CONTROL SYSTEM PROJECT

Group Members:
Aditya Rana - 110118002
Subhankar Biswas - 110118084
Vishal Mandrai - 110118100

PROBLEM STATEMENT

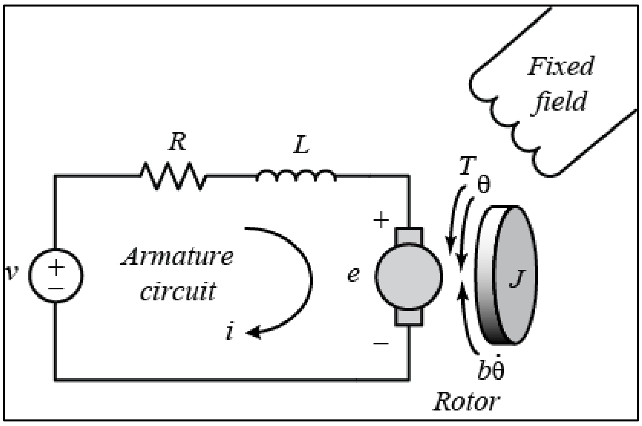
The purpose of this project was to control the angular rate of the load (shaft position) of a DC motor by varying the applied input voltage.

For a step input of 1 rad/sec, we will design a controller to manipulate motor speed as per our need.

INTRODUCTION

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in the below figure.

For this example, we will assume that the input of the system is the voltage source (V) applied to the motor's armature, while the output is the rotational speed of the shaft $\dot{\theta}$. The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity.



The physical parameters for our example are:

- moment of inertia of the rotor (J) = 0.02 kg.m^2/sec^2
- motor viscous friction constant (b) = 0.2 N.m.sec
- electromotive force constant (K_e) = 0.02 V/rad/sec
- motor torque constant (K_t) = 0.02 N.m/Amp
- electric resistance (R) = 2 Ohm
- electric inductance (L) = 0.4 H

System Equations:

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. In this example we will assume that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current i by a constant factor K_t as shown in the equation below. This is referred to as an armature-controlled motor.

$$T = K_t i \tag{1}$$

The back emf, e, is proportional to the angular velocity of the shaft by a constant factor K_e.

$$e = K_e\dot{\theta} \qquad [2]$$

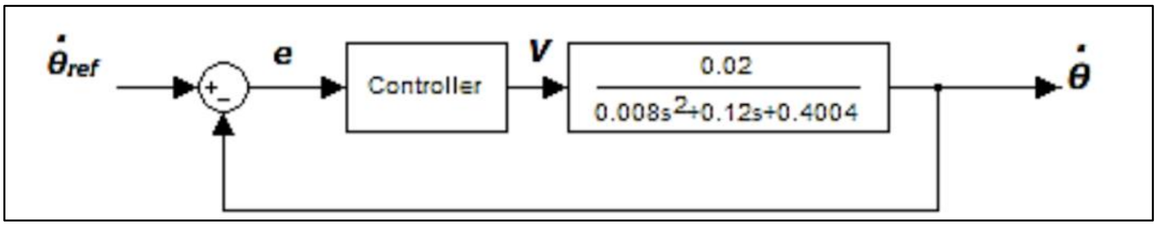
In SI units, the motor torque and back emf constants are equal, that is, $K_t = K_e$; therefore, we will use K to represent both the motor torque constant and the back emf constant.

From the figure above, we can derive the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law.

$$J\ddot{\theta} + b\dot{\theta} = Ki \qquad [3]$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \qquad [4]$$

Block diagram of the closed loop system labelling all the signals



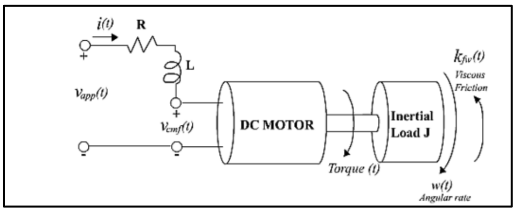
Closed loop block diagram

Transfer Function:

Applying the Laplace transform, the above modelling equations can be expressed in terms of the Laplace variable *s*.

$$s(Js + b)\Theta(s) = KI(s) \qquad [5]$$

$$(Ls + R)I(s) = V(s) - Ks\Theta(s) \qquad [6]$$



A simple model of a DC motor driving an inertial load

We arrive at the following open-loop transfer function by eliminating $I(s)$ between the two above equations, where the rotational speed is considered the output and the armature voltage is considered the input.

$$P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \qquad [\frac{rad/sec}{V}] \qquad [7]$$

Entering the above transfer function in Matlab:

```
% Open loop Transfer function of DC motor
clc; clear;
J=0.02;
b=0.2;
kt=0.02;
ke=0.02;
R=2;
L=0.4;
num=[kt];
den=[J*L J*R+b*L b*R+ke*kt ];
TF_DC=tf(num,den)
```

$$\frac{\dot{\boldsymbol{\theta}}(\boldsymbol{s})}{\boldsymbol{V}(\boldsymbol{s})}=\frac{\mathbf{0.02}}{\mathbf{0.008}\boldsymbol{s}^{2}+\mathbf{0.12}\boldsymbol{s}+\mathbf{0.4004}}$$

The closed loop transfer function that indicates the relationship between $\dot{\theta}_{ref}$ and $\dot{\theta}(t)$ can be determined from the following block diagram in the below figure.

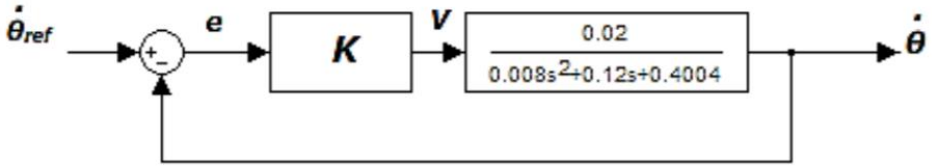


Figure 6.Closed loop block diagram of DC motor

$$\frac{\dot{\theta}_{out}}{\dot{\theta}_{ref}}=\frac{K.G}{1+KG}$$

$$=\frac{K\dfrac{0.02}{0.008s^{2}+0.12s+0.4004}}{1+K\dfrac{0.02}{0.008s^{2}+0.12s+0.4004}}$$

$$\frac{\dot{\boldsymbol{\theta}}_{\boldsymbol{out}}}{\dot{\boldsymbol{\theta}}_{\boldsymbol{ref}}}=\frac{\mathbf{0.02}\boldsymbol{K}}{\mathbf{0.008}\boldsymbol{s}^{2}+\mathbf{0.12}\boldsymbol{s}+(\mathbf{0.02}\boldsymbol{K}+\mathbf{0.4004})} \quad (8)$$

State Space Representation:

We can also represent the system using the state-space equations. The following additional MATLAB commands create a state-space model of the motor and produce the output shown below when run in the MATLAB command window.

```
% State Space representation
A = [-(R/L) -(ke/L) ; (kt/J) -(b/J)];
B = [1/L; 0];
C = [0 1];
D = 0;
motor_ss = ss(A , B , C , D)
```

The above state-space model can also be generated by converting your existing transfer function model into state-space form. This is again accomplished with the ss command as shown below.

```
motor_ss = ss(TF_DC);
```

The open loop and closed loop responses of the DC motor without any controller are shown in below figures:

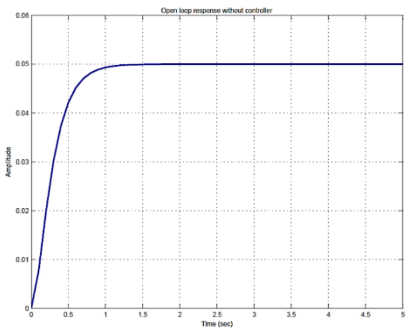


Figure 7. Open loop response of DC motor without controller

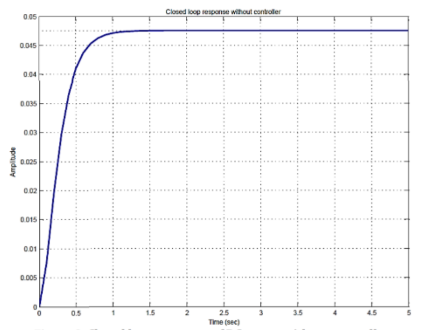


Figure 8. Closed loop response of DC motor without controller

As it can be seen from the system response, we need a controller to greatly improve performance, i.e. steady state and settling time. Before designing the controller, we need to check the stability of the system as well as controllability and observability.

Check the stability of the open-loop and closed-loop systems:

The Routh-Hurwitz criterion which uses the coefficients of the characteristic equation was used to test the stability of the system. For this reason, the following Matlab code was written and used.

```
=======================================================
%Routh-Hurwitz Stability Criterion
% The Routh-Hurwitz criterion states that the number of roots of D(s)
with positive real part is equal to the number of changes in sign of the
first column of the root array.
% The necessary and sufficient requirement for a system to be "Stable"
is that there should be no changes in sign in the first column of the
Routh array. ======================================================
clc;
disp('         ')
D=input('Input coefficients of characteristic equation,i.e:[an an-1 an-2
... a0]= ');
l=length (D);
 disp(' ')
disp('----------------------------------------')
disp('Roots of characteristic equation is:')
roots(D)
%%                        Program Begin
```

```
%%=======================Program Begin=========================
%%----------------Begin of Building array---------------------
if mod(l,2)==0
         m=zeros(l,l/2);
         [cols,rows]=size(m);
         for i=1:rows
              m(1,i)=D(1,(2*i)-1);
              m(2,i)=D(1,(2*i));

          end

    else

m=zeros(l,(l+1)/2);

[cols,rows]=size(m);

for i=1:rows

    m(1,i)=D(1,(2*i)-1);

end


  for i=1:((l-1)/2)
         m(2,i)=D(1,(2*i));
    end
  end
  for j=3:cols
    if m(j-1,1)==0
         m(j-1,1)=0.001;
    end
    for i=1:rows-1
    m(j,i)=(-1/m(j-1,1))*det([m(j-2,1) m(j2,i+1);m(j-1,1) m(j-1,i+1)]);
    end
  end

  disp('--------The Routh-Hurwitz array is:--------'),m
  % --------------------End of Bulding array--------------
  % Checking for sign change

  Temp=sign(m);a=0;
  for j=1:cols
    a=a+Temp(j,1);
  end
  if a==cols
     disp(' ----> System is Stable <----')
  else
    disp(' ----> System is Unstable <----')
  end
```

Checking for controllability and observability

The following Matlab code was written to test the controllability and observability of the system.

Checking the stability of the open-loop transfer function in Matlab using the above code:

$$\Delta = 0.008s^2 + 0.12s + 0.4004$$

```
Input coefficients of characteristic equation,i.e:[an
an-1 an-2 ... a0]= [0.08 0.12 0.4004]
----------------------------------------
Roots of characteristic equation is:
ans =
  -0.7500 + 2.1077i
  -0.7500 - 2.1077i
--------The Routh-Hurwitz array is:--------
m =
    0.0800    0.4004
    0.1200         0
    0.4004         0
          ----> System is Stable <----
```

```
% Checking Controlability and Observability
if det(ctrb(A,B))==0
    disp('          ----------> System is NOT
Controllable <----------')
else
    disp('          ----------> System is Controllable
<----------')
end
if det(obsv(A,C))==0
    disp('          ----------> System is NOT Observable
<----------')
else
    disp('          ----------> System is Observable
<----------')
end

     ----------> System is Controllable <----------
     ----------> System is Observable   <----------
```

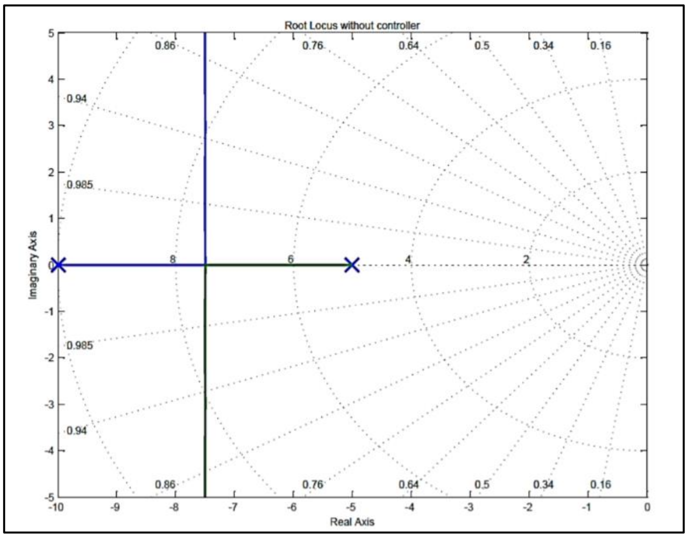
The system is both Controllable and Observable.

Draw the root locus of the given system

The root locus of the DC motor transfer function is shown in Figure 9. It can be seen that we have two real poles at P1 = -5.01 and P2 = -9.99 which repel each other at -7.5 and one goes to positive infinity and the other goes to negative infinity.

```
% Drawing Root Locus of the
given System
>> rlocus(num,den)
```

Root locus plot of DC motor transfer function



Design a Lead-Lag Compensator

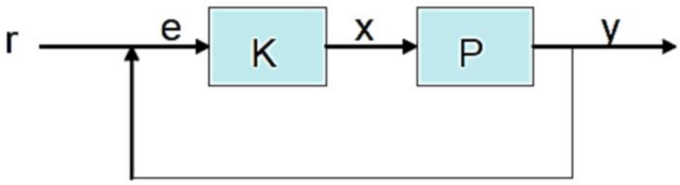


Figure 10. Closed loop block diagram

$$P(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{0.02}{0.008s^2 + 0.12s + 0.4004} = \frac{2.5}{s^2 + 15s + 50.05}$$

From the closed loop block diagram in the above figure and the system transfer function, the actual characteristic equation of the closed loop system is:

$$\boldsymbol{\Delta_A = s^2 + 15s + 50.05 = (s + 9.99)(s + 5.01)}$$

Design criteria are:

Settling time ≤ 1sec
Overshoot ≤ 5%
Steady-state error ≤ 0.4%.
Using the equations for settling time and percent overshoot. We can determine the desired damping ratio (ζ) and natural frequency (ω_n).

The calculated values of Damping ratio and Natural Frequency are:

$\zeta = 0.69010673$ $\omega_n = 5.7962$

Using the desired natural frequency and damping ration, the desired characteristic equation to achieve 1 second settling time and maximum 5% of overshoot will become:

$$\Delta_D = s^2 + 2\zeta\omega_n s + \omega_n^2$$
$$\boldsymbol{\Delta_D = s^2 + 8s + 33.5960 = (s + 4 + 4.194j)(s + 4 - 4.194j)}$$

Now our goal is to place closed loop poles at $S_{1,2} = -4 \pm 4.194j$. In order to do that, we should first check if we need a simple constant gain (K) or a lead/lag compensator to place poles at the desired locations. This can be checked either from root-locus or angle condition. It can be seen from root locus diagram, that the root locus does not go through the desired poles.

Using the angle condition, we can also see that the desired poles does not satisfy the angle condition for the actual characteristic equation.

$$\angle P(s) = \angle -1 = \pm 180(2q+1), \qquad \forall q = 0,1,2,3,\ldots$$

or

$$\angle zeros - \angle poles = \pm 180$$

$$(-\angle s + 9.99 - \angle s + 5.01)|_{-4+4.194j}$$

$$= -\angle 5.99 + 4.194j - \angle 1.01 + 4.194j$$

$$-tg^{-1}\left(\frac{4.194}{5.99}\right) - tg^{-1}\left(\frac{4.194}{1.01}\right)$$

$$= -34.99 - 76.45 = -111.44 \neq \pm 180$$

From root locus and the location of desired closed loop pole, it can be found that a lag compensator is needed to shift the current root locus to right.

The transfer function of a lag compensator is of the form

$$G_1(s) = K\frac{s+a}{s+b}, \qquad |b| < |a|.$$

Multiplying the lag compensator transfer function to the DC motor TF yields;

$$G_1(s)P(s) = \frac{2.5K(s+a)}{(s+b)(s^2+15s+50.05)}$$

Using coefficient matching, we can place the closed loop poles at the desired locations;

$$\Delta_A = \Delta_D$$

$$\Delta_A = (s+b)(s^2+15s+50.05) + 2.5K(s+a)$$

$$= \boldsymbol{s^3 + (15+b)s^2 + (15b + 50.05 + 2.5K)s + (50.05b + 2.5Ka)}$$

$$\Delta_D = (s^2 + 8s + 33.595)(s+c)$$

$$= \boldsymbol{s^3 + (8+c)s^2 + (33.595 + 8c)s + 33.595c}$$

$$15 + b = 8 + c$$

$$15b + 50.05 + 2.5K = 33.595 + 8c$$

$$50.05b + 2.5Ka = 33.595c$$

Since we have three equations and four unknowns $(\mathrm{a,b,c,K})$, we have one degree of design flexibility. Let's assume $\mathbf{a = 14}$ therefore:

$$\boldsymbol{K = 4.8832}, \qquad \boldsymbol{b = 3.9054}, \qquad \boldsymbol{c = 10.9054}$$

and

$$G_1(s) = 4.8832\frac{s+14}{s+3.9054}$$

Checking for steady state error (SSE< 0.1):

$$E(s) = \frac{R(s)}{1+G_1(s)P(s)}$$

where step input: $R(s) = \frac{1}{s}$.

Final value theorem:

$$\lim_{s\to 0} sE(s) = \lim_{s\to 0}\frac{sR(s)}{1+G_1(s)P(s)}$$

$$= \lim_{s\to 0}\frac{1}{1+4.8832\frac{s+14}{s+3.9054}\frac{2.5}{s^2+15s+50.05}} = 0.533 > 0.004$$

Since the steady state error is more than the design limit, we need to add another lag compensator to achieve the design criteria. The transfer function of the 2nd lag compensator is of the form:

$$G_2(s) = \frac{s+C}{s+D}$$

$$\lim_{s\to 0}\frac{1}{1+4.8832\frac{s+14}{s+3.9054}\frac{2.5}{s^2+15s+50.05}\frac{s+C}{s+D}} = 0.004$$

Assuming the zero of the lag compensator $\boldsymbol{C = 2.9}$, we have:

$$\left.\frac{1}{1+4.8832\frac{s+14}{s+3.9054}\frac{2.5}{s^2+15s+50.05}\frac{s+2.9}{s+D}}\right|_{s=0} = 0.004$$

$\boldsymbol{D = 0.02174}$

and

$$G_2(s) = \frac{s+2.9}{s+0.02174}$$

So, the transfer function of the overall lag compensator becomes:

$$\boldsymbol{G(s) = G_1(s)G_2(s) = \left(4.8832\frac{s+14}{s+3.9054}\right)\left(\frac{s+2.9}{s+0.02174}\right)}$$

Testing the performance of the lag controller:

The following Matlab code was written to test the performance of the system with the designed lag controller. The corresponding plots are shown in figure 6 through 12.

```
% Design Criteria
Ts=1;                    % Settling time < 1 second
PO=0.05;                 % Overshoot < 5%
SSE=0.4;                 % Steady state error < 0.4%

abs(roots([1+(((-log(PO))/pi)^2) 0 -(((-
log(PO))/pi)^2)]));      % Damping ratio
Damp=ans(1);
Wn=4/(Ts*Damp);          % Natural frequency

disp('Desired Damping ratio is:'),Damp
disp('Desired Natural Frequency is:'),Wn

% Desired Characteristic Equation:
dend=[1 2*Wn*Damp Wn^2];
disp('Desired Characteristic Equation is:'),dend

% Desired Poles location
Dp=roots(dend);
disp('Desired Pole locations:'),Dp

% From root locus and the location of desired closed loop pole,
it can be found that a lag compensator is needed to shift the
current root locus to right.

%Designing Lag compensator to meet the desired Settling time
and Overshoot
% --------------------------------------------------%
z1 = 14;           % Assuming zero of the first lag compensator

% Finding pole of the first lag compensator
num=num/den(1)
den=den/den(1)
ANS=inv([den(1) -dend(1) 0;den(2) -dend(2) num(1);den(3)-dend(3)
num(1)*z1])*[dend(2)-den(2);dend(3)-den(3);0];

disp('Pole of the first lag compensator is:')
p1=ANS(1)
c=ANS(2);
disp('Gain of the first lag compensator is:')
K=ANS(3)

% TF of the first lag compensator G1(s)=K(s+z1)/(s+p1)
numlag1=K*[1 z1];
denlag1=[1 p1];
disp('Transfer function of the first Lag compensator to
improve Ts and PO%:')
tf(numlag1,denlag1)
```

```
% DC motor Transfer function with Lag compensator
disp('DC motor Transfer function with Lag compensator')
NUM=conv(numlag1,num);
DEN=conv(denlag1,den);
TF=tf(NUM,DEN)
figure
rlocus(TF),grid on

% Open loop responce of the system with Lag compensator 1
figure
step(TF,0:0.1:5),grid on
title('Open loop response with lag compensator 1')

% Closed loop responce of the system with Lag Compensator 1
[numc,denc]=cloop(NUM,DEN)
figure
step(numc,denc,0:0.1:5),grid on
title('Closed loop response with Lag compensator 1 that
improves Ts & PO%')

% Improving SSE by adding a second lag compensator
z2=2.9;           % Assuming zero of the 2nd lag compensator
SSE=0.004;        % Steady State Error design criteria

% Solving for pole of the 2nd lag compensator
p2=(1+((K*z1*num(1)/denlag1(2))/den(3)))*z2*SSE
numlag2=[1 z2];
denlag2=[1 p2];
NumLag=conv(numlag1,numlag2);
DenLag=conv(denlag1,denlag2);

disp('The 2nd Lag compensator Transfer function to
improve SSE:')
tf(numlag2,denlag2)

disp('The overal Lag compensator transfer function
(lag1*lag2):')
tf(NumLag,DenLag)

% DC motor transfer function with Lag compensator that improves Ts,
PO% & SSE
NumDC=conv(NumLag,num);
DenDC=conv(DenLag,den);
disp('Open loop TF of the DC motor with final Lag
compensator (improved Ts, PO% & SSE) ')
tf(NumDC,DenDC)

% Closed loop TF of the DC motor with Lag compensator
[NumCLP,DenCLP]=cloop(NumDC,DenDC);
disp('closed loop TF of the DC motor with final Lag
compensator (improved Ts, PO% & SSE) ')
tf(NumCLP,DenCLP)
figure
step(NumCLP,DenCLP,0:0.1:5), grid on
title('Closed loop response with final Lag compensator')

%-------------------End of Lag compensator Design-----%
```

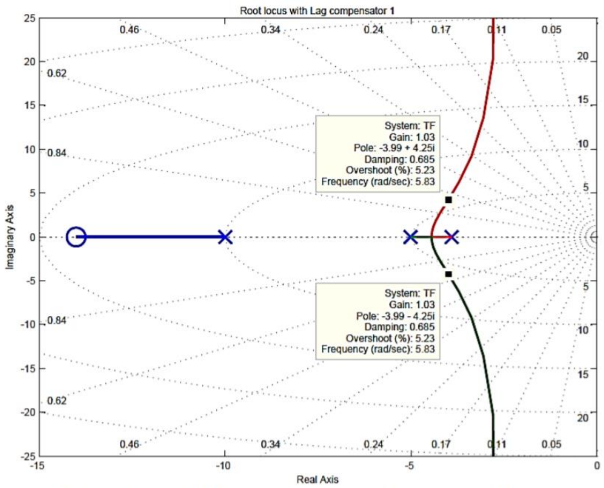


Figure 11. Root locus with lag compensator to improve settling time and percent overshoot.

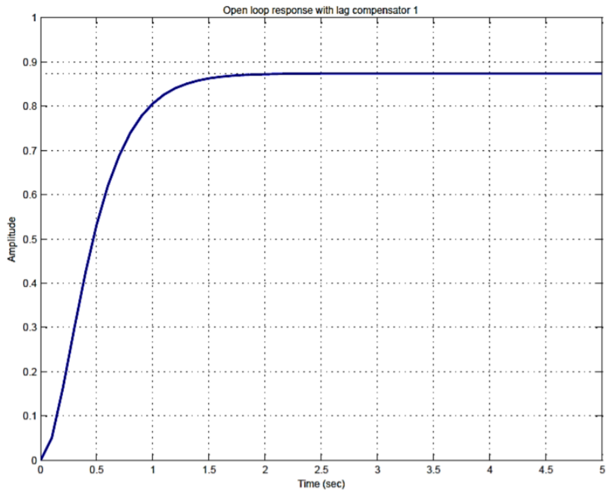


Figure 12. Open loop response of DC motor with Lag compensator1, improved settling time and percent overshoot

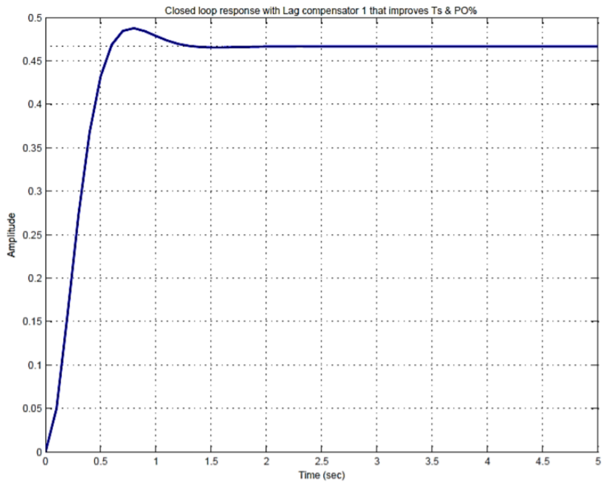


Figure 13. Closed loop response of DC motor with final Lag compensator1

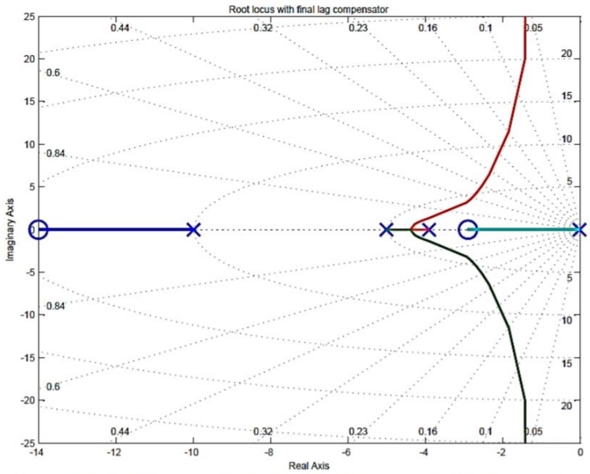


Figure 14. Root locus with the final lag compensator that improves settling time, percent overshoot and steady state error.

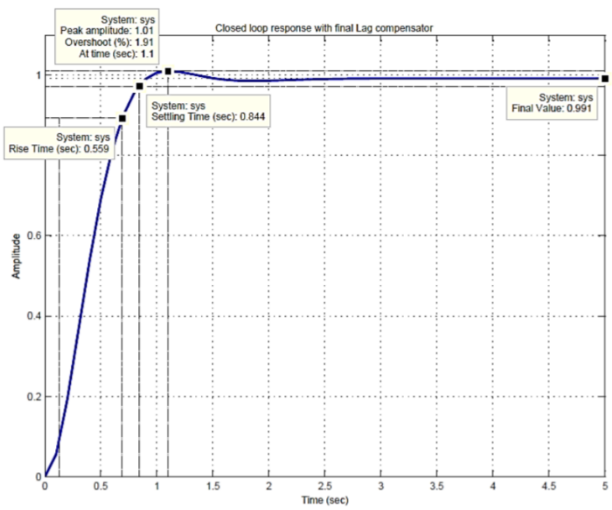


Figure 15. Closed loop response with the final lag compensator that improves settling time, percent overshoot and steady state error.

<u>Result of Lag compensator:</u>

- Settling Time = **0.844 < 1**
- P.O. % = **1.91% < 5%**
- Final value = **0.991** (Steady state error < 0.4%)

Using Bode Plots Determine the Gain and Phase Margins for The Closed-Loop System

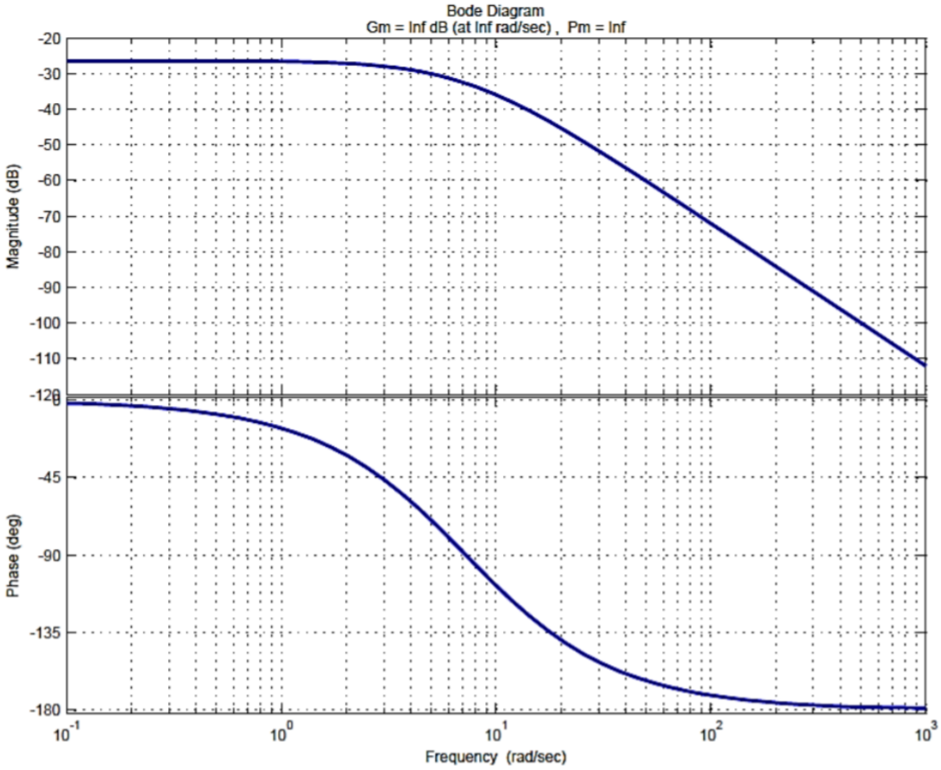
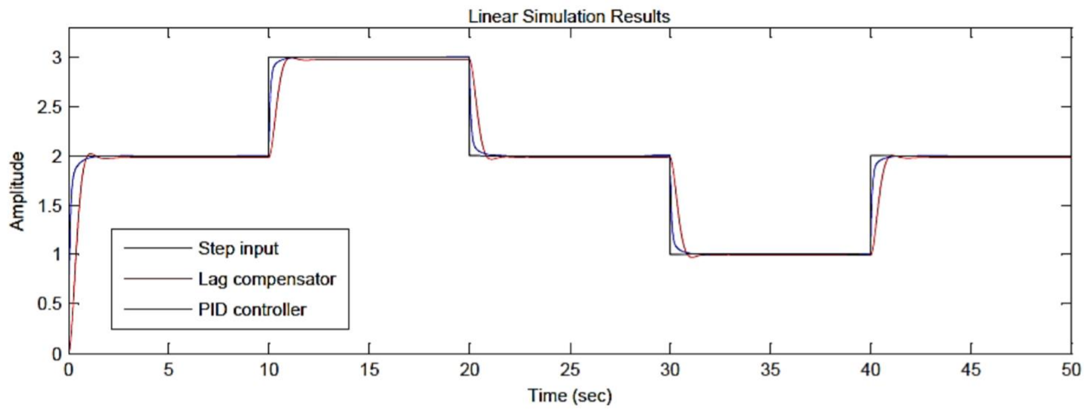


Figure 17. Bode plot, infinite phase and infinite gain margin.

According to the bode plot of the closed loop system, we have infinite gain and infinite phase margin, which means the system will not become unstable with increasing gain.




```
% Bode plot, Determining gain and phase margin figure
margin(numclp,denclp), grid on
figure
margin(numc,denc), grid on
%Bode plot of closed loop TF with lag compensator
```