AE 353 Codes

J.Clements

February 2019

Below, you will find excerpts of code and information on how to solve particular homework problems.

Contents

1 Homework 1.3

Problems 1, 2, and 3. You can use this code for all three problems. Pay extra attention to whether the dot variable is on the top or bottom.

```
clear, clc
%Create a set of variables
syms x xD xDD u

%Input equilibrium value
xE = input('Equilibrium: ');

%Input the equation of motion/system of equations
%eqn1 = {INSERT YOUR EQUATION HERE};

% xD = [ad; solve(eqn1, add)];
xD = [solve(eqn1, add); ad];

%{
INSERT YOUR x AND u EQUATIONS HERE:
Example:
    x = [ \dot{eta} ; eta];
    u = [u];
    y = [ \dot{eta}];
%}

%Jacobian to find values for A, B, C, D
A = vpa(subs(jacobian(xD, x), xE), 3)
B = jacobian(xD, u)
C = jacobian(y, x)
D = jacobian(y, u)
```

2 Homework 2.1

Problem 1

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
expm(A)
```

Problem 2

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
timeDerivative = A*M
```

Problem 3

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
F = (A + B*(-K))
G = (C + D*(-K))
```

Problem 4

```
clear, clc
%% Insert provided data here:

%% Calculations (do not edit)
xt = expm((A - B*K)*(t1-t0)) * x0
y = C*xt + D*(-K * xt)
```

Problem 6

```
clear,clc
%% Insert provided data here:

%% Calculation (do not edit)
syms t
xt = expm((A-B*K)*t) * x0;
u = -K*xt;
y = C*xt + D*u
```

3 Homework 2.2

Problem 4

```
clear,clc
%% Insert provided data here:

%% Calculations (do not edit)
F = inv(V)*(A - B*K)*(V)
G = (C - D*K)*(V)
```

Problem 5

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
[V, F] = eig((A - B*K))
```

Problem 6

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
[V, F] = eig(A)
```

4 Homework 2.3

Problem 5

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
xd1 = eig((A - B*K1))
xd2 = eig((A - B*K2))
xd3 = eig((A - B*K3))
xd4 = eig((A - B*K4))
xd5 = eig((A - B*K5))

% If all the eigenvalues of A have a negative real part, stable
% If one is positive, unstable
```

5 Homework 3.1

Problem 1

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
k_ref = 1 / (-C*(A - B*K)^-1*B)
```

Problem 2

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
k_ref = 1 / (-C*(A - B*K)^-1*B)
```

Problem 3

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
F = (-C * (A - B * K)^-1 * B * kRef - 1)
```

Problem 4

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
E = mat2str((A - B*K))
F = mat2str(B*kRef)

syms x rZ
G = mat2str(C)
H = [0]
```

6 Homework 3.2

Problem 1

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
kRef = kRef(A,B,C,K);
E = mat2str(A-B*K)
F1 = mat2str(B*kRef)
F2 = mat2str(B)
G = mat2str(C-D*K)
H1 = mat2str(D*kRef)
H2 = mat2str(D)
```

Problem 2

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
kRef = kRef(A,B,C,K);
F = (C)*(-inv(A - B*K))*B
```

Problem 3

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
kRef = kRef(A,B,C,K);
syms x y t

um = [r;d];
Am = A - B*K;
Bm = [B*kRef, B];
Cm = C;

x = mat2str(expm(Am*t1)*x0 + inv(Am)*expm(Am*t1)*Bm*um - inv(Am)*Bm*um)
y = mat2str(Cm*(expm(Am*t1)*x0 + inv(Am)*expm(Am*t1)*Bm*um - inv(Am)*Bm*um))
```

Problem 4

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
kRef = kRef(A,B,C,K);
um = [r;d];
Am = A - B*K;
Bm = [B*kRef, B];
Cm = C;
syms t
%%
y(t) = Cm*(expm(Am*t)*x0 + inv(Am)*expm(Am*t)*Bm*um - inv(Am)*Bm*um)
```

Problem 5

```
clear all, close all, clc
%% Insert provided data here:

%% Useful Relations
kRef = kRef(A,B,C,K);
r = 1;
d0 = 0;
d1 = 1;
Am = A-B*K;
Bm0 = [B*kRef, B]*[r; d0];
Bm1 = [B*kRef, B]*[r; d1];
Cm = C;

%% No disturbance
sys1 = ss(Am,Bm0,Cm,D);
step(sys1)
figure(1)
S1 = stepinfo(sys1) %Look at the appended struct to calculate numbers

%% Disturbance
figure(2)
sys2 = ss(Am,Bm1,Cm,D);
step(sys2)

S2 = stepinfo(sys2) %Look at the appended struct to calculate numbers
```

7 kRef Function

The kRef function is necessary to set references in a controller input. The following is the derivation of the kRef:
Consider a state-space system,

$$\dot{x} = Ax + Bu, \quad y = Cx$$

$$u = -Kx + k_{ref}r$$

$$\dot{x} = (A - BK)x + Bk_{ref}r, \quad y = Cx$$

$$0 = (A - BK)x_{ss} + Bk_{ref}r$$

$$x_{ss} = -(A - BK)^{-1}Bk_{ref}r$$

$$y_{ss} = -C(A - BK)^{-1}Bk_{ref}r$$

$$\therefore k_{ref} = \frac{1}{-C(A - BK)^{-1}B}$$

Note, very important function which simplifies above calculations is $kRef()$. Make file, *kRef.m* and place in pathed folder for Matlab to access:

```
function kRef = kRef(A,B,C,K)
%   kRef function helps find the kRef part of the input equation:
%   u = -K * x + kRef * r + d.  Enter in your matrices: A, B, C, K
%   separated by commas.
    kRef = inv(-C*inv(A - B*K)*B);
    kRef;
end
```

8 Homework 3.3

Problem 1

```
clear,clc
%% Insert provided data here:

%% Calculation (do not edit)
kRef = kRef(A,B,C,K);
G = [C 0];
E = mat2str([A-B*K -B*kInt ; G])
F1 = mat2str([B*kRef ; -1])
F2 = mat2str([B ;0])
G = mat2str(G)
H1 = 0
H2 = 0
```

Problem 2

```
clear, clc
%% Insert provided data here:

%% Calculation (do not edit)
eig([A-B*K1, -B*kInt1; C, 0])
eig([A-B*K2, -B*kInt2; C, 0])
eig([A-B*K3, -B*kInt3; C, 0])
eig([A-B*K4, -B*kInt4; C, 0])
eig([A-B*K5, -B*kInt5; C, 0])
```

Problem 3

```
clear , clc
%% Insert provided data here:

%% Useful Relations
kref = -1/(C*(A-B*K)^-1*B);
G = [C 0];
E = [A-B*K -B*kInt ; G];
F1 = [B*kref ; -1];
F2 = [B ;0];

Am = E;
Bm = [F1 F2];
Cm = G;
um = [r; d];
I = eye(length(Am));

%% Equations
z0 = [x0;0];

z = expm(Am*(t1-t0))*z0+Am^-1*(expm(Am*(t1-t0))-I)*Bm*um; %Only the (N-1)
    values
y = Cm*z
```

Problem 4

```
clear, clc
%% Insert provided data here:

%% Useful Relations
kref = -1/(C*(A-B*K)^-1*B);
G = [C 0];
E = [A-B*K -B*kInt ; G];
F1 = [B*kref ; -1];
F2 = [B ;0];

Am = E;
Bm = [F1 F2];
Cm = G;
um = [r; d];
I = eye(length(Am));
syms t
%% Equations
z0 = [x0;0];
z = expm(Am*t)*z0+Am^-1*(expm(Am*t)-I)*Bm*um; %Only the (N-1) values
y = Cm*z
```

Problem 5

```
clear,close all, clc
%% Insert provided data here:

%% Useful Relations
kRef = kRef(A,B,C,K)       %Make sure you have kRef() in your folder/path
G = [C, 0];
E = [A-B*K -B*kInt ; G];
F1 = [B*kRef ; -1];
F1a = [B*kRef+B ; -1];
F2 = [B ;0];
Am = E;
Bm = [F1 F2];
Bma = [F1a F2];
Cm = G;

%% No disturbance
sys1 = ss(Am,Bm,Cm,D);
step(sys1)
figure(1)
S1 = stepinfo(sys1) %Look at the appended struct to calculate numbers


%% Disturbance
figure(2)
sys2 = ss(Am,Bma,Cm,D);
step(sys2)

S2 = stepinfo(sys2) %Look at the appended struct to calculate numbers
```