Engineering 447 Control Systems

Homework #3 #1. #2. #3

#1

Find transfer function $G(s) = \frac{Y(s)}{U(s)}$ and output $y(t)$ when u = unit step input given the following values (note that both y and u are scalars, i.e. their dimension is 1 x 1):

$$\mathbf{x}_0 = 0;\ \mathbf{A} = \begin{bmatrix} -3 & 1 & 0 \\ 2 & -3 & 2 \\ 0 & 1 & -3 \end{bmatrix};\ \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};\ \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix};\ \mathbf{D} = 0$$

Hint: In MATLAB, define symbols: "`syms s t`"; define **A**, **B**, **C**, **D** matrices: e.g. "`A=[-3,1,0;2,-3,2;0,1,-3]`"; then compute $Y(s)$ by equation (5 – 6) page 8 of Handout 5 while substituting $1/s$ for $U(s)$ then take inverse Laplace transform "`yt = ilaplace(Ys)`". Note: 3 × 3 identity matrix in MATLAB is "`eye(3)`".

/ ▸ Users ▸ farnamadelkhani ▸ Documents ▸ MATLAB ▸ 447 HW3

Editor – /Users/farnamadelkhani/Documents/MATLAB/447 HW3/partOne.m

```
syms s t

A = [-3 1 0; 2 -3 2; 0 1 -3];
B = [1; 0; 0];
C = [0 0 1];
D = [0; 0; 0];

%[num,den] = ss2tf(A,B,C,D);
%t = tf(num,den)

% Input U(s) = step of magnitude 0.04;
% recall Laplace transform of unit step = 1/s:
U(s) = 1 / s;

% Type in equation (I-6) without x0 term as x0 = 0;
% eye(3) is 3x3 identity matrix:
Y(s) = (C * inv(s * eye(3) - A) * B + D) * U(s);

% Take inverse Laplace transform:
yt = ilaplace(Y(s))

% Round to 4 decimal places:
vpa(yt,4)
```

lab5Fourier.m | partOne.m | +

Command Window

New to MATLAB? See resources for Getting Started.

```
 exp(-3*t)/6 - exp(-t)/4 - exp(-5*t)/20 + 2/15


ans =

 0.1667*exp(-3.0*t) - 0.05*exp(-5.0*t) - 0.25*exp(-1.0*t) + 0.1333
 0.1667*exp(-3.0*t) - 0.05*exp(-5.0*t) - 0.25*exp(-1.0*t) + 0.1333
 0.1667*exp(-3.0*t) - 0.05*exp(-5.0*t) - 0.25*exp(-1.0*t) + 0.1333
```

Name

A

$$\text{ans} = \frac{1}{6} e^{-3t} - \frac{1}{20} e^{-5t} - \frac{1}{4} e^{-t} + 2/15$$

[redacted] Find transfer function $G(s) = \frac{Y(s)}{U(s)}$ and output $y(t)$ when u = unit step input given the following values (note that both y and u are scalars, i.e. their dimension is 1 x 1):

$$\mathbf{x}_0 = 0;\ \mathbf{A} = \begin{bmatrix} -3 & 1 & 0 \\ 2 & -3 & 2 \\ 0 & 1 & -3 \end{bmatrix};\ \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};\ \mathbf{C} = [0 \quad 0 \quad 1];\ \mathbf{D} = 0$$

Hint: In MATLAB, define symbols: "`syms s t`"; define **A**, **B**, **C**, **D** matrices: e.g. "`A=[-3,1,0;2,-3,2;0,1,-3]`"; then compute $Y(s)$ by equation (5 – 6) page 8 of Handout 5 while substituting $1/s$ for $U(s)$ then take inverse Laplace transform "`yt = ilaplace(Ys)`". Note: 3×3 identity matrix in MATLAB is "`eye(3)`".

>>eig(A)

ans =

$$\begin{bmatrix} -5 \\ -3 \\ -1 \end{bmatrix}$$

$\boxed{2)}$ Convert the continuous time state space system in problem 1 above to a discrete time state space system given sampling period is $T = 0.5$ seconds (zero order hold method). You may use equations (9 – 6) and (9 – 7) to compute $\mathbf{A}_d, \mathbf{B}_d$ by MATLAB or use `c2d` command. Write your answer in the form below;

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d u_k \\ y_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k \end{cases}$$

a. Write down the $\mathbf{A_d}, \mathbf{B_d}, \mathbf{C}, \mathbf{D}$ matrices you found.

b. Compare the Eigen values (system poles) of the continuous time **A** matrix of problem 1 to Eigen values of $\mathbf{A}_d$ matrix you found above; verify the following expression for all Eigen values:

$$s_d = e^{s_c T}$$

Where s_c = a continuous time system pole (same as Eigen value), s_d = the corresponding discrete time system pole and $T = 0.5$(s) sampling period.

>> eig(Ad)

ans = 0.6065
0.2231
0.0821

```
syms s t

A = [-3 1 0; 2 -3 2; 0 1 -3];
B = [1; 0; 0];
C = [0 0 1];
D = [0];
T=.5;

syscont = ss(A,B,C,D) % Define continuous time state space system

sysdisc = c2d(syscont,T) % Convert to discrete time

[Ad,Bd,Cd,Dd,TT] = ssdata(sysdisc) % Extract the discrete time system data
```

Sampling period

```
syscont =

  A =
       x1  x2  x3
   x1  -3   1   0
   x2   2  -3   2
   x3   0   1  -3

  B =
       u1
   x1   1
   x2   0
   x3   0

  C =
       x1  x2  x3
   y1   0   0   1

  D =
       u1
   y1   0

Continuous-time state-space model.
```

```
sysdisc =

  A =
            x1       x2       x3
   x1   0.2837   0.1311  0.06059
   x2   0.2622   0.3443   0.2622
   x3  0.06059   0.1311   0.2837

  B =
            u1
   x1   0.2737
   x2   0.1049
   x3  0.01478

  C =
       x1  x2  x3
   y1   0   0   1

  D =
       u1
   y1   0

Sample time: 0.5 seconds
Discrete-time state-space model.

Ad =

    0.2837    0.1311    0.0606
    0.2622    0.3443    0.2622
    0.0606    0.1311    0.2837

Bd =

    0.2737
    0.1049
    0.0148

Cd =

     0     0     1

Dd =

     0

TT =

    0.5000
```

$$\boxed{S_d = e^{S_c T}}$$

$S_{c_1} T = (-5)(0.5) = -2.5$

$e^{S_c T} = e^{-2.5} = 0.08208$

$S_{c_2} T = (-3)(0.5) = -1.5$

$e^{-1.5} = 0.2231$

$S_{c_3} T = (-1)(0.5) = -0.5$

$e^{-0.5} = 0.6065$

$\therefore S_d = e^{S_c T}$ is matching

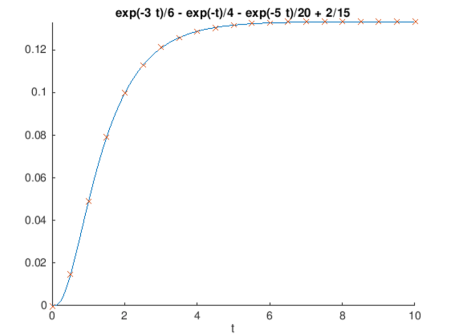
3) Using MATLAB find the response of discrete time system of problem 2 above to a unit step input for 10 seconds with all zero initial conditions ($\mathbf{x}_0 = 0$) by recursion. To do so, compute the values of y_k using the discrete time equations directly as follows. You may use/modify the code below; copy and paste in a m-file:

```
tfinal = 10;          % Simulation end time.
T = 0.5;              % Sampling period.
kfinal = 10/T;        % Final time increment.
D = 0;                % Ad, Bd, C and D matrices must be defined by now.
x = [0; 0; 0];        % Initial values.
y = 0;                % Initial value of output.

u = 1;                % unit step input.

for k = 1: kfinal     % Iterations.

    x = Ad * x + Bd * u;
    ynew = C * x + D * u;
    y = [ y, ynew]    % Put together the new value of y with all previous y's.

end
```

Next, plot the y values found above (use “x” marks) together with the problem 1 answer, in the same figure. You may use `hold`, `plot` and `ezplot` commands:

```
figure                       % Initiate a new figure.
hold                         % Hold plotting.
ezplot(yt,[0,10])            % Plot the problem 1 solution yt for 0 < t < 10 sec.
plot(0:T:tfinal, y, 'x')     % Plot with X's the discrete time system solution.
hold                         % Release hold to plot all on same graph.
axis tight                   % Force plot axes cover the complete plot range.
```

The results of problem 3 should fall on those of problem 1 curve. This is because <u>zero order hold conversion</u> method is <u>step invariant</u>.




```
syms s t

A = [-3 1 0; 2 -3 2; 0 1 -3];
B = [1; 0; 0];
C = [0 0 1];
D = [0];

tfinal = 10; % Simulation end time.
T = 0.5; % Sampling period.
kfinal = 10/T; % Final time increment.
D = 0; % Ad, Bd, C and D matrices must be defined by now.
x = [0; 0; 0]; % Initial values.
y = 0; % Initial value of output.
u = 1; % unit step input.
for k = 1: kfinal % Iterations.

x = Ad * x + Bd * u;
 ynew = C * x + D * u;
 y = [ y, ynew] % Put together the new value of y with all previous y's.
end

% Input U(s) = step of magnitude 0.04;
% recall Laplace transform of unit step = 1/s:
U(s) = 1 / s;

% Type in equation (I?6) without x0 term as x0 = 0;
% eye(3) is 3x3 identity matrix:
Y(s) = (C * inv(s * eye(3) - A) * B + D) * U(s);

% Take inverse Laplace transform:
yt = ilaplace(Y(s))

figure % Initiate a new figure.
hold % Hold plotting.
ezplot(yt,[0,10]) % Plot the problem 1 solution yt for 0 < t < 10 sec.
plot(0:T:tfinal, y, 'x') % Plot with X's the discrete time system solution.
hold % Release hold to plot all on same graph.
axis tight % Force plot axes cover the complete plot range.
```