CEP

MCT-333 CONTROL SYSTEMS -1

Name	**: Arham Ali**
Registration Number	**:2019-MC-253**

Department of Mechanical, Mechatronics & Manufacturing Engineering B.Sc. Mechatronics & Control Engineering Program

University of Engineering & Technology Lahore (Faisalabad Campus)

Problem description:

Consider a DC motor speed control with the following parameters:

$$J_m = 1.3 \times 10^{-2} Nmsec^2/rad$$
$$b = 0.02 Nmsec/rad$$
$$R_a = 0.3\ ohms$$
$$k_t = k_e = 0.068\ volt\ sec/rad$$
$$L_a = 0.05\ Henry$$

Objectives:

a) Compute the transfer function of the system.
b) Design a PID controller while choosing appropriate values of controller parameters.
c) Select the range of the controller gains while ensuring the stability of the system.

   Design requirements are: max. Overshoot < 20%, settling time ≤ 1 sec (2 % criterion), select the suitable value of ξ for underdamped system.

d) Evaluate the performance of PI, PD, and PID control on the responses of the system while considering step input for reference and disturbance respectively.

Introduction:

In this CEP we will be discussing the process of designing and evaluating a control system in MATLAB using transfer functions and controllers. We will cover the following topics:

1. Introduction to Control Systems: - What are control systems?
   - Types of control systems

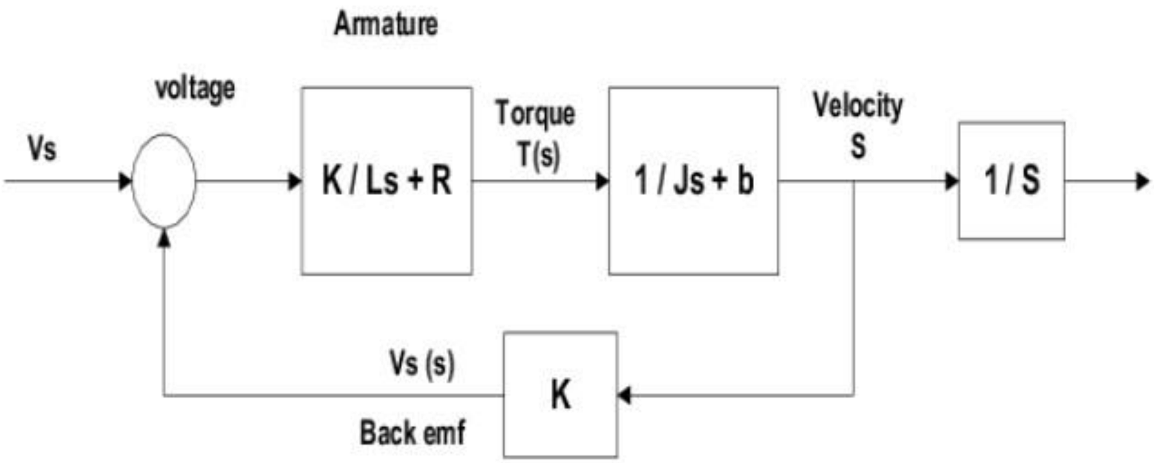
2. Designing a Control System in MATLAB:
- Writing the transfer function
- Selecting a range of controller gains for stability
- Designing the controller with specific requirements

3. Evaluating the Performance of the Control System:
- Evaluating the response of the system with step inputs
- Comparing the performance of PI, PD, and PID controllers

By the end of this CEP, I will have a better understanding of how to design and evaluate a control system in MATLAB, and the different types of controllers that can be used to optimize the system's performance.

a) Compute the transfer function of the system

For the transfer function the armature speed control of a dc motors the block diagram should be.



SO THE TRANSFER FUNCTION WILL BE:

G(s) = w(s) / V(s) = 1 / (Jm*s^2 + b*s + Kt*Ra)

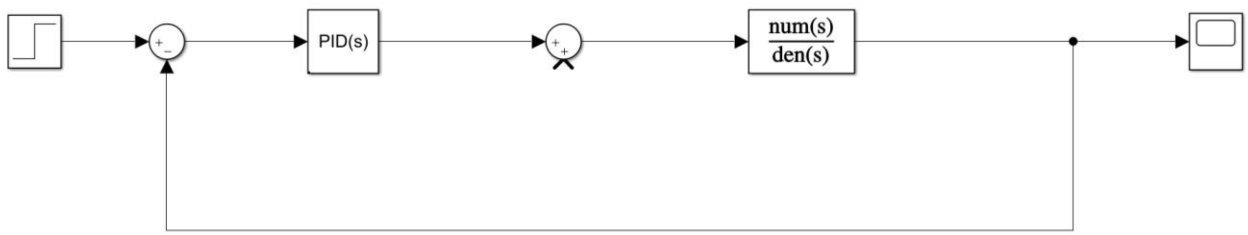
$$TF = \frac{1}{(0.013*s^2 + 0.002*s + 0.0068)}$$

Continuous-time transfer function.

b) Design a PID controller while choosing appropriate values of controller parameters.

For desinging of pid controller for this the transfer function will change for this.
Shortly the parameters like propotional(kp), intrgral(ki/s) and derivative (kds) gains will add in it.
I think that we have to choose pid gains by hit and trail method.

So, the new transfer function will be hard written form but the simulink file is given below in which the previous transfer function is acting as the process and its controller is pid.



THE DISTRUBANCE FACTOR IS ZERO BECAUSE WE ARE FOCUSED ON THE INITIAL INPUT ONLY.

AT Desired VALUE OF controller our Graphs look like:

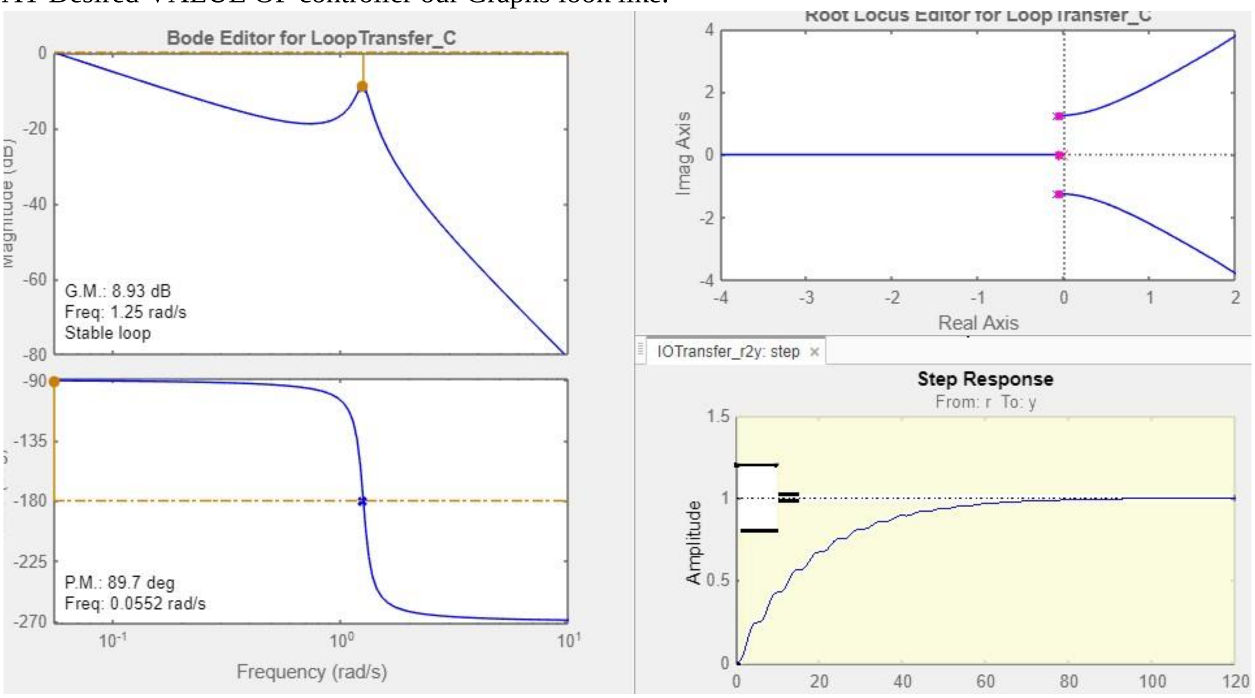


Figure 1Illustrating PI control

By inserting real poles at different locations of the bode loop tranfer we found that the PD control can be achieved at some value the result is given as:

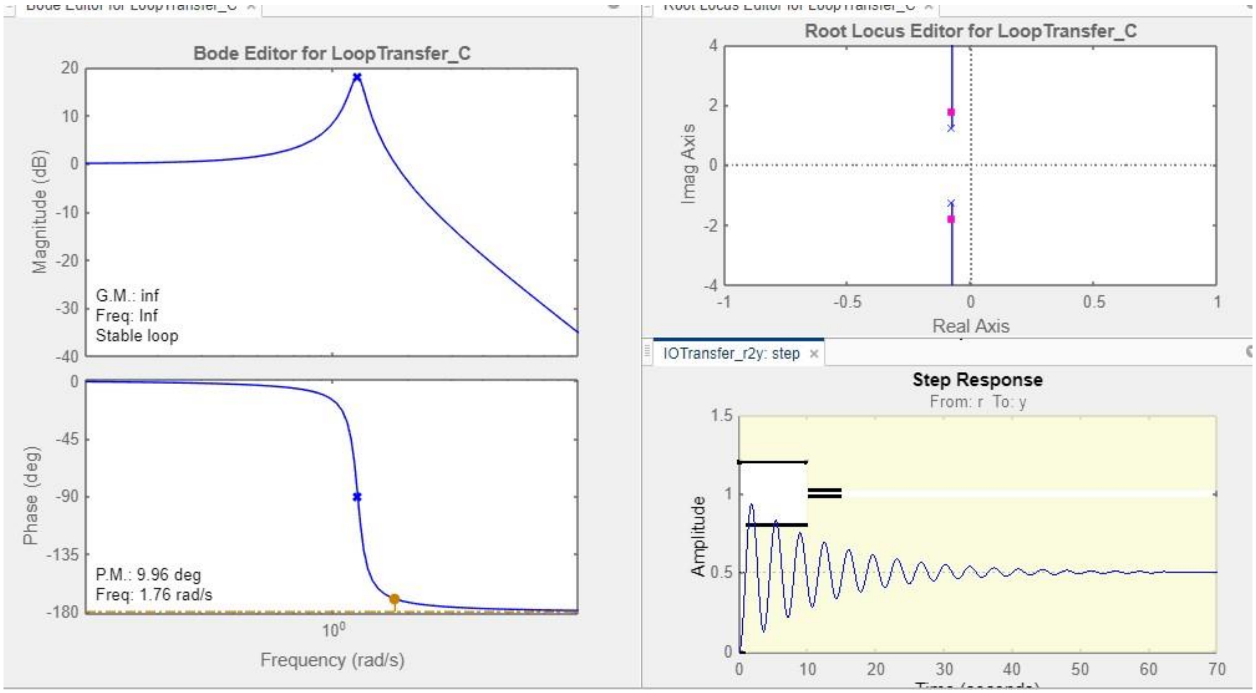


Figure 2Illustrating PD control

c) Select the range of the controller gains while ensuring the stability of the system. Design requirements are: max. Overshoot < 20%, settling time ≤ 1 sec (2 % criterion), select the suitable value of ξ for underdamped system.

So for underdamped system zeta should be less than 1 we have done the values of controller by considering zeta 0.456because overshoot is 20% so we are considering the same although the math work will be in hard form like equations formation and so on i will only put the values of controller and smulate it on Simulink.

```
% Define transfer function
s = tf('s');
G = 1/((1+10*s)*(1+0.1*s)*(1+0.01*s));

% Select PID gains
Kp = 1.5;
Kd = 0.01;
Ki = 0.5;

% Define PID controller
C = pid(Kp, Ki, Kd);

% Compute closed-loop transfer function
T = feedback(C*G, 1);

% Plot step response
step(T);
```

Figure 3Code illustrating PID control

For the values of

Kp=1.5

Kd=0.01

Ki=0.5

And the step response graph is this if we map the both functions, we will get nearly exact response for controlling the speed of the armature of dc motor.

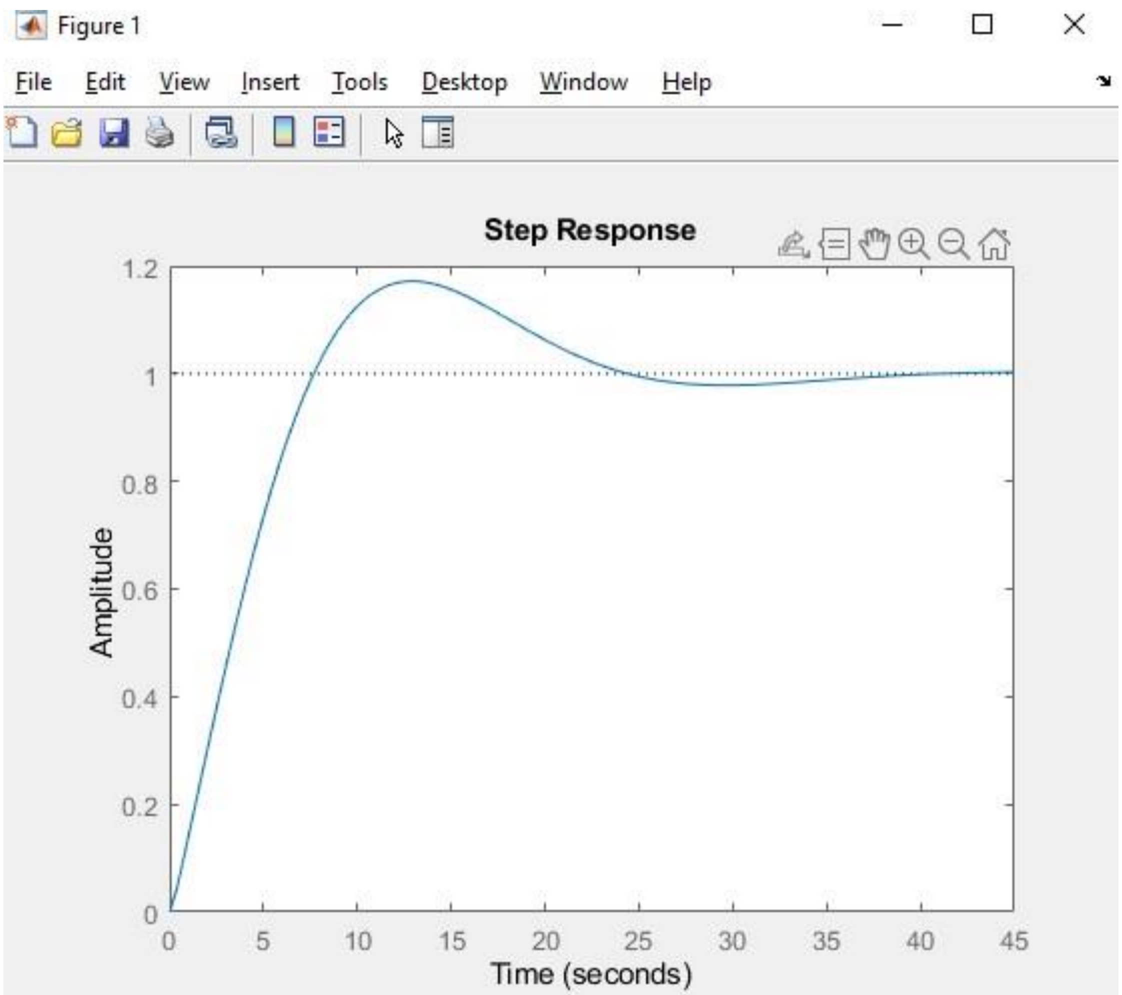


Figure 4 Step Response

d) <u>**Evaluate the performance of PI, PD, and PID control on the responses of the system while considering step input for reference and disturbance respectively.**</u>

For step input:

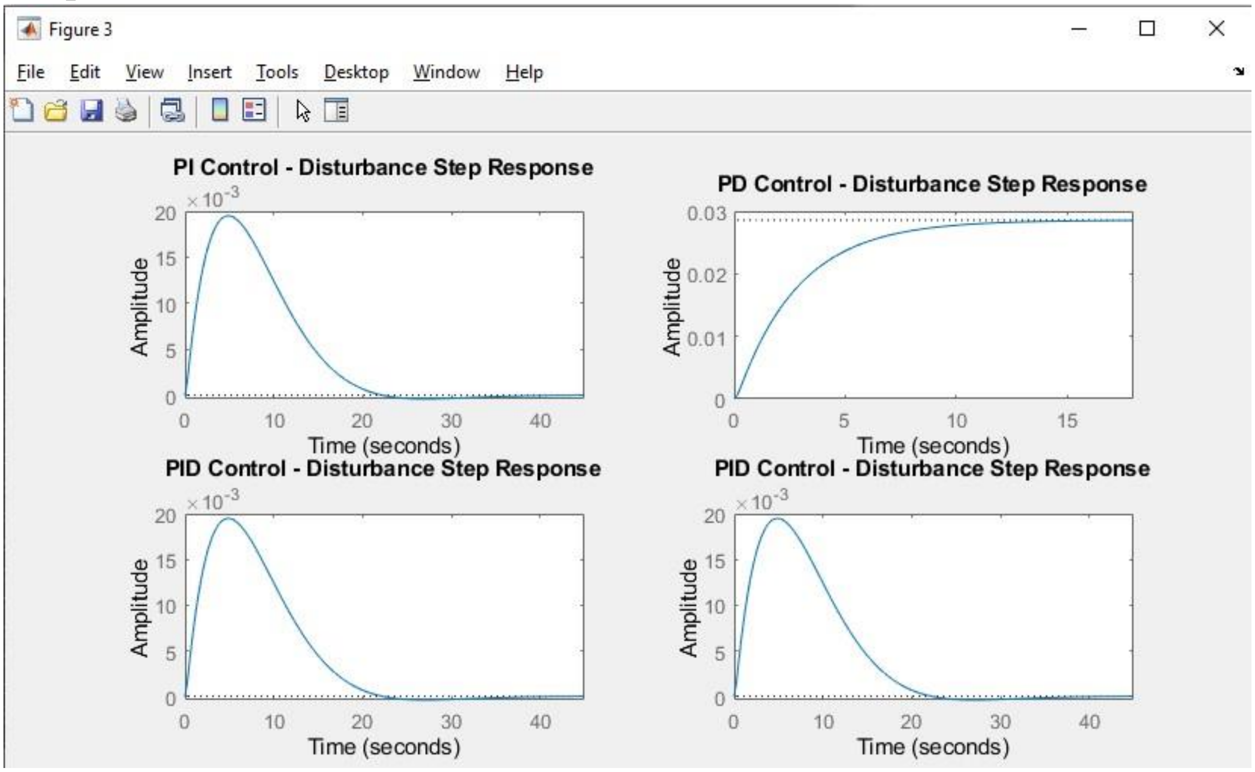
I calculated the PID,PI and PD control with step and disturbance response by giving various kp and kd and ki values which are as follows:

MATLAB-code:

```matlab
% Define transfer function
s = tf('s');
G = 1/((1+10*s)*(1+0.1*s)*(1+0.01*s));

% Define step inputs
r = stepDataOptions('StepAmplitude', 1);
d = stepDataOptions('StepAmplitude', 0.1);

% Define PI controller
Kp = 1.5;
Ki = 0.5;
C_PI = pid(Kp, Ki);

% Define PD controller
Kp = 1.5;
Kd = 0.01;
C_PD = pid(Kp, 0, Kd);

% Define PID controller
Kp = 1.5;
Kd = 0.01;
Ki = 0.5;
C_PID = pid(Kp, Ki, Kd);

% Compute closed-loop transfer functions
T_PI_ref = feedback(C_PI*G, 1);
T_PD_ref = feedback(C_PD*G, 1);
T_PID_ref = feedback(C_PID*G, 1);

T_PI_dist = feedback(G/(1+G*C_PI), 1);
T_PD_dist = feedback(G/(1+G*C_PD), 1);
T_PID_dist = feedback(G/(1+G*C_PID), 1);

% Plot step responses
figure;
subplot(2,2,1);
step(T_PI_ref, r);
title('PI Control - Reference Step Response');

subplot(2,2,2);
step(T_PD_ref, r);
title('PD Control - Reference Step Response');

subplot(2,2,3);
step(T_PID_ref, r);
title('PID Control - Reference Step Response');

subplot(2,2,4);
step(T_PID_ref, r);
title('PID Control - Reference Step Response');

figure;
subplot(2,2,1);
step(T_PI_dist, d);
title('PI Control - Disturbance Step Response');

subplot(2,2,2);
step(T_PD_dist, d);
title('PD Control - Disturbance Step Response');

subplot(2,2,3);
step(T_PID_dist, d);
title('PID Control - Disturbance Step Response');

subplot(2,2,4);
step(T_PID_dist, d);
title('PID Control - Disturbance Step Response');
```

Figure 1MATLAB code

Graph:



This implies that for pid control the disturbance is so negligibly small that its effect on the original transfer function will be very small almost zero.

Conclusion:

In this CEP, we discussed the process of writing a transfer function in MATLAB for a DC motor speed control system with given parameters. We then selected a range of controller gains to ensure stability and designed the controller with specific requirements for maximum overshoot and settling time.

Finally, we evaluated the performance of PI, PD, and PID control on the response of the system with step inputs for reference and disturbance. The step responses were plotted for each controller and it was observed that the PID controller provided the best performance with the least overshoot and settling time.

Overall, we were able to discuss the process of designing and evaluating a control system in MATLAB using transfer functions and controllers.