DYNAMIC RESPONSE OF A MOBILE ROBOT DRIVE USING A PID CONTROL

by

Zulfiqur Aziz

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

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

December, 2011

Certificate of Approval

The thesis titled, **"Dynamic Response of a Mobile Robot Drive Using a PID Control"** submitted by **Zulfiqur Aziz**, Roll No: **100110005 P**, Session: **October 2001**, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN MECHANICAL ENGINEERING** on 21st **December 2011**.

____

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

Chairman
(Supervisor)

____

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

Member
(Ex-Officio)

____

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

Member

____

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

Member
(External)

Candidate's Declaration

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

Zulfiqur Aziz

Table of Contents

List of Tables

List of Figures

List of Symbols

Symbol	
b	Damping constant
D	Derivative
DC	Direct current
$\pm\Delta x$	Tolerance limit
e	Controller Deviation
emf	Electromotive force
ξ	Damping ratio
I	Integral
J	Inertia load
K_C	Critical Gain
K_D	Derivative Gain
K_I	Integral Gain
K_P	Proportional Gain
KS	Process transfer co-efficient
L	Inductance in Henry
MV	Manipulated Variable
P	Proportional
PMDC	Permanent Magnet DC motor
PT1	First order system
PT2	Second order system
PV	Process Variable
PWM	Pulse Width Modulation
R	Resistance in ohm
SP	Set-point
t	Deviation duration

T_a	Approach time
τ	Motor output torque
T_C	Critical Period
T_d	Derivative Period
T_g	Delay time
θ	Shaft angle
T_I	Integral Period
T_n	Reset Time
T_S	Stabilization time
T_U	response time
V	Volt
V_{emf}	Back emf
w	Set-point
ω_n	Undamped natural frequency
x	Process Variable
XP	Proportional band
X_{PC}	Critical proportional band
X_S	Overshoot
y	Manipulated Variable
y_0	Controller output with no error
z	Disturbance

Acknowledgements

I like to express my sincere gratitude and deep respect to my thesis supervisor Dr. Md. Zahurul Haq, Professor, Department of Mechanical Engineering, BUET for his constant guidance, constructive criticisms, encouragement and careful supervision throughout this research without which this thesis would not come to an end.

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

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

Abstract

To work in hazardous environment the popularity of mobile robot is increasing day by day. DC motors are widely used to drive these mobile robots. To study the dynamic response of a mobile robot the mathematical model of a DC motor was made and simulated by SimApp software. Dynamic response of a second order system and DC motor model were studied and simulated by using step input and P, PI and PID control. From results of simulations and comparison between different controllers only PID controller found most suitable to drive a mobile robot for speed and position control. The simulated result was also verified with previous work and this simulated result found similar to that of previous work. By using PID controller the stability could reach quickly, without overshot and no residual error.

Chapter 1

Introduction

With the advent of new technologies the desire for automation and accuracy is increasing day by day. The rise in popularity of the microcontroller and the drastic reduction in size and cost of integrated circuits in recent years have opened up new arenas for creating intelligent robotic systems [1] for industrial automation and manufacturing. Mobile robots have the capability to move around in their environment and they are not fixed to one physical location. A spying robot is an example of a mobile robot which is capable of moving in a given environment. In contrast, industrial robots usually consist of a jointed arm and gripper assembly that is attached to a fixed surface. Mobile robots are the focus of a great deal of current research and almost every major university has one or more labs that focus on mobile robot research. Mobile robots are also found in industry, military and security environments. Mobile robots may be classified by the environment in which they travel and the device they use to move [2]. In the field of mobile robot control many schemes for stabilization and trajectory tracking problem have been proposed [3]. These methods differ in complexity and flexibility. An indispensable component of the control system is the actuator. The actuator is the first system component to actually move and convert electrical energy in to mechanical motion. The most common type actuator is the electric DC motor. DC motors have speed control capability, which means that speed, torque, and even direction of the rotation can be changed at any time to meet the new condition. There are many ways to drive the DC motor among them Pulse Width Modulation (PWM) to drive the motor and PID (proportional–integral–derivative) control algorithm to generate variable pulse width depending on the error signal is getting popularity.

A PID controller is a generic control loop with feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs. The PID parameters used in the calculation must be tuned according to the nature of that specific system. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element. By tuning these three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation. With this prelude the objectives of this paper are given in next paragraph.

1.1 Objectives

The specific objectives of the work are as follows:

- To analyze system response of the drive system controlled by PID control actions with different values of K_P, K_I, and K_D. Hence optimum values of controlled parameter are obtained for the drive system with fixed and inertial loading.
- To analyze the drive system response with optimal control parameters and the system subjected to variable loading and disturbance.
- To verify the actual controlled system response as compared to desired value.

1.2 Scope of the Thesis

Comprehensive modeling and simulation was carried out to investigate the effects of PID control action on a DC motor. In this paper importance of robot, requirement of control its basic, different controller modes and review of previous works are included in Chapter 2. DC motor basics its governing equation, its drive and tuning of controller are described in Chapter 3. Governing equation of system, modeling of a DC motor and its simulation by SimApp software are covered in Chapter 4. Simulation was carried out for different control parameters to investigate the control system response and the results of the simulation were discussed and compared in Chapter 5. Finally the conclusion, recommendations and scope for future works are described in Chapter 6.

Chapter 2

Review of Mobile Robot Fundamentals and Control Basics

2.1 Importance and Classification of Mobile Robot

Mobile robots have the capability to move around in their environment and they are not fixed to one physical location. A spying robot is an example of a mobile robot which is capable of moving in a given environment. In contrast, industrial robots usually consist of a jointed arm and gripper assembly that is attached to a fixed surface. Mobile robots are found in industry, military and security environments. They also appear as consumer products, for entertainment or to perform certain tasks like work in hazardous or vacuum environment and even in household tasks like gardening. Mobile robots may be classified by the environment in which they travel like land or home, aerial, underwater and polar robots, and also are classified by the device they use to move like legged, wheeled or tracks.

2.2 Mobility of a Mobile Robot

Mobile robots are the focus of great interest of current research and almost every major university has one or more labs that focus on mobile robot research. As mentioned earlier they can move on wheel, on track or walk like human or insects. Electric motors are frequently used for position and speed controls. Motors can be classified into two main categories, they are DC motors and AC motors. In practice most motors used in mobile robot drives or modern control systems are DC motors.

2.3 Requirement of Control

Robots perform tasks in hazardous and extreme environments where six degrees of freedom may exist therefore; speed, position and trajectory tracking are the utmost requirement to perform required tasks. The primary task of control system is to bring these parameters to certain pre-defined values under the influence of disturbances and maintain stability. The fundamental laws of control engineering are applied to all control circuits, irrespective of the different forms of equipment and instruments are involved.

A control system consists of different component; these components are interconnected to each other and form a system configuration to provide a desired system response. Control involves comparison of actual value with desired value or set point. Any deviation from the set point leads to change to the energy supply. Therefore, control is the method to force parameters in the environment to have specific value.

Most of the control systems in industry are operated by Proportional-Integrative-Derivative (PID) controllers still nowadays [4-6]. Besides, in most of the situations a PID can perform reasonably well and is indeed all what is required. Consequently, given the widespread use of PID compensators, it is clear that even a small percentage of improvement over the already existing tuning rules could have a considerable impact. On the other hand, the highly developed modern control state of the art theory is usually associated with more sophisticated controllers than those of PID type. This fact helps to understand why many PID tunings are still carried out by means of empirical or numerical optimization-based techniques [7-9].

The need for literature review was felt to know the PID tuning procedures, advantages, disadvantages, applicability, scope for future works and many more on PID. From literature review it reveals that the close-loop ZN tuning is one of the most popular methods to obtain reasonably good initial settings for PID controllers [10-12]. However, ZNPIDs are found to perform quite satisfactorily for first-order processes, but they usually fail to provide acceptable performance for high-order and nonlinear processes [12-14] due to large overshoots and poor load regulation. To overcome such drawbacks several tuning schemes are proposed [12-20]. The Internal Model Control (IMC) based approach for controller design proposed in [21] has gained widespread acceptance in the control industry because a clear tradeoff between the closed loop performance and robustness is achieved by means of a single tuning parameter namely the IMC filter time constant.

In the past four decades, there are numerous papers dealing with the tuning of PID controllers. A natural question arises, how can the PID settings obtained by different methods are compared? A simple answer is to use step responses of the closed-loop systems and compare the overshoot, rise time and settling time. However, these time domain performance measures do not address directly another important factor of closed-loop system robustness. It is a well-known fact that models used for controller tuning or design are often inaccurate, so a PID setting based on optimization assuming an accurate model will generally not be guaranteed to be robust. For a fair comparison of different PID settings, both time domain performance and frequency domain robustness should be considered. With this literature review ZN tuning method was used to simulate and determine the values of K_P, T_i and T_d to tune DC motor model by using a PID control.

2.4 Basic Laws of Control

The control loop consists of, connection from the output of the process to the input of the controller, and from controller the output goes as input to the process, thus forming a control loop. In closed loop control system the output of the process (e.g. speed, temperature etc) is monitored continuously and compared with set point. As a result of these comparisons, the controller takes control action to bring the process variable equal to the set point. The resulting action takes place in a closed circuit and this is known as the control loop as shown in Fig. 2.1.

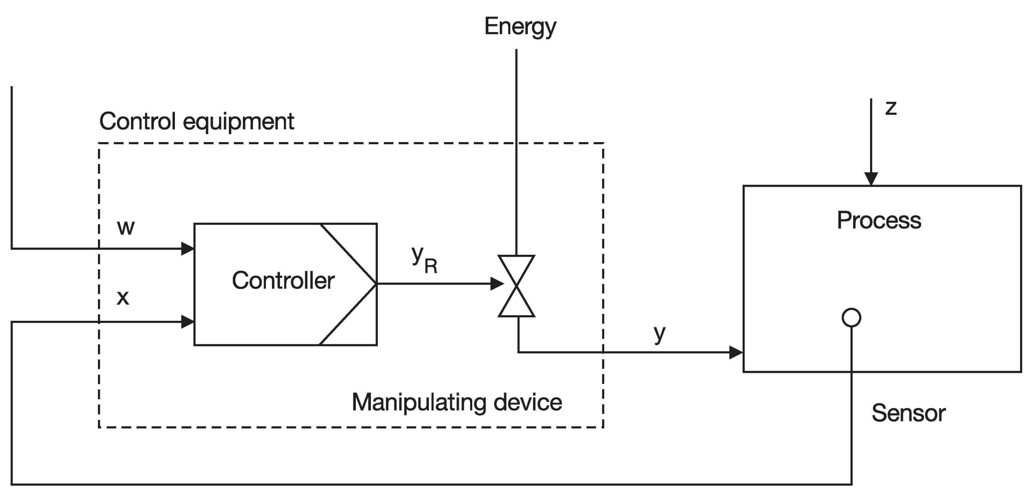

Fig. 2.1. The closed loop control block diagram.

The action of the controller on the control loop is characteristic by, overshoot (X_S), approach time (T_a), stabilizing time (T_S) and agreed tolerance limit is ±Δx. Overshoot is the value of process variable above the set point, approach time is the time taken for the process variable to reach the new set point for the first time. A controller is said to be stabilized when operating with a constant manipulating variable and the process variable is moving within the agreed tolerance limit ±Δx.

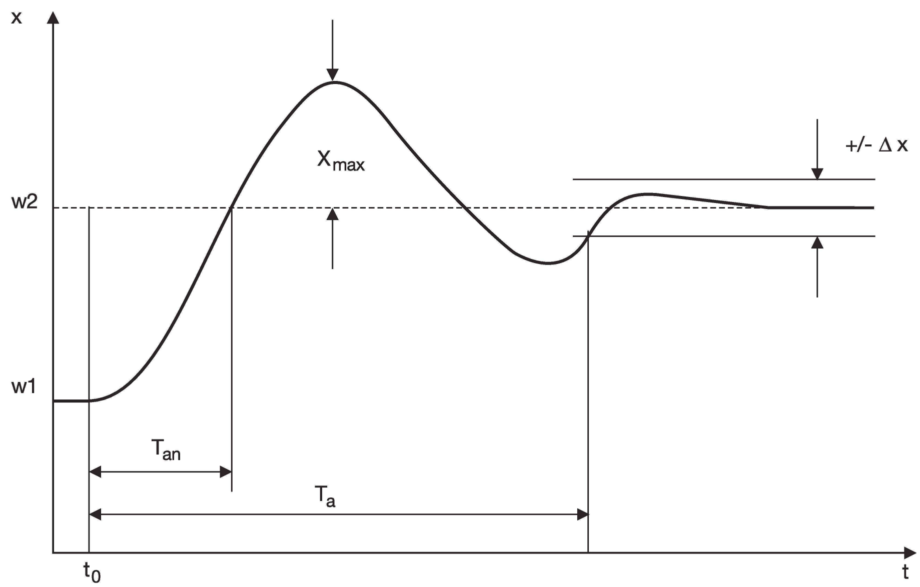

Fig. 2.2. Criteria for control action.

In control theory the output of the process is the first element for any control action. Therefore, knowledge on process, its static and dynamic behaviors are prerequisite for design of controllers and its optimization or tuning.

The process covers a number of elements within a system which has to be controlled in accordance with the application. The static behavior of a process is described by comparing the values of input and output signals at the end of stabilization or settling time. When considering the static behavior of control loop elements, it is of no importance how a particular control element reaches its final state. The static characteristic of a system can be linear and non-linear.

The dynamic response of a process describes the variation in output signal of the transfer element (the process) when input signal varies with time. In theory, it is possible for the output variable to change immediately and to the same extent as the input variable changes. However in many cases, the system responds with certain delays and study on dynamic response has became very important for optimization or tuning.

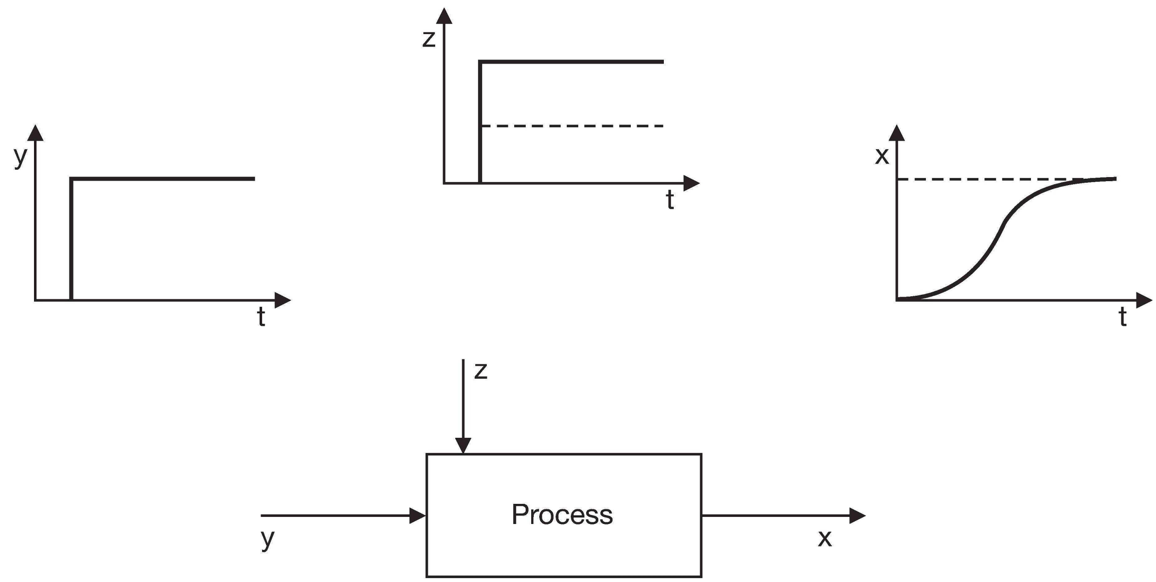

Fig. 2.3. Step response of a process with self-limitation.

The simplest way of establishing behaviors of output signal is to record the variation of process value (x), with time, after a step change to the input of the process, and recording the variation of process value (x) with time. The dynamic behavior of process can be clearly predicted from step response. When there is step change in manipulating variable (e.g. voltage), the process variable (e.g. speed) changes with time and this behavior can be studied by step response.

2.5 Controller Modes

Many control methods are already in use, they differ in complexity and flexibility. Depending on the application, there is a need to have several types of tuning method. The process variable goes to the controller in the form of electrical signal. After processing the process variable the controller output goes to the manipulating device. Controllers are usually made to produce one or a combination of control actions. Different controller modes are:

(1) On-Off or two-position control
(2) Proportional control
(3) Proportional plus integral (PI) control
(4) Proportional plus derivative (PD) control, and
(5) Proportional plus integral plus derivative (PID) control

ON-OFF Control Mode

The controller output has only two states of control usually fully ON and fully OFF. One state is used when the process variable is anywhere above the set point, and the other state is used when the process variable is anywhere below the set point. The process variable must cross the set point to change the output state therefore, the process variable will continually cycling. The peak-to-peak variation and the period of the cycling are mainly dependent on the process response and characteristics. For example Time-Temperature response of an ON-OFF controller is shown in Fig. 2.4.

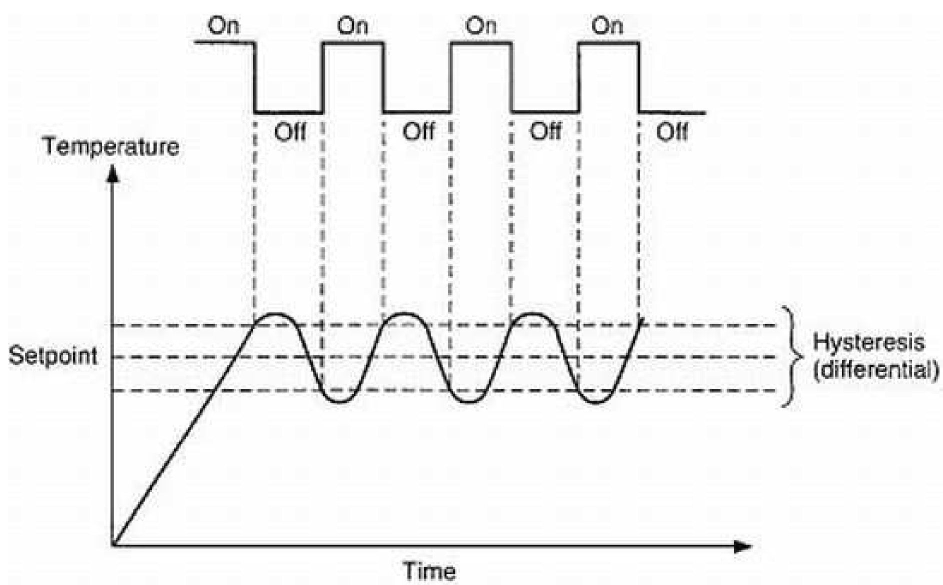

Fig. 2.4. Time-temperature diagram for ON-OFF controller.

Generally ON-OFF control mode is best adapted to large-scale systems with relatively slow process rate. Room heating and air-conditioning system use this ON-OFF control, where the air volume is large and the heating/ cooling effect is relatively slow. The main advantage of an ON-OFF controller is that it is very simple in nature. The problem of ON-OFF control is the cycling, overshoot and undershoot of the process variable. In Fig. 2.4 for Time-Temperature diagram of ON-OFF controller a dead band is a range of temperature where no control action will occur and hysteresis is the behavior of the system for the same value of temperature depending on whether the temperature is increasing or decreasing.

Continuous Controller Modes

Continuous controllers offer advantages for certain control systems since their action on the process can be continuously modified to meet demands imposed by process events. Continuous controllers are used to operate actuators, such as regulating valves, which need a continuous signal. Different continuous controller modes are described below:

Proportional Control Mode

In a P controller the control deviation is calculated from the difference between process value and set point. This control deviation is then amplified to give the manipulating variable and this manipulating variable operates actuator.

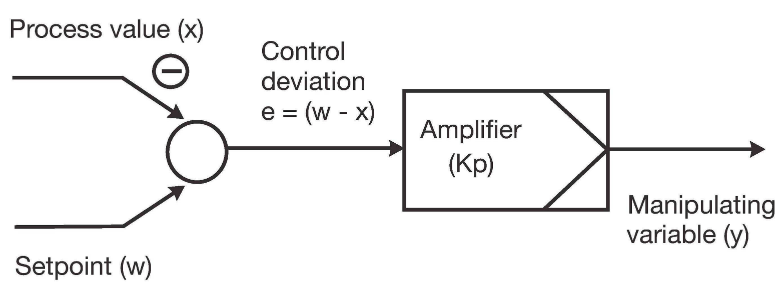

Fig. 2.5. Operating principal of P controller.

The control deviation signal is amplified because it is too small and cannot be used directly as manipulating variable. The gain (K_P) of a P controller must be adjustable, so that the controller can be matched to the process. In a P controller the manipulating variable changes to the same extent as the deviation, though amplified by a factor, P controller can be represented mathematically as given by

$$y = K_p \times (w\text{-}x) = K_p \times e \tag{2.1}$$

The factor K_P is called the proportionality factor and corresponds to the control amplification or gain. The unit of the proportionality factor, K_P, is the unit of the manipulating variable divided by the unit of the process variable. A high proportional gain results in a large change in the output for given change in error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive (or sensitive) controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

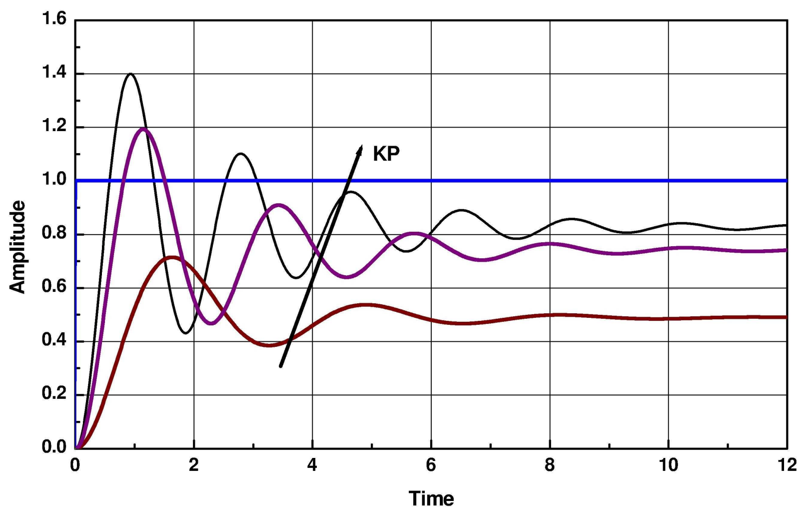

Fig. 2.6. P Controller with various gains.

The proportional band is the range of error for which there is a linear relationship between the error and controller output. Gain is the relationship between control output and range of error. For example an electric furnace is to be controlled by a digital controller. The manipulating variable is to be 100% for a deviation of 10°C then the proportional band XP is 10. The relationship between proportional band and gain or proportionality factor of the controller is given by

$$XP = 1/K_P \tag{2.2}$$

An important characteristic of the proportional control mode is that it produces a permanent residual error when load change occurs. This error is referred to as offset. Incase of ON-OFF control the process variable must cross the set point to change the output state therefore, the process variable will continually cycling. This cycling is removed in proportional controller but it always has permanent control deviation or offset. It can be minimized by a larger proportionality factor (K_P), which also reduces the proportional band or by setting the set point above the true desired value, or corrected by adding integration term.

Integral control mode

An I controller integrates the deviation signals applied to its input over a period of time. The longer there is a deviation on the controller, the larger the manipulating variable of the integral controller becomes. How quickly the controller builds up its manipulating variable depends firstly on the setting of the I controller, and secondly on the magnitude of the deviation. The manipulating variable changes as long as there is a deviation. Thus, over a period of time, even small deviations can change the manipulating variable to such an extent that the process variable corresponds to the required set point. In principle, an I controller can fully stabilize after a sufficiently long period of time. Therefore, I controller removes the offset produces by P controller. The step response of the I controller shows the course of the manipulating variable over time, following a step change in the control difference

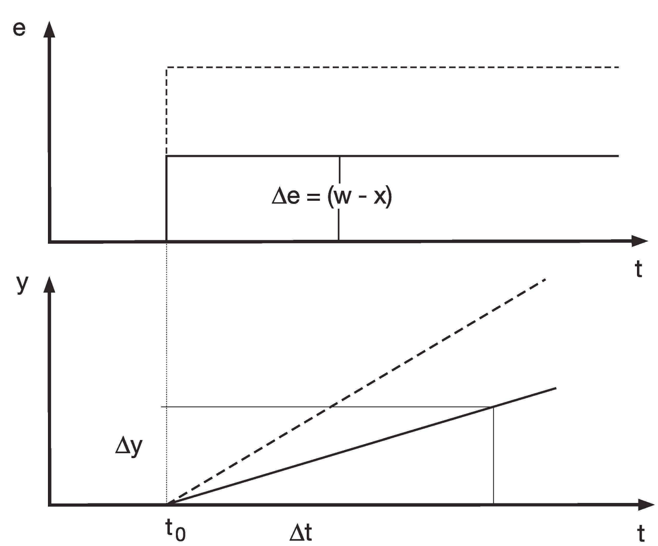

Fig. 2.7. Step response of an I controller.

For a constant deviation Δe, the contribution of an I controller is given by

$$\Delta y = \Delta e \times t \times \frac{1}{T_1} \tag{2.3}$$

Where T_I ≡ The integral time of the I controller

t ≡ The duration of the deviation

It is clear that the change in manipulating variable y is proportional not only to the change in process variable, but also to the duration of the deviation, t. If control deviation varies with time then the contribution for I controller is given by

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

For processes with long time constants, the I component must be set very low, so that the process variable does not oscillate. With this small I component, the I controller works much slowly. For this reason, it is not particularly suitable for processes with long time constants (e.g. temperature control systems) rather I controller is generally used for pressure regulation.

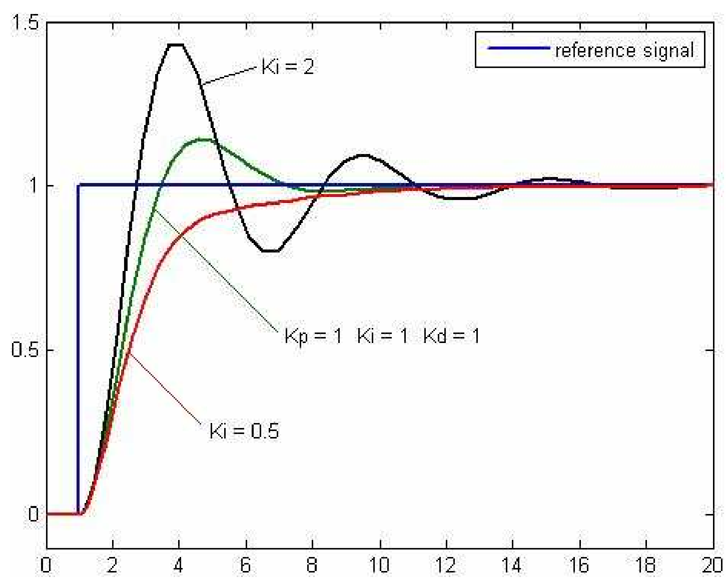

Fig. 2.8. Integral Controller with various gains (Source: Wikimedia)

Derivative control mode

Derivative control action responds to the rate of change of process error. The rate of change of process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_D. The contribution from derivative component is given by $K_D \times \frac{de(t)}{dt}$. The derivative term slows the rate of change of controller output and this effect is most noticeable close to the controller set point. Hence the D controller is used to reduce the magnitude of the overshoot produced by the I component and improved the combine controller stability. Proportional term works on the present error, integral term works on past error and derivative term works on the present and past error to forecast a future response of the system. A notable characteristic in this type of control is when the error is constant, or at the maximum limit, the effect of D component is minimal.

Composite control mode

Real processes are complex and individual controller mode alone cannot fully control the process. By using different controller together we can utilize the advantages of individual controller to eliminate some limitations possess by other controllers. In control engineering PI, PD or PID types composite control modes are widely used in different fields.

Proportional-Integral Control mode (PI)

In case of I controller, it takes a relatively long time before the controller has built up its manipulating variable. Conversely, P controller responds immediately, but is unable to completely remove the offset error. The combination of P controller with I controller forms the PI controller. This PI controller is therefore, much faster than P or I controller.

We can obtain the step response of a PI controller simply by superimposing the responses of P and I controller. As shown in Fig. 2.9 for step input the manipulating variable of a PI controller is projected back and intersected at S with the time axis, then the time, T_n, corresponds to reset time. The reset time is a measure of the extent to which the duration of the control deviation affects the control function. A long reset time means that the I component has little influence, and vice versa. The analytical expression for PI controller can be found by combining the expression of P and I controller as given by

$$y(t) = k_p e + k_p k_I \int_0^t e\,dt + y_I(0) \quad (2.5)$$

The response curve of a PI controller is given in Fig. 2.10. Due to the combine contribution of P and I controller this PI controller is much faster than P or I controller.

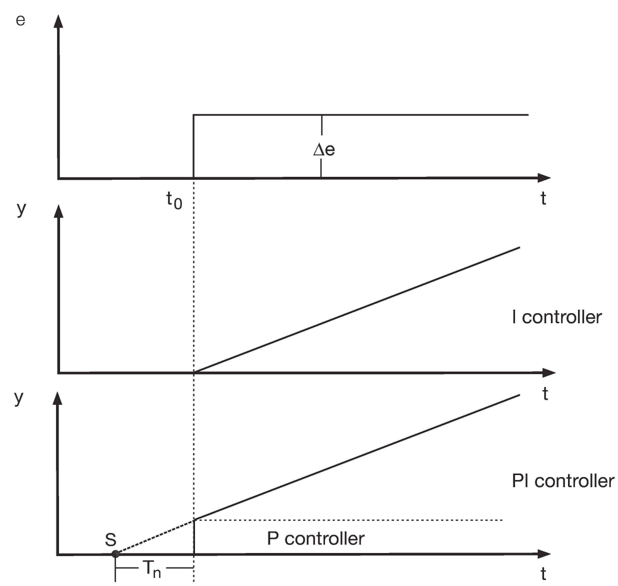

Fig. 2.9. Step response of a PI controller.

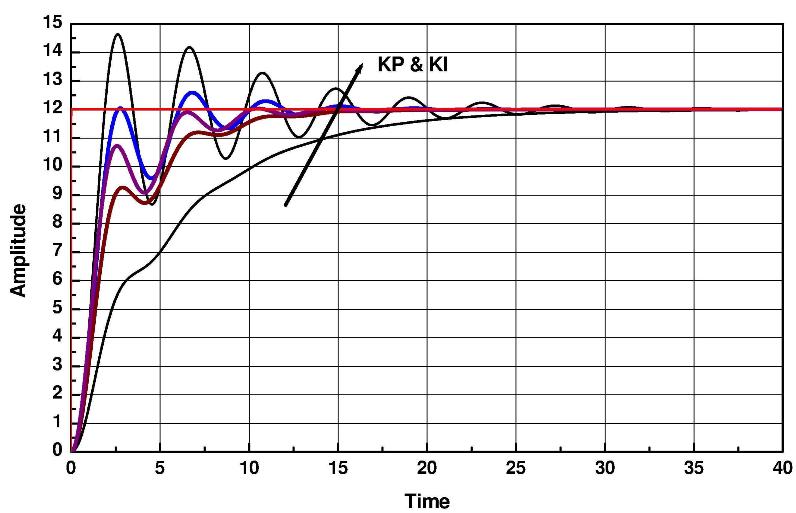

Fig. 2.10. PI Controller with various gains.

Proportional-Derivative Control mode (PD)

For a large disturbance in a manually controlled control loop, the operator will initially try large adjustment to the actuator to cushion the effect of the disturbance. Then he will quickly reduce the adjustment, so that the new equilibrium of the control loop can be approached gradually. A controller which responds in a similar way like this operator is the PD controller: it consists of P and D component with a derivative action. This D component responds not to the magnitude or duration of the control deviation, but to the rate of change of the process variable. The PD controller cannot eliminate the off-set of proportional controllers. The analytical expression for this mode is found from a combination of P and D controller is given by

$$y(t) = k_p e + k_p k_D \frac{de}{dt} + y_t(0) \tag{2.6}$$

PD controller can handle processes quickly for load changes as long as the load change exists and the off-set error is within acceptable range. It is important to note that the effect of derivative action in moving the controller output is related to the rate of change of the error.

Three-mode controller (PID)

The PID controller brings together the best characteristics of the P, I and D controllers. The P component responds with a suitable manipulating variable when a deviation occurs. The D component counteracts when there is a change in the process variable, and increases the stability of the control loop. Finally permanent deviation of a process is removed by I component. The block diagram of a PID controller is given in Fig.2.11.

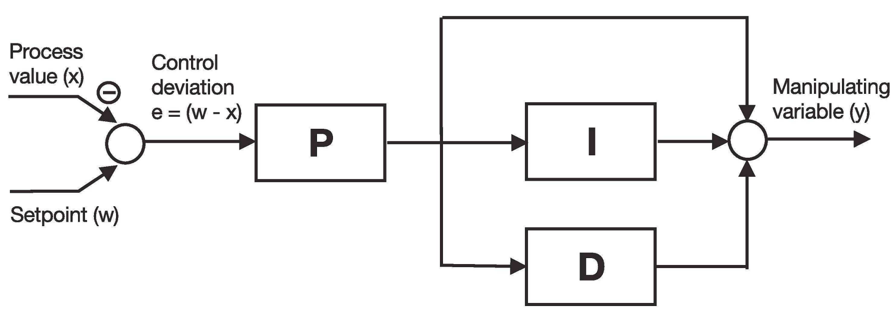

Fig. 2.11. Block diagram of a PID controller.

An ideal PID controller obeys the controller equation as given in Eq. 2.7.

$$y(t) = k_p e + k_p k_I \int_0^t edt + k_p k_D \frac{de}{dt} + y_t(0) \tag{2.7}$$

With PID controller, the XP, T_n, T_d parameters are adjusted for the P,I and D action. These three components can be seen separately for step response to a PID controller as shown in Fig. 2.12.

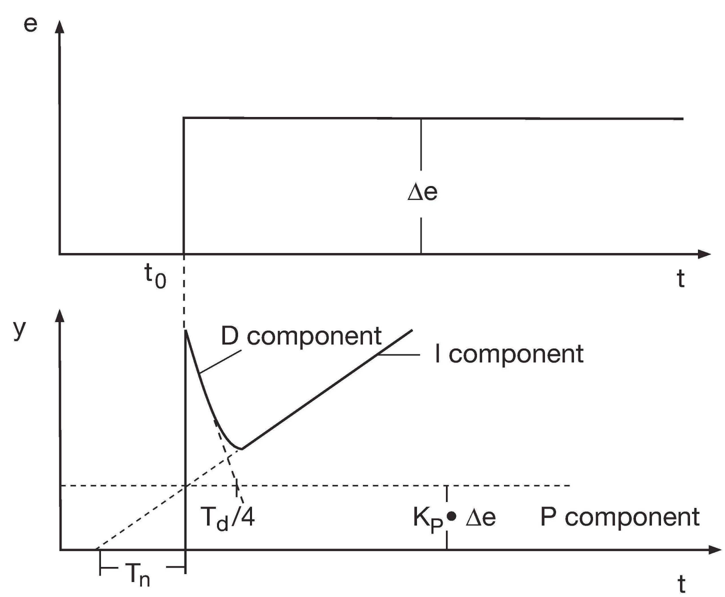

Fig. 2.12. Step response of a PID controller.

The individual parameters ($K_{P,}$ $T_{n,}$ and T_d) have different effects on the individual components.

Larger XP, means smaller K_P and smaller P component.

Larger T_n, means reduced I component.

Larger T_d, means increased D component.

For PID controller the control action is not only influenced by the adjustment of T_n and T_d parameters but also by proportional gain. If the proportional gain of a PID controller is doubled then the controller will doubled the proportional action and, at the same time I and D components will also be increased to double.

Chapter 3

Control of DC Motor Servo-Motors

3.1 Fundamentals of a DC Motor

Actuators are the main component to convert the electrical energy into mechanical energy. DC motors can have permanent or non permanent magnet. In the case of a non permanent DC motor field coil is used to make temporary magnet. DC motors with field coil are classified as series, shunt and compound wound motor. They can also be brush or brushless type. In reality AC motors are more difficult for position control and their characteristics are quite nonlinear which makes the analytical task more difficult. The stepper motor is a digital version of DC motor and is driven by alternating current square waves generated by solid state circuitry. Comparing AC motor, stepper motor and DC motor characteristics finally DC motor is found suitable for control applications in Mobile Robot Drive.

When current is passed through coil placed inside a magnetic field, then force acting on the coil is shown in Fig. 3.1 and force, *F*, is given by

$$F = NBiL \tag{3.1}$$

where N ≡ Nos of coils

B ≡ Density of magnetic flux

i ≡ Current

L ≡ Length of coil

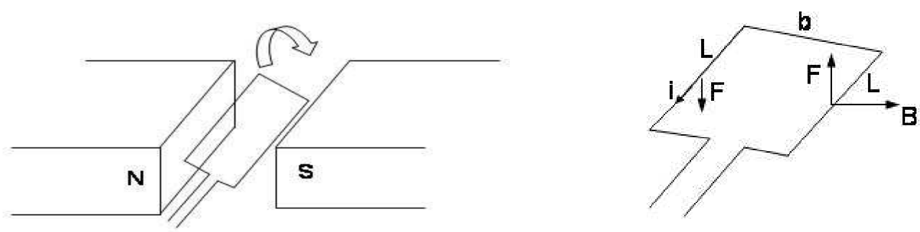

Fig. 3.1. Force on armature coil in a magnetic field.

The torque is given by

$$T = b \times NBiL = K_i \times i \quad (3.2)$$

where b ≡ Diameter of armature coils

K_i ≡ Torque constant

Since armature rotates in magnetic field, an electromagnetic induction will occur and back emf will be induced, which is proportional to the rate at which flux is cut by coil, for higher rotational speed this back emf will be higher. Therefore, the actual current passing through the motor coil will be less at higher speed and torque will be less at higher speed.

3.2 Driving of a DC Motor

In DC motor current is passed through the armature coil and speed depends on this current. The resistance of armature coil is constant i.e. current can be varied by changing the voltage. Therefore, motor speed can be varied by controlling the voltage applied to the armature. In most cases fixed voltage sources are used and variable voltage is obtained by an electronic circuit. Motor converts the electrical energy into mechanical energy (torque) not the speed. In general it is safe to say that increased voltage will result in increased speed. To drive the motor an interface circuit is required to convert the low level control signal from the controller into a signal strong enough to run the motor. The direction rotation can be reversed by regulating the armature or field supply current and this can easily be achieved by using a H-bridge. As shown in Fig. 3.2 when switch A and D are connected then motor rotates clockwise and when switch B and C are connected then motor rotates anticlockwise.

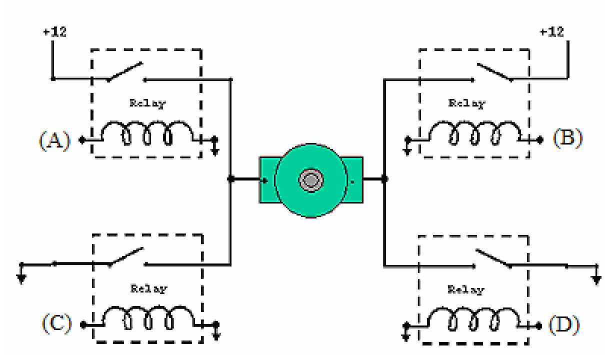

Fig. 3.2. H-bridge circuit for DC motor direction control.

The motor drive can be analogue or digital type. For analog drive, a linear power amplifier amplifies the drive signal from the controller and gives the motor a "Strengthened" analog voltage. For digital control power is supplied to the motor in the form of DC pulses of a fixed voltage. PWM is an entirely different approach to control the torque and speed of a DC motor. Here power is supplied to the motor in a square wavelike signal of constant magnitude but varying pulse width or duty cycle.

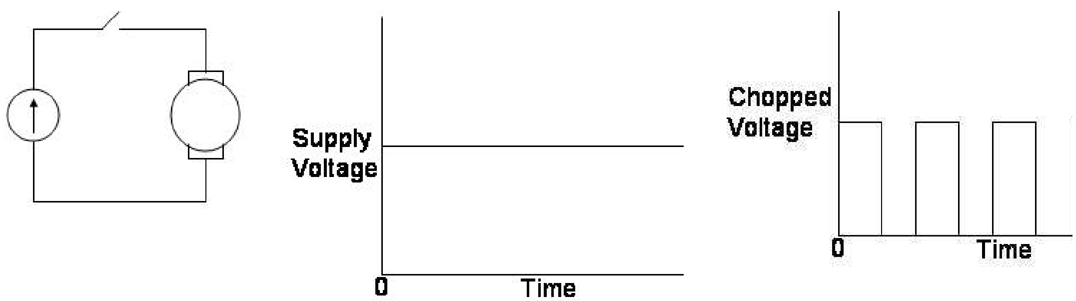

Fig. 3.3. Basic concept of PWM generation.

Duty cycle refers to the percentage of the time the pulse is high or ON per cycle. By changing (modulating) the width of the pulse applied to the DC motor we can increase or decrease the amount of power provided to motor, thereby increasing or decreasing the motor speed. Here the voltage has fixed amplitude but variable duty cycle. The wider the pulse the motor speed will be higher.

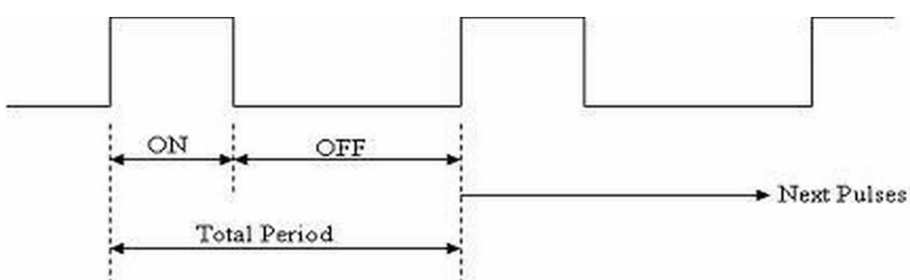

Fig. 3.4. Duty Cycle for a DC source.

To control the speed of a DC motor the input voltage is required to be controlled. This input voltage can be manipulated by the introduction of a suitable controller. This controller cannot control the speed alone unless it is tuned properly.

3.3 Reviews on Tuning

Tuning means the adjustment of the controller to a given process. The control parameters (XP, T_n and T_d) are selected such that optimum control action of the control loop is achieved, under the given operating conditions. This optimum operating condition can be rapid attainment of the set point with a small overshoot or longer stabilization time with no overshoot. Therefore, dynamic behavior of the process i.e. variation of the process value, with time, after a step change to the input of the process is an important criterion for controllability and tuning. The step response of a process can be characterized by:

- Delay time, T_U, and
- Response time, T_g

To determine the transfer function, delay and response time, the course of the process value is recorded and a tangent is drawn to the curve at the point of inflection, and T_U and T_g are determined as shown in Fig. 3.5.

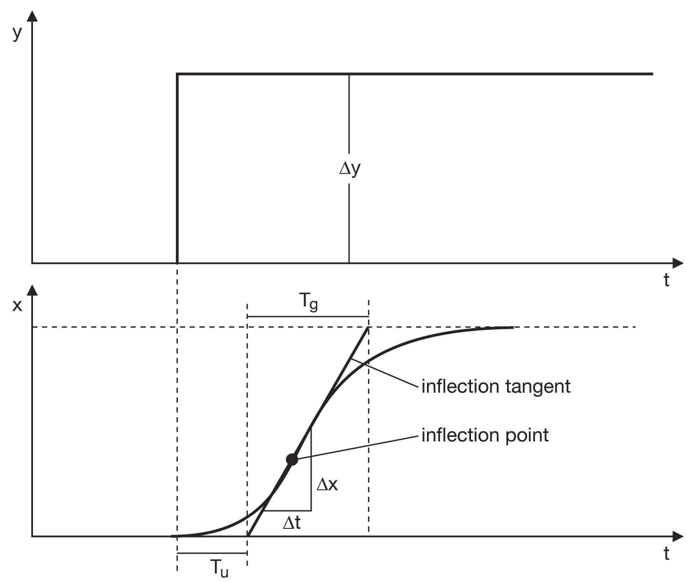

Fig. 3.5. Determining the delay time and response time.

The ratio of delay time to response time gives information about the character of the process and its controllability:

- $\frac{T_g}{T_U}$ more than 10: Process easy to control.

- $\frac{T_g}{T_U}$ between 10 and 3: Process can be controlled.
- $\frac{T_g}{T_U}$ less than 3: Process difficult to control.

A low T_g/T_U ratio corresponds to a steep gradient on the graph, representing a higher order process which is difficult to control because of its tendency to overshoot. For tuning apart from the mathematical derivation of the process parameters and controller data derived from them, various empirical methods are also widely in practice. One method consists of periodically changing the manipulating variable and investigating how the process variable follows these changes. Such test methods are very expensive, involve increased mathematically treatment, and not suitable for practical use. Other controller settings are based on empirical values, obtained from lengthy investigations. Ziegler and Nichols method and that of Chien, Hrones and Reswick formulated method for such controller setting are widely used.

Adjustment by the oscillation method

This method was devised by Ziegler and Nichols, and is mainly applied to processes with short dead times and with a ratio T_g/T_U greater than 3. The control parameters are adjusted until the process variable performs periodic oscillations about the set point. The procedure is used in processes that can actually be made unstable and where an overshoot does not cause danger. The process variable is made to oscillate by initially reducing the controller gain to its minimum value. The controller operates as pure P controller by keeping I and D component switched off. Then the proportional band XP is reduced until the process variable performs undamped oscillations of constant amplitude. From test we can find the critical proportional band, X_{PC}, and the oscillation time, T_C.

The advantage of this process is that the control parameters can be studied under operational conditions. There is no need to open the control loop and recorded data can easily be evaluated. The disadvantage of this method is that it can only be used on processes which can be made unstable.

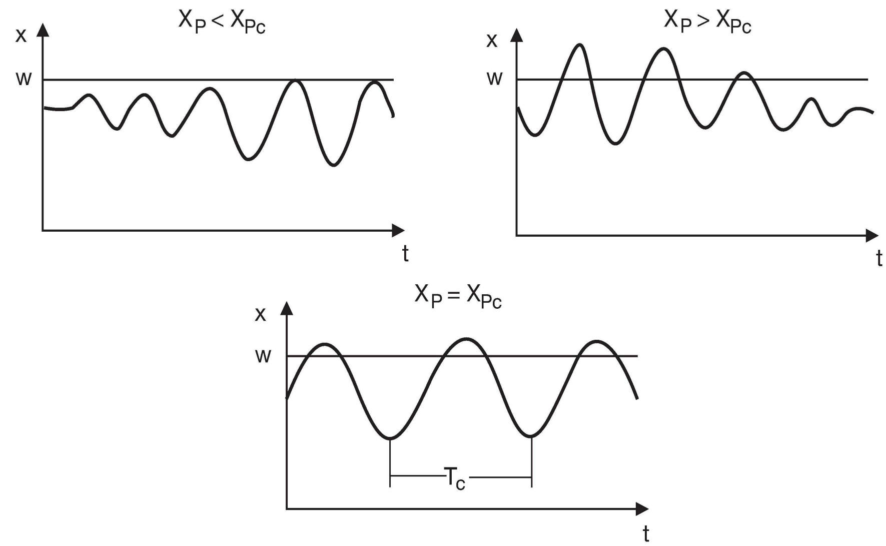

Fig. 3.6. Adjustment by oscillation method by Ziegler and Nichols.

The controller can then be set to values as mentioned in Table 3.1

Table 3.1: Controller Parameter for Oscillation Method by Ziegler and Nichols [23]

Control Mode	**Parameter**
P	$XP = X_{PC}/0.5$
PI	$XP = X_{PC}/0.45$
	$T_n = 0.85 \times T_C$
PID	$XP = X_{PC}/0.6$
	$T_n = 0.5 \times T_C$
	$T_d = 0.12 \times T_C$

Adjustment according to the transfer function or process step response

This method is suitable for processes which cannot be made to oscillate. However, it does require opening the control loop. If possible, the step change in manipulating variable should be made when the process variable is close to the set point. In practice, the graph very rarely shows a very clear point of inflection. Hence, drawing the tangent at the point of inflection can lead to errors in determining the values of T_U and T_g which may or may not be significant. This method is still very useful for first impression of the controller settings. Table 3.2 is used to find out values for different control mode.

$$KS = \text{(Change in process variable)/(Change in manipulating variable)} = \frac{\Delta x}{\Delta y} \qquad (3.3)$$

Where KS ≡ Process transfer co-efficient

T_U ≡ Delay time

T_g ≡ Response time

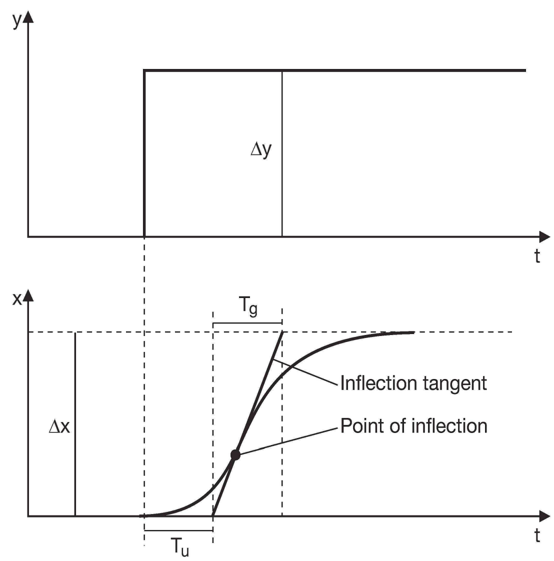

Fig. 3.7. Adjustment according to the step response.

Table 3.2: Controller Parameter for Adjustment According to the Step Response [23]

Control Mode	**Set point**	**Disturbance**
P	$XP = 3.3 \times KS \times \frac{T_g}{T_U} \times 100\%$	$XP = 3.3 \times KS \times \frac{T_g}{T_U} \times 100\%$
PI	$XP = 2.86 \times KS \times \frac{T_g}{T_U} \times 100\%$	$XP = 1.66 \times KS \times \frac{T_g}{T_U} \times 100\%$
	$T_n = 1.2 \times T_g$	$T_n = 2.4 \times T_U$
PID	$XP = 1.66 \times KS \times \frac{T_g}{T_U} \times 100\%$	$XP = 1.05 \times KS \times \frac{T_g}{T_U} \times 100\%$
	$T_n = 1 \times T_g$	$T_n = 2.4 \times T_U$
	$T_d = 0.5 \times T_U$	$T_d = 0.42 \times T_U$

Adjust according to the rate of rise

When it is difficult to determine the response time, T_g, the maximum rate of rise, V_{max}, is evaluated. To do this the manipulating variable is suddenly set to 100% and the output of the process is observed. Very often, the manipulating variable can only be set to either 0 or 100%. The rate of change will increase continuously until the point of inflection is reached. At the point of inflection the process variable approaches its final value slowly as shown in Fig.3.8. In this method, it is necessary to wait until the point of inflection is reached, and then set the manipulating variable back to 0% again. The tangent at the point of inflection is then drawn and V_{max} is determined from the gradient triangle. Delay time is also determined for the step response and the controller settings can be implemented by using the Table 3.3.

Table 3.3: Controller Parameter for Adjustment According to the Rate of Rise [23]

Control Mode	**Set point**	**Disturbance**
P	$XP = V_{max} \times T_U$	$XP = V_{max} \times T_U \times (yH/\Delta y)$
PI	$XP = 1.2 \times V_{max} \times T_U$	$XP = 1.2 \times V_{max} \times T_U \times (yH/\Delta y)$
	$T_n = 3.3 \times T_g$	$T_n = 3.3 \times T_U$
PID	$XP = 0.83 \times V_{max} \times T_U$	$XP = 0.83 \times V_{max} \times T_U \times (yH/\Delta y)$
	$T_n = 2 \times T_U$	$T_n = 2 \times T_U$
	$T_d = 0.5 \times T_U$	$T_d = 0.5 \times T_U$

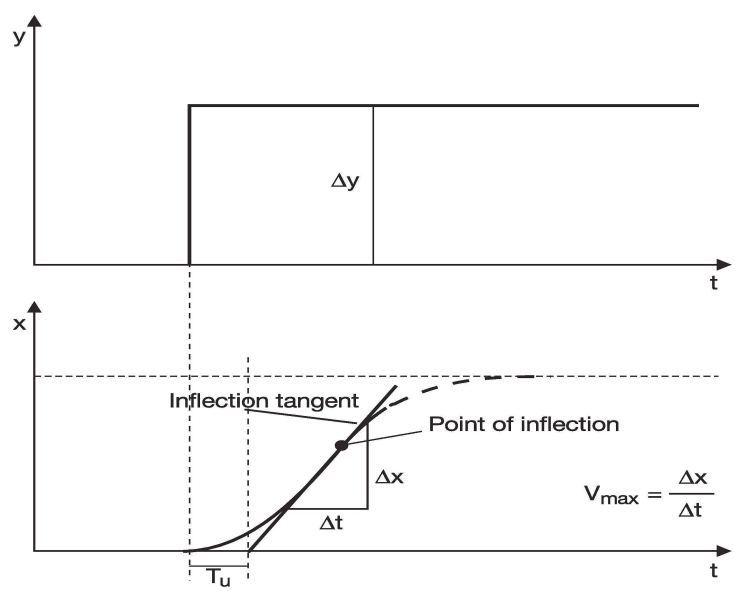

Fig. 3.8. Adjustment according to the rate of rise.

Adjustment without knowledge of the process

Occasionally a controller has to be adjusted to a process where it is simply not possible to record a transfer or to open the control loop. If the process is not very slow, the controller is initially set to a pure P structure with the largest proportional band possible, so that pure P action is achieved.

The set point is set close to the future operating point and the process value indicator on the controller is observed. After some time the process value will stabilize at a value quite some way from the set point. This is because of the low gain through the large proportional band setting, XP is now reduced, as a result of which the deviation from the set point becomes smaller and smaller. As XP is further reduced, a point is eventually reached at which the process value starts to oscillate periodically. There is no point in reducing XP any further, as it would only increase the amplitude of these oscillations. These oscillations are not usually symmetrically about the set point; their mean value is either above or below the set point because the permanent deviation produces by P controller.

The proportional band is now increased once again, until the process value becomes stable. Next, the I component is added (PI structure), and the reset time, T_n, is reduced step by step. The process variable slowly approaches the set point, as a result of I component. Reducing T_n still further accelerates the approach time, but also leads to oscillation. We now apply to disturbance to the process, either by changing the set point or an external disturbance. The approach to the new set point is monitored. If the process value overshoots, we have to increase T_n. If the approach is very slow, the reset time settling can be reduced still further. The D component can be activated next, if required (PID structure), by settling T_d to a value of approximately T_n /4.5. This procedure is widely used in practical and suitable for simple processes.

Chapter 4

Modeling and Simulation of DC Motor

So far we have covered the importance of mobile robot, its application, how it moves, the importance of DC motor as drive, how DC motor works, why motor control is required, basic literature on control, different types of controller, how to tune these controllers. In this Chapter we will carry out mathematical modeling and simulation of a DC motor. To make the mathematical model and simulate that the knowledge on process, system and transfer function are prerequisite

4.1 Transfer function and process

The process is the element of a system which has to be controlled in accordance to the application. Normally the process covers a number of elements within a system. The system can be represented by first, second or higher order differential equation. A transfer function is the transformation of differential equations to the frequency domain. This transformation is done by Laplace transform. The Laplace transform is often interpreted as a transformation from the time-domain, in which inputs and outputs are functions of time, to the frequency-domain, where the same inputs and outputs are functions of complex angular frequency, in radians per unit time. The transfer function is often described in two parts, the static part and dynamic part. The static transfer function describes the input-output relationship when the input is not changing with time. The dynamic transfer function describes the input-output relationship when there is time variation of the input. The dynamic transfer function is also called time response.

4.2 First-Order System

A first order system is described by

$$a_1 \frac{dx}{dt} + a_0 x = f_1(t) \tag{4.1}$$

Here, *x* represents the measured and controlled output variable and *f(t)* the input function. The equation is often rearranged to the form

$$\frac{a_1}{a_0}\frac{dx}{dt} + x = f_2(t) \tag{4.2}$$

$$\tau\frac{dx}{dt} + x = f_2(t) \tag{4.3}$$

τ is designated the time constant of the process.

This equation is linear as long as *f(t)* is not a function of x, thus it can be transformed into a transfer function

$$\tau sX + X = F_2 \tag{4.4}$$

$$X = \frac{1}{\tau x + 1}F_2 \tag{4.5}$$

This type of transfer function is known as a first order lag with a steady state gain of 1.0. The first order has one internal energy store. Therefore, it cannot follow a step change instantaneously.

4.3 Second-Order Systems

Second order differential equations are applicable throughout engineering and science and are used in applications where there is either electrical or mechanical oscillatory motion to be modeled. A second order differential equation takes the general form:

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n\frac{dy}{dt} + \omega_n^2 y = \frac{F(t)}{m} = \frac{\omega_n^2}{k}F(t) \tag{4.6}$$

This is called the characteristic equation of a second order differential equation. The $1/k$ is termed as static sensitivity of the system. The time response of the system is thus a function of the natural frequency of the system, the damping ratio, and the forcing function, *F(t)*.

In order to solve equations of this nature, one can split out the differential terms into variables and solve for them in the same manner they would the roots of a polynomial equation. In a second order differential equation, there are three different cases of roots that can occur, and the root types are conditional based upon the square root value in the numerator. These cases represent something called damping, a phenomenon which reduces the amplitude of waveforms in a system.

4.4 Modeling and Simulation of a DC Motor

Consider a DC motor, whose electric circuit of the armature and the block diagram of the rotor are shown in Fig. 4.1.

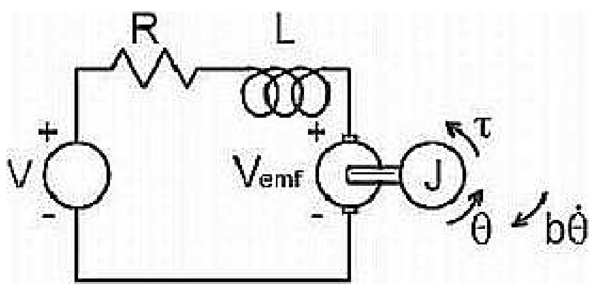

Fig. 4.1. Block diagram of a motor model.

Where	V	: input voltage (V)
	R	: nominal resistance (Ω)
	L	: nominal inductance (H)
	J	: inertial load (kg*m^2/s^2)
	V_{emf}	: back emf voltage (V)
	b	: damping constant (Nms)
	τ	: motor output torque (Nm)
	θ	: motor shaft angle (rad)

The motor torque, τ, is linearly proportional the current through armature and expressed as

$$\tau = K_t i \tag{4.7}$$

As the armature rotates, a voltage difference, proportional to the rotational velocity of the armature shaft, is created. This voltage is called the *back emf* voltage.

$$V_{emf} = K_e\dot{\theta} \tag{4.8}$$

By using Newton's and Kirchoff's Laws, the following equations can be derived

$$J\ddot{\theta} + b\dot{\theta} = Ki \tag{4.9}$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \tag{4.10}$$

By making use of the Laplace transform, the above two equations can be rewritten as

$$s(Js + b)\theta(s) = KI(s) \tag{4.11}$$

$$(Ls + R)I(s) = V - Ks\,\theta(s) \tag{4.12}$$

The last step is to combine Eq. 4.11 and 4.12, and canceling the I(s) term. This yields the transfer function for the DC motor. It is easy to see that the transfer function from the input voltage, *V*, to the angular velocity is given by

$$\frac{\dot{\theta}}{V} = \frac{K}{(Js + b)(Ls + R) + K^2} \tag{4.13}$$

By using SimApp for step input the response of the motor transfer function is shown in Fig 4.2. Here the values are taken from previous works mentioned at reference [22].

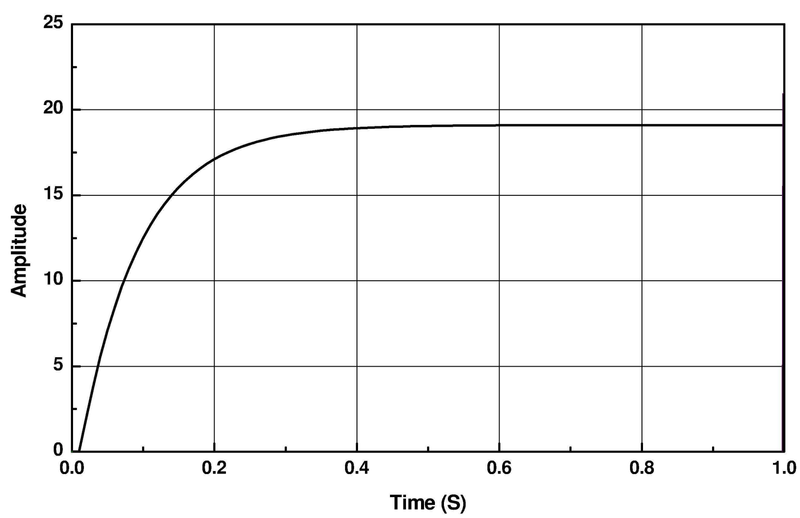

Fig. 4.2. Response to step input for DC motor model.

Where	R	= 0.5 (Ω)
	L	= 0.005 (H)
	J	= 0.05(kg*m^2/s^2)
	K	= 0.5 (Nm/A)

Chapter 5

Result and Discussion

In previous we have seen the governing equation for first and second order system. In this chapter the simulation of first order and second order system for different types of input and controllers' response will be discussed. Finally a DC motor will be simulated for step input using PID control and results will be compared for different control parameter.

5.1 First order system with step input

The block diagram of a first order system with step input was simulated by SimApp simulation software and result of simulation are shown in Fig.5.1 and 5.2 respectively.

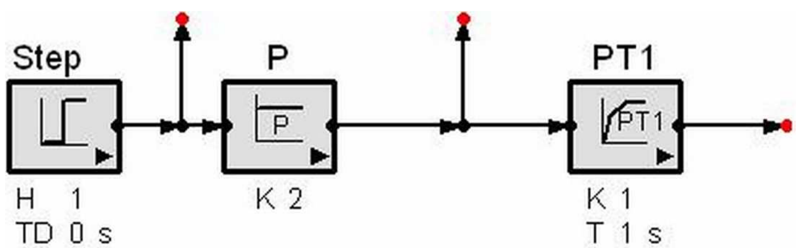

Fig. 5.1. First order system with step input and without feedback.

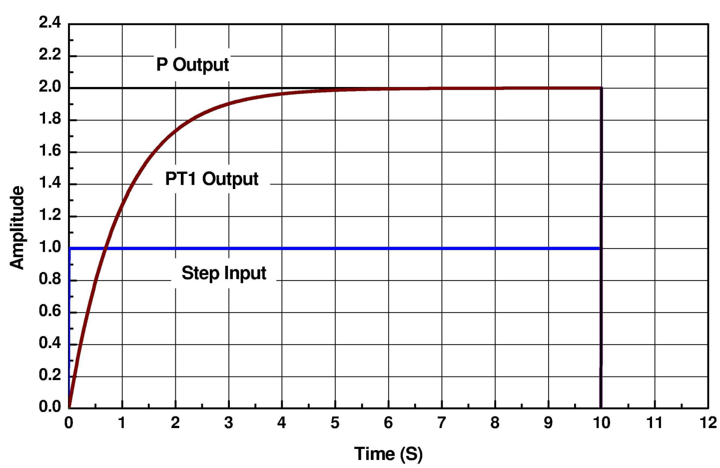

Fig. 5.2. Response curve for first order process with step input and without feedback.

The gain K=1, time constant T=1s, sampling time Ts=0.1s was selected for PT1 system. The step input was taken with step size H=1 and no delay TD=0. A proportional controller with gain K=2 was used as multiplier. After simulation for 10s from response curve we see that a step change in input causes immediate change to output without delay. The output signal is strongly smoothed. The damping is so strong that an overshoot cannot occur. After a long time, the step response follows proportionally the input signal.

5.2 First order system with ramp input

The block diagram of a first order system with ramp input was simulated by SimApp simulation software and result of simulation is shown in Fig.5.3 and 5.4 respectively.

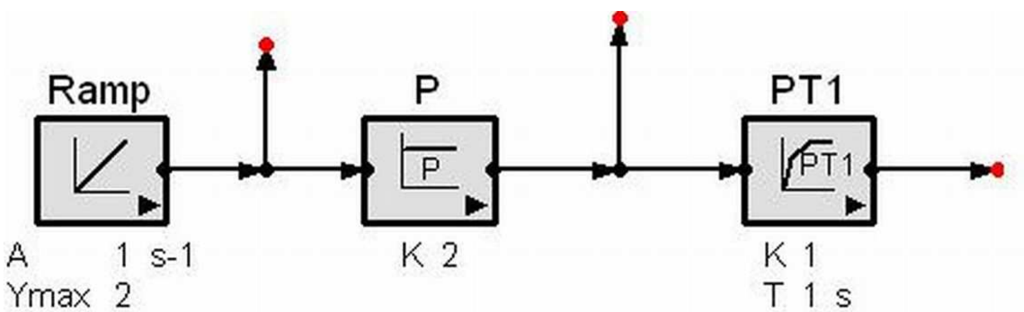

Fig. 5.3. First order system with ramp input and without feedback.

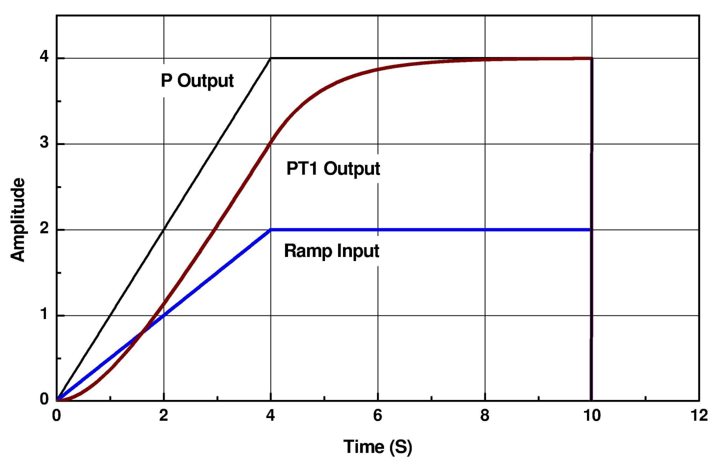

Fig. 5.4. Response curve for first order system with ramp input and without feedback.

The gain K=1, time constant T=1s, sampling time Ts=0.1s was selected for PT1 system. A proportional controller with gain K=2 was used as multiplier. The ramp applied for a constant slope of A=1/s and up to a specified maximum Ymax=2. The saturation is always a positive value and works as a maximum for positive slopes. After simulation for 10s from response curve we see that the error follows ramp input and which is equal to the time constant. The smaller the time constant, the smaller the steady-state error for a first order system with ramp input.

5.3 First order system with step input and P controller

The block diagram of a first order system with step input and feedback for P-controller and response curve after simulation are shown in Fig. 5.5 and 5.6.

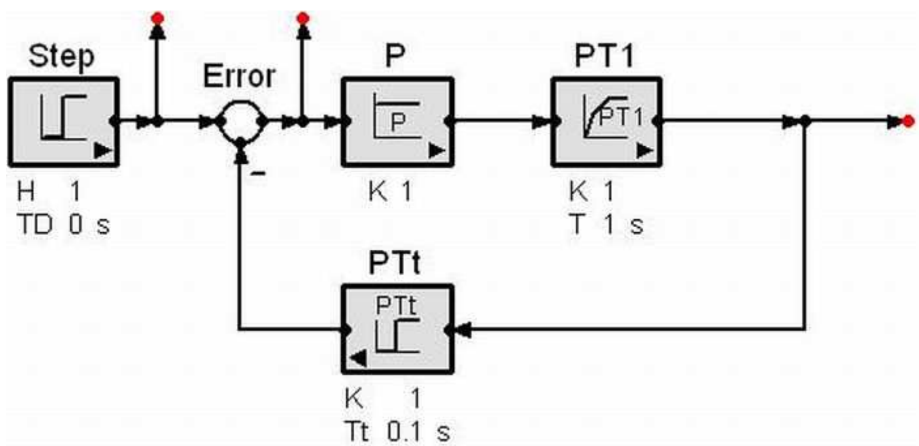

Fig. 5.5. First order system with step input and P controller.

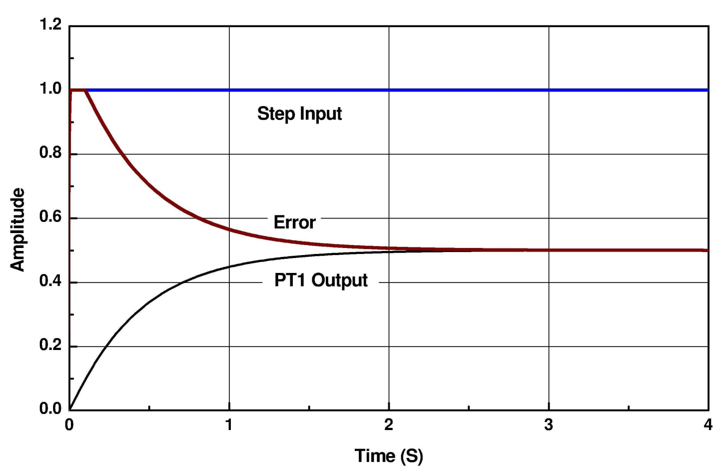

Fig. 5.6. Response curve first order system with step input and P controller.

The gain K=1, time constant T=1s and sampling time Ts=0.1s was selected for a PT1 system. A delay element of gain K=1 and delay 0.01s was placed in the feedback line. An adding element was also used in this system. The step input was taken with step size H=1 and no delay TD=0. After simulation for 10s from response curve we see that with step input the output changes immediately without any delay and the process value reaches to the desire value more quickly for P-controller

5.4 Second order system with step input

The block diagram of a second order system with step input and response curve is shown in Fig. 5.7 and 5.8 respectively.

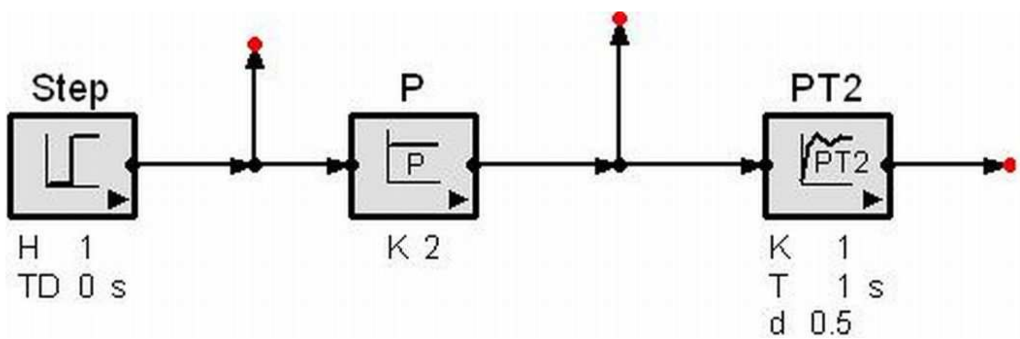

Fig. 5.7. Second order system with step input and without feedback.

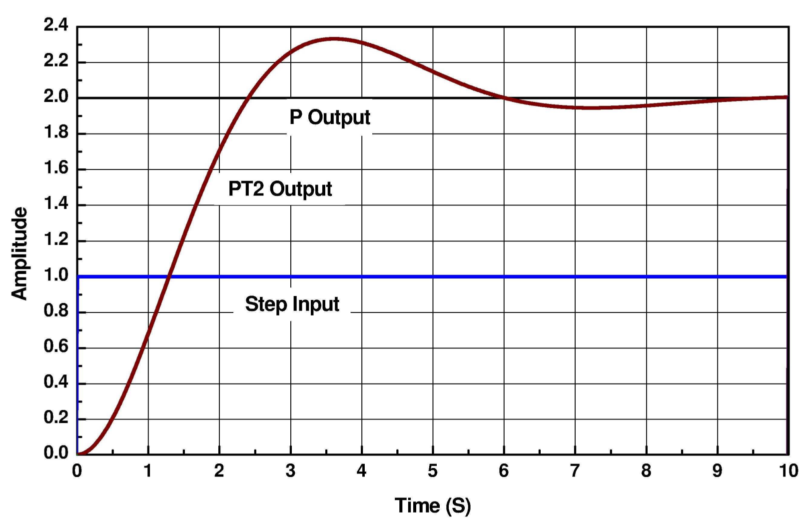

Fig. 5.8. Response curve for second order system with step input.

The gain was K=1, time constant T=1s, damping d=0.5., sampling period Ts=0.1s was selected for PT2 system. The PT2 element contains two independent energy stores. The step response depends on the damping, (d), and the resonant frequency. The step input with step size H=1 and no delay TD=0. A proportional controller with gain K=2 was used as multiplier. After simulation for 10s from response curve we see that three important parameters can easily be determined by applying step input to a second order system. They are rise time, overshoot and damping.

5.5 Second order system with ramp input

The block diagram of a second order system with ramp input is shown in Fig. 5.9. The system was simulated by SimApp software and response curve is shown in Fig 5.10.

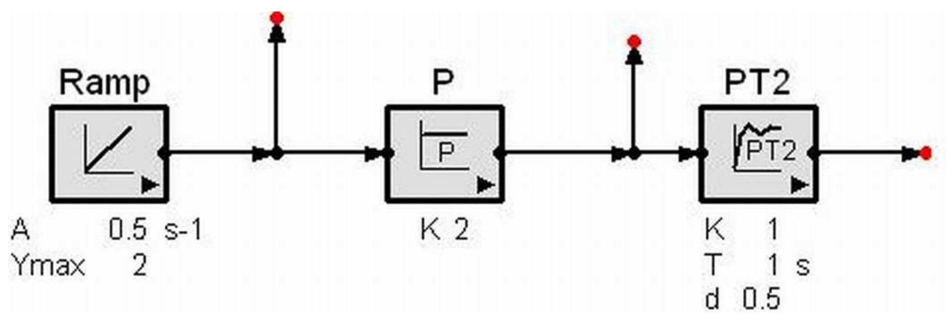

Fig. 5.9. Second order system with ramp input and without feedback.

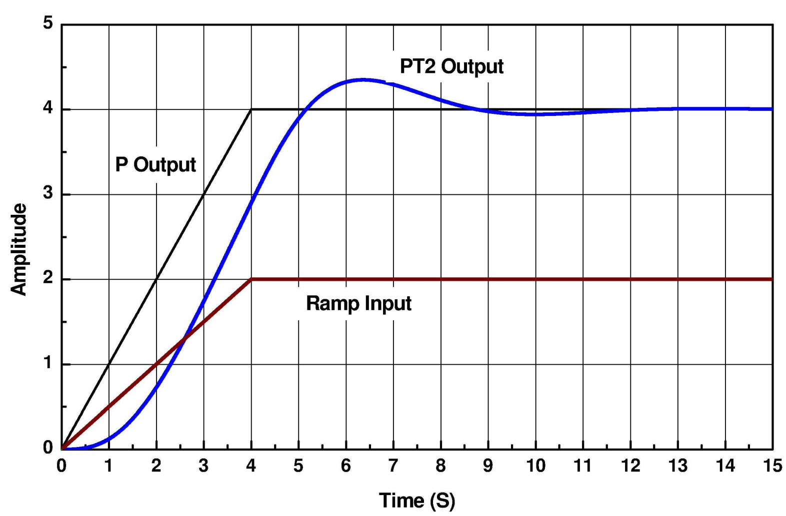

Fig. 5.10. Response curve for second order system with ramp input.

The gain was K=1, time constant T=1s, damping d=0.5, sampling period Ts=0.1s was selected for PT2 system. The PT2 element contains two independent energy stores. The ramp applied for a constant slope of A=1/s and up to a specified maximum Ymax=2. The saturation is always a positive value and works as a maximum for positive slopes. After simulation for 20s response curve was generated and from response curve we see that the ramp response has a slope equal to the slope of the reference input signal. Therefore, the two curves are parallel in steady-state. The steady-state error in the ramp response depends on both the damping ratio and undamped natural frequency as given by

$$e_{ss\_ramp} = \frac{2\xi}{\omega_n} = 0.2672 \qquad (5.1)$$

Where ξ ≡ Damping Ratio

ω_n ≡Undamped Natural Frequency

5.6 Second-Order Systems and Different Controllers

From the response curve of a second order system it is clear that the output oscillates and this oscillation can be controlled by using different types of controller.

Second order system with step and ramp input for P control

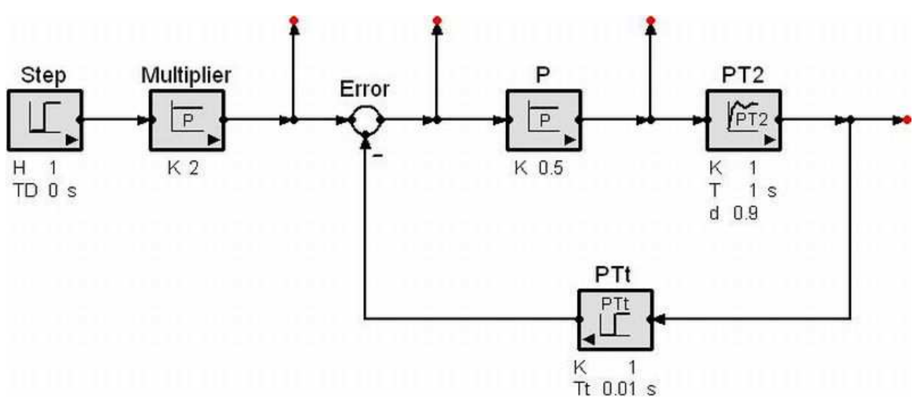

Fig. 5.11. Second order system with step input and P controller.

The block diagram of a second order system with step input was simulated by SimApp simulation software. The gain, K=1, time constant, T=1s, damping, d=0.9 and sampling time, Ts=0.1s was selected for PT2 system. The step input was taken with step size, H=1 and no delay, TD=0. P controller with gain, K=2 was used as multiplier of the step input. P controller placed in the system and simulated for Kp=0.5, 1 and 2. After simulation for 20s from response curve we see that by using P controller the error can be made zero with some overshot and this overshot can be minimized by lowering the values of Kp but the error does not become zero. At lower values of Kp the overshot is less but residual error remains. For step and ramp input the output, residual error and effect of Kp shown in Fig. 5.12 and 5.13.

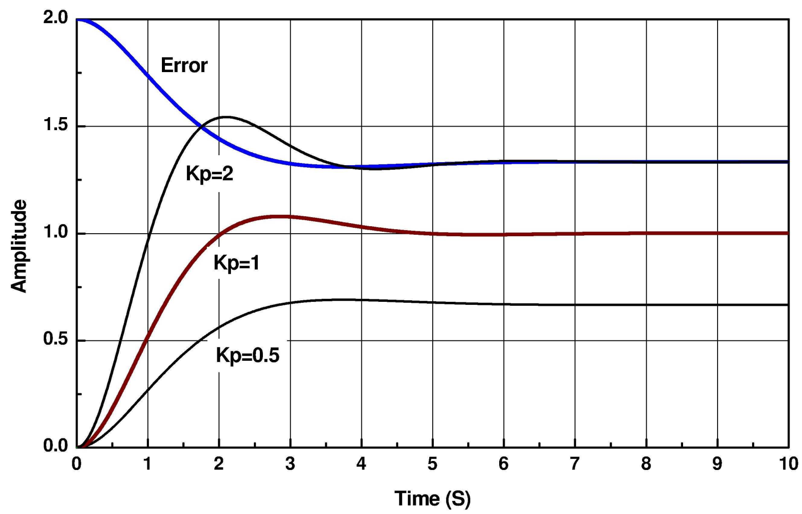

Fig. 5.12. Response curve second order system with step input and P controller.

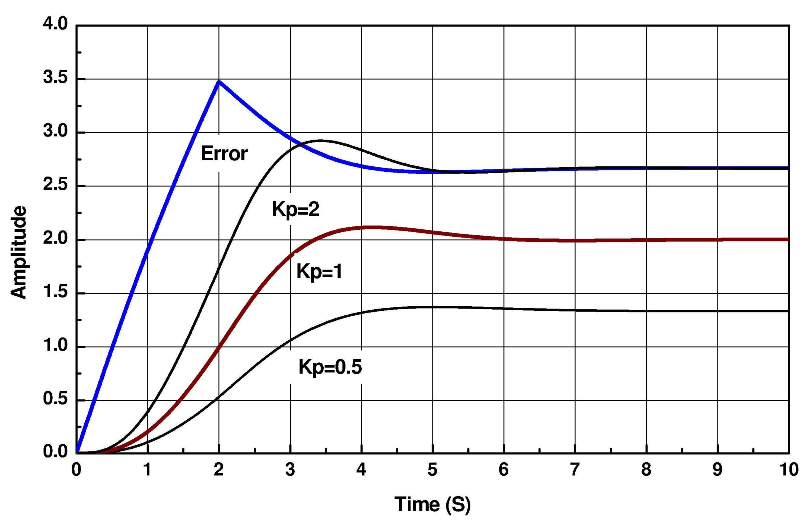

Fig. 5.13. Response curve second order system with ramp input and P controller.

Second order system with step and ramp input for PI control

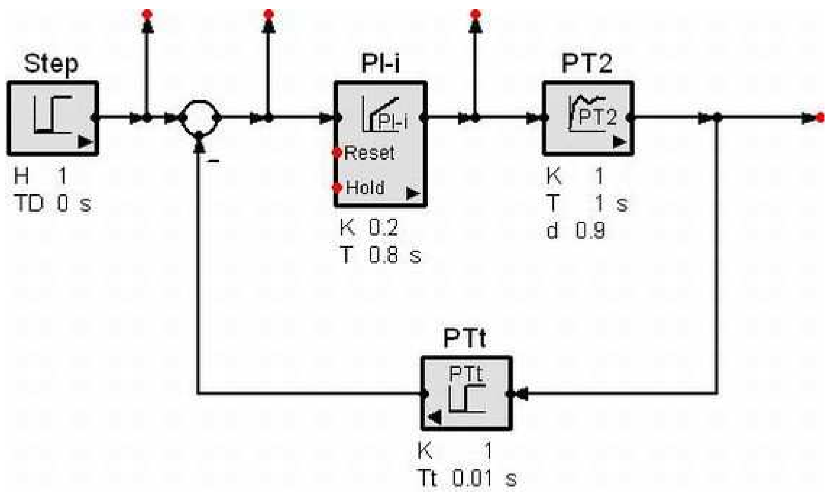

Fig. 5.14. Second order system with step input and PI controller.

Second order system with step input was simulated by SimApp simulation software. The gain, K=1, time constant, T=1s, damping, d=0.9 and sampling time, Ts=0.1s was selected for PT2 system. The step input was taken with step size, H=1 and no delay, TD=0. PI controller with Kp=0.2, T=0.8, Kp=0.2, T=0.3 and Kp=0.1, T=0.5 were selected for simulation. At lower values of Kp the overshot is less and the residual error diminishes due the presence of I controller. For step and ramp input the output, error and effect of Kp and T of a PI controller is shown in Fig. 5.15 and 5.16.

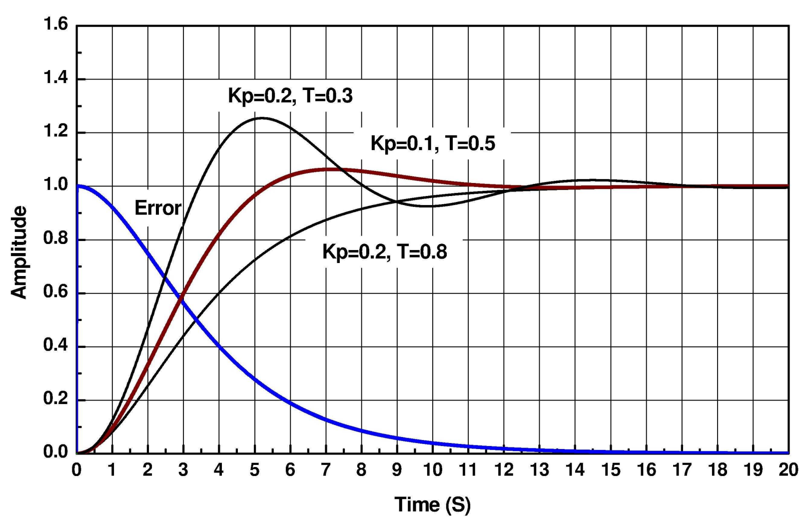

Fig. 5.15. Response Curve for second order system with step input and PI controller.

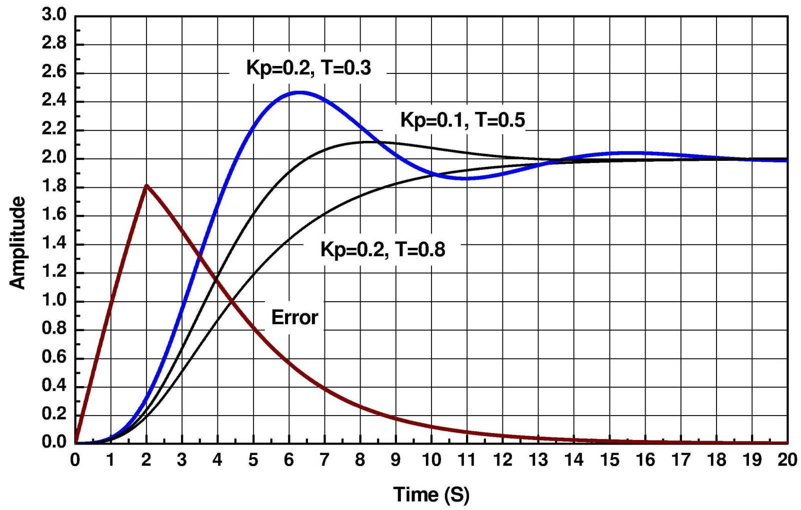

Fig. 5.16. Response Curve for second order system with ramp input and PI controller.

Second order system with step and ramp input for PID control

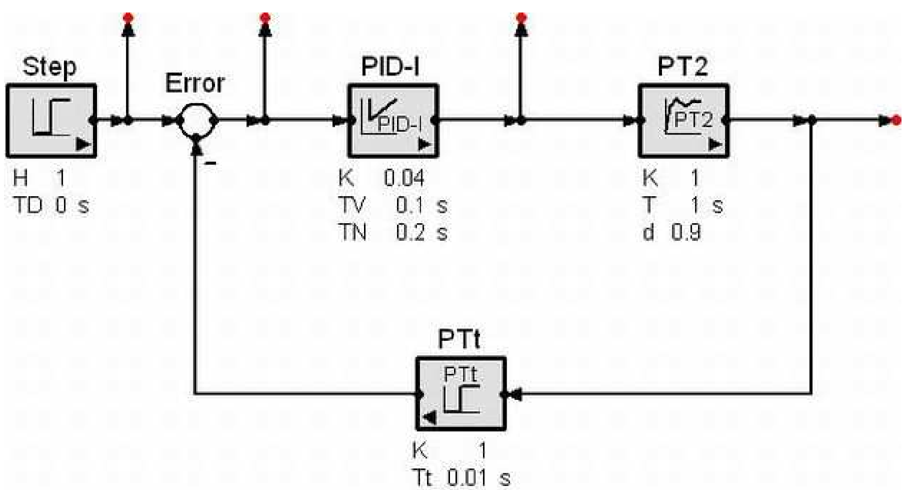

Fig. 5.17. Second order system with step input and PID controller.

Second order system with step input was simulated by SimApp simulation software. The gain K=1, time constant T=1s, damping d=0.9 and sampling time Ts=0.1s was selected for PT2 system. The step input was taken with step size H=1 and no delay TD=0. PID controller placed in the system and simulated for Kp=0.1, Tv=0.1s Tn=0.2s and Kp=0.04, Tv=0.1s Tn=0.2s. After simulation for 30s from response curve we see that by using PID controller the overshot, residual error and settling time can be controlled. At lower values of Kp the overshot is less, the residual error diminishes due the presence of I controller and settling time reduces for D controller. For step and ramp input the output, error and effect of Kp, Tv and Tn of a PID controller is shown in Fig. 5.18 and 5.19.

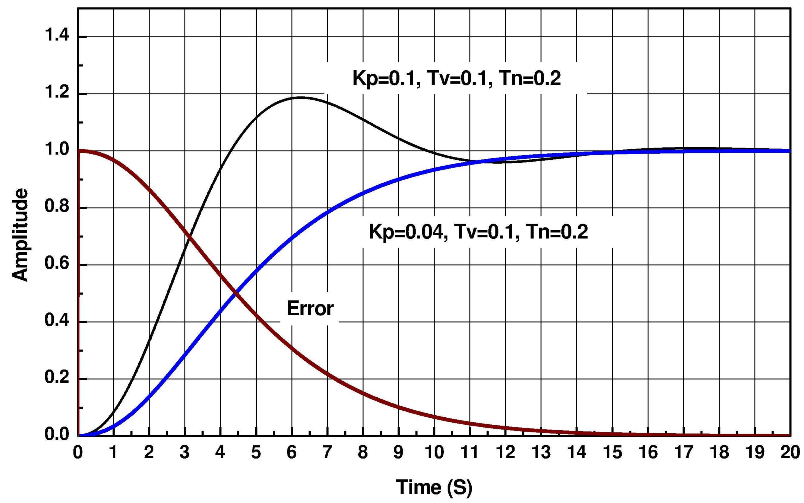

Fig. 5.18. Response Curve for second order system with step input and PID controller.

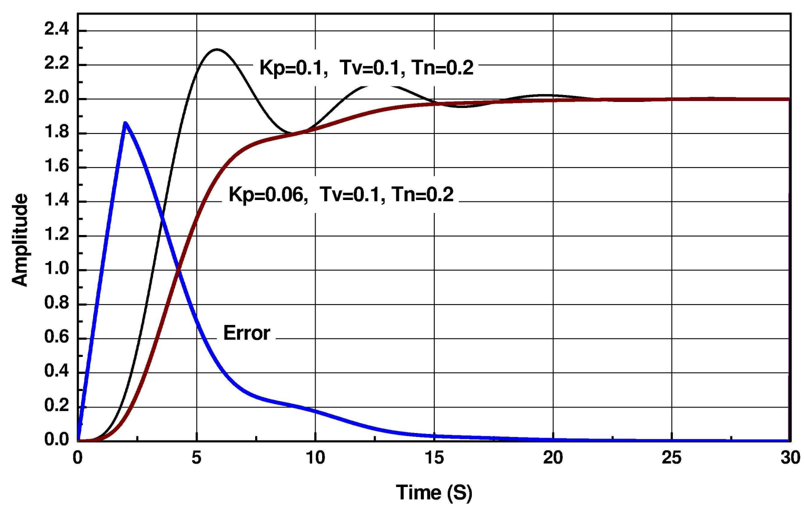

Fig. 5.19. Response Curve for second order system with ramp input and PID controller.

5.7 Comparison of P, PI and PID control for step input

The simulated result for a second order system after tuning plotted and the comparison of P, PI and PID controller shown in Fig. 5.20. It is clearly seen that for P controller the residual error remains and it cannot be removed without overshot. Then the use of PI controller has removed the residual error and overshot. Finally the use of PID controller has removed the overshot, residual error and reduced the settling time.

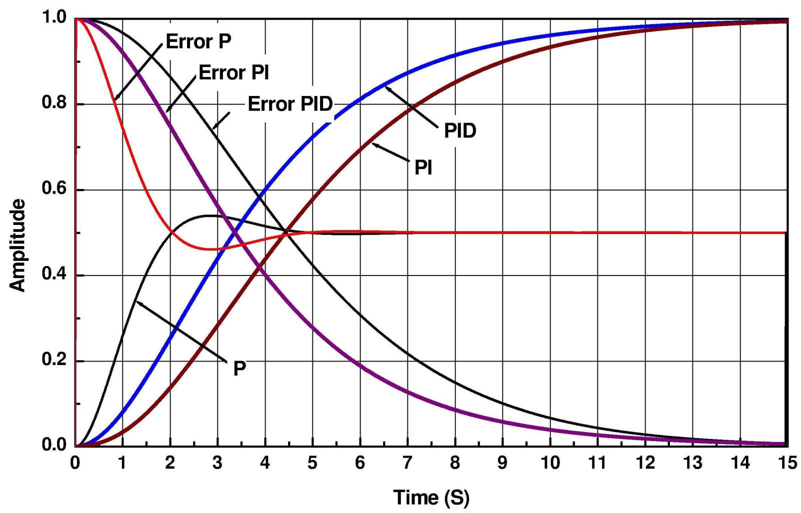

Fig. 5.20. P, PI and PID controller Comparison for step input to a second orders system.

5.8 Simulation of a DC Motor model

A model of DC motor was made for simulation with inductance, La=0.5 H, Ra=0.5 ohm, Kf=0.1Nm/A and J=0.01kgm^2. We know that a DC motor follows second order system therefore the block diagram is shown in Fig. 5.21 and from response curve as shown in Fig. 5.22 it is seen that DC motor follows equation of second order system. To see the effect of controllers, the motor model was simulated with P controller for step and ramp input and a feedback circuit was incorporated with delay of 0.01 sec. By using open loop method for tuning the point of inflection was traced and values of Kp and Tn were calculated. Next this motor model was simulated with PI controller and these calculated values for Kp and Tn were verified. Finally the same motor model was simulated with PID controller for step input

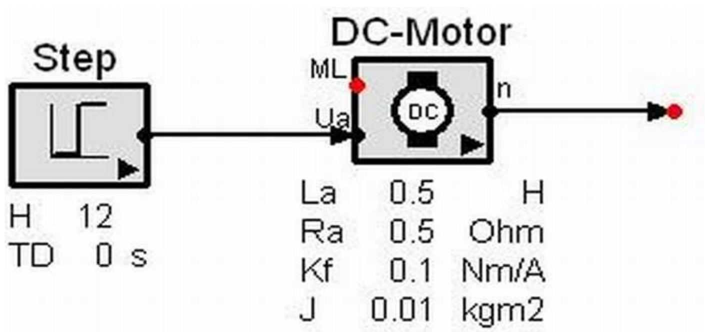

Fig. 5.21. Block diagram of a DC motor model without feedback loop.

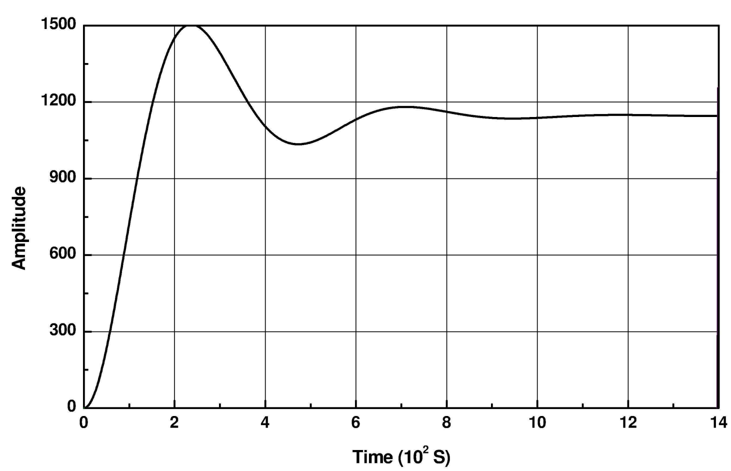

Fig. 5.22. Response curve for a DC motor model with step.

Simulation of DC motor model with step input and P-control

The Block diagram to simulate a DC motor with step input of amplitude 1 and P controller is shown in Fig. 5.23, simulation was carried out for Kp=0.004, 0.006 and 0.009. The comparison of response curves for different curves are shown in Fig. 5.24. From Fig. 5.24 it is clearly seen that overshoot can be reduced at lower value of Kp=0.007 and the output reaches to steady state without overshoot after 1500 ms. But at higher value of Kp=0.009 the output becomes steady after 700 ms with overshot and residual error.

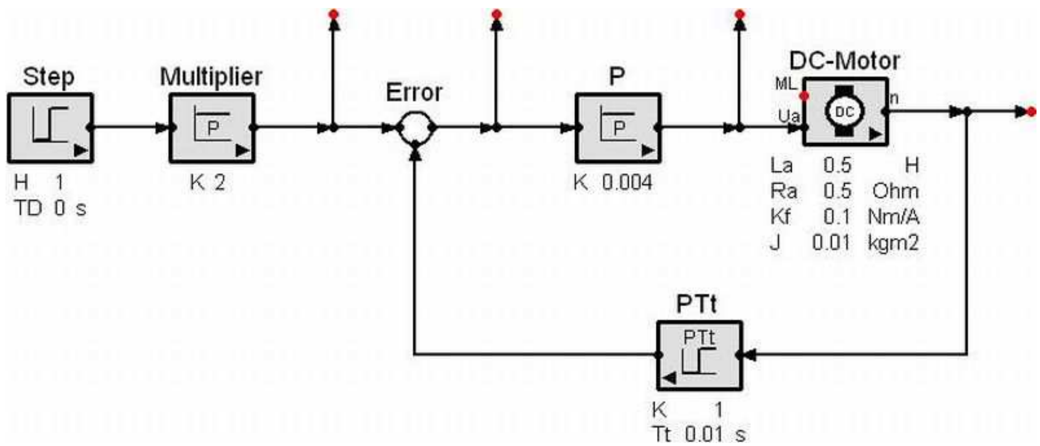

Fig. 5.23. Block diagram of a DC motor model with step input and P-control.

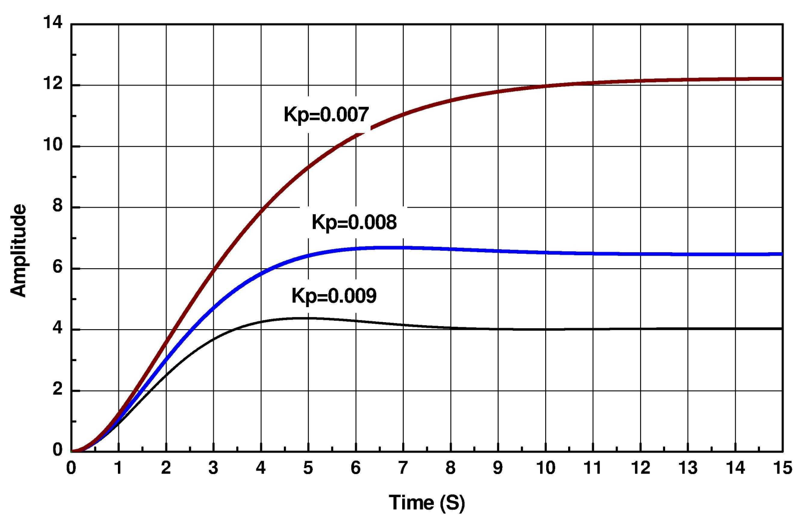

Fig. 5.24. Response curve of a DC motor model with different Kp values.

Simulation of DC motor model with Step input and PI-control

To find out the control parameter the control loop was opened and for step input the inflection point was traced as shown in Fig. 5.25 to determined delay time and response time. The values of Kp and Tn was calculated by using formula given in Table 3.2. These values were placed for simulation and from the response curve it was found that the error became zero and the output became steady as shown in Fig. 5.27.

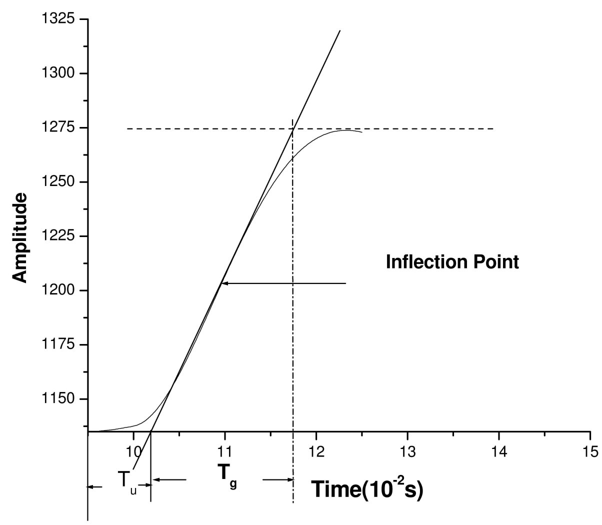

Fig. 5.25. Determination of Inflection point.

The value of Kp was calculated from empirical formula as given by.

$$Xp=2.86 \times Ks \times (Tu/Tg) \times 100\% \qquad (5.2)$$

Where Ks = (Process variable change/Manipulating variable change)=140

Tu = 40

Tg = 160

Xp = 2.86 x 140 x (40/160)=100.1

Kp = 0.00999

Tn = 1.2 x Tg= 1.2 x 160= 144 x 10^{-2} s= 1.92 s

Initially the motor was simulated with Kp=0.00999 and Tn=1.92s as initial approximation then the overshot was reduced by reducing the values of Kp.

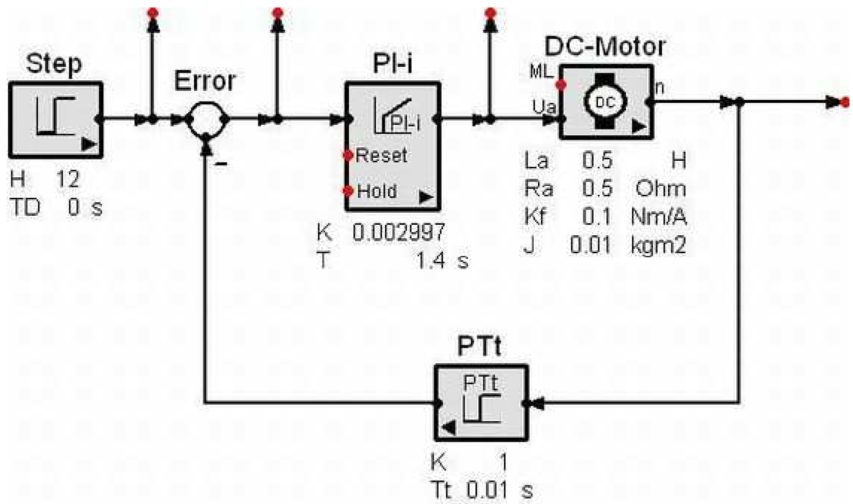

Fig. 5.26. Block diagram of a DC motor model with step input and PI-control.

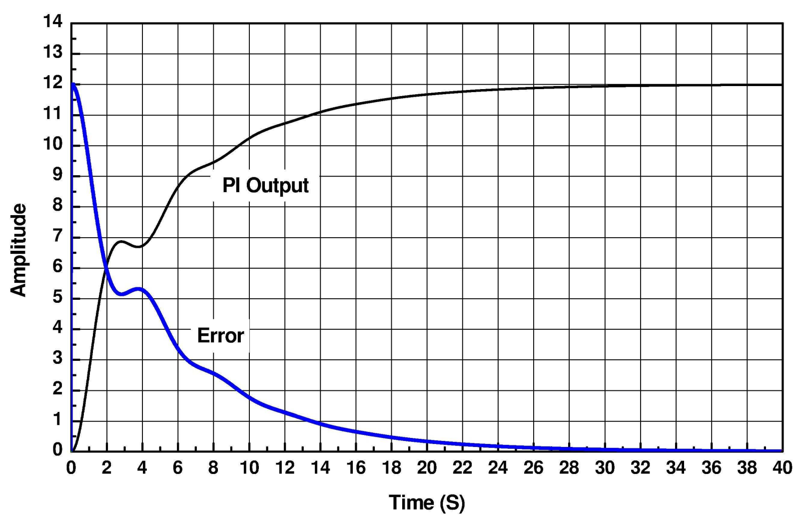

Fig. 5.27. Response curve for a DC motor model with step and PI control.

Finally simulation was carried for DC motor with step input and different values of Kp and Tn to verify the effectiveness of the PI controller. From Fig. 5.28 we can see that for Kp=0.0029 and Tn=1.4s the output has became steady without overshot, whereas for higher value of Kp=0.004 and different value of Tn=0.06s the output initially oscillated and then became steady, and for lower value Kp=0.0004 and Tn=0.06s the output did not oscillated but became steady after a long time. Thus the values of Kp and Tn as derived from open loop method hold good for PI controller to drive a DC motor model.

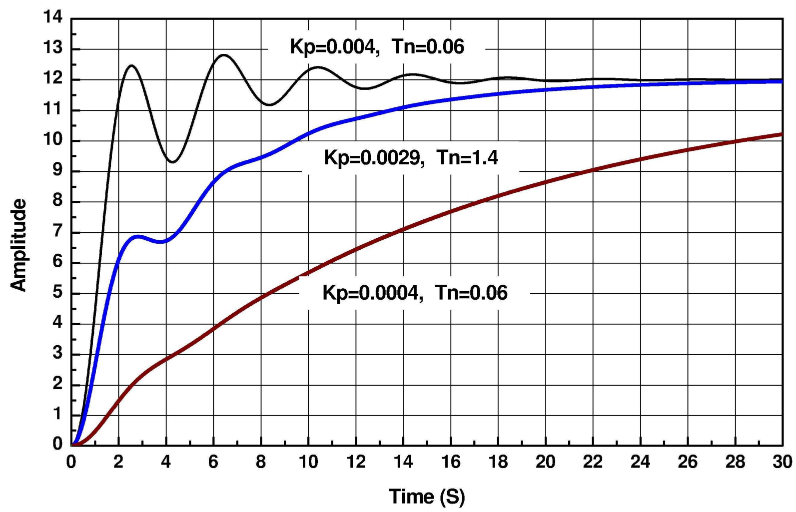

Fig. 5.28. Comparison of a DC motor model with step input and various Kp and Tn.

Simulation of DC motor model with Step input and PID-control

The block diagram to simulate a DC motor with step input of amplitude 12 and PID control is shown in Fig. 5.29

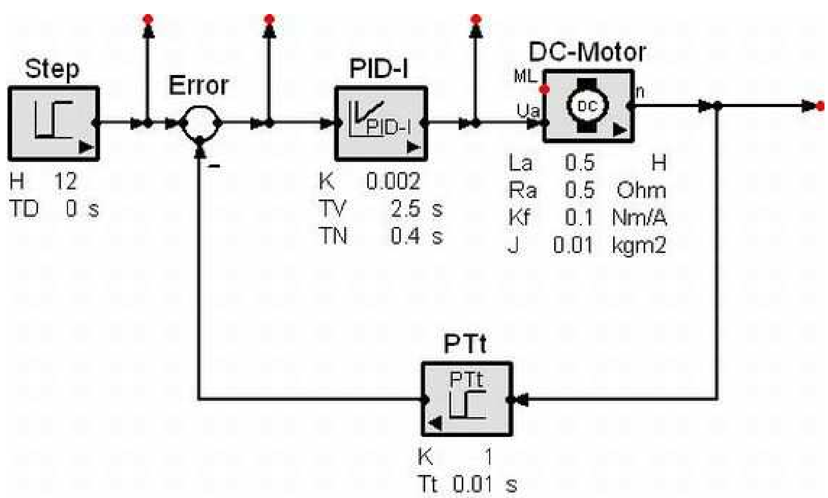

Fig. 5.29. Block diagram of a DC motor model with step input and PID-control.

Simulation was carried out for Kp=0.002, Tv=2.5 and Tn=0.4. The response curves for 20 sec simulation were plotted in Fig. 5.30 and from this figure it is seen that the error diminish without overshoot and the output reaches to steady state without overshot quickly.

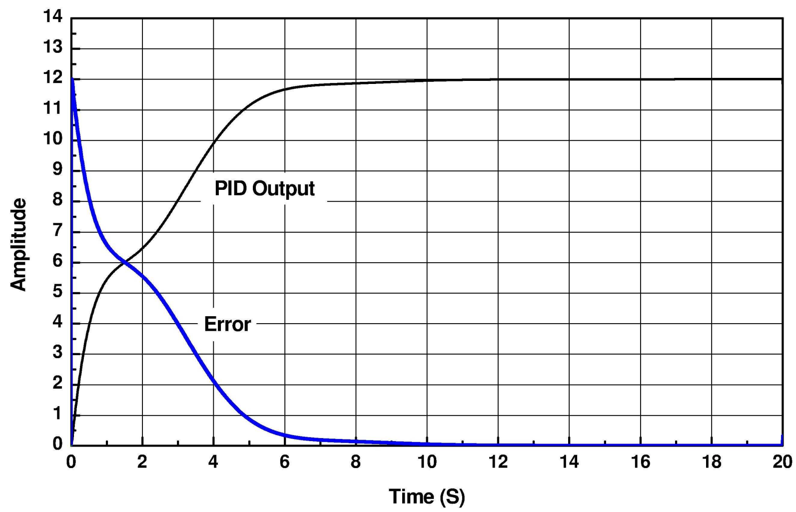

Fig. 5.30. Response curve for a DC motor model with step and PID control.

Response of a PID controlled PMDC subjected to a square input is shown in Fig. 5.31 from [24]. Adequate data to model/simulate [24] is not available, however the reported response to step input indicates very similar trends as reported in Fig. 5.30.

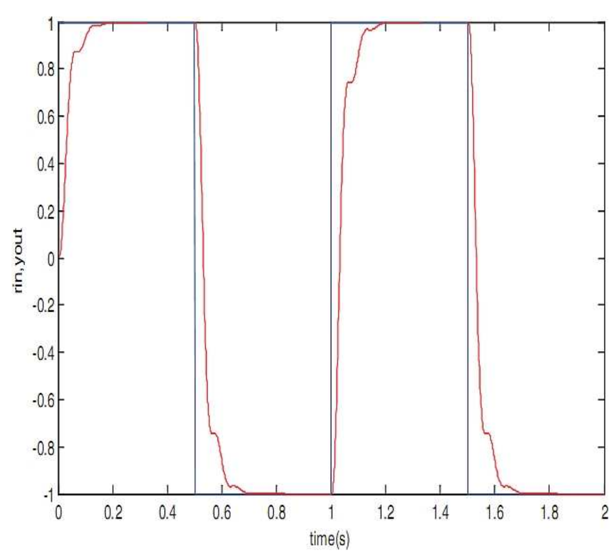

Fig. 5.31. Response curve for a DC motor with square input and PID control [24].

5.9 Comparison of Different PID values for a DC Motor

Simulation was carried for DC motor with step input and different control parameter to verify the effectiveness of the PID controller. From Fig. 5.32 we can see that for Kp=0.002, Tv=2.5s and Tn=0.4s the output became steady without overshot, whereas for different value of Kp=0.003, Tv=2.5s and Tn=0.4s the output initially oscillated and then became steady, and for lower value of Kp=0.001 the output did not oscillate but became steady after a long time.

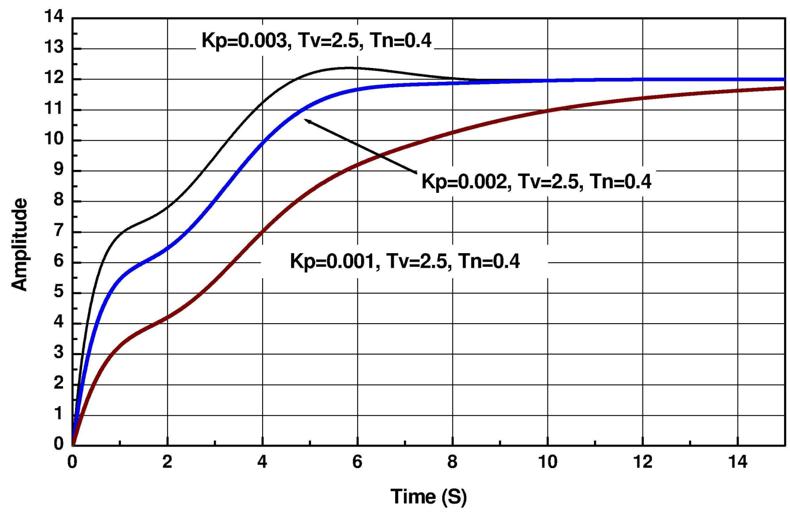

Fig. 5.32. Comparison of DC motor model with step input and various Kp, Tv and Tn.

5.10 Comparison of Different Controllers for DC Motor

The response curves of a DC motor model using P, PI and PID controls were plotted and compared. From Fig. 5.32 it is clearly seen that by using PID control the output reaches to steady state in minimum time, without overshot and no residual error. Therefore, the dynamic response of a DC motor for step input by using PID control found most suitable to drive a mobile robot and it is proved by simulation.

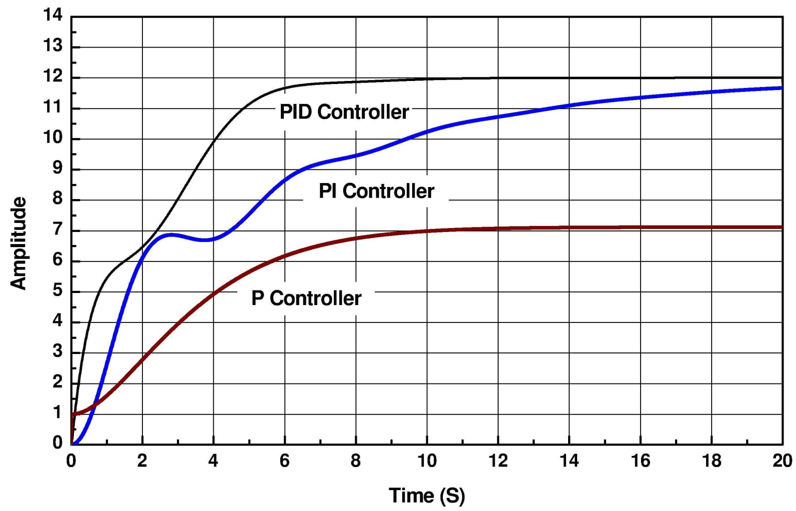

Fig. 5.33. Comparison of a DC motor model with P, PI PID control and step input.

Chapter 6

Conclusion and Scope for Future Work

6.1 Conclusion

The usages of robot are increasing day by day in various field of our daily life. Therefore, the accuracy for speed and position control is also increasing to meet users' requirements. The dynamic response i.e. response with respect to time, of a mobile robot drive is very important for position and speed control. To study this dynamic behavior the basic of control system, drive mechanism and controller were studied initially. Different modes of control were studied to know the advantages and disadvantages of individual controllers. The advantages of combine modes of controller were also discussed to counter the disadvantages of individual controllers. The tuning methods also discussed to know how to tune a controller and find out the initial control parameter.

The DC motors are widely used as drive for mobile robots because DC motors have speed control capability, which means that speed, torque, and even direction of the rotation can be changed at any time to meet the new condition. To make the mathematical model of a DC motor the governing equation for first order and second order systems were studied. Mathematical model of a DC motor was also made and simulated by SimApp software. The governing equation of a DC motor is a second order equation and the response curve of this second order system for step input and without feedback is an oscillating curve which damps with time. The use of control system exhibits different behavior to these curves and by simulation these curves were generated for different control parameters. Different systems and DC motor model were simulated for P, PI, and finally PID controller showed the disadvantages and advantages of each of them. A second order system was tuned for step input initially then the input was changed to ramp input and it was seen from the response curve that the system follows the input signal with significant accuracy. At the same time the effects of different control parameter were compared and discussed.

From the comparison of simulated results of a DC motor for P, PI and PID controller it was clearly seen that by using PID control the output reached to steady state in minimum time, without overshot and no residual error. Therefore, from dynamic response of a DC motor for step input by using PID controller found most suitable to drive a mobile robot and it is proved by simulation.

6.2 Scope for Future Works

DC motors have speed control capability, which means that speed, torque, and even direction of the rotation can be changed at any time to meet the new condition and widely used for mobile robots drive. In this work dynamic response of a second order system and DC motor model were simulated by SimApp software for step input. Therefore; the dynamic response of a DC motor model may be simulated by SimApp simulation software for different PID control parameter with variable PWM drive.

References

[1] Jone, JL and Flynn, AM (1993) Mobile Robots: Inspiration to Implementation, AK Peters.

[2] http://en.wikipedia.org/wiki/Mobile-robot.

[3] Sugisaka, M and Hazry, D. Development of a proportional control method for a mobile robot. Applied Mathematics and Computation 2007;186:74-82

[4] K. Astrom, T. Hagglund, PID Controller: Theory, Design and Tuning, Instrument Society of America, 1995.

[5] K. Astrom, T. Hagglund, Revisiting the Ziegler-Nichols step response method for PID control, J. Process Control 14 (2004) 635-650.

[6] G.J. Silva, A. Datta, S.P. Bhattacharyya, New results on the synthesis of PID controllers, IEEE Trans. Autom. Control 47 (2) (2002) 241-252.

[7] M. Ge, M. Chiu, Q. Wang, Robust PID controller design via LMI approach, J. Process Control 12 (2002) 3-13.

[8] S. Tavakoli, I. Griffin, P. Fleming, Robust PI control design: a genetic algorithm approach, Int. J. Soft Comput. 2 (3) (2007) 401-407.

[9] R. Toscano, A simple PI/PID controller design method via numerical optimization approach, J. Process Control 15 (2005) 81-88.

[10] Ang KH, Chong G, Li Y. PID control system analysis, design, and technology. IEEE Trans Control Syst Technol 2005;13(4):559-76.

[11] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. ASME Trans 1942;64:759-68.

[12] Hang CC, Åström KJ, Ho WK. Refinements of Ziegler_Nichols tuning formula. IEE Proc-D 1991;138(2):111-8.

[13] Åström KJ, Hang CC, Persson P, Ho WK. Towards intelligent PID control. Automatica 1992;28(1):1-9.

[14] Mudi RK, Dey C, Lee TT. An improved auto-tuning scheme for PI controllers. ISA Trans 2008;47:45-52.

[15] Luyben WL. Design of proportional integral and derivative controllers for integrating dead-time processes. Eng Chem Res 1996;35:3480-3.

[16] Mann GKI, Hu BG, Gosine RG. Time-domain based design and analysis of new PID tuning rules. IEE Proc Control Theory Appl 2001;148(3):251-61.

[17] Kristiansson B, Lennartson B. Robust and optimal tuning of PI and PID controllers. IEE Proc Control Theory Appl 2002;149(1):17-25.

[18] Chidambaram M, Padma Sree R. A simple method of tuning PID controllers for integrator/dead-time processes. Comput Chem Eng 2003;27(2):211-5.

[19] Padma Sree R, Srinivas MN, Chidambaram M. A simple method of tuning PID controllers for stable and unstable FOPTD systems. Comp Chem Eng 2004; 28(11):2201-18.

[20] Basilio JC, Matos SR. Design of PI and PID controllers with transient performance specification. IEEE Trans Education 2002;45(4):364-70.

[21] Rivera DE, Morari M, Skogestad S. Internal model control. 4. PID controller design. Ind Eng Chem Process Des Dev 1986;25:252-65.

[22] Jean, DEPREZ. DC Motor Control: MATLAB/Simulink in Drives and Power Electronics.

[23] Manfred S, Frank B. Control Engineering A Guide to Beginner
JUMO GmbH & Co. KG, Fulda, Germany, January 2003

[24] Yan, W., Wang, D., Jia, P. and Li, W..(2012). “The PMW Speed regulation of DC motor based on intelligent control.” System Engineering procedia 3: 259-267.

Appendix A

SimApp (Simulation Software developed by Buesser Engineering)

Introduction

SimApp is a software tool for the analysis and optimization of dynamic systems based on block diagrams. It does not assume any predefined structures. You can draw and simulate any kind of equations that can be represented by block diagrams and by using the available functional elements. SimApp may be used to model systems that can be represented by ordinary linear differential equations. It is especially useful for simulation of feedback systems or automatic control systems.

Linear and nonlinear, time invariant and time variant systems or subsystems can be simulated in the time domain and the results displayed as time and XY plots. In addition, linear and time invariant systems or sub-systems can be investigated in the frequency domain with Bode and Polar (Nyquist, Black) plots and their eigenvalues. The plots and data tables can be printed out or exported to other applications with using the

Windows clipboard

The modeling of the systems, i.e. the drawing of the block diagrams, is done graphically by placing functional elements into the drawing and connecting them with signal lines. The most important parameters (gain, time constants, delays, etc.) can be entered directly in the drawing without the need to open dialog boxes. Additional graphic forms (lines, rectangles, circles, etc.) and text are helpful for documenting the block diagrams.

The object palette of SimApp consists of more than 80 functional elements. Often used subsystems can be grouped and saved into the palette or libraries. You can also create your own specialized blocks – custom blocks - that consist of the available functional elements and other custom blocks. They can be used in the same way as standard elements and let you adapt SimApp to your special needs

The time and frequency response can be captured and analyzed at any system node. Special time and frequency probes are available for extended tasks. For example, you can capture and compare the frequency response of the open and closed loop of a control system in one run. Another special probe allows two-dimensional displays in the time domain. SimApp has a multiple document interface. This means that you can open more than one document at the same time to run simultaneous simulations.

Who can or should use SimApp?

SimApp is suitable for students, technicians, engineers and scientists who want study dynamic systems and would like easy, intuitive operation. SimApp is very simple to use, and enables beginners to quickly obtain usable results within minutes.

Using help

SimApp has several supporting help mechanisms. You can call the SimApp online help application that gives you information in structural form or you can use context-specific help on the various interactive components (menus, controls and toolbars). You will also find this manual in the online help.

Launch help application

Start the online help by using the content and index command in the help menu (?) and search the required information in the contents list. This help resource is best for general information. For detailed information about buttons and entry controls use the context-sensitive help instead.

Context-sensitive help

Contextual information is available anywhere where you discover the ? symbol in a window's title bar or the help button somewhere in the window. First click the ? symbol or the help button and then the object you need more information about. This opens a help pop-up window for that object which delivers detailed information about the subject. Alternatively you may also press the F1 key if the corresponding control has the input focus. You can also pull up a manual page by right-clicking on the object. There are controls with comprehensive information but also such with nothing at all. The amount of information depends of the actual need for additional comments.

System requirements

SimApp runs on Windows 2000, XP and Vista and has no special system prerequisites. But for long and high resolution simulations it is preferable to use a newer PC with significant main memory.

Installation

Before you can run SimApp, you must first install it by launching the setup program. There are two ways by downloading over the Internet or by Purchasing the optional CD-ROM

Overview

After launching, SimApp opens the main window. In the program options (menu Extras+Options) you can select whether you want SimApp to reopen all drawings of the last session. Otherwise it opens a new empty drawing.

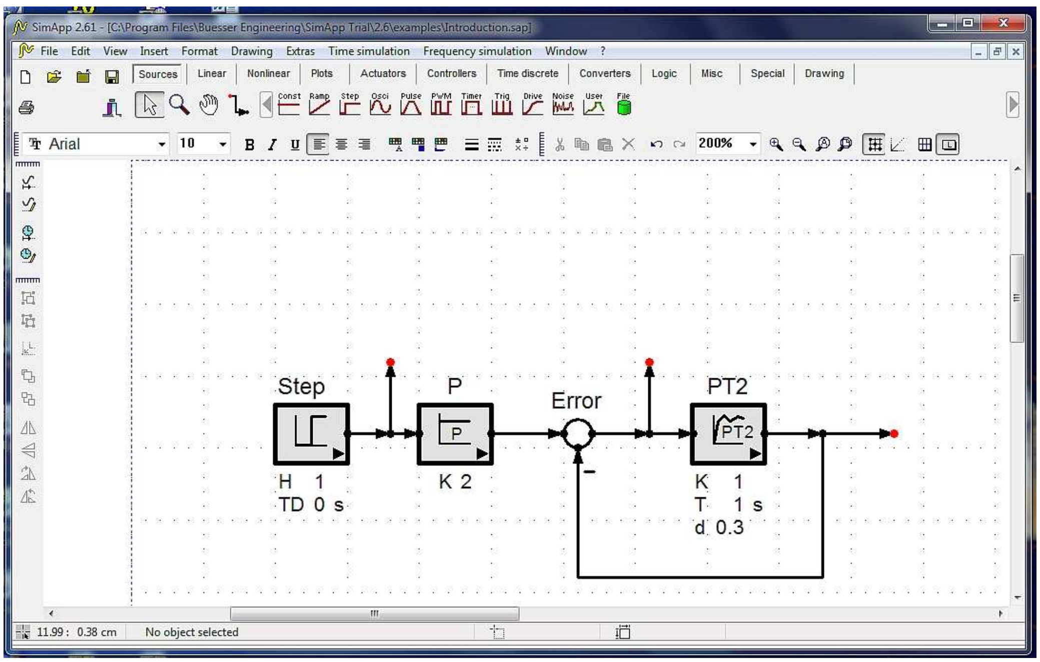

Fig. A1. SimApp main window.

Main menu

In the menu bar across the top of the SimApp application window you will find most of the commands. Some contextual or object specific commands reside only in specific pop-up menus.

Pop-up menus

Right-clicking an object opens its pop-up menu. Objects can be anything you see in the application window; palette, buttons, toolbars, panels and even objects in your drawings. By means of pop-up menus you can access the operations of these objects. Pop-up menus are displayed at the pointer's current location so they eliminate the need to move the pointer to the menu bar or a toolbar.

File toolbar

The file toolbar is in the top left corner of the SimApp main window. It has some buttons for the most important file operations, such as saving, opening, printing, and a button to exit SimApp. This toolbar is not moveable, but you can hide it along with the palette.

Palette

The palette is a multiple page toolbar that contains all standard simulation objects and some drawing tools. Select a page by clicking on a tab. There two types of objects:

Functional elements:

These objects are the building blocks of your block diagrams, i.e. the models of the systems you want to analyze by simulation. Examples are sources, actuators, controllers, signal lines.

Shapes, lines and text (Page Drawing):

In SimApp you can draw simple shapes and lines. These objects are not primary objects because they are not needed for simulations. You can use them if you want to illustrate your drawings and draw the symbol of custom blocks. The most important tools for drawing manipulation (selection, zoom and drag) are on the left side of the palette.

Error bar

The error bar flashes in red if an error is encountered. It has a brief error message. The lines in the error bar may be associated with the block where the error occurred. Some errors may allow you to find the block where the error occurred by clicking the error message.

Introductory Example

This chapter contains a small project to demonstrate step-by-step creation of a block diagram and the simulation of the time and frequency responses. More advanced features are described in the next chapters.

System modeling

Block diagrams (or block schemes) represent the real system. A block diagram mainly consists of blocks and signal lines. The blocks represent the transfer elements that change system data or create new data. The signal lines interconnect the blocks and enable the system's data flow. Each line stands for a system data item with direction indicated by the arrow head.

From real system to block diagram

You are fortunate if your system is already represented by a mathematical description (differential equations) or even by a block diagram. You can just start drawing. Otherwise, you still have a lot to do: Analyze the system, find interfaces to other systems, break it into subsystems and find suitable mathematical equations. If you have found a mathematical description, you can present it in a graphical form by using objects from the palette and simulate it. However, the translation from the real system to a block diagram representation is not part of this manual. Refer to the literature about automatic control systems, control engineering, nonlinear control, dynamic simulation, etc. See the bibliography at the end of this manual. All basic elements are divided into categories in the palette. As a simple tutorial project we draw the following block diagram:

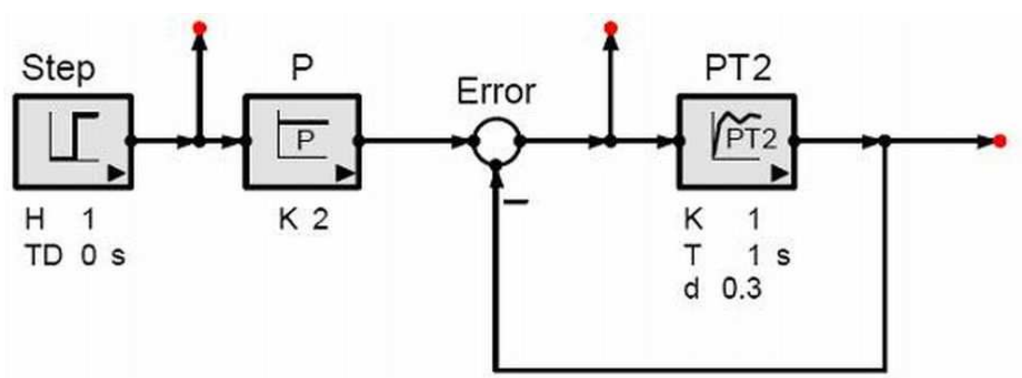

Fig. A2. Control loop with step stimulation.

Be sure that automatic snap is on. (Button in the drawing palette). Get the objects out of the palette by first clicking the tab of the page that contains the category the desired object belongs to and then click the button showing the symbol of the object (release the mouse button again). Position the mouse at the location where you want to paste the object and press the mouse button. If you do not release the button immediately, you see the shape of the object at the pointer's location. You can now exactly position the object by dragging and then releasing the mouse button. When drawing a block diagram it is helpful to place the blocks first, and then connect them.

Simulations

After drawing a block diagram without error, you can perform time and frequency simulations.

Time simulation

First, we are interested in the system's step response. Press the start button for the time simulation new window consisting of several pages appears. On the top page you see the plot of the step response at the system's output nodes. By default, output nodes are named for the blocks that generate those outputs.

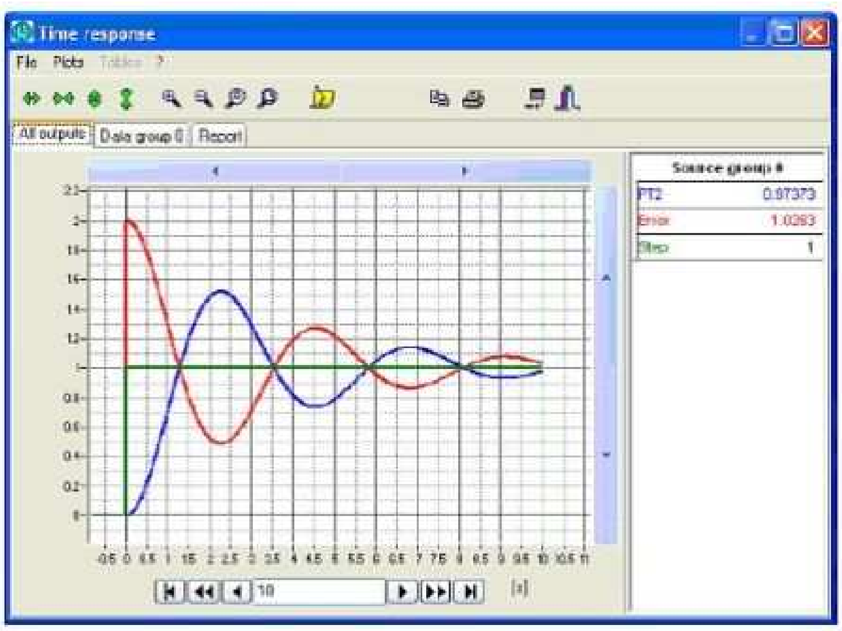

Fig. A3. Time plot.

On the second page you find the simulated values in tabular form. The third page shows the parameters of the current simulation. Tables and plots can be printed out or exported to other applications with the Windows clipboard. The plot contains a vertical measurement line that may be moved by dragging with the mouse or by navigation buttons at the bottom of the page. The point of time is always displayed in the navigator and the corresponding signal values are displayed in the legend. Zoom any diagram area by dragging a zoom frame with the mouse. The curves can be edited by right-clicking the curve or the corresponding legend item. The data in the tables can be selected and copied to other applications by means of the Windows clipboard. Columns are adjustable with the mouse.

Simulation Objects

Description

Simulation objects are the building blocks of the block diagrams. They consist of standard functional elements, the custom blocks and the signal lines. The arrow inside the block symbol indicates the data flow direction through the object. The outputs are generally on the right and the inputs on the left edge. For feed-back loops, blocks can be flipped with the flip command. You can also create custom blocks that consist of a system of basic blocks or any other user defined blocks. The basic blocks usually have two different symbols. The default symbol is a schematic representation of the time or frequency response. The second symbol shows the mathematical transfer function in the time or frequency domain.

The transfer function of a block is controlled by its parameters. Each block has its own set of parameters. Some of them are displayed just below the block in the drawing and can be edited by simply clicking; others can only be changed in the block's simulation properties dialog box. You can determine which parameters appear directly in the drawing. Every parameter has a default value. Parameters of which there actual value differs to the default value are displayed anyway.

Custom Blocks

Large and complex systems can be difficult to understand. Such systems often consist of several subsystems. You could graphically separate subsystems by drawing lines or using colored shapes and include descriptive text, but this would enlarge the drawings even more. Generally, it is not necessary to show all the details of a system simultaneously.

Furthermore, there are some well-known structures in specific engineering disciplines that can be used repeatedly but with different parameter values.

For this purpose, in SimApp you can create your own blocks. You can draw any system or subsystem and pack it into a new block with its own parameters. This is not even restricted to block diagrams. You can put any kind of graphical representation you draw with SimApp into custom block. User-defined blocks have own block symbols and do not differ from basic blocks. You can store them to the palette or collect them in libraries.

There are two methods of creating higher level integrated blocks:

In the first method, you can select all objects in a drawing and gather them into a single block. In the second method, you use a special workshop in SimApp, the block folder. The first method is very fast and is suitable for temporarily simplifying a drawing without any need for re-use. The second method takes more effort, but has no restrictions and is suitable as a long-term solution.

Creating simple custom blocks by selection

You can select any object in a drawing and put it into a new block. After selection, use the menu command Extras + Create custom block. All selected objects are immediately replaced by the new block. It has a default symbol and looks like a basic block.

The input and output nodes result from the connections to non-selected elements or are open nodes of the selected objects. The connections to external elements are not broken.

Custom blocks can be broken up with the menu command Extras + Break up custom block at any time. It is immediately replaced by its inner structure. The symbol is lost, but connections to external blocks are retained.

Creating custom blocks in the block folder

In a block folder, you can create and alter custom blocks. You have access to the inner structure and the block symbol. A special feature is the ability to create new virtual parameters and connect them to the real parameters of the contained blocks.

Introductory example

In example, you will learn the most important steps to create a custom block in the block folder. The lesson is to create a custom block from the block diagram of a DC motor. This block should be supplied with a descriptive symbol.

Designing the block symbol

After you have defined the functional part you need a symbol for the motor block. Change to the Symbol page. You see an empty frame and free nodes that are labeled with the input and output names. If you did not assign names, they will be numbered as (1),(2) and (3). These are the input and output nodes of the new block. Move them onto the frame. Input nodes on the left and output nodes on the right. Do not press the Alt-key for positioning or switch off snapping permanently. The nodes must lie exactly on the grid. If you are not sure if they do so, move them to grid with the button. The empty frame has the standard size of 1.6 by 1.6 cm, but you can enlarge or reduce it. Now paint the block symbol with the SimApp drawing tools.

Block assembly

Now we can join the functional and graphical part to a whole. Select the command Extras + Block Folder + Export Block to Windows clipboard. The ready-to-use block is now in the Window clipboard. Change to another drawing or open a new one and paste the block for further use.

Fig. A4. Block assembly for a DC motor model.

Therefore, by using SimApp software mathematical model can be created and that model can also be simulated. Thus it is very good and simple simulation software for students and professionals.