```
%Project Part 3
% Aly Khater
%Lawrence Lai
%Leroy Joseph


clc;
clear;
close all;

%-------------------------------------%
%Chapter 1. State-Space
%-------------------------------------%
%Initial Variables
L = 0.0006;
R = 1.40;
Kb = .00867;
Jm = .00844;
Bm = .00013;
Kt = 4.375;
n = 200;
Jl = 1;
Bl = 0.5;
%Concatenated Variables
J = Jm + Jl/(n^2);
b = Bm + Bl/(n^2);

%Matrices in State-Space
A = [0 1 0;0 0 1;0 -(R*b+Kt*Kb)/(L*J) -(L*b+R*J)/(L*J)];
B = [0;0;Kt/(L*J*n)];
C = [1 0 0];
D = [0];

%State-Space representation
Rbt = ss(A,B,C,D);

%Transfer Function
Rbttf = tf(Rbt);
%Zero-Pole
Rbtzpk = zpk(Rbt);

[num,den] = tfdata(Rbt, 'v');
[z,p,k] = zpkdata(Rbt,'v');
Rbtss = ss(Rbttf);

%-------------------------------------%
%Chapter 2. Simulation of State-Space
%-------------------------------------%
%time
t = [0:.01:4];
U = [zeros(size(t))];
%Initial State Variables
X0 = [0;1;0];

%Characteristic Polynomial
CharPoly = poly(A)
```

```
%Poles
Poles = roots(CharPoly)
%Eigenvalues
Eigs0 = eig(A)

damp(A);
%Open Loop Response
[Yo,t,Xo] = lsim(Rbt,U,t,X0);
Xo(101,:);

X1 = expm(A*1)*X0;

%Open loop plots
figure;
subplot(2,1,1), plot(t,Xo(:,1));grid;
title('Unit Step Input Armature Voltage');
axis([0 4 -0.2 0.5]);
set(gca,'FontSize',18);
ylabel('{\itx}_1 (\itrad)')
subplot(2,1,2), plot(t,Xo(:,2)); grid; axis([0 4 -2 1]);
title('Zero Input Armature Voltage');
set(gca,'FontSize',18);
xlabel('\ittime (sec)'); ylabel('{\itx} 2 (\itrad/s)');

%-------------------------------------------------------
% Chapter 2. Coordinate Transformations and Diagonal
% Canonical Form
%-------------------------------------------------------
[Tdcf,E] = eig(A); % Transform to DCF via
% formula
Adcf = inv(Tdcf)*A*Tdcf;
Bdcf = inv(Tdcf)*B;
Cdcf = C*Tdcf;
Ddcf = D;
[Rbtm,Tm] = canon(Rbt,'modal');
% Calculate DCF using
% MATLAB canon
Am = Rbtm.a
Bm = Rbtm.b
Cm = Rbtm.c
Dm = Rbtm.d

%-------------------------------------------------------
% Chapter 3. Controllability
%-------------------------------------------------------
P = ctrb(Rbt); % Calculate controllability
% matrix P
if (rank(P) == size(A,1)) % Logic to assess
% controllability
disp('System is controllable.');
else
disp('System is NOT controllable.');
end
P1 = [B A*B]; % Check P via the formula
```

```
CharPoly =
```

```
   1.0e+03 *

    0.0010    2.3334    7.5075         0


Poles =

   1.0e+03 *

         0
   -2.3301
   -0.0032


Eigs0 =

   1.0e+03 *

         0
   -0.0032
   -2.3301


   Pole        Damping       Frequency       Time Constant
                           (rad/TimeUnit)     (TimeUnit)

  0.00e+00    -1.00e+00       0.00e+00              Inf
 -3.22e+00     1.00e+00       3.22e+00          3.10e-01
 -2.33e+03     1.00e+00       2.33e+03          4.29e-04

Am =

   1.0e+03 *

         0         0         0
         0   -0.0032         0
         0         0   -2.3301


Bm =

    0.0717
    0.1363
    4.2134


Cm =

    8.0000   -4.2153    0.0002


Dm =

     0

System is controllable.
```

Published with MATLAB® R2022b