EML 4312: Control of Mechanical Engineering Systems
Fall 2009, University of Florida

A Lecture Note on DC Motor Speed Control

Dr. Redmond Ramin Shamshiri
redmound@ufl.edu

m-file download

I. Physical Setup and System Equations

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 transitional motion. The electric circuit of the armature and the free body diagram of the rotor are shown in the following figure.

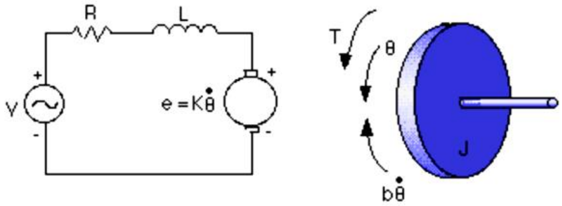


Figure 1. Schematic view of a DC motor

Assume the following values for the physical parameters:

- Moment of inertia of the rotor, $J = 0.02 \frac{kg.m^2}{s^2}$
- Damping ratio of the mechanical system $b = 0.2\ Nms$
- Electromotive force constant $k = k_e = k_t = 0.02 \frac{Nm}{Amp}$
- Electric resistance $R = 2\ ohm$
- Electric inductance $L = 0.4\ H$
- Input (V): Source Voltage
- Output (θ): position of shaft
- The rotor and shaft are assumed to be rigid

The motor torque, T , is related to the armature current, i, by a constant factor k_t. The back emf, e, is related to the rotational velocity by the following equations:

$$T = k_t i$$
$$e = k_e \dot{\theta}$$

In SI units, k_t (armature constant) is equal to k_e (motor constant). From the figure above the following equations based on Newton's law combined with Kirchhoff's law can be obtained:

$$J\ddot{\theta} + b\dot{\theta} = T$$
$$L\frac{di}{dt} + Ri = V - e \qquad (1)$$

II. Design Criteria

- Settling time $\leq 1\ s$
- Overshoot $\leq$ 5 %
- Steady-state error $\leq$ 0.4 %

III. Problem Statement

For a step input ($\dot{\theta}_{ref}$) of $1 \frac{rad}{sec}$, design a controller so that the motor speed satisfies the above requirements. Specifically, you are required to:

1. Draw the block diagram of the closed loop system labeling all the signals (e.g., $\dot{\theta}_{ref}$, $\dot{\theta}, V, e, T$ etc)
2. Determine the transfer function between $\dot{\theta}_{ref}$ and $\dot{\theta}(t)$ using the set of equations in (1)
3. Check the stability of the open-loop and closed-loop systems.
4. Draw the root locus of the given system
5. Design a Lead-Lag Compensator
6. PID Controller
7. Using Bode plots determine the gain and phase margins for the closed-loop system.

Report:

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. A linear differential equation describing the electromechanical properties of a DC motor to model (transfer function) the relation between input (V) and output ($\dot{\theta}$) was first derived using basic laws of physic. This transfer function was used to analyze the performance of the system and to design proper controllers (Lag compensator and PID) to meet the design criteria. The locations of the desired poles were found from the design criteria (settling time, percent overshoot). Using root locus, it was found that a lag compensator is required to meet this design criteria and place poles in the desired locations. A second lag compensator was also designed to meet the steady state requirement of the problem. The result of the final lag compensator on the closed loop response of the system is shown in figure 2. A settling time of 0.844 seconds, percent overshoot of 1.91% and steady state error of 0.1% were achieved using the designed lag compensator. PID controller was also used in this problem to meet the design criteria. Using a trial and error approach, PID gains were first tuned and implemented. The closed loop response of the PID controller is shown in figure 3. A settling time of 0.8 seconds without any percent overshoot and zero steady state error were achieved with PID controller. Details of the design procedure and MATLAB code are shown in the following pages.

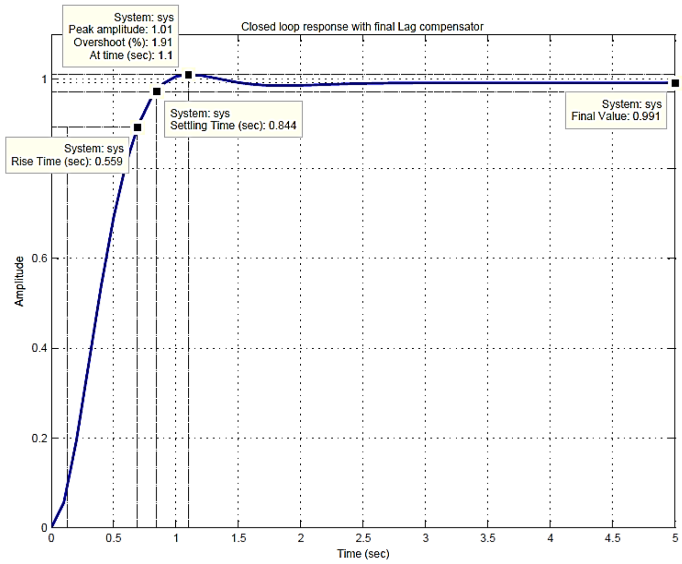


Figure 2. DC motor control with Lag compensator.

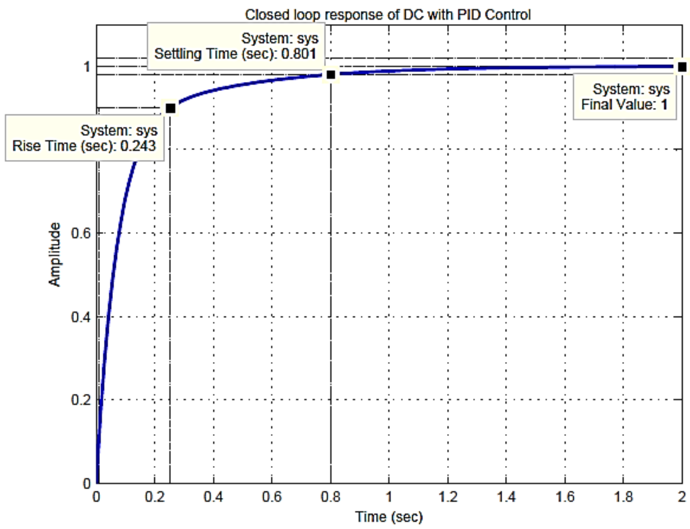


Figure 3. DC motor control with PID.

I. Block diagram of the closed loop system labeling all the signals (e.g., $\dot{\theta}_{ref}$, $\dot{\theta}, V, e, T$ etc)

The block diagram of the closed loop system is shown in figure 4.

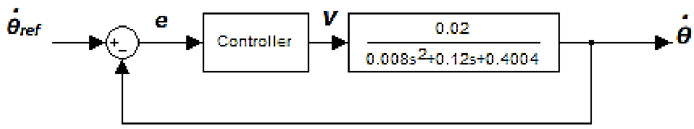


Figure 4. Closed loop block diagram.

II. Determine the transfer function between $\dot{\theta}_{ref}$ and $\dot{\theta}(t)$ using the set of equations in (1)

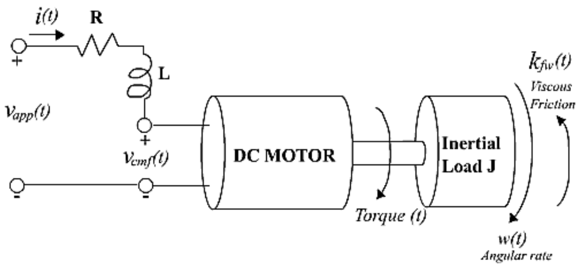


Figure 5. A simple model of a DC motor driving an inertial load

$$T = k_t i$$
$$e(t) = k_e \dot{\theta}$$
$$J\frac{d\omega}{dt} = \sum T_i = K_t i(t) - b\omega(t)$$
$$V(t) - e(t) = L\frac{di}{dt} + Ri(t)$$
$$V(t) = L\frac{di}{dt} + Ri(t) + k_e\dot{\theta} \qquad (2)$$

$$J\ddot{\theta} + b\dot{\theta} = K_t i(t) \qquad (3)$$

Equations (2) and (3) can be solved using Laplace transform as follow:

$$V(s) = LSI(s) + RI(s) + k_e\dot{\theta}(s) \qquad (4)$$

$$JS\dot{\theta}(s) + b\dot{\theta}(s) = K_t I(s) \qquad (5)$$

Eliminating $I(s)$ from (4),(5) we will have:

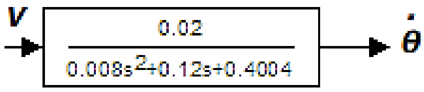
$$\frac{V(s) - k_e\dot{\theta}(s)}{(LS+R)} = I(s)$$
$$\frac{(JS+b)\dot{\theta}(s)}{K_t} = I(s)$$
$$\frac{V(s) - k_e\dot{\theta}(s)}{(LS+R)} = \frac{(JS+b)\dot{\theta}(s)}{K_t}$$
$$\frac{V(s)}{\dot{\theta}(s)} = \frac{(JS+b)(LS+R)}{K_t} + k_e$$

or

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{K_t}{(JS+b)(LS+R) + k_e K_t}$$

$$\boldsymbol{\frac{\dot{\theta}(s)}{V(s)} = \frac{K_t}{jLs^2 + (jR + bL)s + bR + k_e K_t}} \qquad (6)$$

Replacing the given property values of the DC motor in equation (6) yields the **open-loop transfer function** with $V(s)$ as input and $\omega = \dot{\theta}(s)$ as output. This equation indicates the behavior of the motor speed for a given voltage.



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)
```

$$\boldsymbol{\frac{\dot{\theta}(s)}{V(s)} = \frac{0.02}{0.008s^2 + 0.12s + 0.4004}} \qquad (7)$$

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 figure 6.

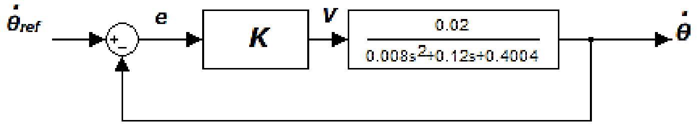


Figure 6.Closed loop block diagram of DC motor

$$\frac{\dot{\theta}_{out}}{\dot{\theta}_{ref}} = \frac{K.G}{1+KG}$$
$$= \frac{K\frac{0.02}{0.008s^2 + 0.12s + 0.4004}}{1 + K\frac{0.02}{0.008s^2 + 0.12s + 0.4004}}$$
$$\boldsymbol{\frac{\dot{\theta}_{out}}{\dot{\theta}_{ref}} = \frac{0.02K}{0.008s^2 + 0.12s + (0.02K + 0.4004)}} \qquad (8)$$

The following Matlab code can was used to determine the closed loop transfer function with a constant gain K=1.

```
% Closed loop Transfer function of DC motor
cloop(num,den);
```

State Space representation:
The two sets of differential equation given in (2) and (3) could also be written as a matrix system of equation format called state space representation. Re-arranging these two equations, we will have:

$$\frac{di}{dt} = -\frac{R}{L}i(t) - \frac{k_e}{L}\dot{\theta}(t) + \frac{1}{L}V(t)$$
$$\ddot{\theta} = \frac{K_t}{J}i(t) - \frac{b}{J}\dot{\theta}$$
$$\begin{bmatrix} \frac{di}{dt} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{k_e}{L} \\ \frac{K_t}{J} & -\frac{b}{J} \end{bmatrix} \begin{bmatrix} i \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V(t)$$
$$\dot{\theta} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ \dot{\theta} \end{bmatrix} \qquad (9)$$

State space representation in Matlab

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

$$\begin{bmatrix} \frac{di}{dt} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -5 & -0.05 \\ 1 & -10 \end{bmatrix} \begin{bmatrix} i \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 2.5 \\ 0 \end{bmatrix} V(t)$$
$$\dot{\theta} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ \dot{\theta} \end{bmatrix} \qquad (10)$$

The open loop and closed loop responses of the DC motor without any controller are shown in figure 7 and 8.

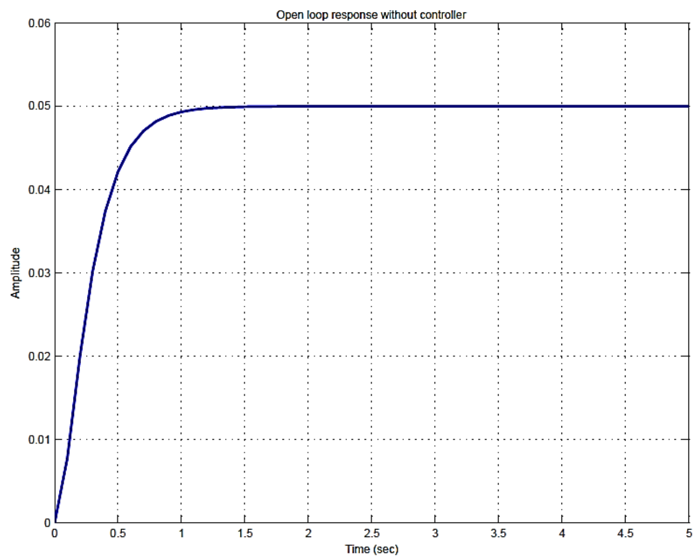


Figure 7. Open loop response of DC motor without controller

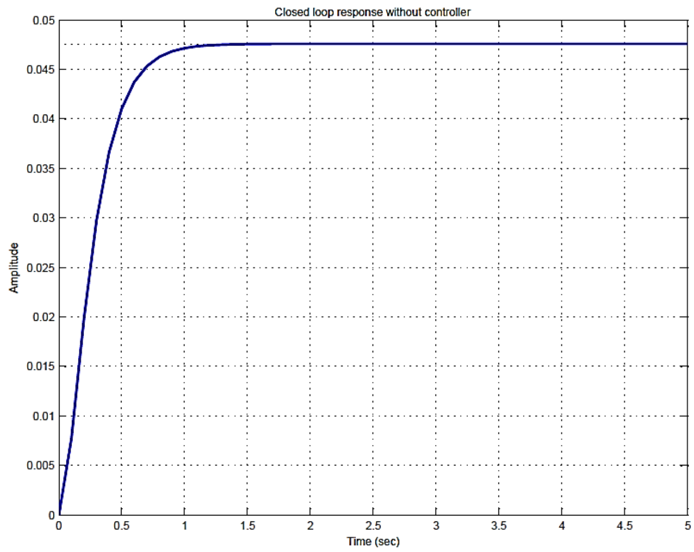


Figure 8. Closed loop response of DC motor without controller

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.

III. 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
% Reference:
% Modern Control system, Richard Dorf, 11th Edition,
pages 360-366
%
% 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.
%==========================================================
% ======Example 6.2. Page 363 ======
% Checking the stability of D=s^3 +s^2 + 2s +24
% Solution:
% >> D=[1 1 2 24];
% ==================================
%++++++++++++++++++++++++++++++++++
%   Nov.25.2009, Ramin Shamshiri  +
%   ramin.sh@ufl.edu              +
%   Dept of Ag & Bio Eng          +
%   University of Florida         +
%   Gainesville, Florida          +
%++++++++++++++++++++++++++++++++++
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==========================

% -------------------Begin of Bulding 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(j-
2,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

%=======================Program
Ends==========================
```

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

$$\Delta = \mathbf{0.008s^2 + 0.12s + 0.4004}$$

```
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 the stability of the closed loop transfer function with a constant gain of K=10;

$$\Delta = \mathbf{0.008s^2 + 0.12s + 10.4004}$$

```
Input coefficients of characteristic equation,i.e:[an an-1 an-2 ... a0]= [0.08
0.12 10.4004]
 ----------------------------------------
Roots of characteristic equation is:
ans =
  -0.7500 +11.3773i
  -0.7500 -11.3773i
--------The Routh-Hurwitz array is:--------
m =
    0.0800   10.4004
    0.1200         0
   10.4004         0
              ----> System is Stable <----
```

Checking for controllability and observability

The following Matlab code was written to test the controllability and observability of the system. The system is both controllable and observable

```
% 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   <----------
```

IV. 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 $P_1 = -5.01$ and $P_2 = -9.99$ which repel each other at -7.5 and one goes to positive infinity and the other goes to negative infinity.

```
>> rlocus(num,den)
```

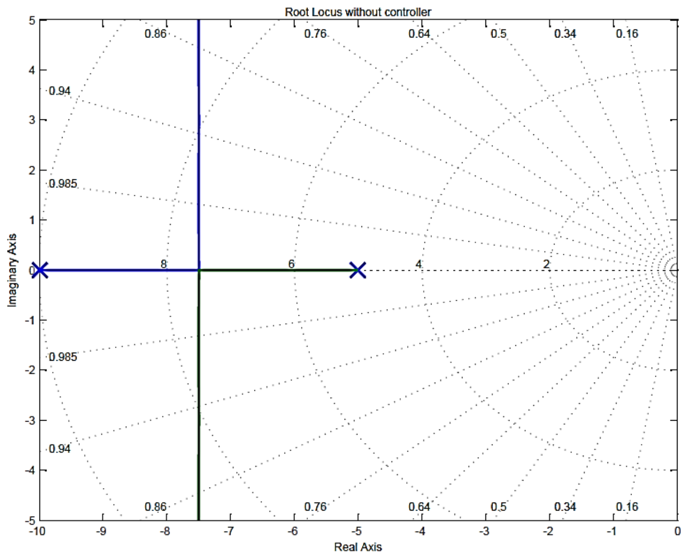


Figure 9. Root locus plot of DC motor transfer function

We can decide about using lead/lag compensator from the root locus plot. If the root locus should be shifted to the right, we will need a lag compensator and if the root locus should be shifted to the left to place poles at the desired locations, we will need lead compensator.

V. Design a Lead-Lag Compensator

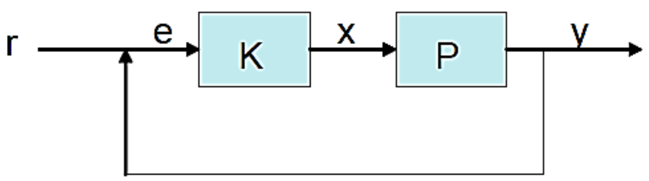


Figure 10. Closed loop block diagram

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

From the closed loop block diagram in figure 10 and the system transfer function, the actual characteristic equation of the closed loop system is:

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

Design criteria are settling time $\leq 1\ s$, overshoot $\leq$ 5 % and steady-state error $\leq$ 0.4 %. Using the equations for settling time and percent overshoot given in (11) and (12), we can determine the desired damping ratio (ζ) and natural frequency (ω_n) as follow:

$$T_s = \frac{4}{\omega_n \zeta} = 1\ sec \qquad (11)$$

$$P.O. = 100e^{-\pi\zeta/\sqrt{1-\zeta^2}} \qquad (12)$$

Therefore:

$$100e^{-\pi\zeta/\sqrt{1-\zeta^2}} = 5$$

$$e^{-\pi\zeta/\sqrt{1-\zeta^2}} = 0.05$$

$$-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\ln(e) = \ln(0.05)$$

$$\frac{\zeta}{\sqrt{1-\zeta^2}} = 0.953571199$$

$$\boldsymbol{\zeta = 0.69010673}$$

$$\boldsymbol{\omega_n = 5.7962}$$

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:

$$\Delta_D = s^2 + 2\zeta\omega_n s + \omega_n^2$$

$$\Delta_D = \boldsymbol{s^2 + 8s + 33.5960 = (s + 4 + 4.194j)(s + 4 - 4.194j)} \qquad (13)$$

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 figure (4) 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 (a, b, c, K), we have one degree of design flexibility. Let's assume $\boldsymbol{a = 14}$ 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-----%
```

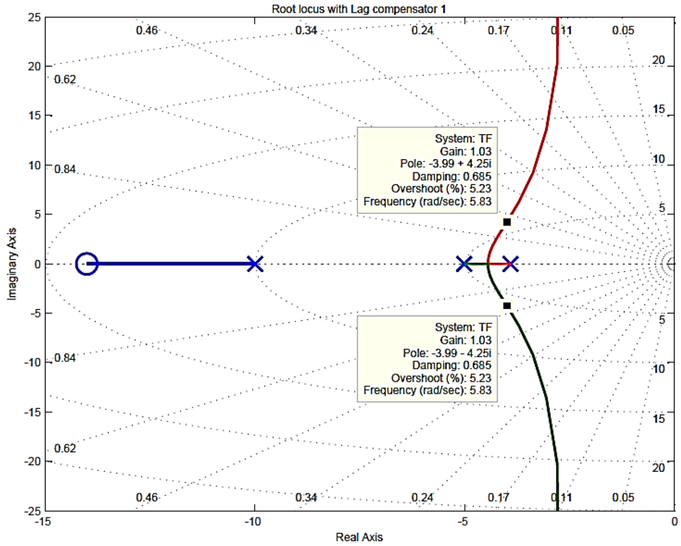


Figure 11. Root locus with lag compensator to improve settling time and percent overshoot.

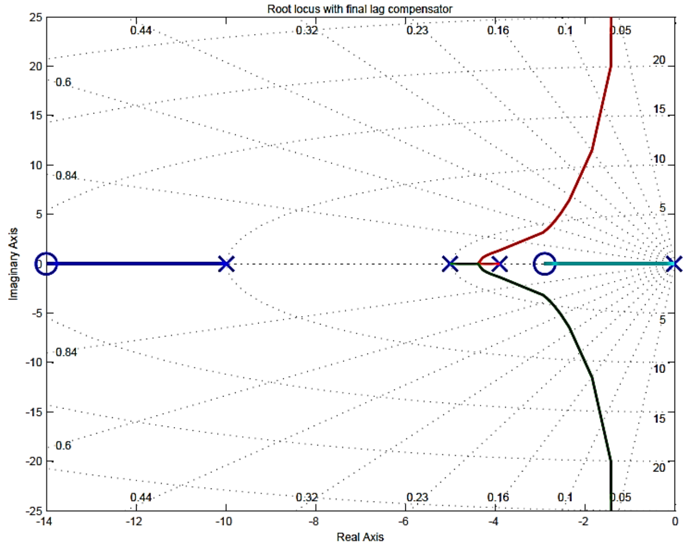


Figure 14. Root locus with the final lag compensator that improves settling time, percent overshoot and steady state error.

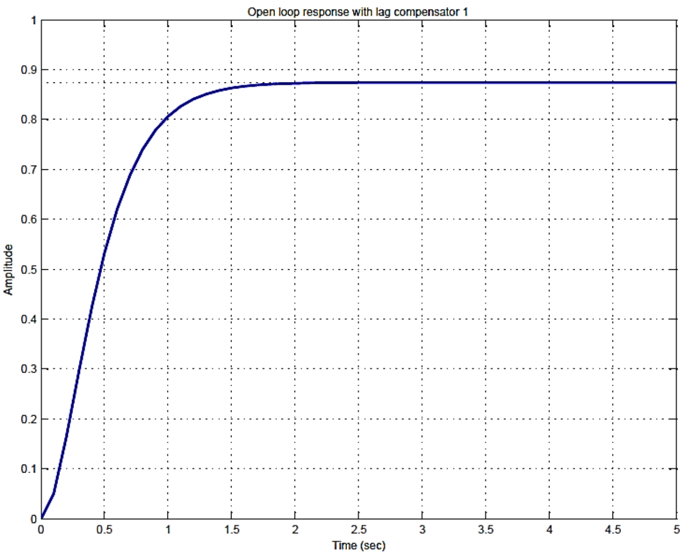


Figure 12. Open loop response of DC motor with Lag compensator1, improved settling time and percent overshoot

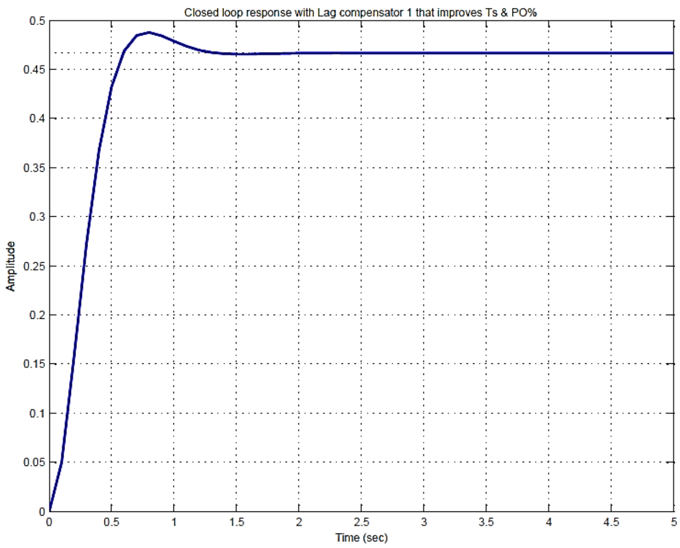


Figure 13. Closed loop response of DC motor with final Lag compensator1

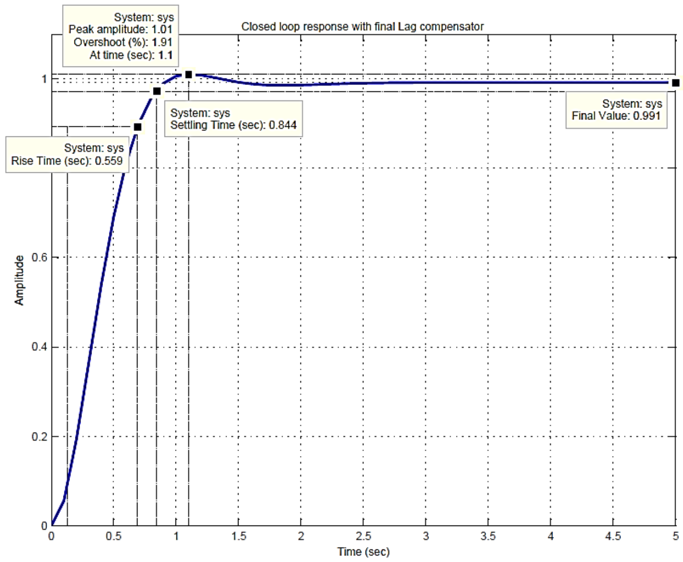


Figure 15. Closed loop response with the final lag compensator that improves settling time, percent overshoot and steady state error.

Result of Lag compensator:

Settling time= 0.844 < 1
PO%=1.91%<5%
Final value=0.991 (Steady state error<0.4%)

VI. PID Controller

The transfer function for a PID controller is:

$$K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s}$$

where K_p is the proportional gain, K_D is the derivative gain and K_I is the integrator gain. The effect of each term in PID controller is summarized in table 1.

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
K_p	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	Small Change	Decrease	Decrease	Small Change

Table 1. Effects of PID terms on response

Using a trial and error approach, the following gains were selected:
$K_P = 70$;
$K_I = 170$;
$K_D = 5$;
Therefore the transfer function for the PID controller becomes:

$$G_{PID}(s) = \frac{5s^2 + 70s + 170}{s}$$

The open loop transfer function of the system with PID controller is:

$$\frac{\dot{\theta}_{out}}{\dot{\theta}_{ref}} = G_{PID}(s)P(s)$$
$$= \frac{5s^2 + 70s + 170}{s} \cdot \frac{0.02}{0.008s^2 + 0.12s + 0.4004}$$
$$= \frac{12.5\ s^2 + 175\ s + 425}{s^3 + 15\ s^2 + 50.05\ s}$$

And the closed loop transfer function of the system with PID controller is:

$$\frac{\dot{\theta}_{out}}{\dot{\theta}_{ref}} = \frac{G_{PID}(s)P(s)}{1 + G_{PID}(s)P(s)}$$
$$= \frac{G_{PID}(s)P(s)}{1 + G_{PID}(s)P(s)}$$
$$= \frac{12.5\ s^2 + 175\ s + 425}{s^3 + 27.5\ s^2 + 225.1\ s + 425}$$

The following Matlab code was implemented to derive the open-loop and closed loop transfer function of the DC motor with PID controller and to plot the step response. The results are shown in figure through.

```
%----------------------------PID control-----------%
% PID control gain, using trial and error
kp=70;
ki=170;
kd=5;

% PID transfer function
numPID=[kd kp ki];
denPID=[1 0];

% Open loop TF of DC motor with PID controller
num_DC_PID=conv(num,numPID);
den_DC_PID=conv(den,denPID);
disp('Open loop TF of DC motor with PID controller')
tf(num_DC_PID,den_DC_PID)

% Closed loop TF of DC motor with PID controller
[NumPID_CLP,DenPID_CLP]=cloop(num_DC_PID,den_DC_PID);
disp('Closed loop TF of DC motor with PID controller')
tf(NumPID_CLP,DenPID_CLP)
figure
step(NumPID_CLP,DenPID_CLP), grid on
title('Closed loop response of DC with PID Control')
%---------------------End of PID control-----------%
```

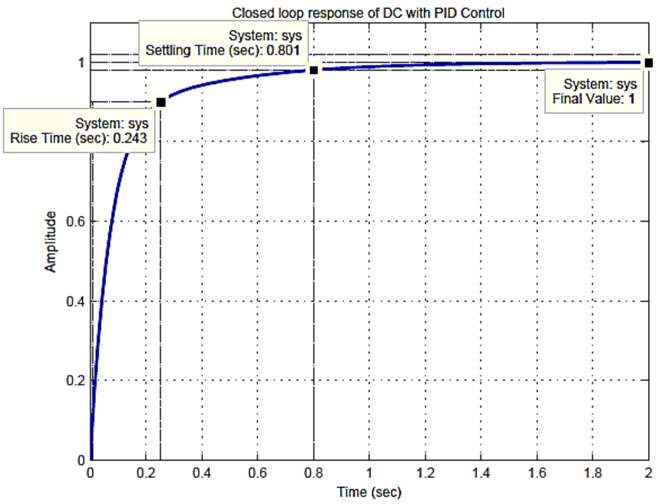


Figure 16. Step response of DC motor with PID controller.

Result of PID controller:

Settling time= 0.801< 1
PO%=0 < 5%
Final value=1 (Steady state error =0 <0.4%)

VII. Using Bode plots determine the gain and phase margins for the closed-loop system.

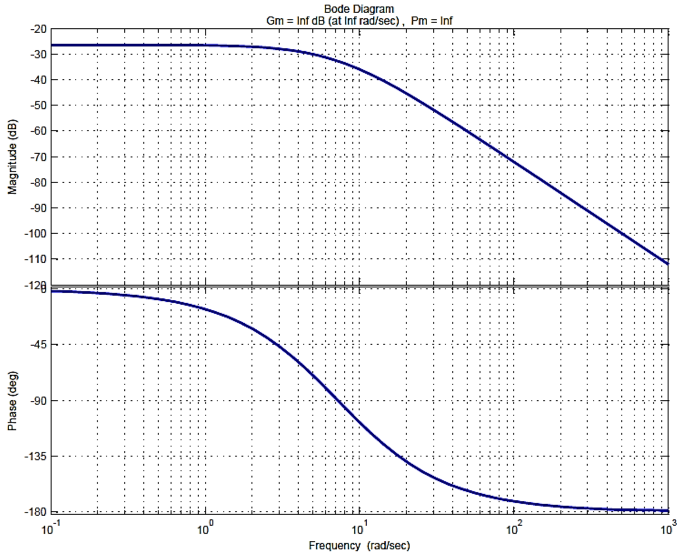
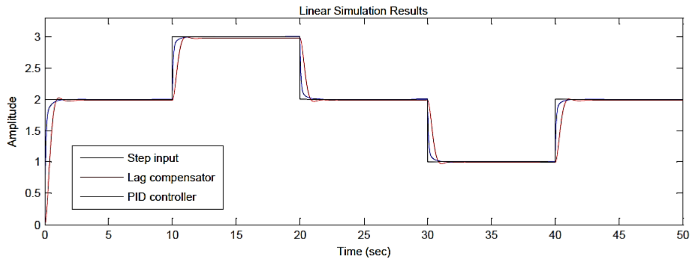


Figure 17. Bode plot, infinite phase and infinite gain margin.

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.



m-file download

```
%==========================================================
% DC motor Control using lag compensation and PID
% To customize this code you need to:
% 1- Change the values of DC motor constants
% 2- Change the zeros of lag compensation
% 3- Change the gains of PID controller
%==========================================================
%++++++++++++++++++++++++++++++++++
%   Nov.29.2009, Redmond R. Shamshiri
%   Dept of Ag & Bio Eng, University of Florida
%++++++++++++++++++++++++++++++++++
clc;
clear;
close all
% DC motor constants
J=0.02;
b=0.2;
kt=0.02;
ke=0.02;
R=2;
L=0.4;
% Transfer Function
num=[kt];
den=[J*L J*R+b*L b*R+ke*kt ];
disp('Open loop Transfer function without controller')
TF_DC=tf(num,den)
[numclp,denclp]=cloop(num,den,-1);
disp('Closed loop Transfer Function without controller')
tf(numclp,denclp)
% Open loop response without controller
step(num,den,0:0.1:5),grid on
title('Open loop response without controller')
% Closed loop response without controller
figure
step(numclp,denclp,0:0.1:5),grid on
title('Closed loop response without controller')
% State Spapce representation
A=[-(R/L) -(ke/L);(kt/J) -(b/J)];
B=[1/L; 0];
C=[0 1];
D=0;
disp('State Space representation:')
SYS_DC=ss(A,B,C,D)
% 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
% Drawing Root locus
figure
rlocus(num,den),grid on
title('Root Locus without controller')
% 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 1
disp('DC motor Transfer function with Lag compensator')
NUM=conv(numlag1,num);
DEN=conv(denlag1,den);
TF=tf(NUM,DEN)
% Root locus with Lag compensator 1
figure
rlocus(TF),grid on
title('Root locus with Lag compensator 1')
% Open loop response of the system with Lag compensator1
figure
step(TF,0:0.1:5),grid on
title('Open loop response with lag compensator 1')
% Closed loop response 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
disp('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)
% Root locus with final lag compensator
figure
rlocus(NumDC,DenDC), grid on
title('Root locus with final lag compensator')
% 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------

%-------------------------------PID control------------
% PID control gain, using trial and error
kp=70;
ki=170;
kd=5;

% PID transfer function
numPID=[kd kp ki];
```

```
denPID=[1 0];
% Open loop TF of DC motor with PID controller
num_DC_PID=conv(num,numPID);
den_DC_PID=conv(den,denPID);
disp('Open loop TF of DC motor with PID controller')
tf(num_DC_PID,den_DC_PID)
% Closed loop TF of DC motor with PID controller
[NumPID_CLP,DenPID_CLP]=cloop(num_DC_PID,den_DC_PID);
disp('Closed loop TF of DC motor with PID controller')
tf(NumPID_CLP,DenPID_CLP)
figure
step(NumPID_CLP,DenPID_CLP), grid on
title('Closed loop response of DC with PID Control')
%-------------------------End of PID control------------
% 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
figure
margin(NumPID_CLP,DenPID_CLP), grid on % Bode plot of
closed loop TF with PID controller
```

MATLAB workspace result:

```
Open loop Transfer function without controller
            0.02
---------------------------
0.008 s^2 + 0.12 s + 0.4004

Closed loop Transfer Function without controller
           0.02
---------------------------
0.008 s^2 + 0.12 s + 0.4204
State Space representation:
a =
          x1     x2
   x1     -5  -0.05
   x2      1    -10
b =
        u1
   x1  2.5
   x2    0
c =
       x1  x2
   y1   0   1
d =
       u1
   y1   0

Continuous-time model.
          ----------> System is Controllable <----------
          ----------> System is Observable   <----------
Desired Damping ratio is:
Damp =
    0.6901
Desired Natural Frequency is:
Wn =
    5.7962
Desired Characteristic Equation is:
dend =
    1.0000    8.0000   33.5960
Desired Pole locations:
Dp =
  -4.0000 + 4.1948i
  -4.0000 - 4.1948i
Pole of the first lag compensator is:
p1 =
    3.9054
Gain of the first lag compensator is:
K =
    4.8832
Transfer function of the first Lag compensator to
improve Ts and PO%:
 4.883 s + 68.36
---------------
   s + 3.905
DC motor Transfer function with Lag compensator
          12.21 s + 170.9
---------------------------------
s^3 + 18.91 s^2 + 108.6 s + 195.5
pole of the 2nd lag compensator
p2 =
    0.0217
The 2nd Lag compensator Transfer function to improve
SSE:
   s + 2.9
-----------
s + 0.02174
The overal Lag compensator transfer function
(lag1*lag2):
 4.883 s^2 + 82.53 s + 198.3
---------------------------
  s^2 + 3.927 s + 0.08491
Open loop TF of the DC motor with final Lag compensator
(improved Ts, PO% & SSE)
       12.21 s^2 + 206.3 s + 495.6
-------------------------------------------
s^4 + 18.93 s^3 + 109 s^2 + 197.8 s + 4.25
closed loop TF of the DC motor with final Lag
compensator (improved Ts, PO% & SSE)
          12.21 s^2 + 206.3 s + 495.6
---------------------------------------------
s^4 + 18.93 s^3 + 121.3 s^2 + 404.1 s + 499.9

Open loop TF of DC motor with PID controller
 12.5 s^2 + 175 s + 425
----------------------
s^3 + 15 s^2 + 50.05 s
Closed loop TF of DC motor with PID controller
    12.5 s^2 + 175 s + 425
------------------------------
s^3 + 27.5 s^2 + 225.1 s + 425
```