Alexandria University
Faculty of Engineering
Division of Communications & Electronics

EE 381: Automatic Control Systems

Lab 3 : PID Controllers

PID Controllers

Structures involving PID Controllers are simple yet versatile feeback control structures. PID is short for ***Proportional-Integral-Derivative*** controller. As the name describes, ID controllers perform their jobs by scaling the error signal, its derivative and integral.

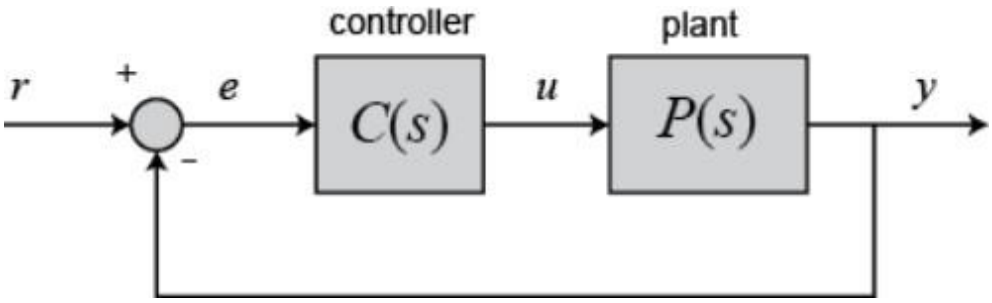
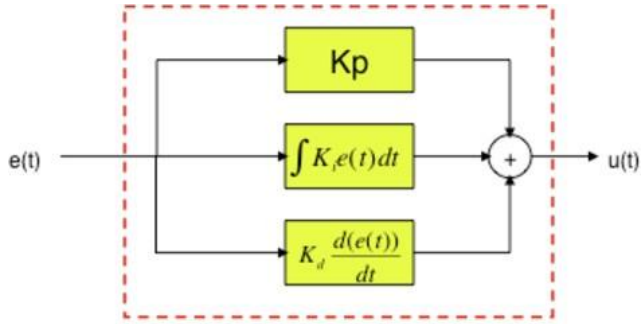


Figure 1: Feedback Block for PID Controller



$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{d(e(t))}{dt}$$

Figure 2: PID Controller Components

If we consider a unity feedback system as shown in the Figure 1. The system consists of the System to be controlled (The Plant ***P***(***s***)) and the Controller ***C***(***s***) and a negative unity feedback. The desired value for the output ***y*** is input to the system as ***r***. The output of the PID controller, equal to the control input to the plant, in the time-domain is as follows:

$$u(t) = K_p\, e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt} \qquad (1)$$

This control signal (***u***) is sent to the plant, and the new output (***y***) is obtained. The new output (***y***) is then fed back and compared to the reference to find the new error signal (***e***). The controller takes this new error signal and computes its derivative and its integral again, and so on.

The transfer function of a PID controller is found by taking the Laplace transform of the above equation to obtain:

$$\frac{U(s)}{E(s)} = K_p + K_d\, s + \frac{K_i}{s} = \frac{s^2 K_d + K_p\, s + K_i}{s} \qquad (3)$$

A proportional controller (***Kp***) will have the effect of reducing the rise time and will reduce but never eliminate the steady-state error. An integral control (***Ki***) will have the effect of eliminating the steady-state error for a constant or step input, but it may make the transient response slower. A derivative control (***Kd***) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

The effects of each of controller parameters ***Kp***, ***Ki*** and ***Kd*** on a closed-loop system are summarized in the table below:

	RISE TIME	OVERSHOOT	SETTLING TIME	Steady ERROR
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	No Change

Note that these correlations may not be exactly accurate, because **Kp**, **Ki**, and **Kd** are dependent on each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when you are determining the values for **Kp**, **Ki** and **Kd**.

Hint:

Kp: proportional gain.

Kd=KpTd : Derivative gain

Ki=Kp/Ti : Integral gain

Example 1: mass-spring-damper system.

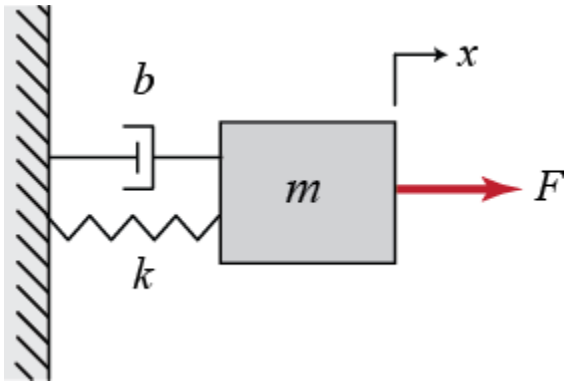


Figure 3 : Mass-Spring-Damper System

The free-body diagram for this system is shown below. The spring force is proportional to the displacement of the mass x, and the viscous damping force is proportional to the velocity of the mass, $v = \dot{x}$.
Both forces oppose the motion of the mass and are, therefore, shown in the negative x-direction. Note also that $x = 0$ corresponds to the position of the mass when the spring is unstretched.

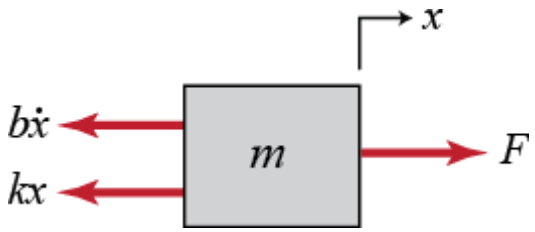


Figure 4 : Mass-Spring-Damper System

Physical Model

The physical model of the system flows from applying Newton's second Law in the X-direction.The system equation becomes:

$$\Sigma F_x = F(t) - b\dot{x} - kx = m\ddot{x} \tag{4}$$

where:
x = displacement of the mas
b = damping coefficient
F(t) = input applied force
k= spring constant

The previous equations are in time domain. Using Laplace transform on the system model, we arrive at the following equations:

$$ms^2X(s) + bsX(s) + kX(s) = F(s) \tag{5}$$

and the plant transfer function is therefore:

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2+bs+k}$$

Let:

m = 1 kg
b = 10 N s/m
k = 20 N/m
F = 1 N

if we apply step response to the plant pen loop system we get the following output

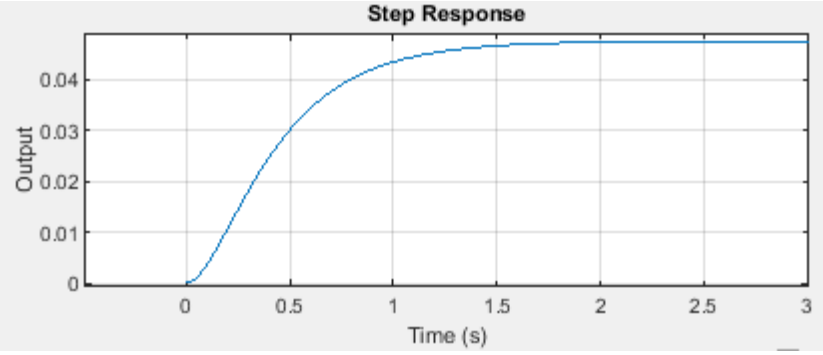


Figure 5: open loop step response

This step response of the open loop plant show the following results the rise time is about 1 sec , the steady state error is about 95% as the final value is below 0.05 ,and the settling time is around 1.5 sec.

The goal of this example is to show how each of the terms Kp, Ki ,Kd contributes to obtaining better performance with the common goals of:

- Fast rise time
- Minimal overshoot
- Minimal steady-state error

Proportional Controller only:

We will apply Proportional controller with Kp=350 within closed loop unity feedback system ,then the o/p is

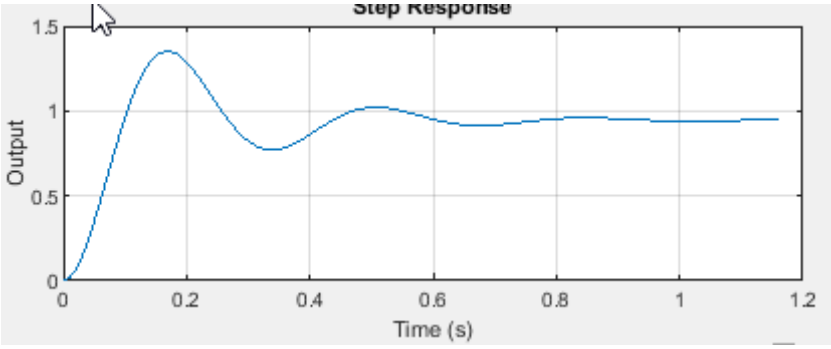


Figure 6: P controller step response

We can observe that the rise time has decreased to be about 0.066 sec, the steady state error also decreased but the problem is that we have large overshoot value which reaches about 42% in that case. Now we will apply the derivative controller with the proportional one to solve for the overshoot problem

Proportional Derivative Controller :

We will apply Proportional derivative controller with Kp=350 and Kd=50 within closed loop unity feedback system ,then the o/p is :

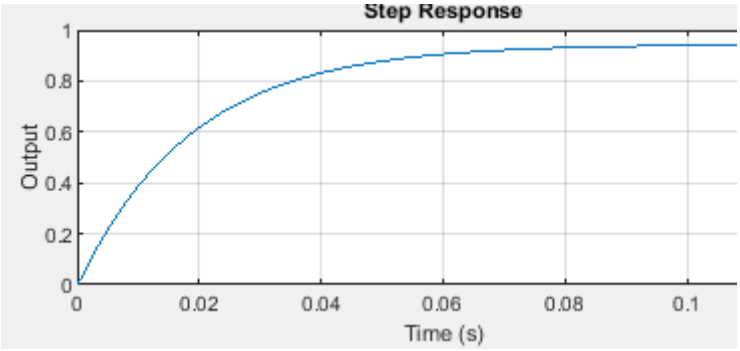


Figure 7: PD controller step response

We can observe that by using the derivative controller beside the proportional controller we could decrease the overshoot of the feedback system, with almost no change with rise time and steady state error value.
Finally, we could apply the integral controller with both proportional and integral to eliminate the steady state error.

Proportional Integral Derivative Controller :

We will apply Proportional integral derivative controller with Kp=350 KI=300 Kd=50 within closed loop unity feedback system ,then the o/p is :

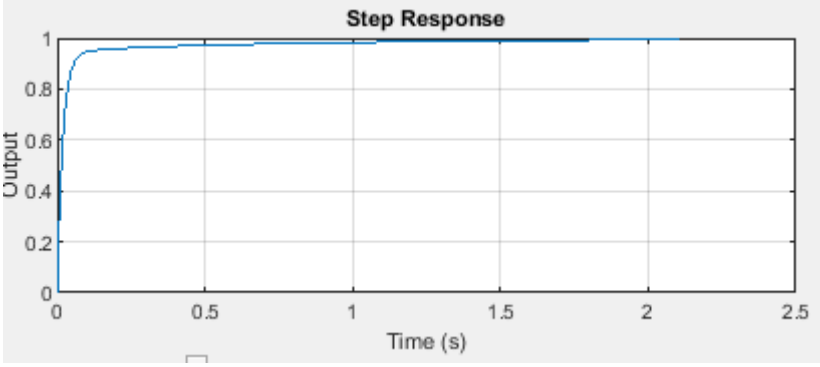


Figure 8: PID controller step response

We can observe that by using PID controller (and tuning the Kp, Ki,Kd)we could get a very good closed loop performance with around no steady state error ,rise time of around 0.0548 sec, with almost no overshoot and about 0.83 sec settling time.

Example 2 : Cruise Control using PID Controllers

Automatic ***cruise control*** is an excellent example of a feedback control system found in many modern vehicles. The purpose of the cruise control system is to maintain a constant vehicle speed despite external ***disturbances***, such as changes in wind or road grade. This is accomplished by measuring the vehicle speed, comparing it to the desired or ***reference*** speed, and automatically adjusting the throttle according to a ***control law***.

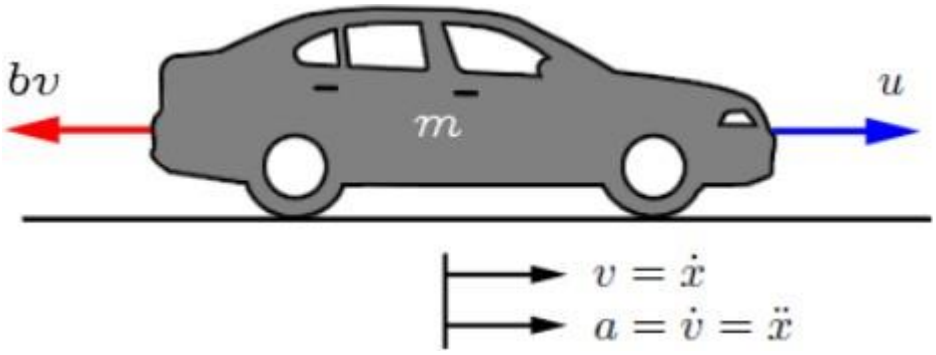


Figure 9 : Free Body Diagram for a Vehicle

we consider a simple model of the vehicle dynamics, shown in the free-body diagram (FBD) above. The vehicle, of mass m, is acted on by a control force, (***u***). The force (***u***) represents the force generated at the road/tire interface. For this simplified model we will assume that we can control this force directly and will neglect the dynamics of the powertrain, tires, etc., that go into generating the force. The resistive forces, (***bv***), due to rolling resistance and wind drag, are assumed to vary linearly with the vehicle velocity, (***v***), and act in the direction opposite the vehicle's motion.

Physical Model

The physical model of the system flows from applying Newton's second Law in the X-direction.The system equation becomes:

$$\sum F(t) = ma(t)$$
$$u(t) - bv(t) = m\,\dot{v}(t)$$
$$m\,\dot{v}(t) + bv(t) = u(t) \qquad (6)$$

Where :

m = *mass of the car*
b = *damping coefficient*
v*(*t*)* = *car velocity at time* t

The previous equations are in time domain. Using Laplace transform on the system model, we arrive at the following equations:

$$m.\ s\ V(s) + b.\ V(s) = U(s) \qquad (7)$$

and the plant transfer function is therefore:

$$\frac{V(s)}{U(s)} = \frac{1}{ms+b} \qquad (8)$$

Lab Requirement

For example 2 : Cruise Control using PID controller

The parameters for the system are as follows:
- (m) vehicle mass = 1000 kg
- (b) damping coefficient = 50 N.s/m

1. Model the Car cruise open loop system shown in Figure 9 and Equation (8) and get its open loop step response using attached MATLAB PID Simulator attached with the pdf. (if your MATLAB supports it), and by writing a MATLAB code.

2. Adjust the PID parameters to get the following system specifications for a step response closed loop negative unity feedback system:

- Rise time < 5 sec
- Overshoot < 10%
- Steady-state error < 2%

Using MATLAB PID simulator (if your MATLAB supports it) and by writing a MATLAB code.

Useful MATLAB Hints

Useful functions: *tf, feedback,pid ,step,stepinfo*

Submission rules

Deadline Date	17th December 2022 23:59
Submission e-mail	control.systems.alexedu.2022@gmail.com subject of the email is your Name & ID
Submission files	- MATLAB (.m file) - Simulink (.slx file) - A Report that includes screenshots for all the requirements and figures.
Note	Write your full name in ARABIC in the body of your email and your ID