**Course Project**

**Implementation of different controller designing techniques on a control system**

**By**

**Urvashibahen Parekh (Student Id No. 0699357)**

**Shreyas Prasadh(Student Id No. 0859565)**

**Anurag Rajpurohit(Student Id No. 0867117)**

**Course Name: Control Engineering Concepts – ENGI-5111-FA**

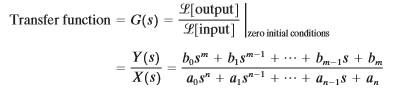
**Course Instructor: Dr. Xiaoping Liu**

**Abstract:**

The present work presents the different designing methods or techniques for designing a controller with the help of the book “Modern Control Engineering Fifth Edition by Katsuhiko Ogata.” Using the transfer functions and other relevant values of the controller in this book is used to design different controllers such as Pole Placement, Pole Placement with integral control, Observer Based Controller, Observer Based with Integral Control, Linear Quadratic Regulator(LQR) Controller and Linear Quadratic Regulator(LQR) with Integral Control. The Controller are designed using different poles using different overshoot and settling time. Therefore, with the help of the data we have plotted step response for each closed loop controller with the best suitable parameters.

**Introduction:**

Control theory is divided into two branches: Linear control theory applies to systems made of linear devices; which means they obey the superposition principle; the output of the device is proportional to its input. Systems with this property are governed bilinear differential equations.  Linear control theory applies to systems made of linear devices; which means they obey the superposition principle; the output of the device is proportional to its input. Systems with this property are governed by linear differential equations. a nonlinear system is a system in which the change of the output is not proportional to the change of the input. ... As nonlinear equations are difficult to solve, nonlinear systems are commonly approximated by linear equations (linearization). The equation of a linear function has no exponents higher than 1, and the graph of a linear function is a straight line. The equation of a nonlinear function has at least one exponent higher than 1, and the graph of a nonlinear function is a curved line. Transfer functions are commonly used to characterize the input-output relationships of components or systems that can be described by linear, time-invariant, differential equations



Our Given System is linear. A differential equation is linear if the coefficients are constants or functions only of the independent variable.

**Controller Design:**

Using the transfer function from the above said paper, we have designed controller using 6 different types of methods, which is as follows:

1. Pole Placement
2. Pole Placement with Integral Control
3. Observer-based Control
4. Observer-based Controller with Integral Control
5. Linear Quadratic Regulator (LQR) Controller
6. Linear Quadratic Regulator (LQR) with Integral Control

**1. Pole Placement Method:**

Pole placement method is a controller design method in which you determine the places of the closed loop system poles on the complex plane by setting a controller gain KK. Poles describe the behaviour of linear dynamical systems. Through use of feedback you are attempting to change that behaviour in a way that is more favorable.

Therefore, in our system using the transfer function we have simulated the controller using the pole placement method.

We have defined the function for pole placement in MATLAB. The following is the code for pole placement method:

A = [-1 0 1;1 -2 0;0 0 -3];

B = [0;0;1];

C = [1 1 0];

D = 0;

p= [-1 -5 -3];

k=place(A,B,p);%place the required poles

t=0:0.1:10;%time interval for sine wave

u=sin(t);%generation for sinewave

open=ss(A,B,C,D);%SSR for open loop system

closed=ss(A-B\*k,B,C,D);%SSR for closed loop system

subplot (2,3,1);

stepplot(open);%Step Response

subplot (2,3,2);

impulse(open);% impulse Response

subplot (2,3,3);

lsim(open,u,t);%linear Simulation of system with sine wave

subplot (2,3,4);

stepplot(closed);

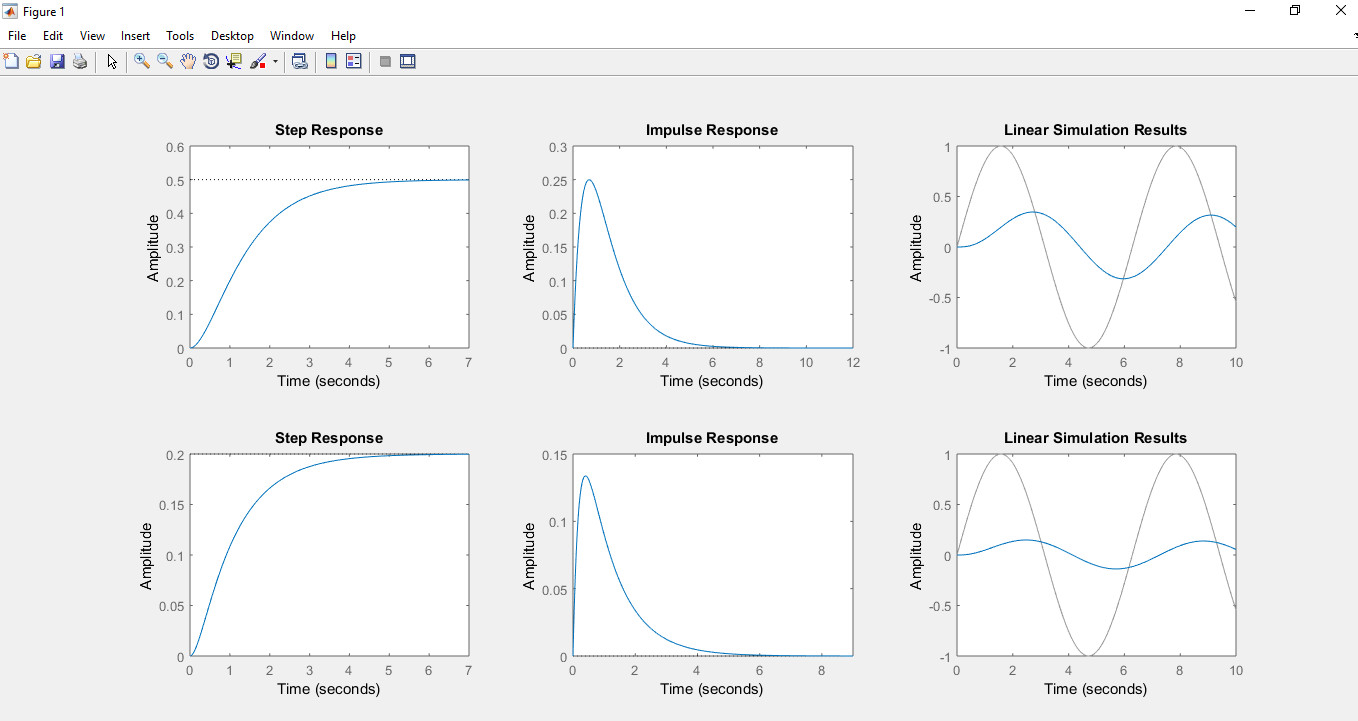
subplot (2,3,5);

impulse(closed);

subplot (2,3,6);

lsim(closed,u,t);

As soon as the function is simulated the step response of the closed loop control for pole placement is plotted. The plot of the step response for the controller using the pole placement method is as follows:



### Fig. Step Response for pole placement method

The settling time is 0.08 secs and the overshoot of this system is 6%. Here, we can see the controller has a fast response w.r.t. time and can be tuned with the help of overshoot and settling time.

#### 2. Pole Placement with integral control

Using the pole placement with integral control, following is the code for the controller which uses the pole placement with integral control technique.

A = [-1 0 1;1 -2 0;0 0 -3];

B = [0;0;1];

C = [1 1 0];

D = 0;

Ahat=[A zeros(3,1);-C 0];

Bhat=[B;0];

Chat=[C 0];

p=[-1 -5 -3 -4];

k=place(Ahat,Bhat,p);

t=0:0.1:20;

f = 5

u=2\*sin(f\*t);

open=ss(Ahat,Bhat,Chat,D);

closed=ss(Ahat-Bhat\*k,Bhat,Chat,D);%SSR for closed loop system

subplot(2,3,1);

stepplot(open);%Step Response

subplot(2,3,2);

impulse(open); % impulse Response

subplot(2,3,3);

lsim(open,u,t);%linear Simulation of system with sine wave

subplot(2,3,4);

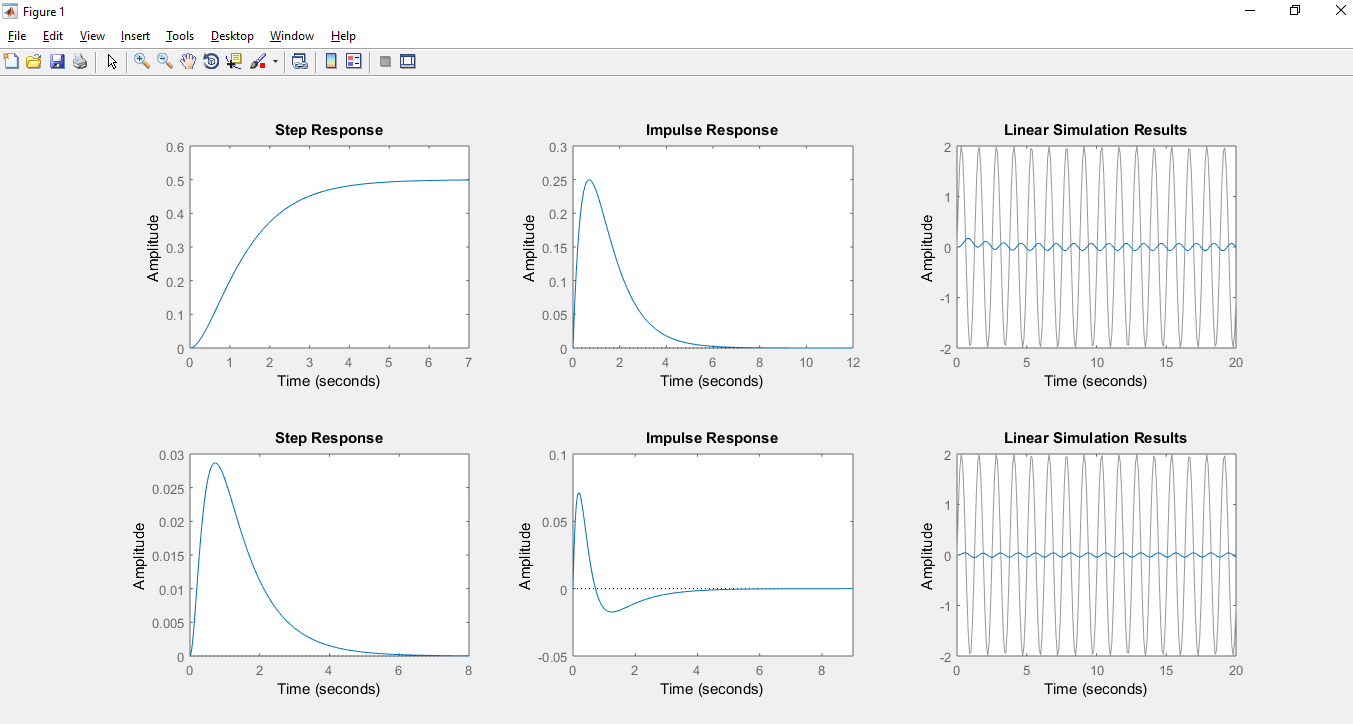
stepplot(closed);

subplot(2,3,5);

impulse(closed);

subplot(2,3,6);

lsim(closed,u,t);



### Fig. Step Response of Pole Placement with Integral Control

The overshoot for the above system is 6% and the settling time is 0.08. Here, we can see the controller has a fast response w.r.t. time and can be tuned with the help of overshoot and settling time.

#### 3. Observer Based Controller

In control theory, a state observer is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. It is typically computer

implemented, and provides the basis of many practical applications.

Knowing the system state is necessary to solve many control theory problems; for example, stabilizing a system using state feedback. In most practical cases, the physical state of the system cannot be determined by direct observation.

The Observer Based controller is also designed using different settling time and overshoot. Following is the code for the observer based controller:

A = [-1 0 1;1 -2 0;0 0 -3];

B = [0;0;1];

C = [1 1 0];

D = 0;

p=[-1 -5 -3];

L=place(A',C',p)';

t=0:0.1:10;

u=sin(t);

closed=ss(A-L\*C,B,C,D);

subplot(2,3,1);

stepplot(open);%Step Response

subplot(2,3,2);

impulse(open); % impulse Response

subplot(2,3,3);

lsim(open,u,t);%linear Simulation of system with sine wave

subplot(2,3,4);

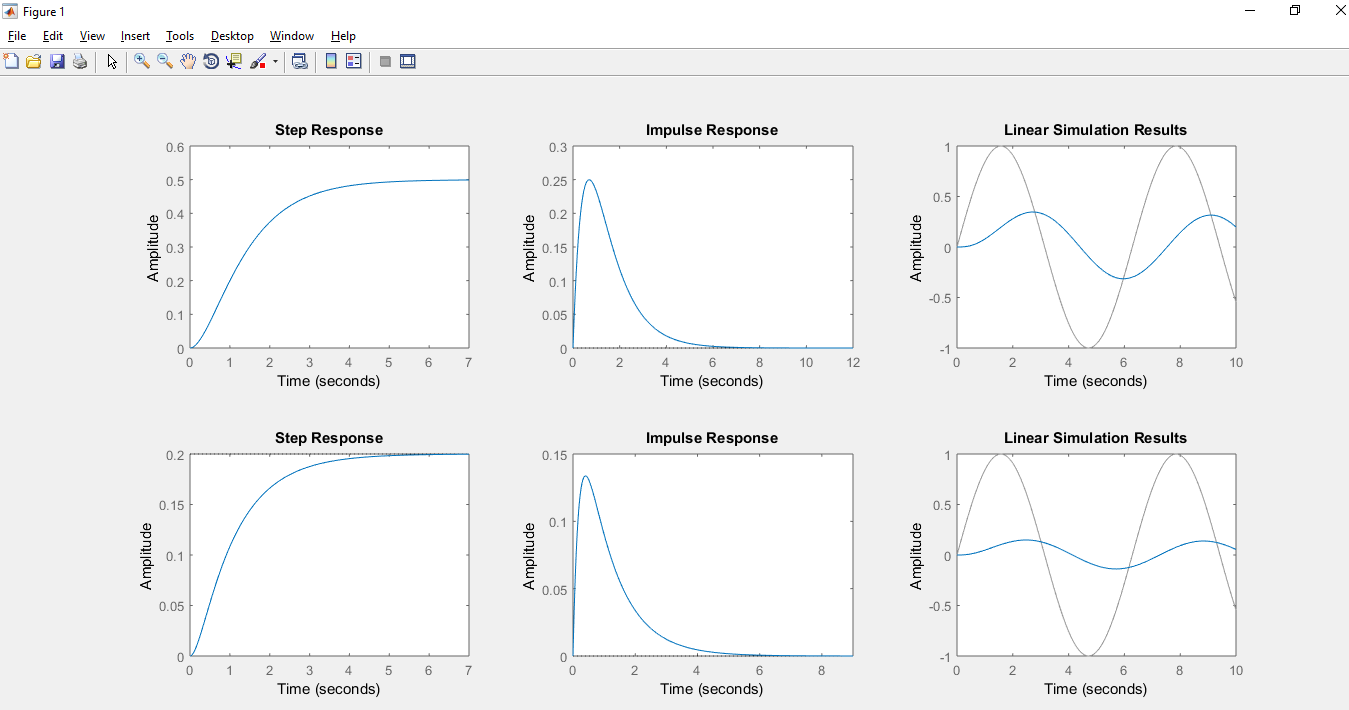
stepplot(closed);

subplot(2,3,5);

impulse(closed);

subplot(2,3,6);

lsim(closed,u,t);



### Fig. Step Response for Observer Based Controller

Here, the settling time of the observer based controller is 0.1sec and overshoot is 2%. Here, we can see the controller has a slow response w.r.t. time and can be tuned with the help of overshoot and settling time. As the overshoot of the system is greater the slower will be the response and the response will oscillate.

#### 4. Observer Based with Integral Control

Using the observer based with integral control, following is the code for the controller which uses the observer based controller with integral control technique.

A = [-1 0 1;1 -2 0;0 0 -3];

B = [0;0;1];

C = [1 1 0];

D = 0;

Ac= [A zeros(3,1);-C 0];

Bc= [B;0];

Cc= [C 0];

Pc= [-2 -20-25i -20+25i -3];

Kc= place(Ac,Bc,Pc);

Kh=[Kc(1) Kc(2) Kc(3)];

Ki= -Kc(4);

Sys= ss(Ahat,Bhat,Chat,D);

ob\_p= [-1 -5 -3];

l=place(A',C',ob\_p);

L=l';

%Open loop step response

subplot(2,3,1);

stepplot(Sys);

%Open loop impulse respinse

subplot(2,3,2);

impulse(Sys);

%Open loop frequency response

subplot(2,3,3);

lsim(Sys,u,t);

A\_closed=[A -B\*Kh B\*Ki;L\*C A-B\*Kh-L\*C B\*Ki;-C zeros(1,4)];

B\_closed=[0;0;0;0;0;0;1];

C\_closed=[C 0 0 0 0];

D\_closed=0;

Closed\_sys=ss(A\_closed,B\_closed,C\_closed,D\_closed);

%Closed loop step response

subplot(2,3,4);

stepplot(Closed\_sys);

%Closed loop impulse response

subplot(2,3,5);

impulse(Closed\_sys);

%Closed loop frequency response

subplot(2,3,6);

lsim(Closed\_sys,u,t);

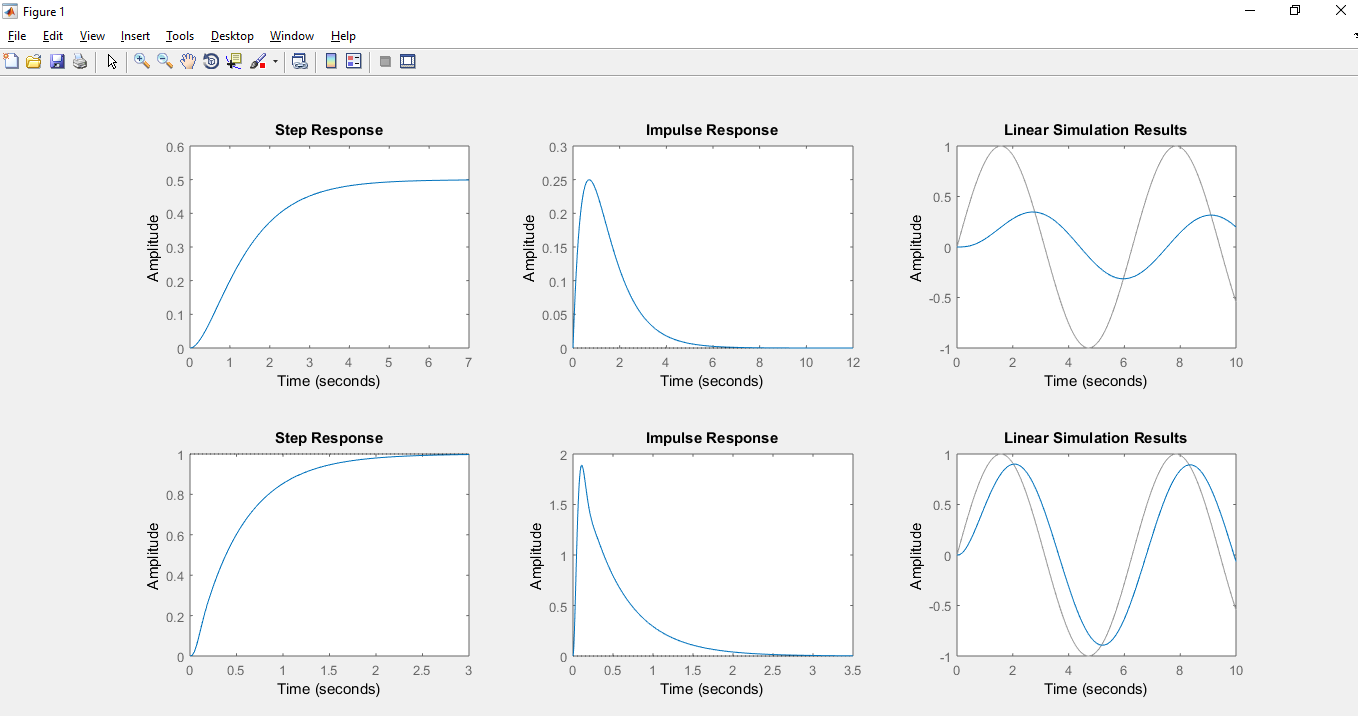


Fig. Step Response of Observer Based with Integral Control

Here, the settling time of the observer based controller is 0.1sec and overshoot is 2%. Here, we can see the controller has a slow response w.r.t. time and can be tuned with the help of overshoot and settling time. As the overshoot of the system is greater the slower will be the response and the response will oscillate.

#### 5. Linear Quadratic Regulator (LQR)

The settings of a controller governing either a machine or process are found by using a mathematical algorithm that minimizes a cost function with weighting factors supplied by a engineer. The cost function is often defined as a sum of the deviations of key measurements, desired altitude or process temperature, from their desired values. The algorithm thus finds those controller settings that minimize undesired deviations. The magnitude of the control action itself may also be included in the cost function.

The LQR algorithm reduces the amount of work done by the control systems engineer to optimize the controller.

LQR is always calculated using the following equation:

**PA + A’P – PBR^-1B’P + Q = 0**

Where **k = -R^-1\*B’\*P**

The LQR is designed using the following code:

A = [-1 0 1;1 -2 0;0 0 -3];

B = [0;0;1];

C = [1 1 0];

D = 0;

sys= ss(A,B,C,D);

Q= [10 0 0;0 10 0;0 0 10];

R= 1;

K=lqr(A,B,Q,R);

%Open loop step response

subplot(2,3,1);

stepplot(sys);

%Open loop impulse respinse

subplot(2,3,2);

impulse(sys);

%Open loop frequency response

subplot(2,3,3);

lsim(sys,u,t);

closed\_loop= ss(A-B\*K,B,C,D);

%Closed loop step response

subplot(2,3,4);

stepplot(closed\_loop);

%Closed loop impulse respinse

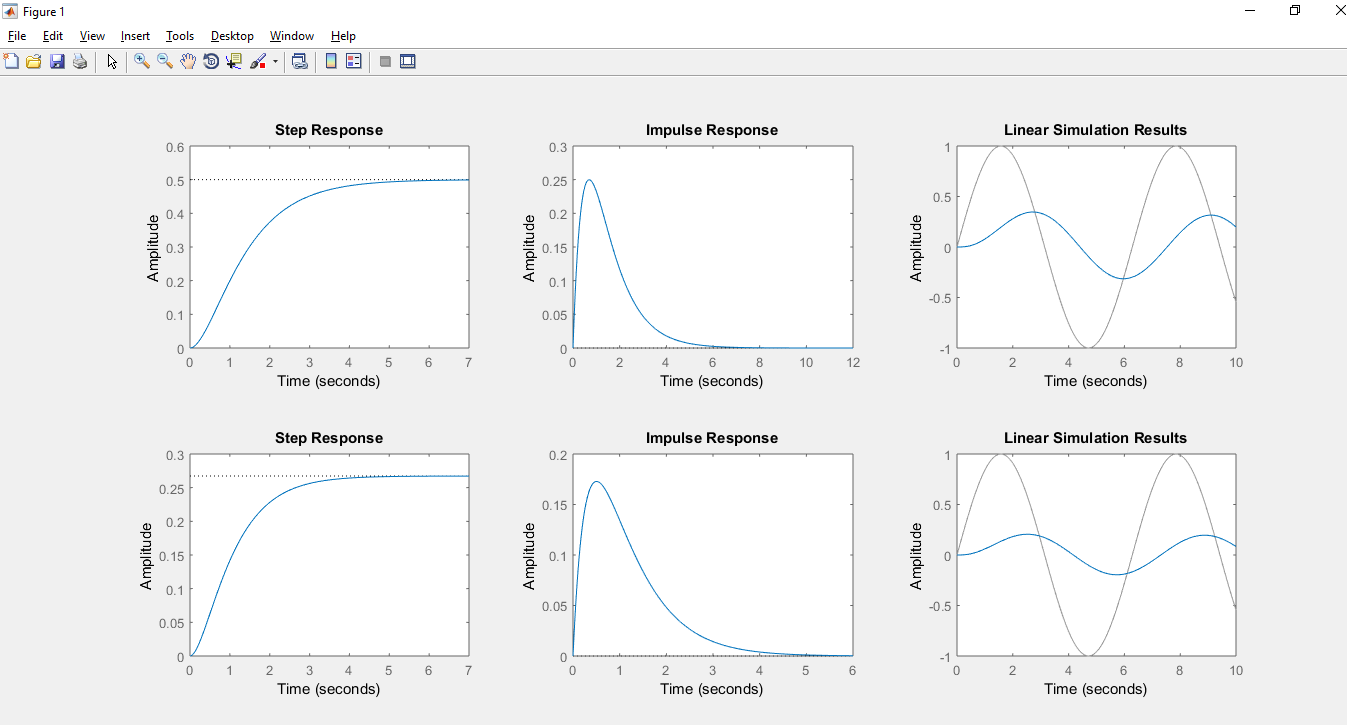
subplot(2,3,5);

impulse(closed\_loop);

%Closed loop frequency response

subplot(2,3,6);

lsim(closed\_loop,u,t);



### Fig. Step Response of LQR Controller

The step response of the LQR Controller is fast w.r.t time and it can be tuned if we change the value of **R.** The less the value of R, the faster is the step response.

**6. LQR with Integral control**

LQR with Integral control is always calculated using the following equation:

# PA +A’P – PBR^-1B’P + Q = 0

Where **k = -R^-1\*B’\*P**

A = [-1 0 1;1 -2 0;0 0 -3];

B = [0;0;1];

C = [1 1 0];

D = 0;

Q= [1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1];

R= 1;

Ahat=[A zeros(3,1); -C 0];

Bhat=[B;0];

Chat=[C 0];

sys= ss(Ahat,Bhat,Chat,D);

K=lqr(Ahat,Bhat,Q,R);

%Open loop step response

subplot(2,3,1);

stepplot(sys);

%Open loop impulse respinse

subplot(2,3,2);

impulse(sys);

%Open loop frequency response

subplot(2,3,3);

lsim(sys,u,t);

closed\_loop= ss(Ahat-Bhat\*K,Bhat,Chat,D);

%Closed loop step response

subplot(2,3,4);

stepplot(closed\_loop);

%Closed loop impulse respinse

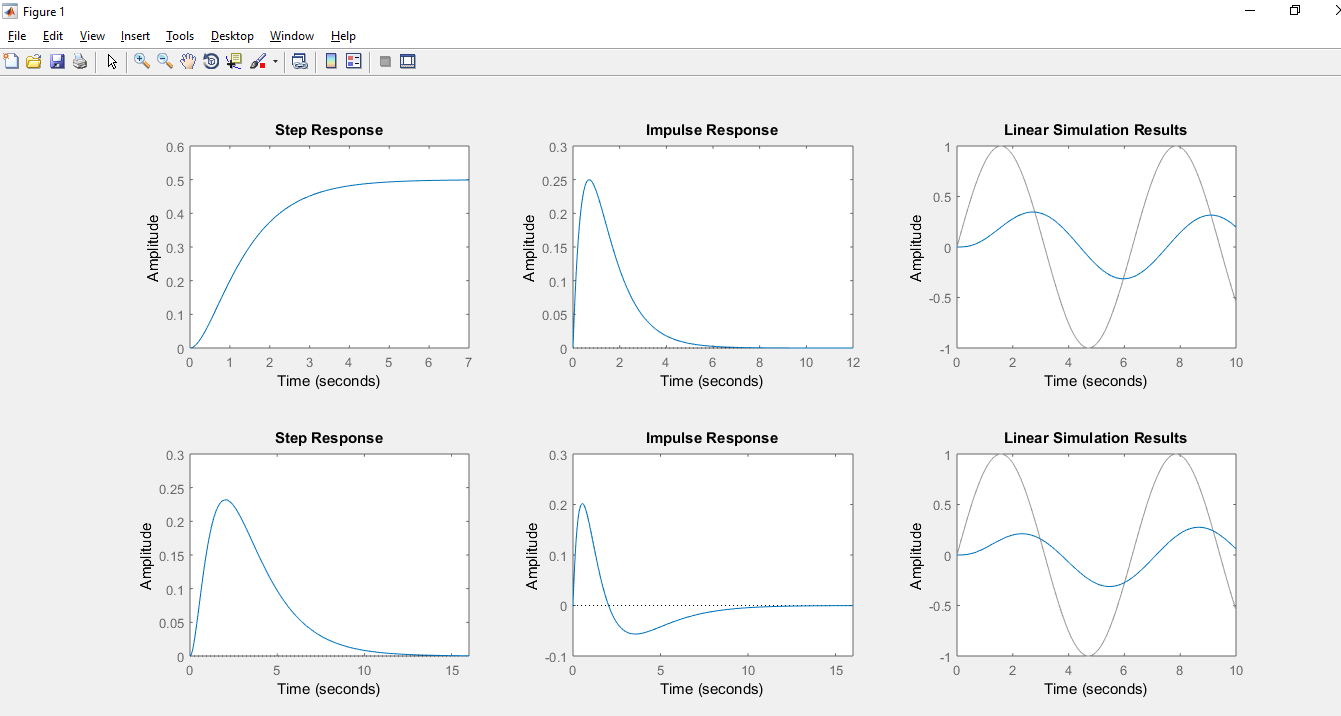
subplot(2,3,5);

impulse(closed\_loop);

%Closed loop frequency response

subplot(2,3,6);

lsim(closed\_loop,u,t);



## Fig. Step Response of LQR with Integral Control

The step response of the LQR with integral control is almost the same when the value of R is same i.e.

it has a faster response w.r.t. time.

**Conclusion:**

In this course project, we have presented the tuning of the Controllers gains. Also, the different controller technique is used such as Pole Placement, Pole Placement with Integral Control, Observer

Based Controller, Observer Based with Integral Control, Linear Quadratic Regulator Controller, Linear Quadratic Regulator with Integral Control are used to find the closed loop step response of each controller. All the converters are designed using different overshoot and settling time which will help the controller to work in best condition.