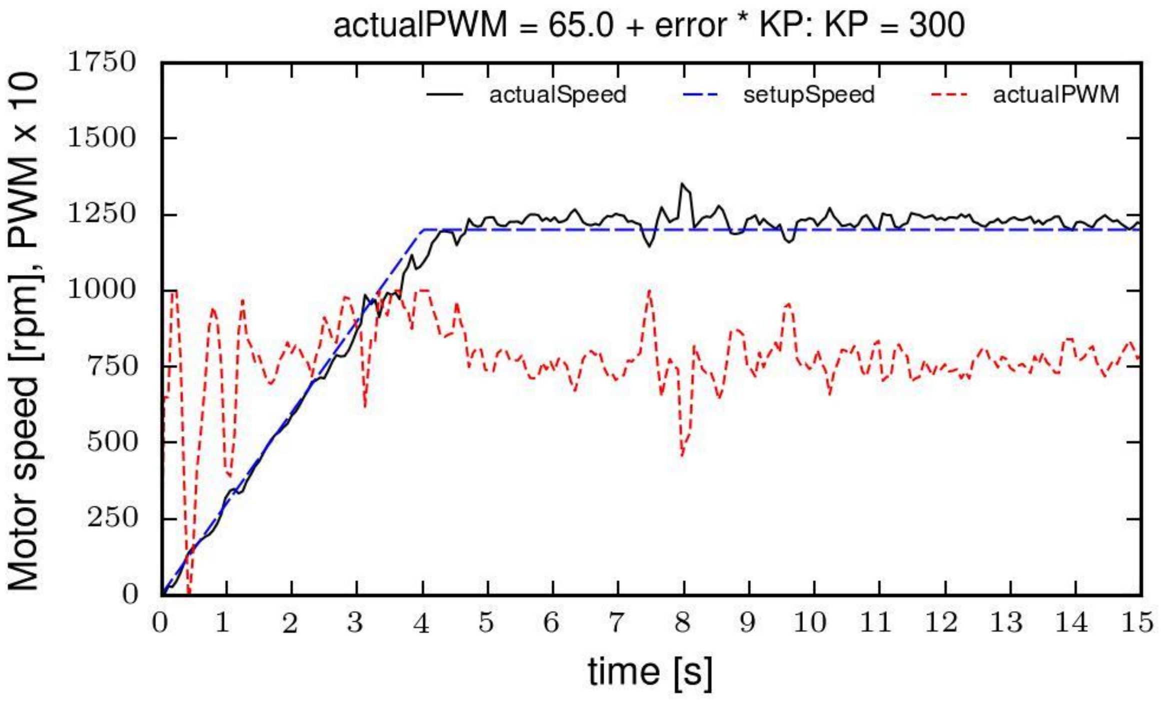


Figure 6-13: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 65% duty cycle, Kp = 300)

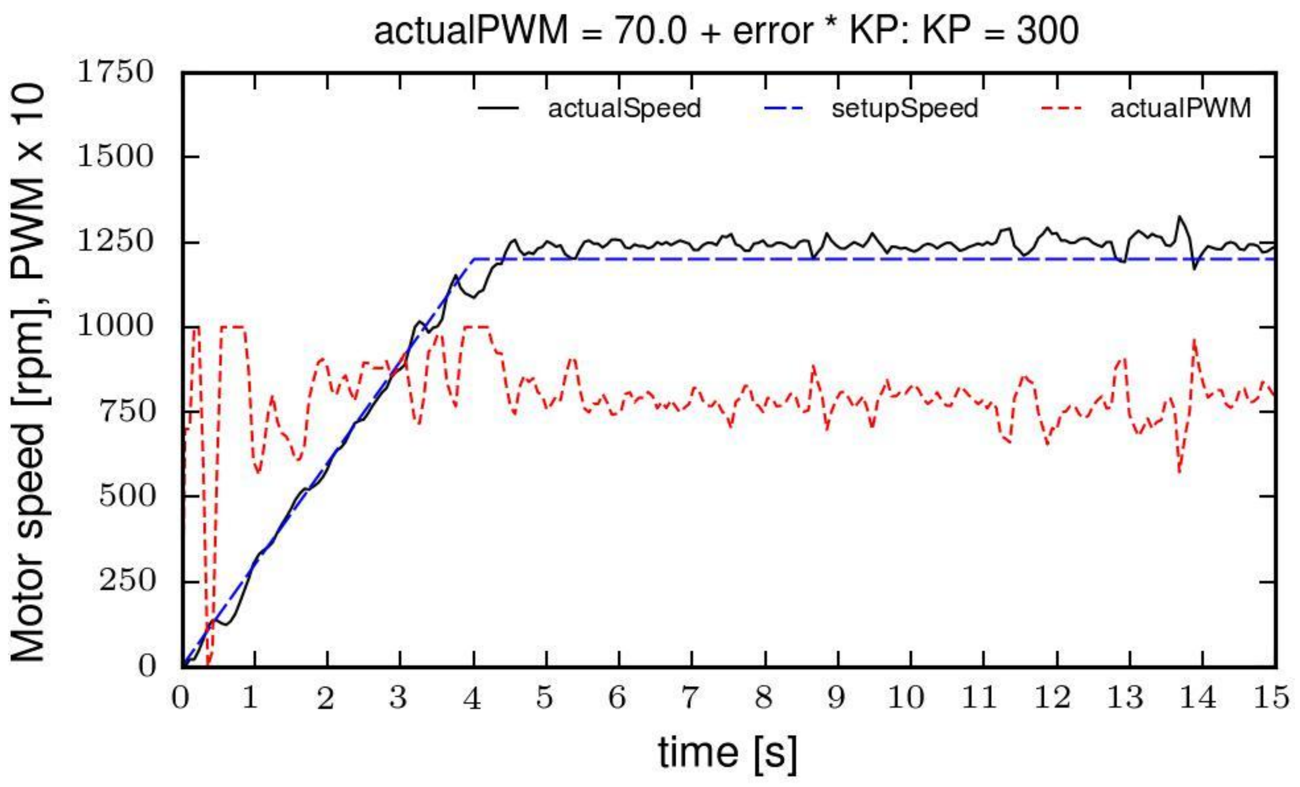


Figure 6-14: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 300)

Figure 6-13 and Figure 6-14 depict better response in terms of following and reaching the desired speed but inferior response in terms system stability. Also at higher initial value permanent overshoot is visible as here both the initial value and Kp value are high. If we further increase the initial value to 75% and 80% of duty cycle which is shown in the Figure 6-15 and Figure 6-16, we can find that at initial value of 75% and 80% motor follows the setup speed profile perfectly up to 4 sec. and after reaching the 1200 rpm value overshoot and oscillatory pattern increases with higher initial value. Though from Figure 6-15 and Figure 6-16 it is evident that with higher initial value, motor follow the ramp with greater accuracy. So, now we keep initial value fixed at 80% of duty cycle and reduce the Kp value to get optimum result.

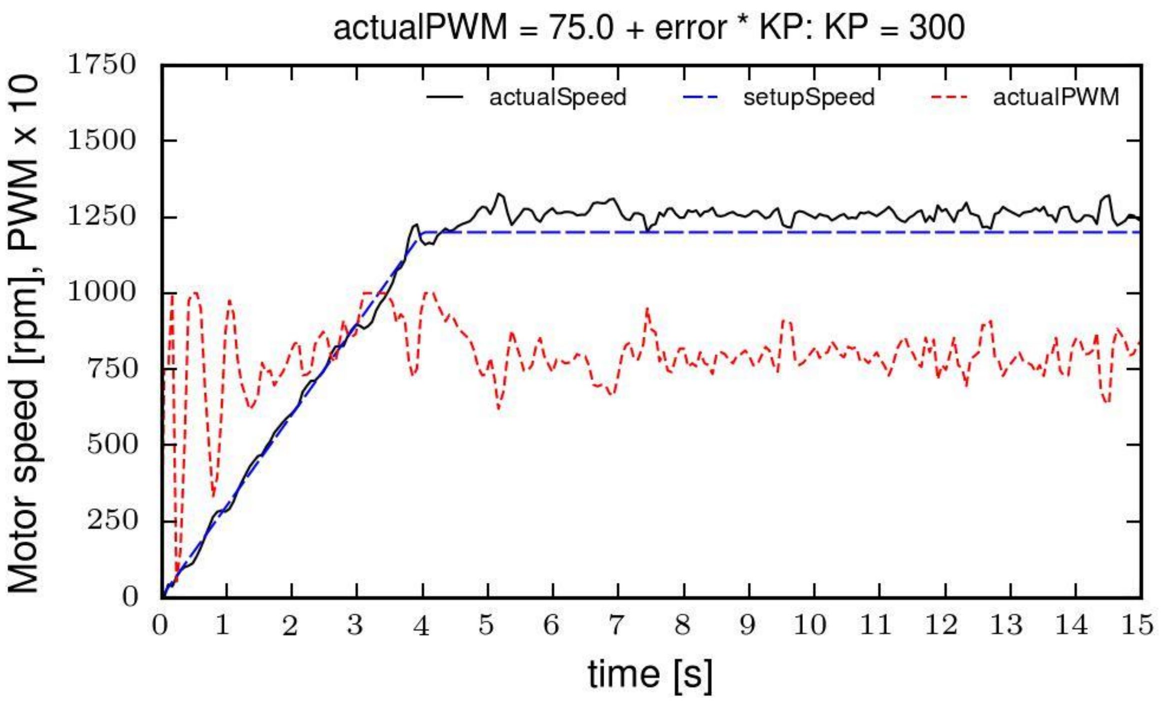


Figure 6-15: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 75% duty cycle, Kp = 300)

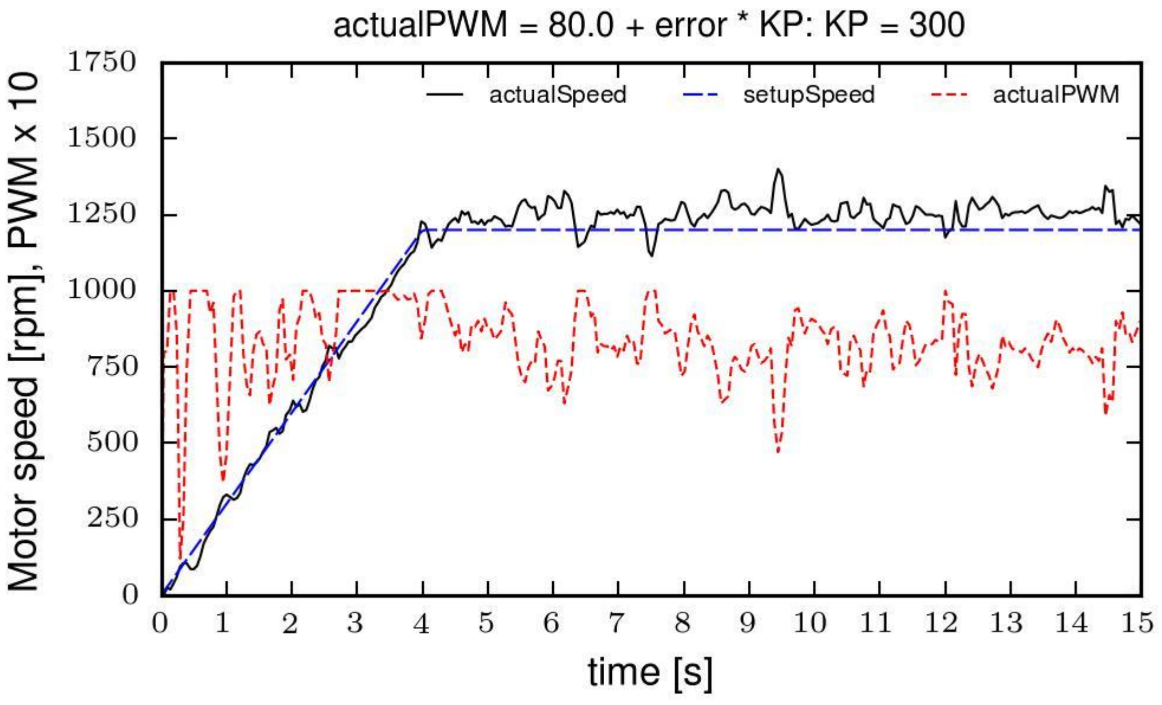


Figure 6-16: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 80% duty cycle, Kp = 300)

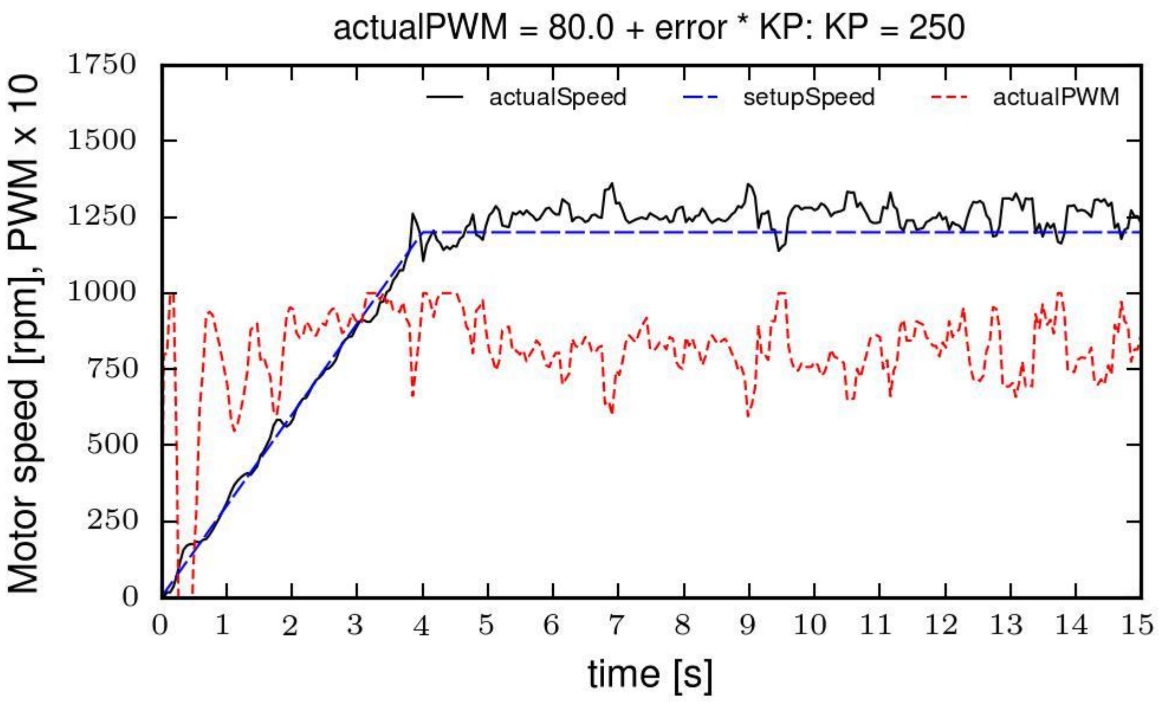


Figure 6-17: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 80% duty cycle, Kp = 250)

Figure 6-17, Figure 6-18, Figure 6-19, Figure 6-20 show response of the motor with initial value kept fixed at 80% of duty cycle and Kp=250, 200, 150 and 100 respectively. It is found that with Kp=250, motor perfectly follows the desired speed, then overshoots and keeps oscillating around the reference speed. With Kp=100, it overshoots at the beginning and then tracks back, but after 3 sec. lagged behind the setup speed. The reason of initial overshoot at lower Kp is that motor starts with higher duty cycle and because of low Kp, it takes time to track back. At lower Kp oscillation in the actualPWM curve is greatly reduced i.e. control action is more stable. This stable control action provides less oscillation in the motor actual speed curve. Response curve with Kp=150 depicts better result in terms of tracking the reference speed and oscillation of the system. Although all these response curve show permanent overshoot after reaching the 1200 rpm. So, a Kp value in the vicinity of 150 is now kept fixed as optimum value and initial value is varied from 65% of duty cycle to reduce oscillatory pattern.

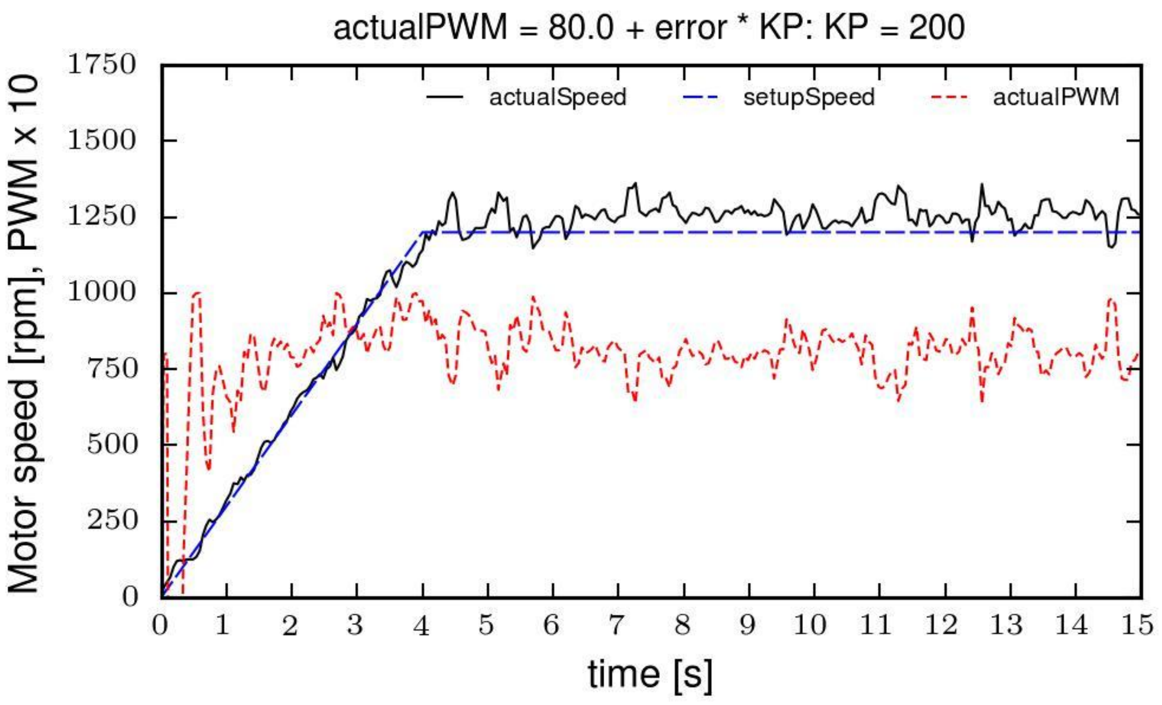


Figure 6-18: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 80% duty cycle, Kp = 200)

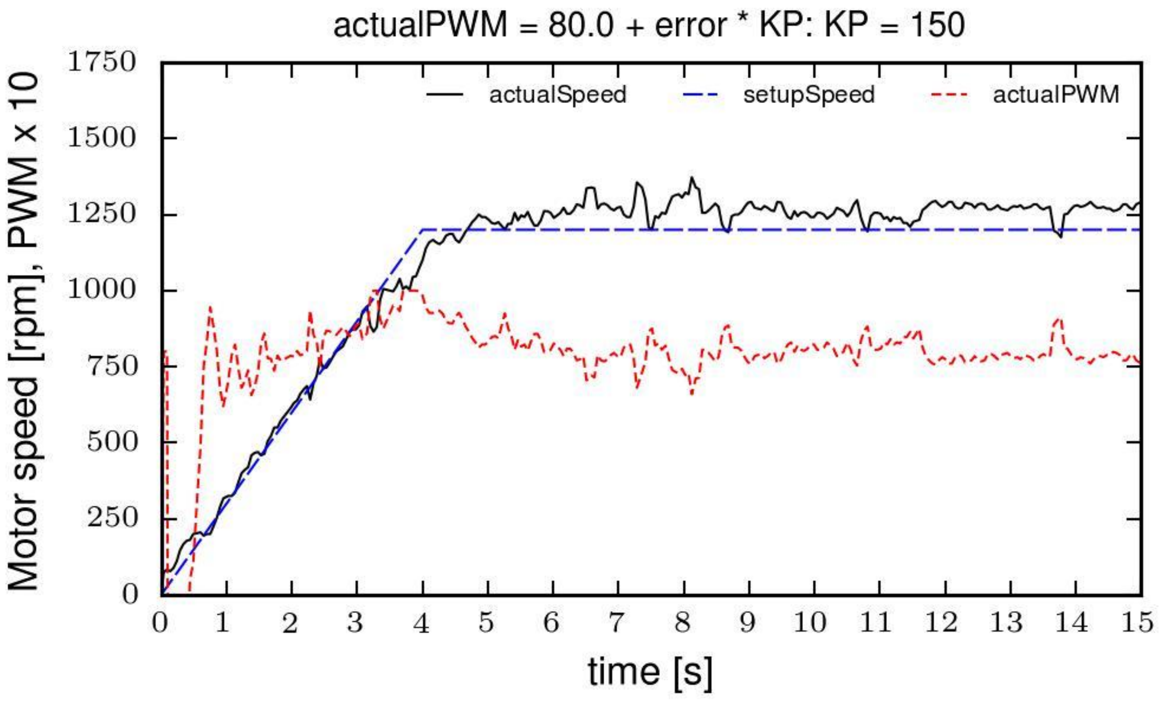


Figure 6-19: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 80% duty cycle, Kp = 150)

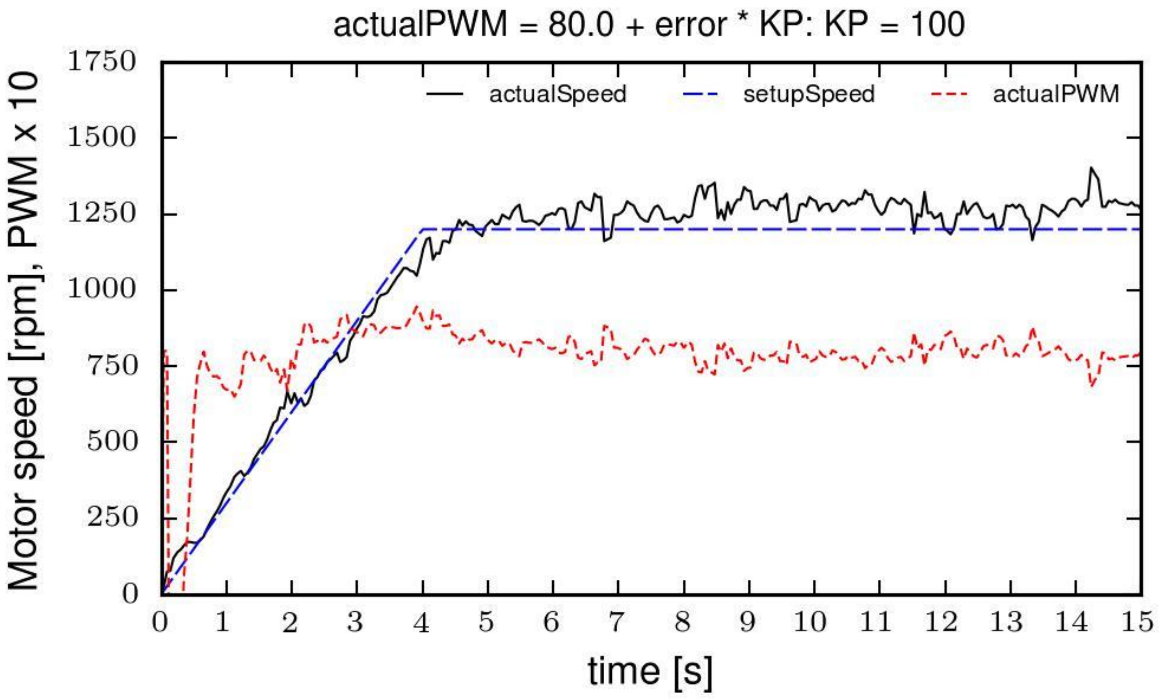


Figure 6-20: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 80% duty cycle, Kp = 100)

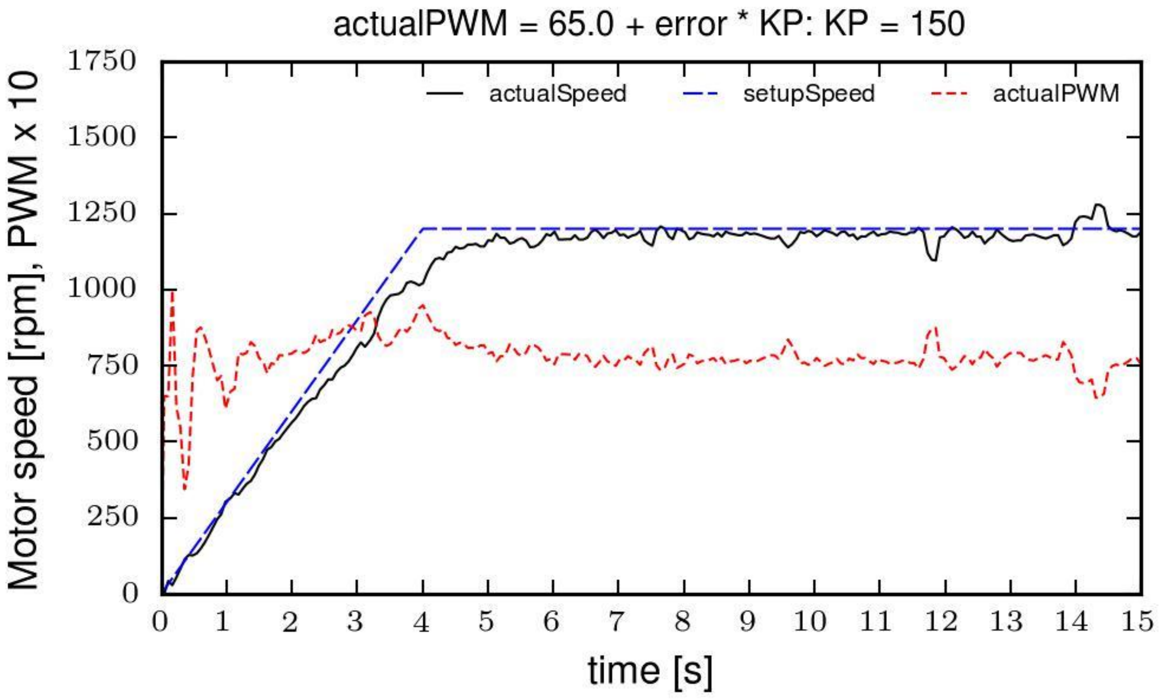


Figure 6-21: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 65% duty cycle, Kp = 150)

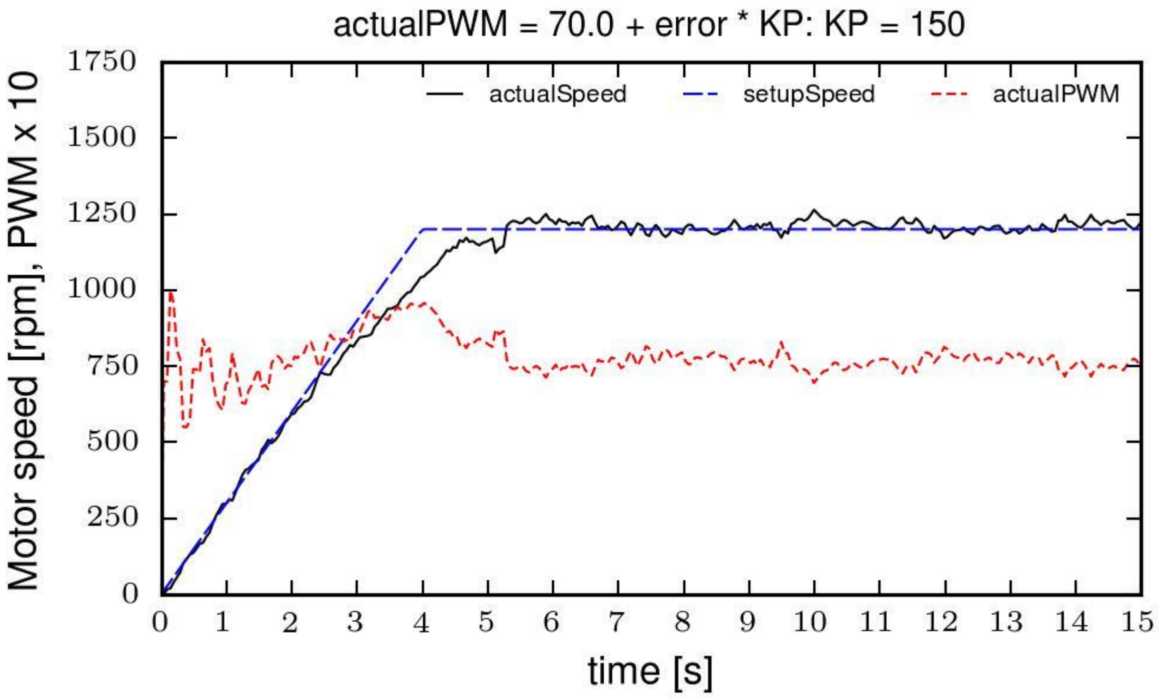


Figure 6-22: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150)

Figure 6-21 depicts at initial value of 65% of duty cycle and Kp=150 instabilty in the control action and the motor response is almost gone. The motor tracks the reference speed perfectly but needs a long time (5 sec.) to reach the reference speed. So, we just increase the initial value to 70% of duty cycle and from Figure 6-22 it is evident that motor tracks the reference speed perfectly with minimum oscillation. The response starts lagging behind the reference speed after 3 sec. and reaches it again at 5 sec. mark. It is evident from the above figures that further increase of initial value or proportionality constant value would increase the system instability and overshoot. So, now we keep the initial value and Kp fixed at 70% of duty cycle and 150 as optimum value and introduce I controller.

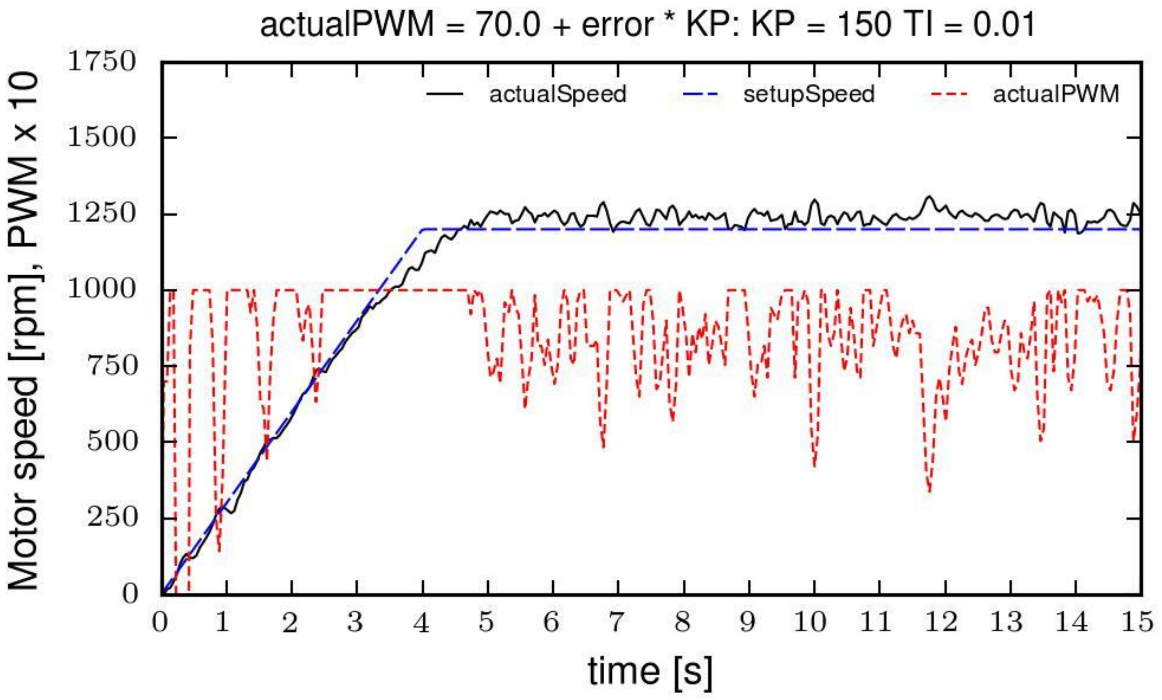


Figure 6-23: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.01)

The introduction of integral control in a control system reduces the error that was present in the 3 to 5 sec. interval for the P controller, with optimum setting. Integral control creates a restoring force that is proportional to the sum of all past errors multiplied by time. Although the addition of integral feedback eliminates the steady-state-error problem, it reduces the overall stability of the system. The problem occurs because integral feedback tends to make the system overshoot, which may lead to oscillations. Figure 6-23 shows motor speed now tracks the reference speed in a better way than Figure 6-22, but if we closely look the control action of these two figure, it is evident that higher value of integral action makes the system more oscillatory here. So, now the Ti is increased to 0.05 and 0.08, which means reducing the integral action. Figure 6-24 and Figure 6-25 depict that reducing the integral action greatly eliminates the oscillation in the control action, hence the motor actual speed.

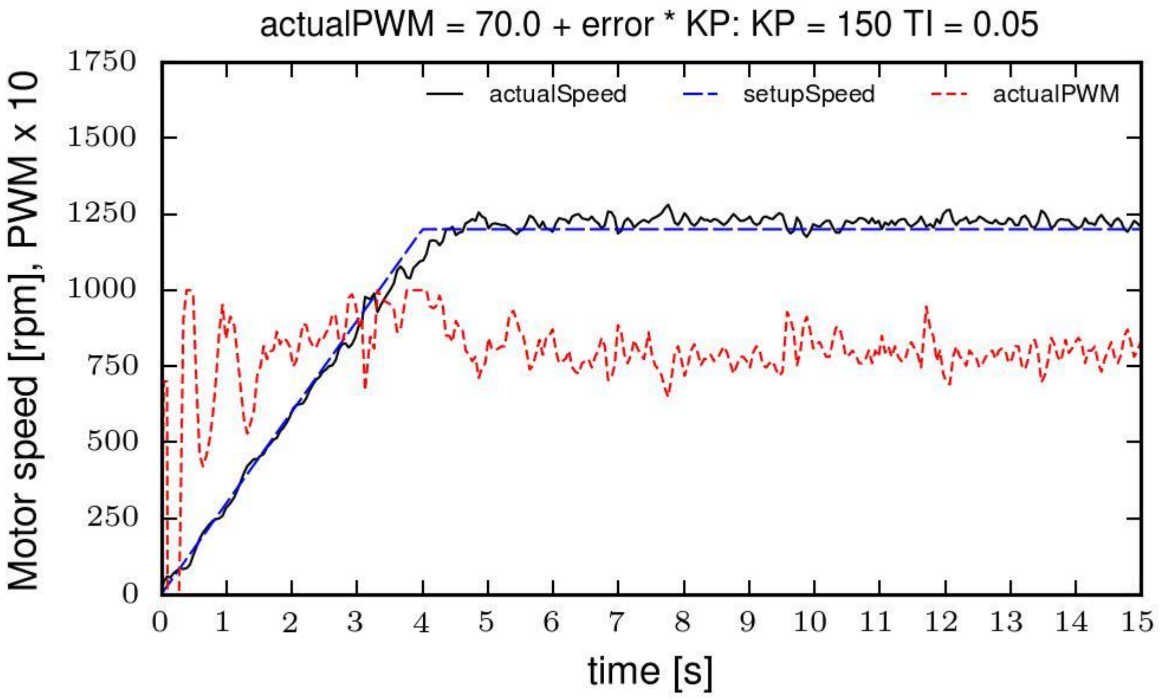


Figure 6-24: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.05)

From Figure 6-24 and Figure 6-25, it is found that with integral time of 50 ms, system's response was better in terms of tracking the reference speed during the initial ramp, but it slightly overshoots after 4 seconds. On the other hand, when the integral time was further increased to 80 ms overshoot and oscillation was reduced greatly, but after 2.5 sec., actual speed started to deviate from the reference speed. So, the Ti was kept fixed at 0.05 and derivative controller was introduced. Derivative control "applies the brakes," slowing the controlled variable just before it reaches its destination, thus eliminates the overshoot problem.

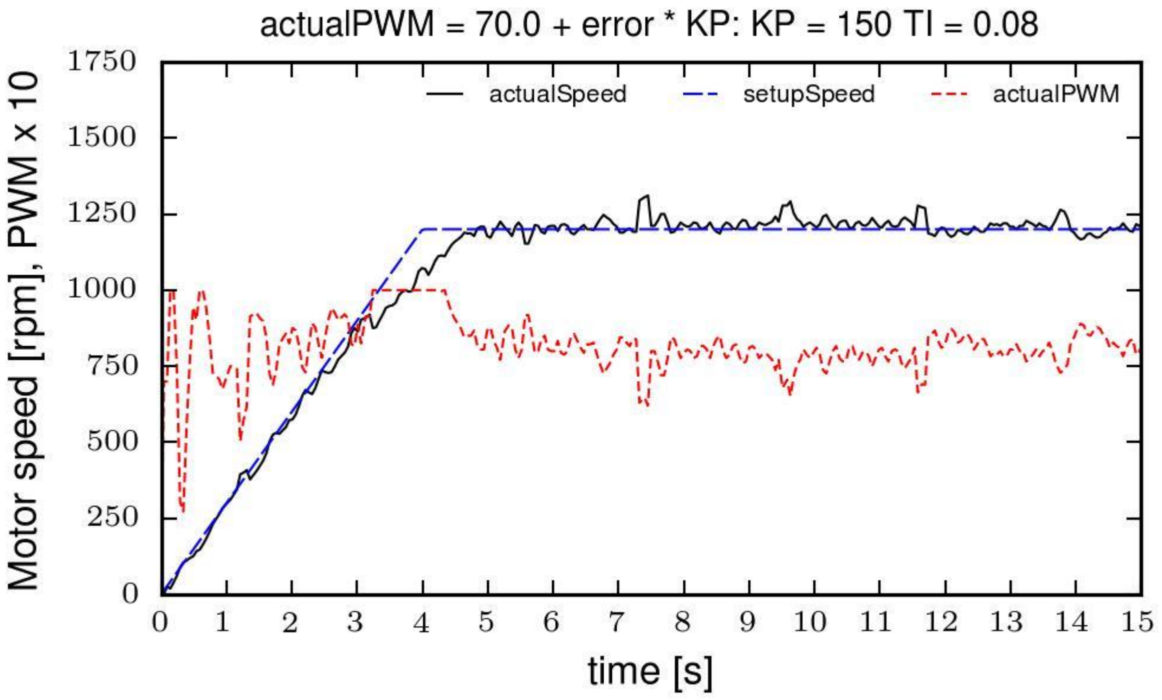


Figure 6-25: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.08)

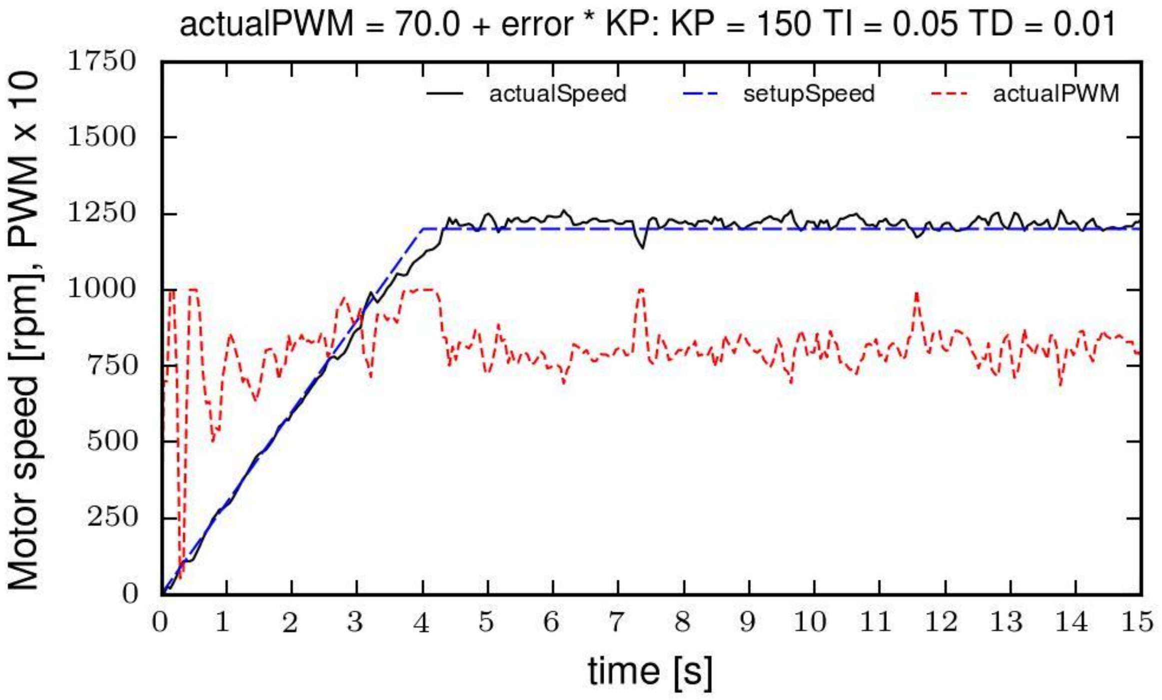


Figure 6-26: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.05, Td=0.01)

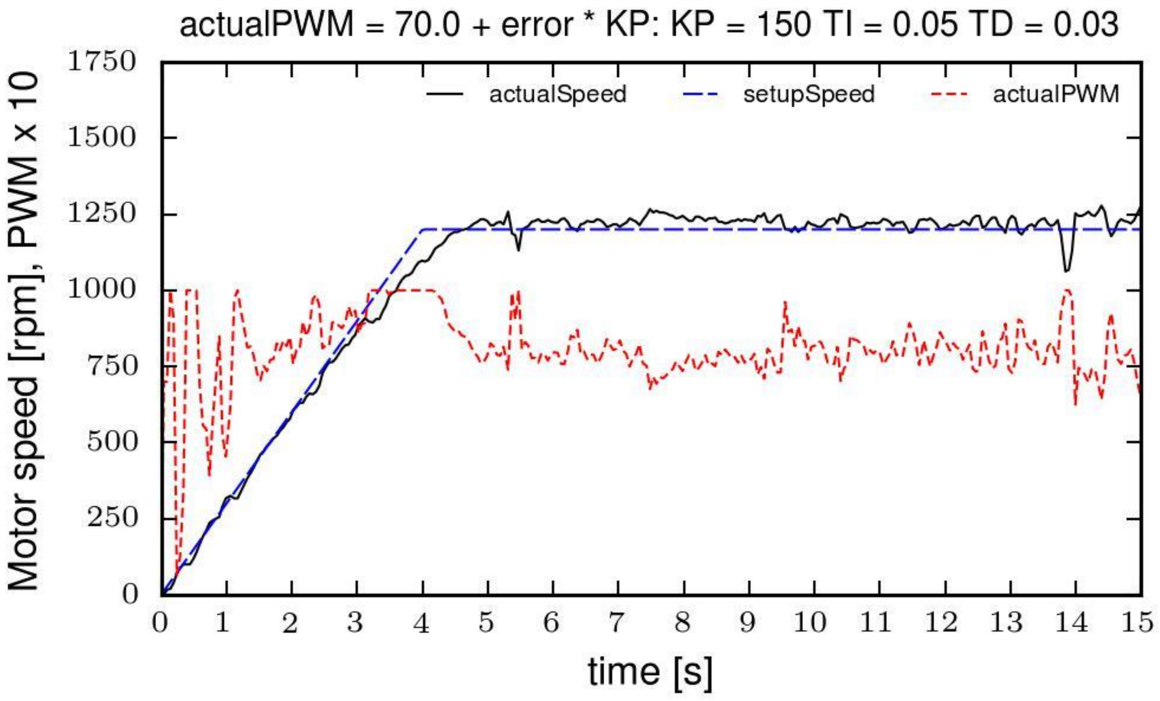


Figure 6-27: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.05, Td=0.03)

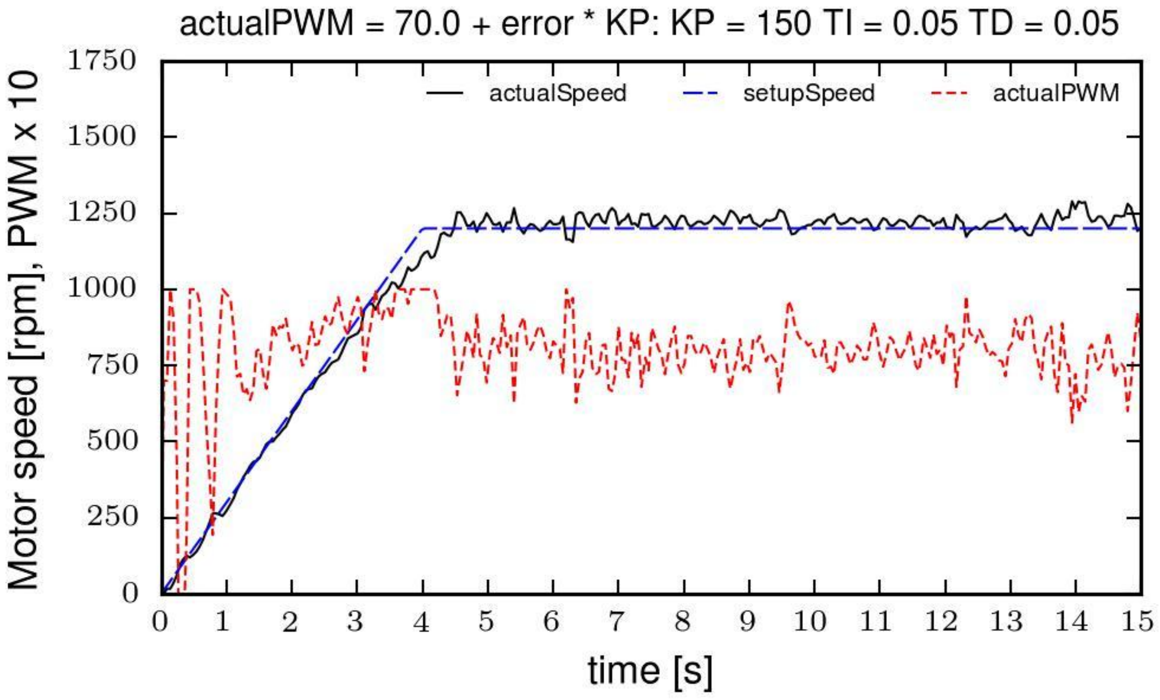


Figure 6-28: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.05, Td=0.05)

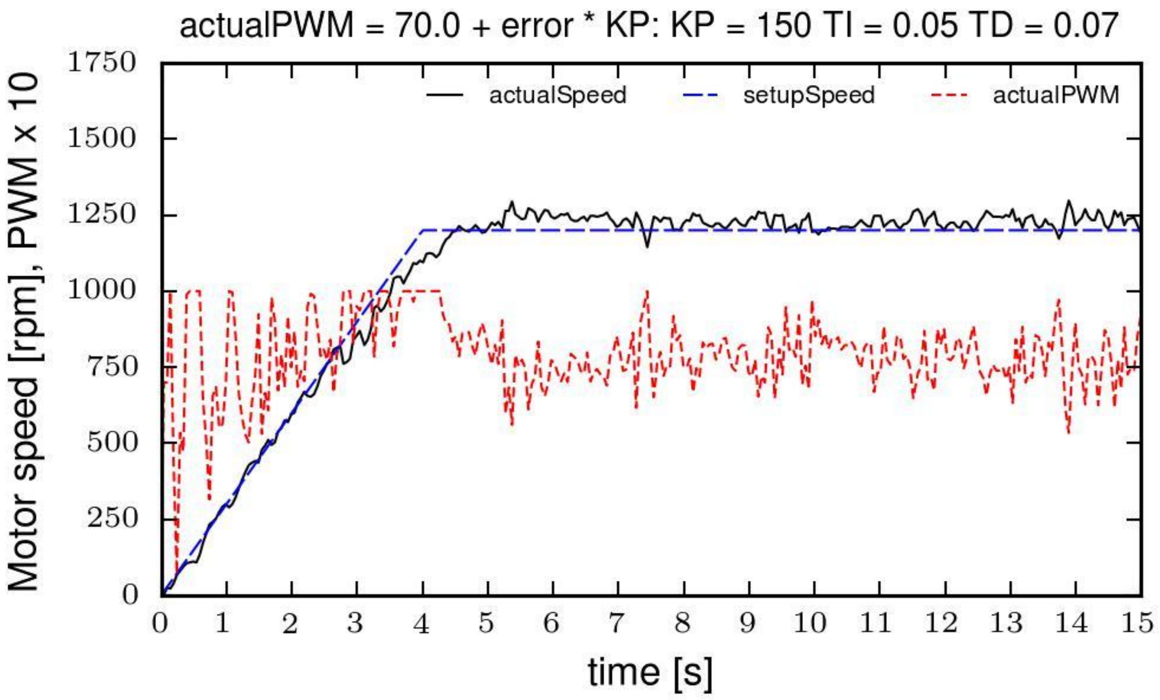


Figure 6-29: Variation of Motor Speed, actualPWM and actualSpeed with Time (Initial Value = 70% duty cycle, Kp = 150, Ti=0.05, Td=0.07)

Comparing Figure 6-24 and Figure 6-26, it is found that derivative action reduces the oscillation in the response. It also reduces the overshoot problem which was the case for PI controller. Though with higher value of T_d, control signal becomes jagged. As very with high derivative action small amount of high frequency noise could cause very large derivatives, which would appear like amplified noise. Observing Figure 6-26 to Figure 6-29, optimum response was found with T_d=0.03. Now, keeping integral time and derivative time fixed at 0.05 and 0.03 respectively, a little fine tuning was done with Kp, which will affect all three controller mode (P, I, and D) equally. These final settings of controller parameters were then tested with all three reference speed profile (setup speed profile-1, 2, 3).

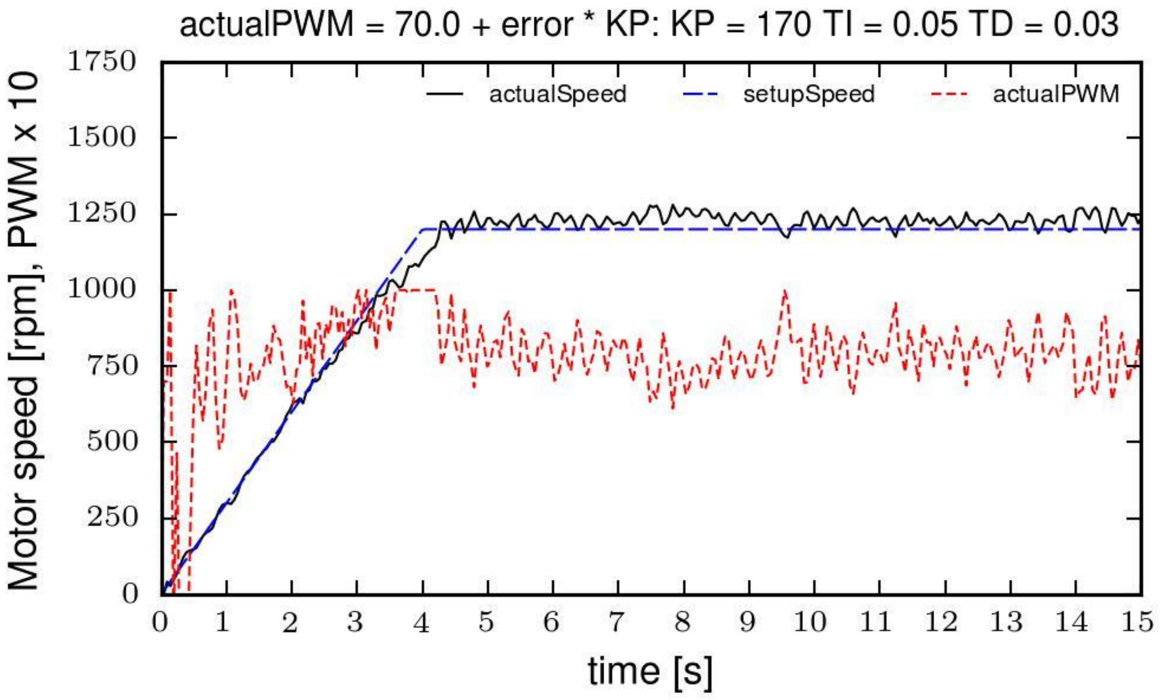


Figure 6-30: Controller response with optimum setting to setup speed profile-1 (medium acceleration)

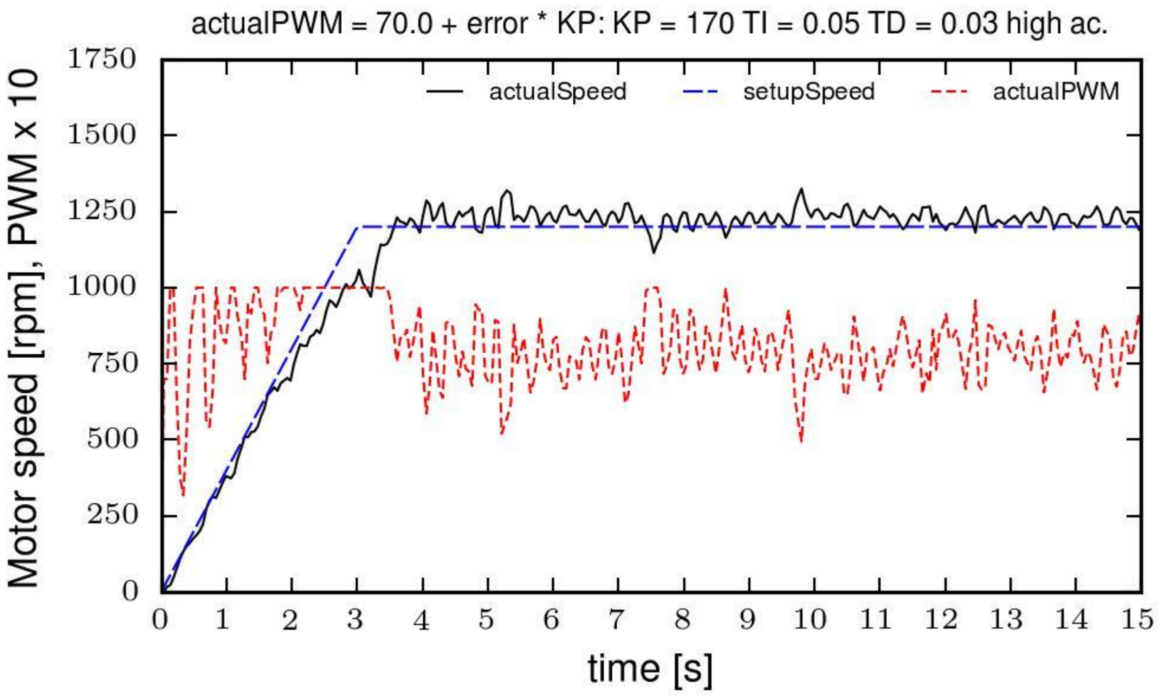


Figure 6-31: Controller response with optimum setting to setup speed profile-2 (high acceleration)

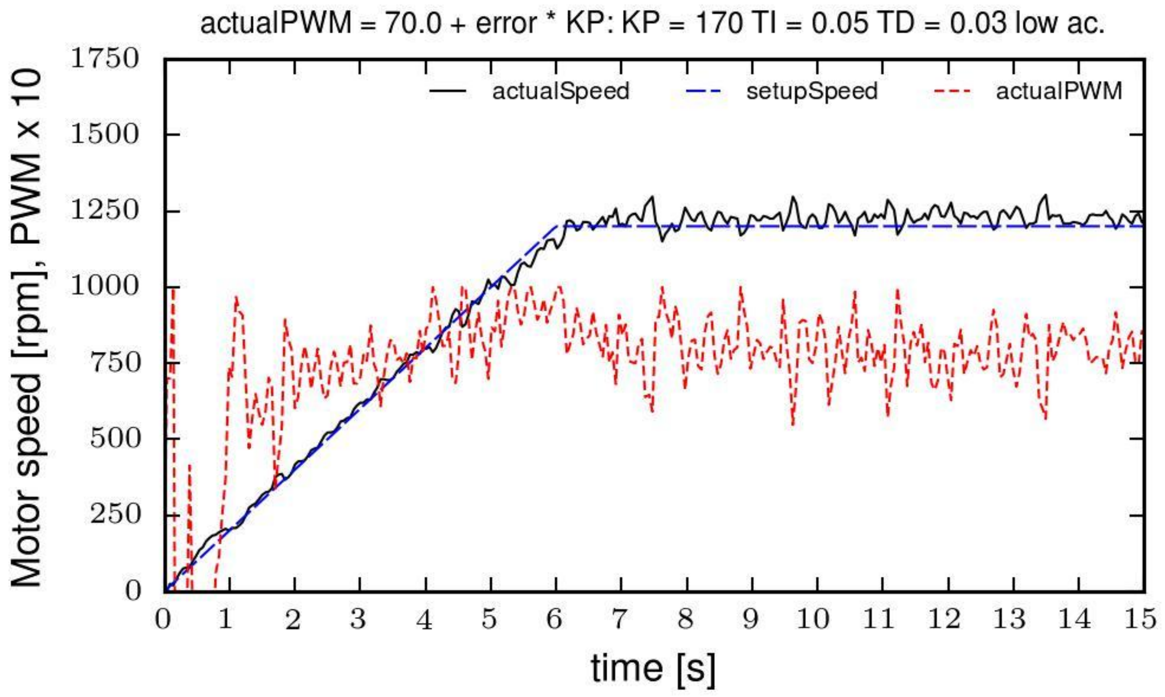


Figure 6-32: Controller response with optimum setting to setup speed profile-3 (low acceleration)

Figure 6-30, Figure 6-31, and Figure 6-32 show that the controller was working fine for each of the three speed profile. For, low acceleration (Figure 6-32), actual speed perfectly matches the reference speed. For medium and high acceleration (Figure 6-30 and Figure 6-31 respectively), there is a slight deviation in the actual speed from the reference speed, just before the ramp reaches its maximum point. Though during this deviation, controller tries to catch the reference speed by providing maximum control action which is 100% of duty cycle. For, medium acceleration this happened for 3.5 to 4 seconds and for high acceleration, 2 to 3.5 seconds. This event can be explained using the Figure 6-33. It shows motor need some finite time to reach a specific speed at different duty cycle. No matter what is the operating condition or the control action is, motor could not be able to reach a certain speed instantaneously.

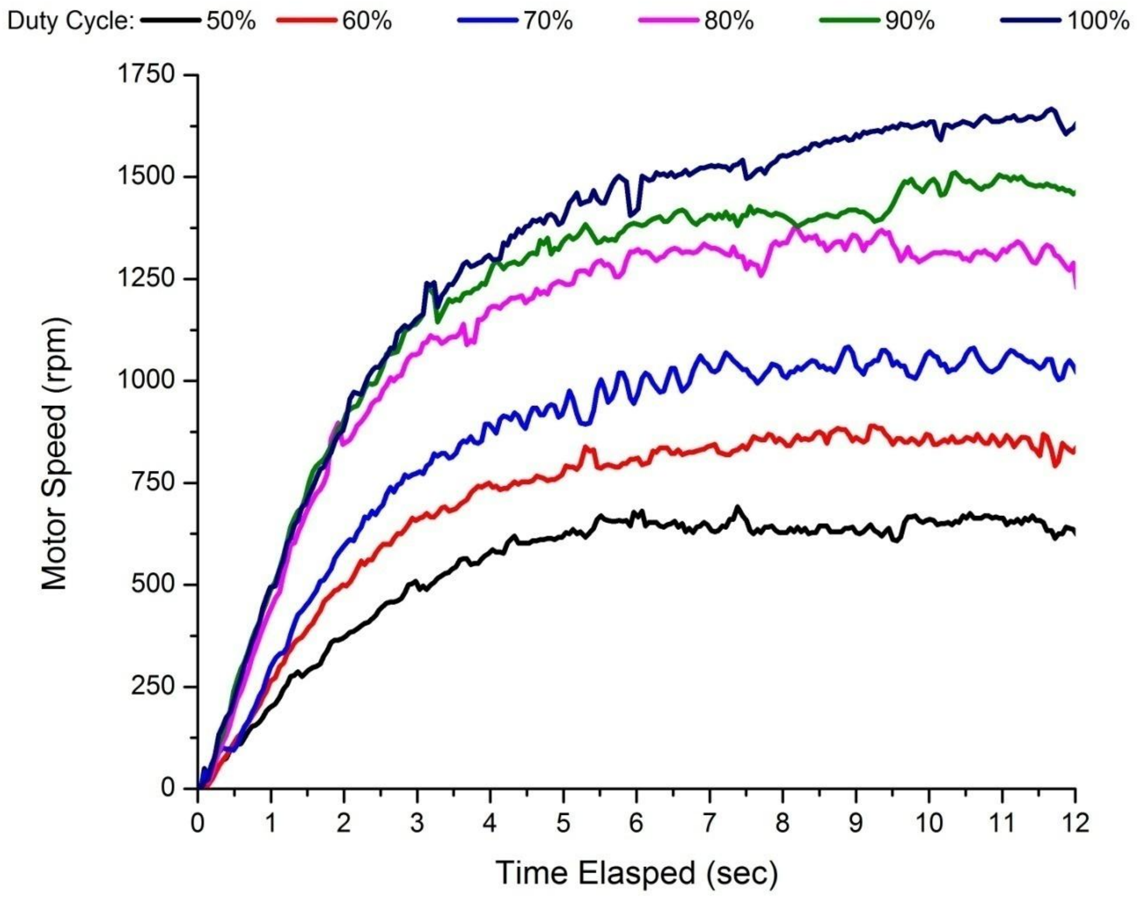


Figure 6-33: Response curve of the motor at different duty cycle

Chapter 7 : CONCLUSION

7.1 Concluding Remarks

This thesis has described the current development of the Servo-motor Control Test Bench at BUET. The test bench has been used to simulate practical environment that motor drive faces during its operation. Different control strategies were studied and applied to find the optimum control for the dc servo-motor while following a predefined velocity profile.

A simulation environment was created in SimApp simulation software to observe the influence of different controller parameters for second order systems. A dc motor model was developed whose characteristics parameter were chosen such that, it's no load behavior closely approximates our practical dc servo-motor. Different controller moods were tested with this model and their parameters were varied to find the optimum response of the model.

The thesis also gave details of the hardware and control electronics used in the test bench. A new generation data acquisition system- LabJack was used and interfaced for the system. Details of the LabJack's interfacing with computer and the test bench were published here. Real time data acquisition was done and monitored for screening the response of the drive motor under varying reference speed profile.

The controller performances on the hardware were very satisfactory when the correct value of controller parameters were used. Though it shows some instabilities arising from the uncertainties associated with the measurement system. Effect of sampling time on this instabilities were studied. It is found that fluctuation in motor's actual speed was greatly reduced with the increase of sampling time.

7.2 Future Work

- Fractional order controllers could be experimented with this setup. Auto tuning is another feature that can achieve greater stability and better response while the setpoint is constantly varying.
- Proper system modeling could be done to find mathematical model of the test bench. That would greatly help in tuning the controller parameters.

- Some vibration was observed during the experiment. So, measures should be taken to minimize this because due to vibration there is a probability of missing some signals from optical encoder. A torque measuring instrument could be implemented on the setup.

REFERENCES

[1] Mellodge, P., (2002), “Feedback Control of Path Following Robotic Car”, M.S. Thesis, Virginia Tech, USA.

[2] Angeles, J., (2007), Fundamentals of Robotic Mechanical System Theory, Methods and Algorithms, Springer.

[3] Kilian, C.,T., (2001), Modern Control Technology: Components and Systems, Delmar Thompson Learning.

[4] Shinskey, F.G., (1994), Feedback Controllers for The Process Industries. NewYork, McGraw-Hill.

[5] Sugisaka, M., &Hazzy, D., (2007), “Development of A Proportional Control Method for A Mobile Robot”. Applied Mathmatics and Computation 186, pp. 74-82.

[6] Ali, M.Y., Hossain, S.G.M., Jamil, H., Haq, M. Z., “ Development of Automatic Guided Vehicles for Industrial Logistic Applications in Developing Countries Using Appropriate Technology”, International Journal of Mechanical and Mechatronics Engineering, Vol:10 No:02.

[7] Blasinger, F., Scheleicher, M., (2003), “Control Engineering”.

[8] Romeo, J. A., Sanchis, R., and Balaguer, P.,(2011) "PI and PID auto-tuning procedure based on simplified single parameter optimization." Journal of Process Control, pp. 840-851.

[9] Sugisaka, M, andHazzy, D., (2007) "Development of a proportional control method for a mobile robot." Applied Mathmatics and Computation, 2007: 74-82.

[10] Ziegler, J.G., Nichols, N.B. (1942), “Optimum Settings for Automatic Controllers”, Trans. ASME 64, pp. 759–768.

[11] Grear,B., Cafuta,P.,Kumin,L., Jezernik,K., (1999), “Robust servo-drive control for dynamic load perturbations”, In Proceedings of the IEEE international symposium on industrial electronics.

[12] H. Panagopoulos, K.J. Astrom, T. Hagglund, Design of PID controllers based on constrained optimization, IEE Proc. Control Theory Appl. 149 (2002) 32–40.

[13] B. Kristiansson, B. Lennartson, Evaluation and simple tuning of PID controllers with high-frequency robustness, J. Proc. Control 16 (2006) 91–102.

[14] T.B. Sekara, M.R. Matausek, Revisiting the Ziegler–Nichols process dynamics characterization, J. Proc. Control 20 (2010) 360–363.

[15] Skogestad, S., “Simple analytic rules for model reduction and PID controller tuning”, Journal of Process Control 13 (2003), pp 291–309.

[16] Quevedo, J., and Escobet, T., “Digital control: Past, present and future of PID control,” in Proc. IFAC Workshop, Terrassa, Spain, Apr. 5, 2000.

[17] I.E.E. Digest, “Getting the best out of PID in machine control,” in Digest IEE PG16 Colloquium (96/287), London, UK, Oct. 24, 1996.

[18] Marsh, P., “Turn on, tune in—Where can the PID controller go next,” New Electron., vol. 31, no. 4, pp. 31–32, 1998.

[19] Ali, S. M. (2011), “ Development and Tuning of a PID Control System for Mobile Robot Drive, M.S. Thesis, Bangladesh University of Engineering and Technology, Bangladesh.

[20] Aziz, Z. (2011), “ Dynamic Response of a Mobile Robot Drive Using a PID control, M.S. Thesis, Bangladesh University of Engineering and Technology, Bangladesh.

[21] O'Dwyer, A.,(2003), Hand Book of PI and PID Controller Tuning Rules. London, Imperial College Press.

[22] Astrom, K.J., Hagglund, T. (1995), “New Tuning Methods for PID Controllers”, European Control Conference, Rome, Italy, pp. 2456–2462.

[23] Li, Y., Feng, W., Tan, K.C., Zhu, X.K., Guan, X., and Ang, K.H., “PIDeasy and automated generation of optimal PID controllers,” in Proc. 3rd Asia-Pacific Conf. Control and Measurement, Dunhuang, P.R. China, 1998, pp. 29–33.

[24] Yun li, kiamheongang, and gregoryc.y. chong, “PID Control System Analysis and Design-Problems, Remedies, And Future Directions”, IEEE Control Systems Magazine, 2006

[25] D.E. Rivera, M. Morari, S. Skogestad, Internal model control. 4. PID controller design, Ind. Eng. Chem. Res. 25 (1) (1986) 252–265.

[26] C.A. Smith, A.B. Corripio, Principles and Practice of Automatic Process Control, John Wiley & Sons, New York, 1985.

[27] B.D. Tyreus, W.L. Luyben, Tuning PI controllers for integrator/ dead time processes, Ind. Eng. Chem. Res. (1992) 2628–2631.

[28] K.J. Astrom, T. Hagglund, PID Controllers: Theory, Design and Tuning, 2nd Edition, Instrument Society of America, Research Triangle Park, 1995.

[29] I.L. Chien, P.S. Fruehauf, Consider IMC tuning to improve controller performance, Chemical Engineering Progress (1990) 33–41.

[30] I.G. Horn, J.R. Arulandu, J. Gombas, J.G. VanAntwerp, R.D. Braatz, Improved filter design in internal model control, Ind. Eng. Chem. Res. 35(10) (1996) 3437–3441

[31] FabrizioPadula, Antonio Visioli, “Tuning rules for optimal PID and fractional-order PID controllers”, Journal of Process Control 21 (2011), pp. 69–81.

[32] Ying Luo, Yang QuanChenc, Chun Yang Wangc,d, You Guo Pi, “Tuning fractional order proportional integral controllers for fractional order systems”, Journal of Process Control 20 (2010), pp. 823–831.

[33] Godfrey, C. O., Mechatronics Principles and Application (2008).

[34] Altmann, W**.,** Practical Process Control for Engineers and Technicians, Elsevier

[35] Johnson, C. D., Process Controll Instrumentation Technology (2007)

[36] K.J. Astrom, T. Hagglund, PID Controllers: Theory, Design and Tuning, 2nd Edition, Instrument Society of America, Research Triangle Park, (1995).

[37] Ogata, K., Modern Control Engineering, Fifth Edition (2010)

[38] LabJack Corporation, “U3-HV User Manual”, USA

[39] Buesser Engineering, “SimApp User Manual v2.6”, Switzerland