Table of Contents

Swenson MAE 673 HW 3

```
clear; close all; clc;
```

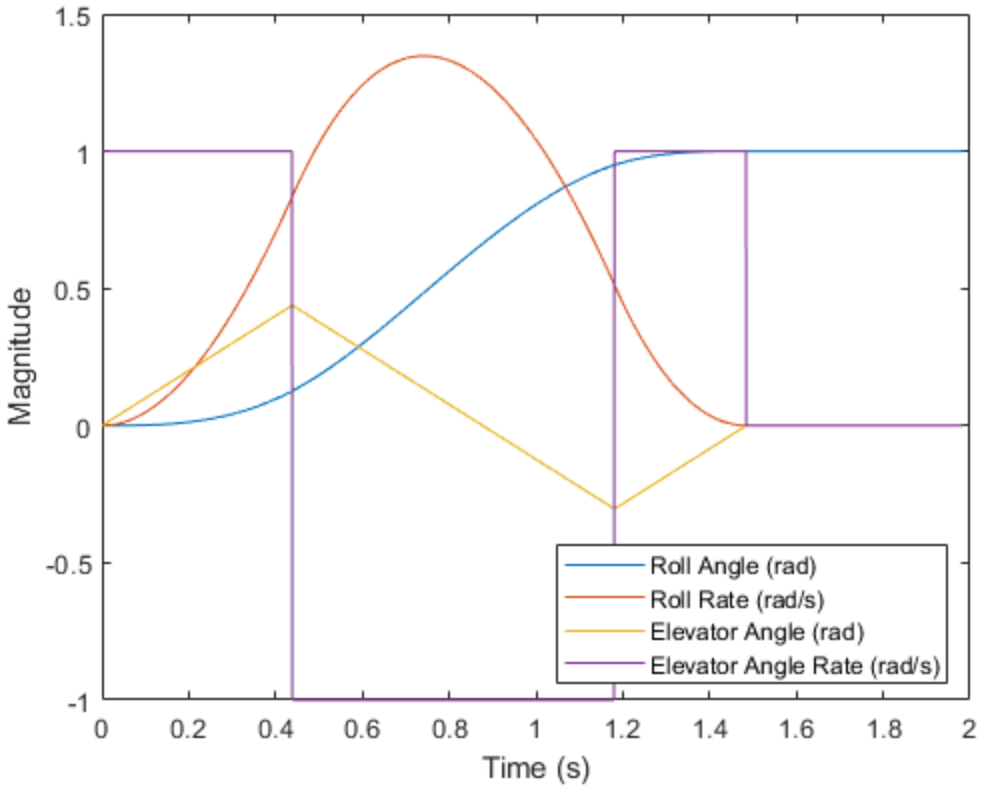
Part 1: Optimal Time Filter with Known K

System Parameters

```
Tau = 1;
c = 1/Tau;
Q = 10;

% A is expanded to include control state
A = [0 1 0; 0 -c c*Q; 0 0 0];
B = [0 0 1]';
C = [1 1 1];
D = 0;
As = -A';
As2 = As*As; As3 = As2*As; As4 = As3*As;

sys = ss(A,B,C,D);;
[num, den] = ss2tf(A,B,C,D);


% Need to do parameter optimazation
optfun = @(x) x(3);
nonlincon = @(x) nonlcon(x);

% Constraints
Aineq = [-1 0 0 ;
         1 -1 0 ;
         0 1 -1 ];
Bineq = [0 0 0 ]';
Aeq = [2 -2 1];
Beq = 0;
LB = [0 0 0 ]; UB = [20 20 20 ];
x0 = [1 0 0];

xout = fmincon(optfun,x0,Aineq,Bineq,Aeq,Beq,LB,UB,nonlincon)
```

Need to check that these are the optimal switches

```
T1 = xout(1);
T2 = xout(2);
T3 = xout(3);

P = [10*( exp(T1)-1-T1 ) -10*( exp(T1)-1 ) 1;
     10*( exp(T2)-1-T2 ) -10*( exp(T2)-1 ) 1];
lambda0 = null(P);

Ant = -A';
% Check that my swtiching fnction equals zero (to machine precision)
Switch1 = B'*expm(Ant.*T1)*lambda0;
Switch2 = B'*expm(Ant.*T2)*lambda0;


t = linspace(0,T3+.5,100001)';
U = 1 - 2*heaviside(t-T1) + 2*heaviside(t-T2) - heaviside(t-T3);

[YY,TT,XX] = lsim(sys,U,t,[0 0 0]');

figure();
plot(t,XX(:,1),t,XX(:,2),t,XX(:,3),t,U)
xlabel('Time (s)');
ylabel('Magnitude');
legend('Roll Angle (rad)','Roll Rate (rad/s)','Elevator Angle (rad)',...
       'Elevator Angle Rate (rad/s)','location','best')
```



Part 2

```
% An attempt at using syms
% syms dydt ddydtt dvdt v dphidc ddphidcc c Q
% As = [0 1 0 0; 0 -c c*Q 1; 0 0 0 0; 0 -1 Q -c]
% Bs = [0 0 1 0]';
% Cs = [1 0 0 0];
% Ds = 0;
% [num,den] = ss2tf(As,Bs,Cs,Ds)

Q = 10;
c = 1/Tau;

As = [0 1 0 0; 0 -c c*Q 0; 0 0 0 0; 0 -1 Q -c];
;
% As = [0 1 0 0; 0 -c c*Q 1; 0 0 0 0; 0 0 0 -1];
Bs = [0 0 1 0]';
Cs = [1 0 0 0];
Ds = 0;

[lambdas] = eig(As);
lamsort = sort(lambdas);
omegaC = lamsort(end,:);

sys2 = ss(As,Bs,Cs,Ds);
```

```
[num,den] = ss2tf(As,Bs,Cs,Ds)
roots(den)
```

Parameter Optimization

Need to do parameter optimazation

```
optfun2 = @(x) x(4);
nonlincon2 = @(x) nonlcon2(x,omegaC);

% Constraints
Aineq2 = [-1 0 0 0 ;
         1 -1 0 0 ;
         0 1 -1 0
         0 0 1 -1];
Bineq2 = [0 0 0 0]';
Aeq2 = [2 -2 2 -1];
Beq2 = 0;
LB2 = [0 0 0 0]; UB2 = [20 20 20 20];
x02 = [1 1 1 1];

xout2 = fmincon(optfun2,x02,Aineq2,Bineq2,Aeq2,Beq2,LB2,UB2,nonlincon2)
```

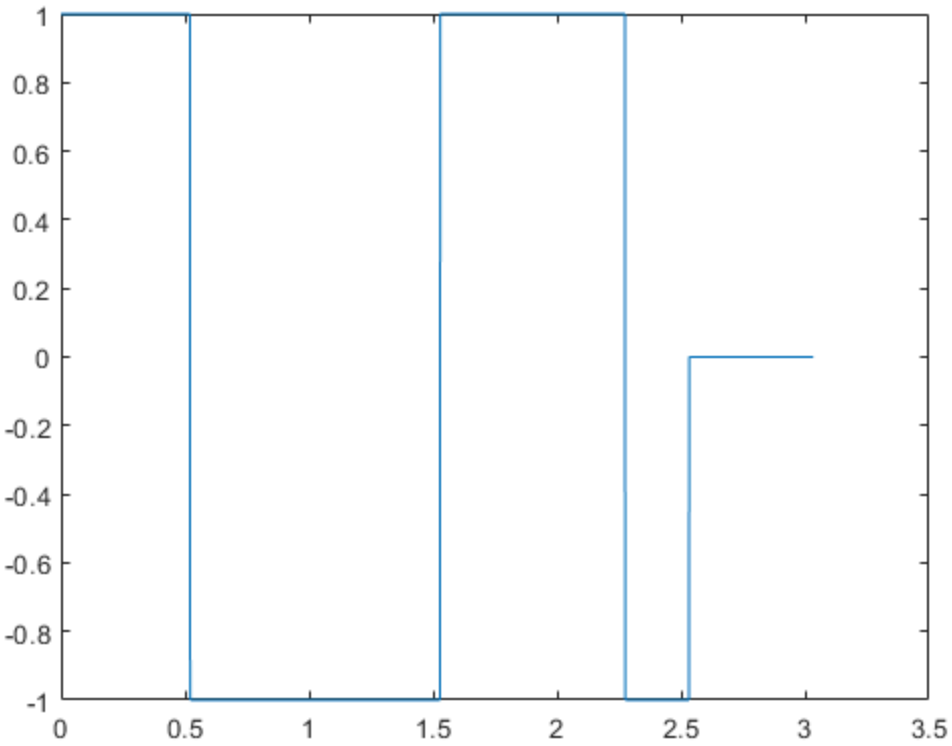
Need to check that these are the optimal switches

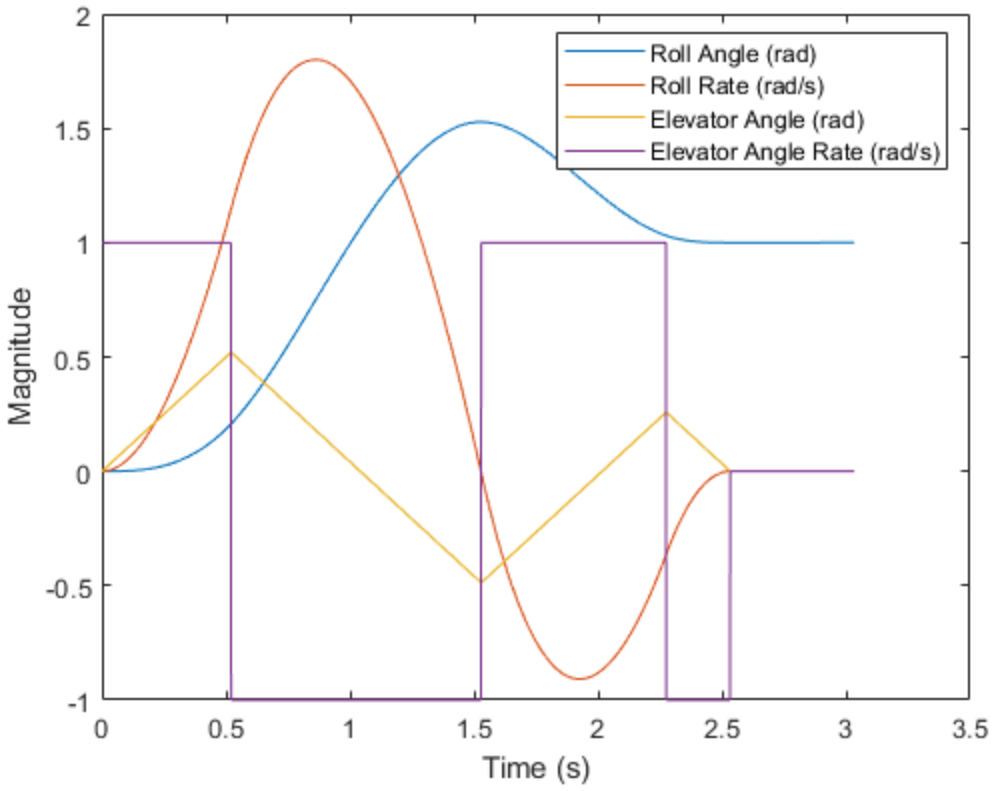
```
T12 = xout2(1);
T22 = xout2(2);
T32 = xout2(3);
T42 = xout2(4);

Ant = -As';

P2 = [Bs'*expm(Ant*T12);
      Bs'*expm(Ant*T22);
      Bs'*expm(Ant*T32);];
lambda02 = null(P2);



Switch12 = Bs'*expm(Ant*T12)*lambda02
Switch22 = Bs'*expm(Ant*T22)*lambda02
Switch32 = Bs'*expm(Ant*T32)*lambda02

% tvec = 0:.001:5';
% for j = 1:length(tvec)
%     switchfun(j) = Bs'*expm(-As'*tvec(j))*lambda02;
% end
%
% figure();
% plot(tvec,switchfun)
```

```
t2 = 0:.00001:T42+.5;
U2 = (1 - 2*heaviside(t2-T12) + 2*heaviside(t2-T22) - 2*heaviside(t2-T32) +
 heaviside(t2-T42));

figure()
plot(t2,U2)

[YY2,TT2,XX2] = lsim(sys2,U2,t2,[0 0 0 0]');

figure();
plot(t2,XX2(:,1),t2,XX2(:,2),t2,XX2(:,3),t2,U2)
xlabel('Time (s)');
ylabel('Magnitude');
legend('Roll Angle (rad)','Roll Rate (rad/s)','Elevator Angle (rad)',...
       'Elevator Angle Rate (rad/s)','location','best')
```

```
num =

         0         0         0   10.0000   10.0000


den =

     1     2     1     0     0


ans =

     0
     0
    -1
    -1


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


xout2 =

    0.5207    1.5290    2.2749    2.5332


Switch12 =

   3.3307e-16


Switch22 =
```

```
   1.7764e-15

Switch32 =

     0
```

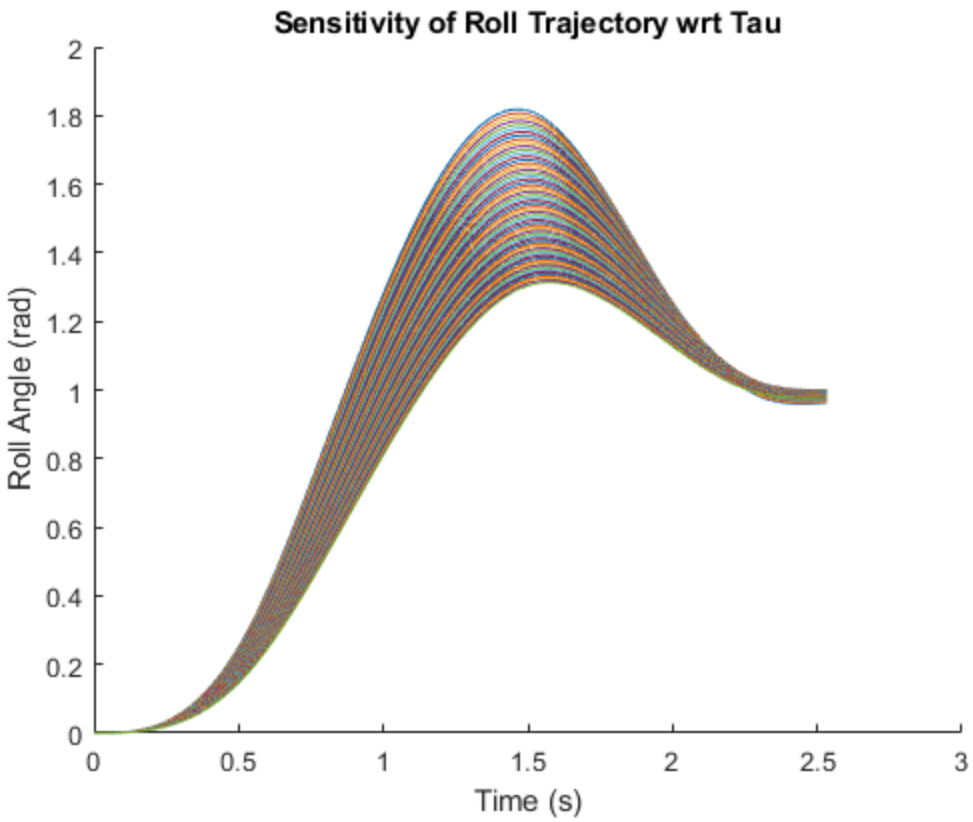


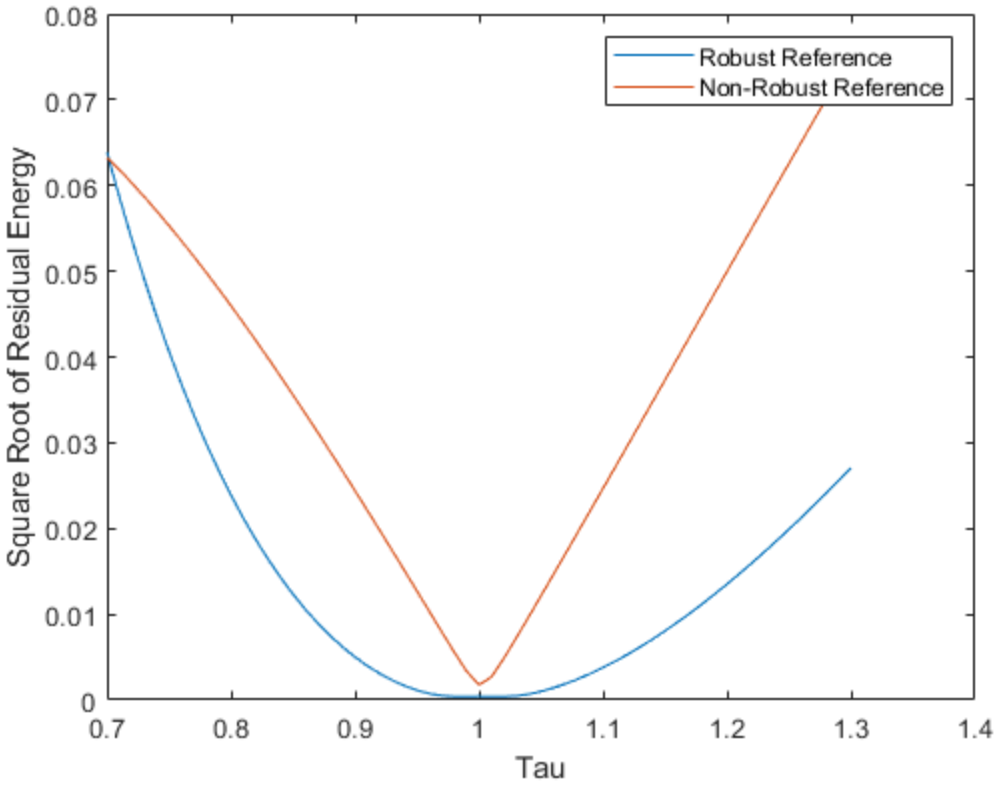


Sensitivity to variations in tau (C)

```
tauvec = .7:.01:1.3';

tvec = 0:.0001:T42;
Uvec1 = (1 - 2*heaviside(tvec-T12) + 2*heaviside(tvec-T22) - 2*heaviside(tvec-
T32) + heaviside(tvec-T42));
Uvec2 = 1 - 2*heaviside(tvec-T1) + 2*heaviside(tvec-T2) - heaviside(tvec-T3);

Jcost = zeros(length(tauvec),1);
Jcost2 = zeros(length(tauvec),1);

figure();
hold on;
for jj = 1:length(tauvec)

    Tau = tauvec(jj);
    c = 1/Tau;
    Q = 10;

    % A is expanded to include control state
    A = [0 1 0; 0 -c c*Q; 0 0 0];
    B = [0 0 1]';
    C = [1 0 0];
    D = 0;
```

```
    systemp = ss(A,B,C,D);

    [YYtemp,TTtemp,XXtemp] = lsim(systemp,Uvec1,tvec,[0 0 0]');
    [YYtemp2,TTtemp2,XXtemp2] = lsim(systemp,Uvec2,tvec,[0 0 0]');

    Jcost(jj,1) = (XXtemp(end,1)-1)^2 + (XXtemp(end,2))^2 + XXtemp(end,3)^2;
    Jcost2(jj,1) = (XXtemp2(end,1)-1)^2 + (XXtemp2(end,2))^2 +
 XXtemp2(end,3)^2;


    plot(tvec,XXtemp(:,1))


end
hold off
xlabel('Time (s)'); ylabel('Roll Angle (rad)'); title('Sensitivity of Roll
 Trajectory wrt Tau');


figure();
plot(tauvec,sqrt(Jcost),tauvec,sqrt(Jcost2))
xlabel('Tau'); ylabel('Square Root of Residual Energy');
legend('Robust Reference','Non-Robust Reference');
%}
```





Function Land

```
function [C,Ceq] = nonlcon(x)

    C = [];

    T1 = x(1);
    T2 = x(2);
    T3 = x(3);

    Ceq(1,1) = 1*(-10*T1^2 + 10*T2^2 - 5*T3^2 - 1);
    Ceq(1,2) = 1*(1 - 2*exp(T1) + 2*exp(T2) - exp(T3));


end

function [C2,Ceq2] = nonlcon2(x,omegaC)

    C2 = [];
    omg = omegaC;

    T1 = x(1);
    T2 = x(2);
    T3 = x(3);
    T4 = x(4);
```

```
    % Ceq2(1,1) = real(1*( 1 - 2*exp(-omg*T1) + 2*exp(-omg*T2) - 2*exp(-
omg*T3) + exp(-omg*T4) ));
    % Ceq2(1,2) = imag(1*( 1 - 2*exp(-omg*T1) + 2*exp(-omg*T2) - 2*exp(-
omg*T3) + exp(-omg*T4) ));
    % Ceq2(1,3) = 10*( (T1-T1^2)+(T2^2-T2)+(T3-T3^2) ) + 5*(T4^2-T4) - 1;

    Ceq2(1,1) = 1 - 2*exp(T1) + 2*exp(T2) - 2*exp(T3) + exp(T4);
    Ceq2(1,2) = 2*T1*exp(T1) - 2*T2*exp(T2) + 2*T3*exp(T3) - T4*exp(T4);
    Ceq2(1,3) = 10*(4*T1 - 4*T2 + 4*T3 - 2*T4 -2*T1^2 + 2*T2^2 - 2*T3^2 +
 T4^2) - 2;
end
```

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


xout =

    0.4385    1.1808    1.4847
```

Published with MATLAB® R2023a