16-642 Manipulation, Estimation, and Control
Problem Set 2

David Wong (Andrew ID: DBWONG)
collaborator: Kelvin Kang

Q1 (a)

G(s)= $\frac{200}{s^3+22s^2+141s+2}$

T(s)= $\frac{200}{s^3+22s^2+141s+202}$

MATLAB code:

```
K = 1 %negative unity feedback assumed by default for feedback function in MATLAB
G = tf([0 200],[1 22 141 2])
T = feedback(G,K)
```

Q1(b)

Poles =

-10.0000 + 1.0000i

-10.0000 - 1.0000i

-2.0000 + 0.0000i

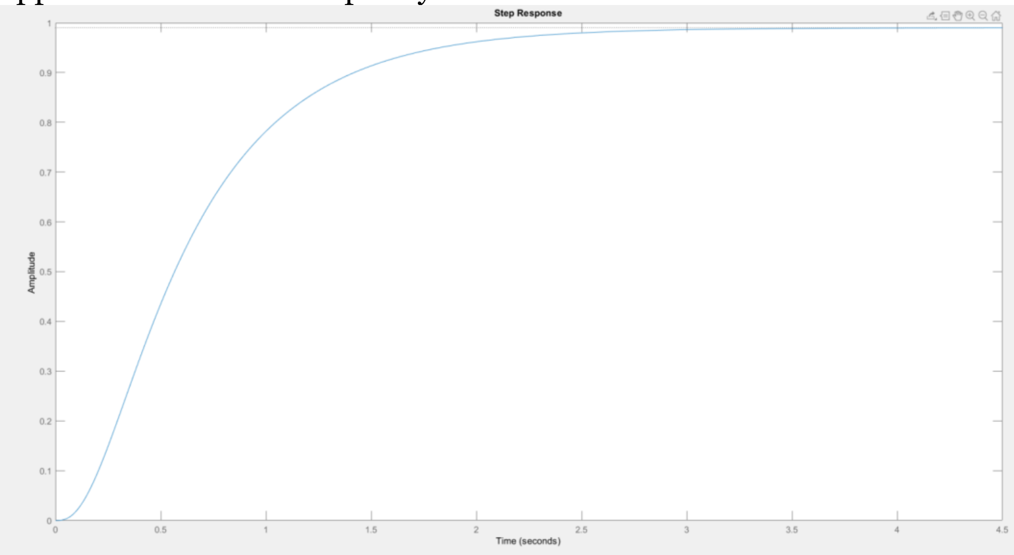
No Zeros (function returns a 0×1 empty double column vector)

MATLAB code:

```
%Q1b find poles and zeros
sys = T
P = pole(sys)
Z = zero(sys)
```

Q1(C)
Plot y(t) using step. Discuss which poles dominate the response
-2.0000 + 0.0000i

The pole that dominates the response is the one nearest to the imaginary axis. This appears to be a overdamped system.




```
MATLAB CODE:
%Q1c
step(sys)
```

Q1 (D)
Steady state value with Final Value Theorem

T(s)= $\frac{200}{s^3+22s^2+141s+202}$

Final value theorem:

$$\lim_{s=0} = sT(s) = 200/202$$

Steady state value is 0.9901

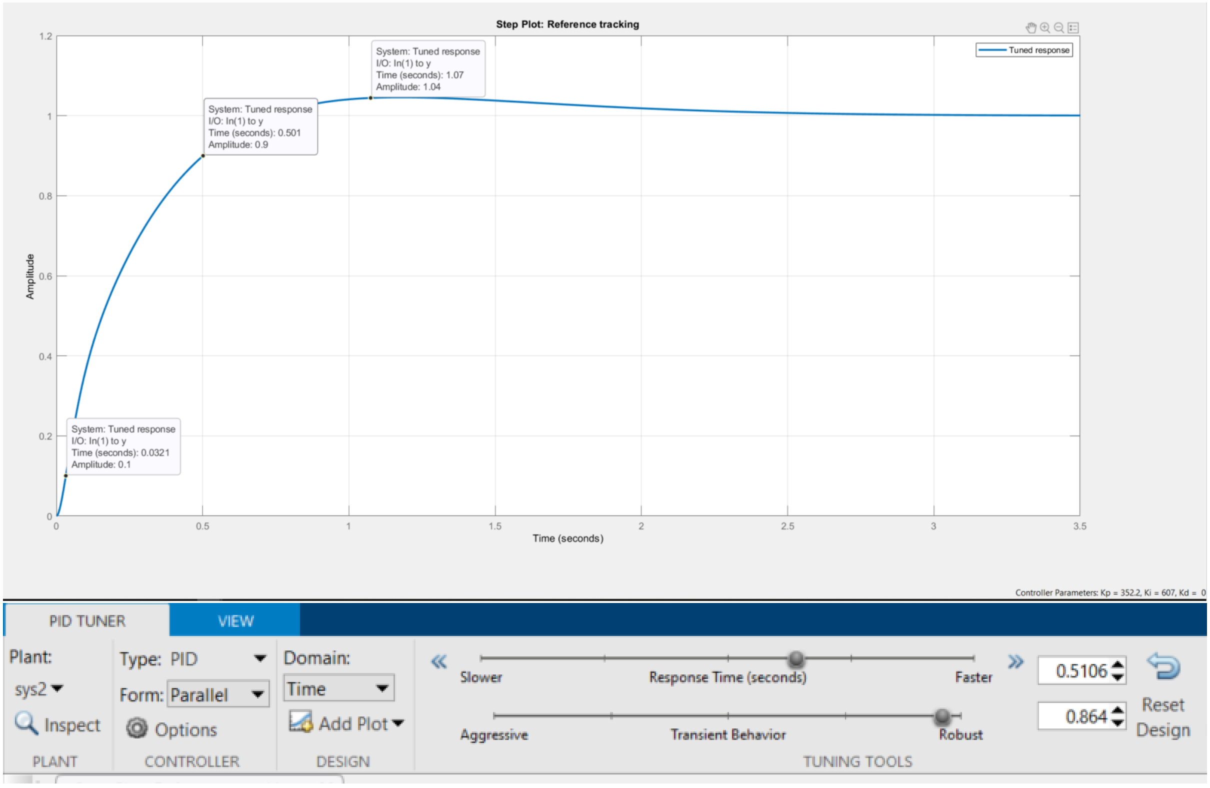
Q2

Rise time = 0.471s
%overshoot = 4%

Controller Parameters: Kp = 352.2, Ki = 607, Kd = 0

```
%Q2
numerator = [1 10];
denominator = [1 71 1070 1000];
sys2 = tf(numerator,denominator)

pidTuner(sys2,'PID')
```



Q3 Observer for cart-pendulum system

Employed MATLAB to construct observability matrix and used tank command to check for system observability:

Observability = 4. (full rank, hence system is observable)

System poles:

0

-3.3301

1.1284

-0.7984

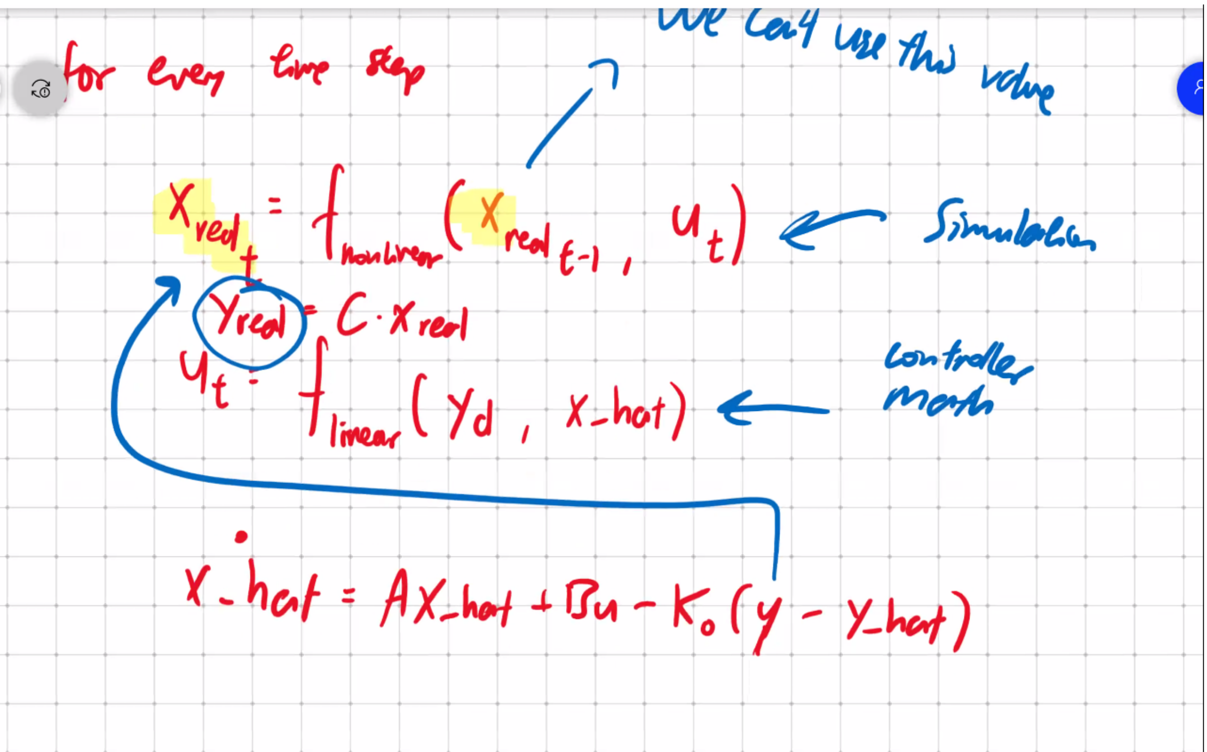
Therefore observer poles via pole placement(> 5x system poles):

obsPole = -20 -30 -40 -50

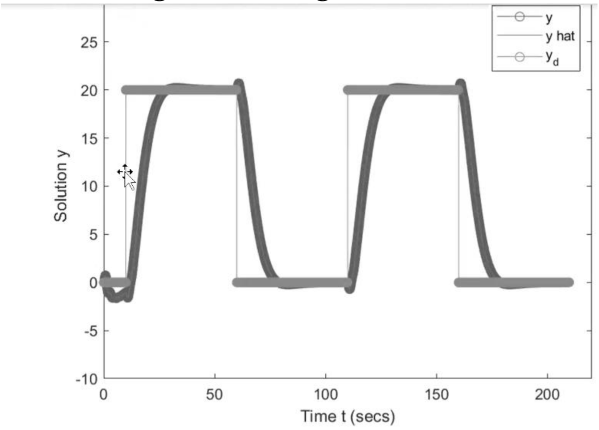
Ko =

1.0e+04 *

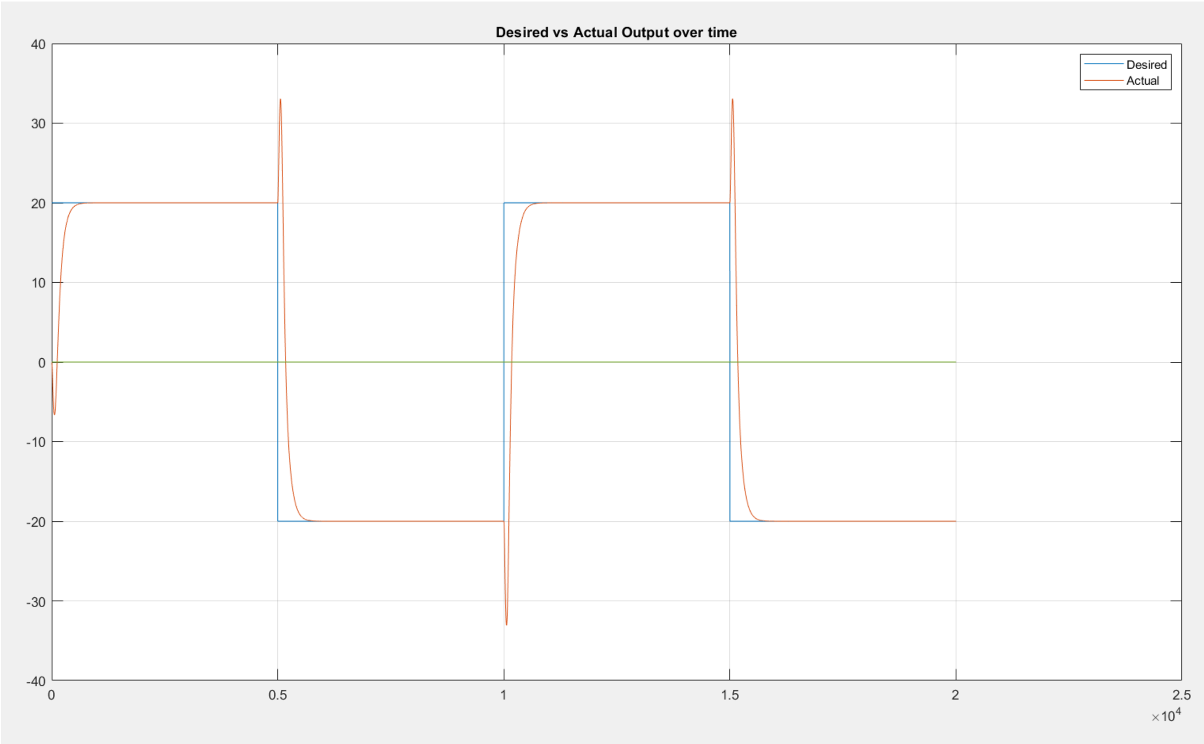
0.0004, 0.3956, 0.0172, 3.1123



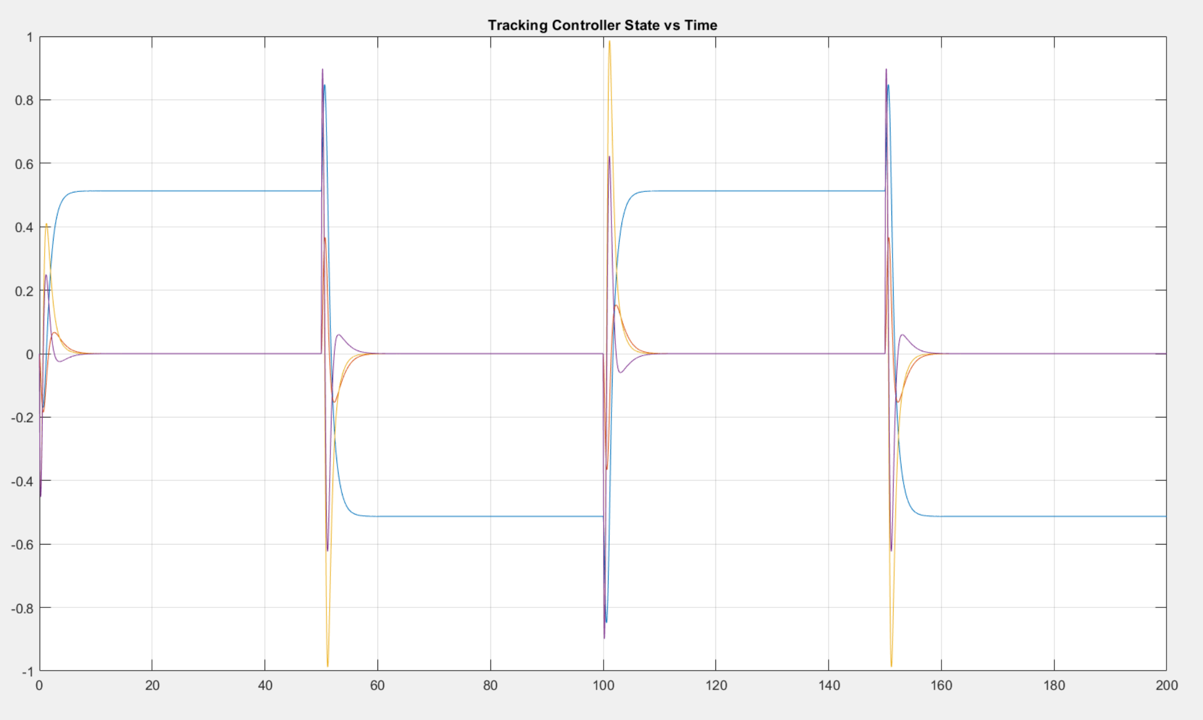
Expected plot should look something like this (which exhibits a worse tracking performance than the original tracking controller as we are tracking xhat instead of x directly)



Original Tracking controller plot of outputs with desired square wave



Tracking controller state vs time




```
% MEC PS2
%David Wong (Andrew ID: DBWONG)

%Q1a
clear all;close all;

K = 1 %negative unity feedback; feeback function assumes
-1 by default
G = tf([0 200],[1 22 141 2])
T = feedback(G,K)

%Q1b find poles and zeros
sys = T
P = pole(sys)
Z = zero(sys)

%Q1c
step(sys)


%Q2
numerator = [1 10];
denominator = [1 71 1070 1000];
sys2 = tf(numerator,denominator)
```

```
pidTuner(sys2,'PID')

%% Q3
clear all;
close all;
%Ref HW1 Q2h
tspan = 0:0.01:200;

gamma = 2;
alpha = 1;
beta = 1;
D = 1;
mu = 3;

R = 10;
Q = [1000 0 0 0 ; 0 5 0 0 ; 0 0 1 0 ; 0 0 0 5];
x_00 = [0; 0; 0; 0];         % initial condition for 2g

A = [0,0,1,0;0,0,0,1;0,1,-3,0;0,2,-3,0]
B = [0;0;1;1]
C = [39,0,0,0];
D = [0,0,0,0; 0,0,0,0; 0,0,0,0; 0,0,0,0];

[K,S,e] = lqr(A,B,Q,R)

[tout,xout] = ode45(@(t,x)
trackingController(t,x,K,C,A,B), tspan, x_00);

figure
%plot(tout,xout)
grid
title('Tracking Controller State vs Time')

yout = C.*xout;
xhatdot = A*xhat + B*u - Ko(yout-yhat)
%%PARTIAL CODE

tf=200;
T = 0.01;
yd1= 20*ones(1,(tf/4)/T);
yd2= -20*ones(1,(tf/4)/T);
yd = cat(2, yd1,yd2,yd1,yd2);

figure
plot(yd)
hold on
%plot(yout)
grid
```

```
title('Desired vs Actual Output over time')
legend('Desired','Actual')
hold off

% OBSERVER PLACEMENT
sys3 = ss(A,B,C,0)
ob = obsv(sys3)
observability = rank(ob)
sysPoles = eig(A) %system poles


A_transp = transpose(A);
C_transp = transpose(C);

%Kc from LQR :
Kc = [-9.99999999999901, 34.1720722633026, -
28.5448907859479, 33.1569959170274]

%define observer poles as >5x dominant system poles
obsPole = [-20, -30, -40, -50]


L = place(A_transp, C_transp, obsPole)
Ko = transpose(L)

At = [A-B*.K, B.*K;
       zeros(size(A)), A-L.*C];

Ct = [C, zeros(size(C))];

sys = ss(At,Bt,Ct,0);
lsim(sys,zeros(size(t)),t,[x0 x0]);

%%%%%%%%%%%%%%%%


tf = 200;
step = 0.01;
numIteration = ceil (tf/step);
x = zeros(1,numIteration);
xhat = zeros(1,numIteration);
xdot = zeros(1,numIteration);
xhatdot = zeros(1,numIteration);
zdot = [A, -B*Kc; Ko*C, A-B*Kc-Ko*C]

A_fb = zdot;
disp('Q5.2 resultant Poles of A_fb:')
disp(eig(A_fb));
```

```matlab
for k = 1:numIteration-1
    k1 = T*(A_fb*x(:,k));
    k2 = T*(A_fb*(x(:,k)+0.5*k1));
    k3 = T*(A_fb*(x(:,k)+0.5*k2));
    k4 = T*(A_fb*(x(:,k)+k3));
    x(:,k+1) = x(:,k) + k1/6 + k2/3 + k3/3 + k4/6;
    t(k+1) = t(k) + T;
    z
end
```

```matlab
function xdot = trackingController(t,x,K,C,A,B);

x_2 = x(2);%phi
x_3 = x(3);%xcdot
x_4 = x(4);%phidot
yd=0;
if t<50
        yd = 20;
    end
    if t>=50 && t<100
        yd = -20;
    end
    if t>=100 && t<150
        yd = 20;
    end
    if t>=150 && t<200
        yd = -20;
    end

ABK= A-B*K;
ABKinv =inv(ABK);

v_partial = -(C*(ABKinv)*B)^(-1);
v = v_partial * yd;
F = v-K*x;

xcddot=(sin(x_2)*x_4^2 - F + 3*x_3)/(cos(x_2)^2 - 2) -
(cos(x_2)*sin(x_2))/(cos(x_2)^2 - 2)
phiddot=(cos(x_2)*(sin(x_2)*x_4^2 - F +
3*x_3))/(cos(x_2)^2 - 2) - (2*sin(x_2))/(cos(x_2)^2 - 2)
```

```
xdot = [x_3;x_4;xcddot;phiddot];
end
```