Problem 1

Table of Contents

Define Symbols

```
syms m1 m2 q1 q2 q3 a b c g
syms dq1 dq2 dq3 ddq1 ddq2 ddq3
syms tau1 force2 force3
```

Define State Vectors

```
q = [q1; q2; q3];
dq = [dq1; dq2; dq3];
ddq = [ddq1; ddq2; ddq3];
```

Put Torque vector into MCG Form

```
tau = [ddq1*m2*(b + q2)^2;-m2*(b + q2)*dq1^2 + ddq2*m2; m2*(ddq3 - g)]

M11 = simplify((tau(1) - subs(tau(1),ddq(1),0))/ddq(1));
M12 = simplify((tau(1) - subs(tau(1),ddq(2),0))/ddq(2));
M13 = simplify((tau(1) - subs(tau(1),ddq(3),0))/ddq(3));
M21 = simplify((tau(2) - subs(tau(2),ddq(1),0))/ddq(1));
M22 = simplify((tau(2) - subs(tau(2),ddq(2),0))/ddq(2));
M23 = simplify((tau(2) - subs(tau(2),ddq(3),0))/ddq(3));
M31 = simplify((tau(3) - subs(tau(3),ddq(1),0))/ddq(1));
M32 = simplify((tau(3) - subs(tau(3),ddq(2),0))/ddq(2));
M33 = simplify((tau(3) - subs(tau(3),ddq(3),0))/ddq(3));

M = [M11 M12 M13;
 M21 M22 M23;
 M31 M23 M33];
M = simplify(expand(M))

G = subs(tau, {ddq(1),ddq(2), ddq(3),dq(1),dq(2), dq(3)},
{0,0,0,0,0,0})

C1 = simplify(expand(tau(1) - M(1,:)*[ddq1 ddq2 ddq3].' - G(1)));
C2 = simplify(expand(tau(2) - M(2,:)*[ddq1 ddq2 ddq3].' - G(2)));
C3 = simplify(expand(tau(3) - M(3,:)*[ddq1 ddq2 ddq3].' - G(3)));
```

```
C = [C1;C2;C3]
```

```
tau =

             ddq1*m2*(b + q2)^2
 - m2*(b + q2)*dq1^2 + ddq2*m2
                  m2*(ddq3 - g)


M =

[ m2*(b + q2)^2,  0,  0]
[             0, m2,  0]
[             0,  0, m2]


G =

     0
     0
 -g*m2


C =

                  0
 -dq1^2*m2*(b + q2)
                  0
```

Torque and Force equations

```
torqEq1 = M(1,1)*ddq1+C(1)*dq1+G(1) == tau1;
forceEq2 = M(2,2)*ddq2+C(2)*dq2+G(2) == force2;
forceEq3 = M(3,3)*ddq3+C(3)*dq3+G(3) ==force3;
```

Change to State Space Format

establish state vector

```
x1 = q1; x2 = dq1;
x3 = q2; x4 = dq2;
x5 = q3; x6 = dq3;

% find state space
dx1 = x2;
dx2 = solve(torqEq1,ddq1);
dx3 = x4;
dx4 = solve(forceEq2,ddq2);
dx5 = x6;
dx6 = solve(forceEq3,ddq3);
```

```
stateSpace = [dx1;dx2;dx3;dx4;dx5;dx6]
```

```
stateSpace =

                                        dq1
                  tau1/(m2*(b + q2)^2)
                                        dq2
 (dq2*m2*(b + q2)*dq1^2 + force2)/m2
                                        dq3
                    (force3 + g*m2)/m2
```

Published with MATLAB® R2018b

Question 2

Table of Contents

Establish symbolic variables

```
syms a b c d
```

Part A

```
A1 = [   -2  1   0   0   0;...
        0   -1  1   0   0;...
        0   0   -2  1   0;...
        0   0   0   -2  1;...
        0   0   0   0   -1]
B1 = [0 0; 0 0; a b ; c 0; 0 d]

testM = [B1 A1*B1 (A1^2)*B1 (A1^3)*B1 (A1^4)*B1]
rank1 = rank(testM)
% In order for A1 and B1 to allow reachability, the matrix testM must
 be Full Row Rank
% In this case testM has 5 rows, so rank(testM) must return 5
% If a, b, c, d are all diferent values then the pair will always be
% reachable. However if some of their values are identical they
% pair may or may not lead to reachability.
% If b and d or c and d equal zero then testM loses rank and becomes
% not reachable.
```

```
A1 =

    -2     1     0     0     0
     0    -1     1     0     0
     0     0    -2     1     0
     0     0     0    -2     1
     0     0     0     0    -1


B1 =

[ 0, 0]
[ 0, 0]
[ a, b]
[ c, 0]
[ 0, d]
```

```
testM =

[ 0, 0,       0,     0,        a,       b,    c - 5*a,        -5*b,
 17*a - 7*c,     17*b + d]
[ 0, 0,       a,     b,    c - 3*a,    -3*b,  7*a - 5*c,     7*b + d,
 17*c - 15*a, - 15*b - 6*d]
[ a, b, c - 2*a, -2*b, 4*a - 4*c, 4*b + d, 12*c - 8*a, - 8*b - 5*d,
 16*a - 32*c,  16*b + 17*d]
[ c, 0,    -2*c,     d,      4*c,    -3*d,       -8*c,         7*d,
     16*c,        -15*d]
[ 0, d,       0,    -d,        0,       d,          0,          -d,
        0,            d]


rank1 =

     5
```

Part B

```
A2 = [   -1  1   0   0   0;...
        0   0  1   0   0;...
        0   0   -1  0   0;...
        0   0   0   -1  1;...
        0   0   0   0   0];
B2 = [0 0; 0 0;a b; 0 0; c d];

testM2 = [B2 A2*B2 (A2^2)*B2 (A2^3)*B2 (A2^4)*B2 ]
rank2 = rank(testM2)

% In order for A2 and B2 to allow reachability, the matrix testM must
 be Full Row Rank
% In this case testM2 has 5 rows, so rank(testM2) must return 5
% If a, b, c, d are all diferent values then the pair will always be
% reachable. However if some of their values are identical they
% pair may or may not lead to reachability.
% If a=c and b=d pair is not reachable. If a and b or a and c or b and
 d or c and d equal zero then testM loses rank and becomes
% not reachable.


testM2 =

[ 0, 0,  0,  0,  a,  b, -2*a, -2*b, 3*a, 3*b]
[ 0, 0,  a,  b, -a, -b,    a,    b,  -a,  -b]
[ a, b, -a, -b,  a,  b,   -a,   -b,   a,   b]
[ 0, 0,  c,  d, -c, -d,    c,    d,  -c,  -d]
[ c, d,  0,  0,  0,  0,    0,    0,   0,   0]
```

```
rank2 =

     5
```

Published with MATLAB® R2018b

Table of Contents

Question 3

```
%%Setup
syms k1 k2 lambda
A = [0 1; 1 0];
B = [0;1];
```

Make M Matrix

M matrix is based on linear state feedback control

```
M = A - B*[k1 k2]
```

```
M =

[      0,   1]
[ 1 - k1, -k2]
```

Make Characteristic Equation and Subsitute estimator poles

```
charEq = det(M - lambda*[1 0;0 1]) == 0
eq1 = subs(charEq, lambda, -2 )
eq2 = subs(charEq, lambda, -3 )
```

```
charEq =

lambda^2 + k2*lambda + k1 - 1 == 0


eq1 =

k1 - 2*k2 + 3 == 0


eq2 =

k1 - 3*k2 + 8 == 0
```

Solve for Ks

```
eqs = [eq1 eq2];
ks = [k1 k2]
[solvK1, solvK2] = solve(eqs, ks)

K = [solvK1; solvK2]

% The controller is a linear state feedback controller
% The output state is used to affect input
% input to the system becomes U = -K*x
```

```
ks =

[ k1, k2]


solvK1 =

7


solvK2 =

5


K =

 7
 5
```

Published with MATLAB® R2018b

Question 4

Table of Contents

Part A Obtain State space

I took the solved dyanmics from 1.7.4 of Zaks textbook, but with a point mass and no friction adjustments

```
syms m M I l theta dtheta ddtheta ddx g fc u a

%a = 1/(m+M)

diffdiffx = -m*a*l*ddtheta*cos(theta)+m*a*l*dtheta^2*sin(theta)-a*fc
+a*u

eq = I*ddtheta == m*g*l*sin(theta)-m*l^2*ddtheta-
m*diffdiffx*l*cos(theta)

diffdifftheta = solve(eq,ddtheta)
diffdifftheta = simplify(subs(diffdifftheta, I, m*l^2))
diffdifftheta = simplify(subs(diffdifftheta, fc, 0))

diffdiffx = subs(diffdiffx, ddtheta, diffdifftheta)
diffdiffx = simplify(subs(diffdiffx, I, m*l^2))
diffdiffx = simplify(subs(diffdiffx, fc, 0))

syms x1 x2 x3 x4

output = x1

dx1 = x2
dx2 = diffdiffx
dx2 = subs(dx2, [theta dtheta], [x1 x2])
pretty(dx2)
dx3 = x4
dx4 = diffdifftheta
dx4 = subs(dx4, [theta dtheta], [x1 x2])

g1 = 0
f1 = dx1

g2 = coeffs(dx2, u);
g2 = g2(2)
f2 = simplify(dx2-g2*u)

g3 = 0
f3 = dx3
```

```
g4 = coeffs(dx4, u);
g4 = g4(2)
f4 = simplify(dx4-g4*u)
```

diffdiffx =

*a*l*m*sin(theta)*dtheta^2 - a*fc + a*u - a*ddtheta*l*m*cos(theta)*

```
eq =

I*ddtheta == g*l*m*sin(theta) - ddtheta*l^2*m + l*m*cos(theta)*(-
 a*l*m*sin(theta)*dtheta^2 + a*fc - a*u + a*ddtheta*l*m*cos(theta))


diffdifftheta =

-(l*m*cos(theta)*(a*l*m*sin(theta)*dtheta^2 - a*fc + a*u) -
 g*l*m*sin(theta))/(I + l^2*m - a*l^2*m^2*cos(theta)^2)


diffdifftheta =

-(l*m*cos(theta)*(a*l*m*sin(theta)*dtheta^2 - a*fc + a*u) -
 g*l*m*sin(theta))/(2*l^2*m - a*l^2*m^2*cos(theta)^2)


diffdifftheta =

(a*l*m*cos(theta)*sin(theta)*dtheta^2 - g*sin(theta) +
 a*u*cos(theta))/(l*(a*m*cos(theta)^2 - 2))


diffdiffx =

a*u - a*fc + a*dtheta^2*l*m*sin(theta) -
 (a*m*cos(theta)*(a*l*m*cos(theta)*sin(theta)*dtheta^2 - g*sin(theta)
 + a*u*cos(theta)))/(a*m*cos(theta)^2 - 2)


diffdiffx =

-(a*(2*u - 2*fc + a*fc*m*cos(theta)^2 + 2*dtheta^2*l*m*sin(theta) -
 g*m*cos(theta)*sin(theta)))/(a*m*cos(theta)^2 - 2)


diffdiffx =

-(a*(2*l*m*sin(theta)*dtheta^2 + 2*u - g*m*cos(theta)*sin(theta)))/
(a*m*cos(theta)^2 - 2)
```

```
output =

x1


dx1 =

x2


dx2 =

-(a*(2*l*m*sin(theta)*dtheta^2 + 2*u - g*m*cos(theta)*sin(theta)))/
(a*m*cos(theta)^2 - 2)


dx2 =

-(a*(2*l*m*sin(x1)*x2^2 + 2*u - g*m*cos(x1)*sin(x1)))/(a*m*cos(x1)^2 -
 2)

                   2
  a (l m sin(x1) x2  2 + 2 u - g m cos(x1) sin(x1))
- -------------------------------------------------
                              2
                   a m cos(x1)  - 2


dx3 =

x4


dx4 =

(a*l*m*cos(theta)*sin(theta)*dtheta^2 - g*sin(theta) +
 a*u*cos(theta))/(l*(a*m*cos(theta)^2 - 2))


dx4 =

(a*l*m*cos(x1)*sin(x1)*x2^2 - g*sin(x1) + a*u*cos(x1))/
(l*(a*m*cos(x1)^2 - 2))


g1 =

     0


f1 =

x2
```

```
g2 =

-(2*a)/(a*m*cos(x1)^2 - 2)


f2 =

(a*m*sin(x1)*(- 2*l*x2^2 + g*cos(x1)))/(a*m*cos(x1)^2 - 2)


g3 =

     0


f3 =

x4


g4 =

(a*cos(x1))/(l*(a*m*cos(x1)^2 - 2))


f4 =

-(sin(x1)*(- a*l*m*cos(x1)*x2^2 + g))/(l*(a*m*cos(x1)^2 - 2))
```

Part B

```
syms x1e x3e u1e u3e

dx2 = subs(dx2, a, (1/(m+M)))
dx4 = subs(dx4, a, (1/(m+M)))

dx2 = subs(dx2, [l m M g], [1 0.1 1 10])
dx4 = subs(dx4, [l m M g], [1 0.1 1 10])

dxAll = [dx1;dx2;dx3;dx4]

A = simplify([diff(dxAll,x1),
 diff(dxAll,x2),diff(dxAll,x3),diff(dxAll,x4)])

B = simplify(diff(dxAll,u))


dx2 =

-(2*l*m*sin(x1)*x2^2 + 2*u - g*m*cos(x1)*sin(x1))/((M +
 m)*((m*cos(x1)^2)/(M + m) - 2))
```

```
dx4 =

((l*m*cos(x1)*sin(x1)*x2^2)/(M + m) - g*sin(x1) + (u*cos(x1))/(M +
 m))/(l*((m*cos(x1)^2)/(M + m) - 2))


dx2 =

-(10*((sin(x1)*x2^2)/5 + 2*u - cos(x1)*sin(x1)))/(11*(cos(x1)^2/11 -
 2))


dx4 =

((cos(x1)*sin(x1)*x2^2)/11 - 10*sin(x1) + (10*u*cos(x1))/11)/
(cos(x1)^2/11 - 2)


dxAll =


        x2
        -(10*((sin(x1)*x2^2)/5 + 2*u - cos(x1)*sin(x1)))/
(11*(cos(x1)^2/11 - 2))

        x4
 ((cos(x1)*sin(x1)*x2^2)/11 - 10*sin(x1) + (10*u*cos(x1))/11)/
(cos(x1)^2/11 - 2)


A =

[

                                    0,
 1, 0, 0]
[                                   - (2*x2^2*cos(x1) - 20*cos(x1)^2 +
 10)/(cos(x1)^2 - 22) - (20*cos(x1)*sin(x1)*((sin(x1)*x2^2)/5 + 2*u -
 cos(x1)*sin(x1)))/(cos(x1)^2 - 22)^2,   (4*x2*sin(x1))/(sin(x1)^2 +
 21), 0, 0]
[

                                    0,
 0, 0, 1]
[ (22*cos(x1)*sin(x1)*((cos(x1)*sin(x1)*x2^2)/11 - 10*sin(x1)
 + (10*u*cos(x1))/11))/(cos(x1)^2 - 22)^2 - (110*cos(x1) -
 2*x2^2*cos(x1)^2 + 10*u*sin(x1) + x2^2)/(cos(x1)^2 - 22),
 (2*x2*sin(2*x1))/(cos(2*x1) - 43), 0, 0]


B =
```

```
                    0
   -20/(cos(x1)^2 - 22)
                    0
(10*cos(x1))/(cos(x1)^2 - 22)
```

Published with MATLAB® R2018b