Subhabrata Ganguli

Control System Design with an Industrial Perspective:

With MATLAB Examples

Table of Contents

[Acknowledgement 1](#_Toc154305037)

[Introduction 2](#_Toc154305038)

[Chapter 1: PID Control Implementation 5](#_Toc154305039)

[Overview 5](#_Toc154305040)

[Implementation of PID Controller 6](#_Toc154305041)

[Set Point Weights 7](#_Toc154305042)

[Implementation of the Integral Term 9](#_Toc154305043)

[Disable integrator until the system reaches near set-point 10](#_Toc154305044)

[Limit the min/max value of the integrator state 10](#_Toc154305045)

[Position and Velocity Control Using Acceleration Input 11](#_Toc154305046)

# Acknowledgement

Some text

# Introduction

Some text

LTI Plant Without Uncertainty

In this part of the book, we will concentrate on Linear Time Invariant (LTI) plant which does not have any modeling uncertainty. Our goal is to gain insight into some of the key aspect of control design in a simplistic setup.

We will look into a simple but fundamental plant model called a *double integrator*. This type of plant appears in various industrial problems. One of them is related to velocity and position control of Adaptive Cruise Control (ACC) system in automotive vehicles. In ACC, velocity control is required to maintain a set speed during normal operations. In traffic, the vehicle may not be able to operate at the set speed and may need to drive at a lower speed by maintaining a fixed distance from the lead traffic vehicle.

We will use various control design techniques to obtain position and velocity tracking. We will start with a classical control method called Proportional-Integral-Derivative (PID) control. We will also show how a Lead Compensation can be used as an alternate technique. Finally, we will delve into various control analysis techniques to check the stability and tracking performance of the closed loop system.

# Chapter 1: PID Control Implementation

## Overview

PID control, which stands for Proportional-Integral-Derivative control, is a widely used feedback control mechanism in engineering and industrial applications.

PID Control is a type of closed-loop control system that continuously calculates an error value as the difference between a desired setpoint and a measured process variable. The PID controller then adjusts the control input to the system based on three main components: proportional, integral, and derivative terms.

Here is a brief overview of the different components of the PID controller:

**1. Proportional (P) Term**:

* The proportional term is directly proportional to the current error. It contributes to the control output in proportion to the magnitude of the error.
* The goal of the proportional term is to reduce the steady-state error, which is the difference between the desired setpoint and the actual process variable when the system has stabilized.

**2. Integral (I) Term**:

* The integral term is proportional to both the magnitude and the duration of the error. It accumulates the error over time.
* The integral term helps eliminate any residual steady-state error that may be present after the proportional control has brought the system close to the setpoint.

3. **Derivative (D) Term:**

* The derivative term is proportional to the rate of change of the error. It anticipates future behavior based on the current rate of change of the error.
* The derivative term helps to dampen the system's response, preventing overshooting and oscillations.

The overall control output (u) of the PID controller is calculated as the sum of the three terms:

,

where *u*(*t*) and *e*(*t*) are the control and error signals, and *KP*, *KI* and *K­D* are the PID gains.

## Implementation of PID Controller

Note that although all the above terms on the right-hand side of the equation uses the error term *e*(*t*) as the feedback signal, it is not an ideal approach to implement the controller as such. Using the error term in the closed loop system adds a zero in the transfer function between the controlled signal and the reference signal which can cause an unnecessary fast response in the beginning and can even cause an overshoot. Hence, an idea called *set point weights* is typically used in the actual implementation.

Let us try understanding the problem of implementation of the PID controller with the error terms for all the three terms with an illustrative example. Let the plant dynamics be given by

Consider the PID gains be *KP* = 2, *KI* = 1 and *K­D* = 0 (no derivative term). The implementation in Simulink is shown in Figure 1. If we look at the step response in Figure 2, we see that it has a very fast initial response before the output settles down to one. This is due the presence of a fast zero in the transfer function of the closed loop system. Note that the closed loop transfer function is

We observe that the closed loop system has a zero at s = rad/sec = rad/sec which causes the fast response at the beginning. The closed loop system has a real pole at 0.25 rad/sec and two complex conjugate poles with frequency 14 rad/sec and 0.53 damping ratio.

A diagram of a computer

Description automatically generated

Figure 1: PI Controller with error term as feedback for both K­P and K­­I.

A graph with a blue line

Description automatically generated

Figure 2: Step response of closed loop for PI controller with error signal feedback for both K­P and K­­I.

### Set Point Weights

A better way to implement the PID controller is to weight the set-point/reference signal. We scale the proportional term feedback as and derivative term feedback as where *r*(*t*) and *y*(*t*) are the reference and the feedback signals respectively and and are positive scalars. These scalars terms are designer’s choices. For example, if we set in our PI controller example. we get a well behaved second order response of the closed loop system. The closed loop transfer function is:

which shows no zero in the numerator. The Simulink implementation of the controller is shown in Figure 3, and the step response is shown in Figure 4.

A diagram of a computer

Description automatically generated

Figure 3: PI controller with set point weight for KP.

A graph with a curve

Description automatically generated

Figure 4: Step response of closed loop for PI controller with set point weight for KP.

## Implementation of the Integral Term

The integral term in a PID controller is rarely implemented as is. This is due to the integrator windup phenomenon. Integrator windup occurs when the error term keeps accumulating over time and the system is unable to implement the control signal due to physical limitation or saturation. This can lead to a significant overshoot or deviation from the setpoint when the system eventually becomes capable of responding.

There are several strategies which can be implemented to prevent integrator windup. These are listed below:

* 1. Disable integrator until the system reaches near set-point.
  2. Limit the min/max value of the integrator state.
  3. Implementation of an anti-windup scheme.

Consider the PI control problem discussed above, but now we add an actuator before the plant. The actuator transfer function is and has a saturation limit of as shown in Figure 5. If we input a step of 1 at 1 sec followed by another step of -1 at 5 sec, we observe that the output responds to the change in second step at 6.4 sec instead of 5 sec. This is due to integrator windup.

A diagram of a complex function

Description automatically generated

Figure 5: Plant with actuator saturation

A graph with a line

Description automatically generated

Figure 6: Step response for plant with actuator saturation

### Disable integrator until the system reaches near set-point

### Limit the min/max value of the integrator state

One of the strategies to avoid the integrator windup is to add a limit to the min/max value of the error feedback to the integrator as shown in Figure 7. In this example we limit the error input to the integral term to be . With this we see that the output signal immediately responds to the change in the step input at 5 sec as shown in Figure 8.

A diagram of a program

Description automatically generated

Figure 7: Integral control with error signal limit

A graph with a line

Description automatically generated

Figure 8: Step response with error limit for integral control

## Position and Velocity Control Using Acceleration Input

Let us look at a classic problem of position control using acceleration input. The dynamics of a position control system using acceleration can be formulated as:

where *x* is the position and *u* is the control input. We will consider two cases (a) where both position and velocity can be measured, and (b) where we only have position measurement available.

Using Laplace transform, the above equation can be written as

In other words, there are two *pure integrators* in the plant. We call this the *double integrator* problem.

The control solution to the double integration is not straight forward.