# MECH 6V29 - MPC - Homework 3

## Table of Contents



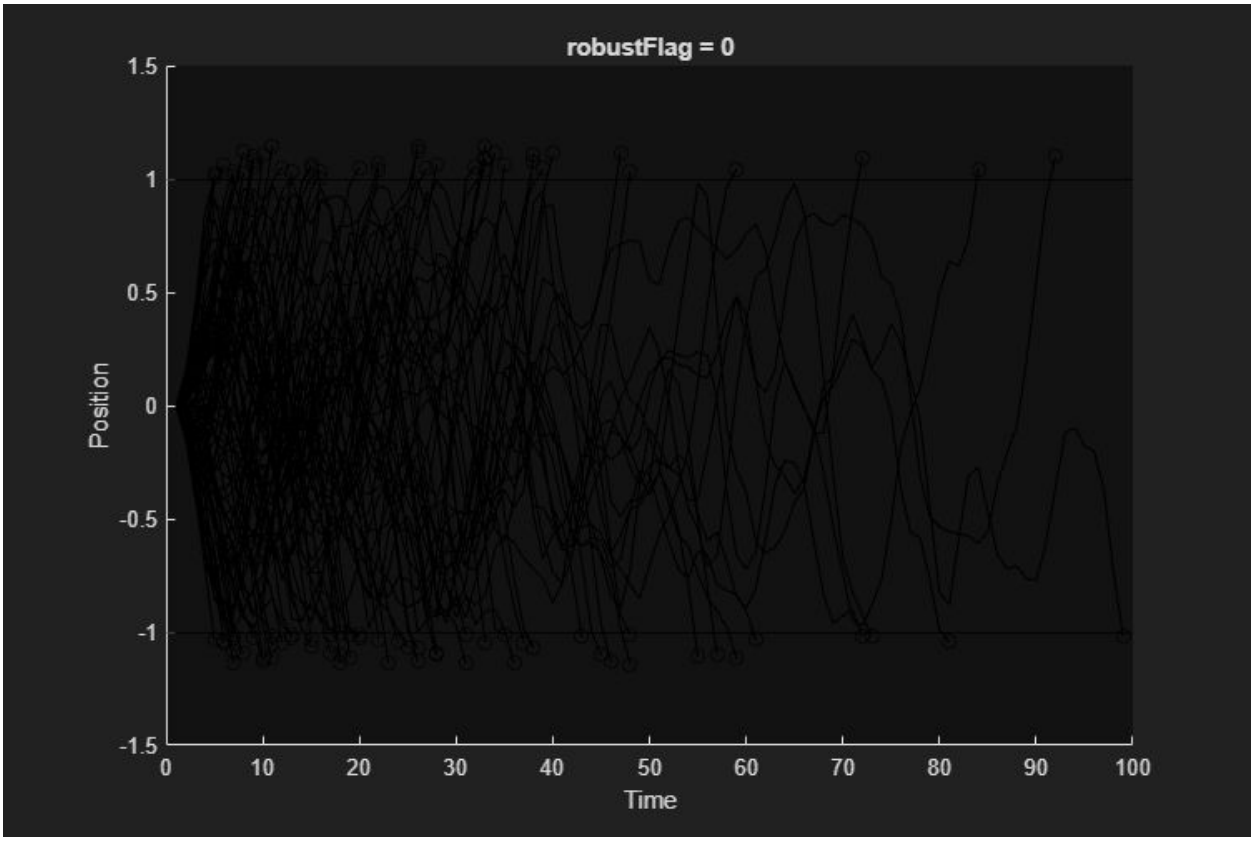
# Problem 1

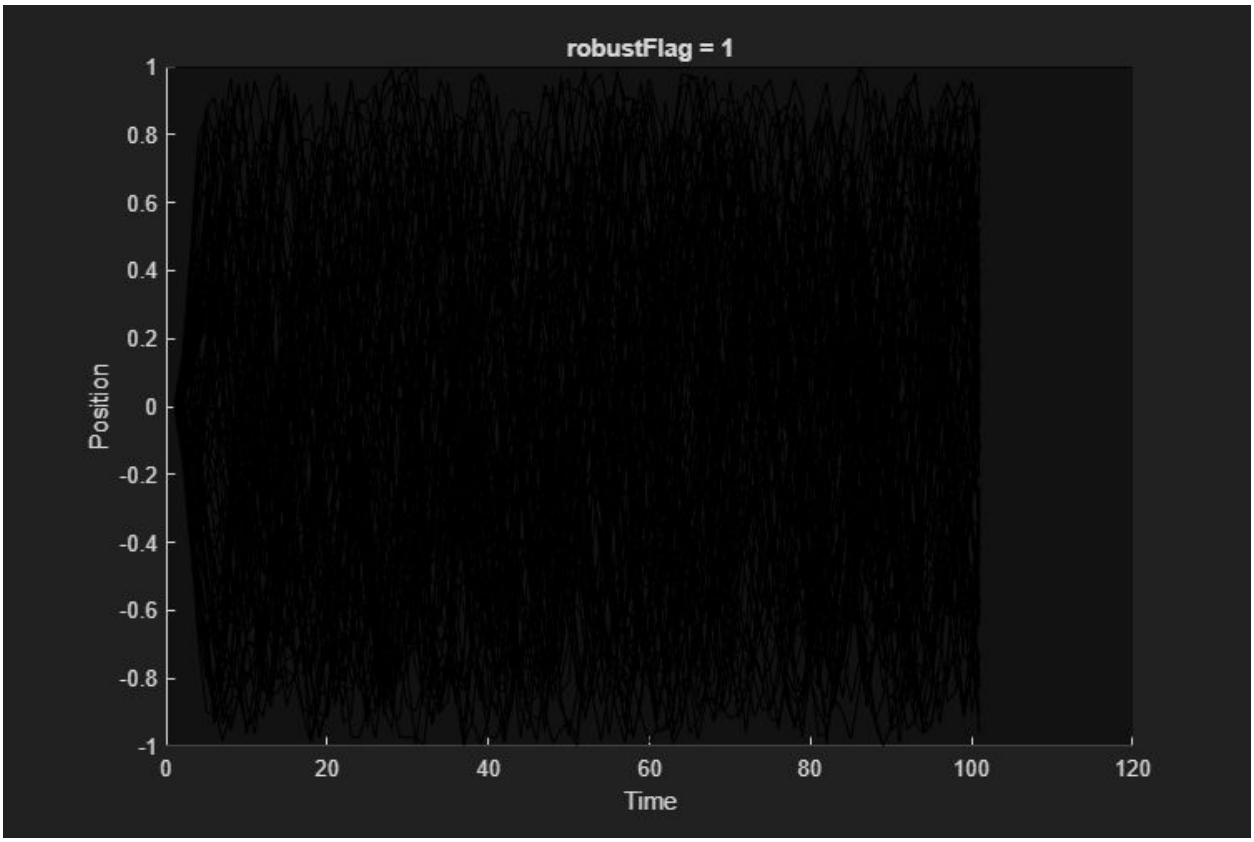
```
% 1a
A = [1, 1;
     0, 1];
B = [0.5;
     1];
C = eye(3,2);%<--- [1,0;0,1;0,0]
D(3,1) = 1; %<--- [0;0;1]
sys = ss(A,B,C,D,1)

N = 10;

% sizes
nx = size(A,1);
nu = size(B,2);
ny = size(C,1);


% 1b
K = -acker(A,B,zeros(nx,1))

% 1c
Y = Polyhedron('A', [eye(ny);-eye(ny)], 'b', ones(2*ny,1));
W = B*Polyhedron('A',[1;-1],'b',[0.3;0.3]);

Y_{1} = Y;% - (C+D*K)*W;
for j = 1:N
    Y_{j+1} = Y_{j} - (C+D*K)*(A+B*K)^(j-1)*W;
end

for robustFlag = [false, true]
```

# 1d ----- Setup Controller

```
P=0;
Q = 1e-3*eye(nx);
R = 100;

yalmip('clear'); clear('controller');
```

```
u_ = sdpvar(repmat(nu,1,N),ones(1,N));
x_ = sdpvar(repmat(nx,1,N),ones(1,N));

constraints = []; objective = 0;
for k = 1:N-1
    objective = objective + x_{k}'*Q*x_{k} + u_{k}'*R*u_{k};
    constraints = [constraints, x_{k+1} == A*x_{k} + B*u_{k}];
    if robustFlag
        constraints = [constraints, Y_{k}.A*(C*x_{k}+D*u_{k})<= Y_{k}.b];
    else
        constraints = [constraints, Y.A*(C*x_{k}+D*u_{k})<= Y.b];
    end
end
k = k + 1;
constraints = [constraints, x_{k} == 0];
if robustFlag; constraints = [constraints, Y_{k}.A*(C*x_{k}+D*u_{k})<=
Y_{k}.b]; end
objective = objective + x_{k}'*P*x_{k};

opts = sdpsettings;
controller = optimizer(constraints,objective,opts,x_{1},u_{1});


% simulate and plot
fig = figure(...
        WindowStyle="normal",...
        Position=[0 0 750 500]);
hold on
for i = 1:100
    rng(i);
    x0 = zeros(nx,1); tf = 100;
    V = num2cell(0.6*rand(nx,tf)-0.3);
    [X{i},U{i},~] = run_sim(A,B,V,controller, x0, tf);
    k_fail = find(~isfinite(U{i}),1,"first");
    plot(X{i}(1,:),'k')
    plot(k_fail,X{i}(1,k_fail),'ko')
end
yline(1,'k'); yline(-1,'k');
ylabel('Position');
xlabel('Time');
title(sprintf('robustFlag = %d',robustFlag))
saveas(fig,strcat('figs',filesep,sprintf('pblm1_robust=%d',robustFlag),'.png'
));
```


robustFlag = 0
Position
Time
1.5
1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
60
70
80
90
100


robustFlag = 1
Position
Time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
20
40
60
80
100
120

# Result Analysis

Cost

```
J_{100} = [];

for i = 1:100
    J_{i} = 0;
    for k = 1:tf-1
        J_{i} = J_{i} + X{i}(:,k)'*Q*X{i}(:,k) + U{i}(:,k)'*R*U{i}(:,k);
    end
    J_{i} = J_{i} + X{i}(:,k+1)'*P*X{i}(:,k+1);
end
J = [J_{:}];

J_mean = mean(J)
J_max = max(J)
```

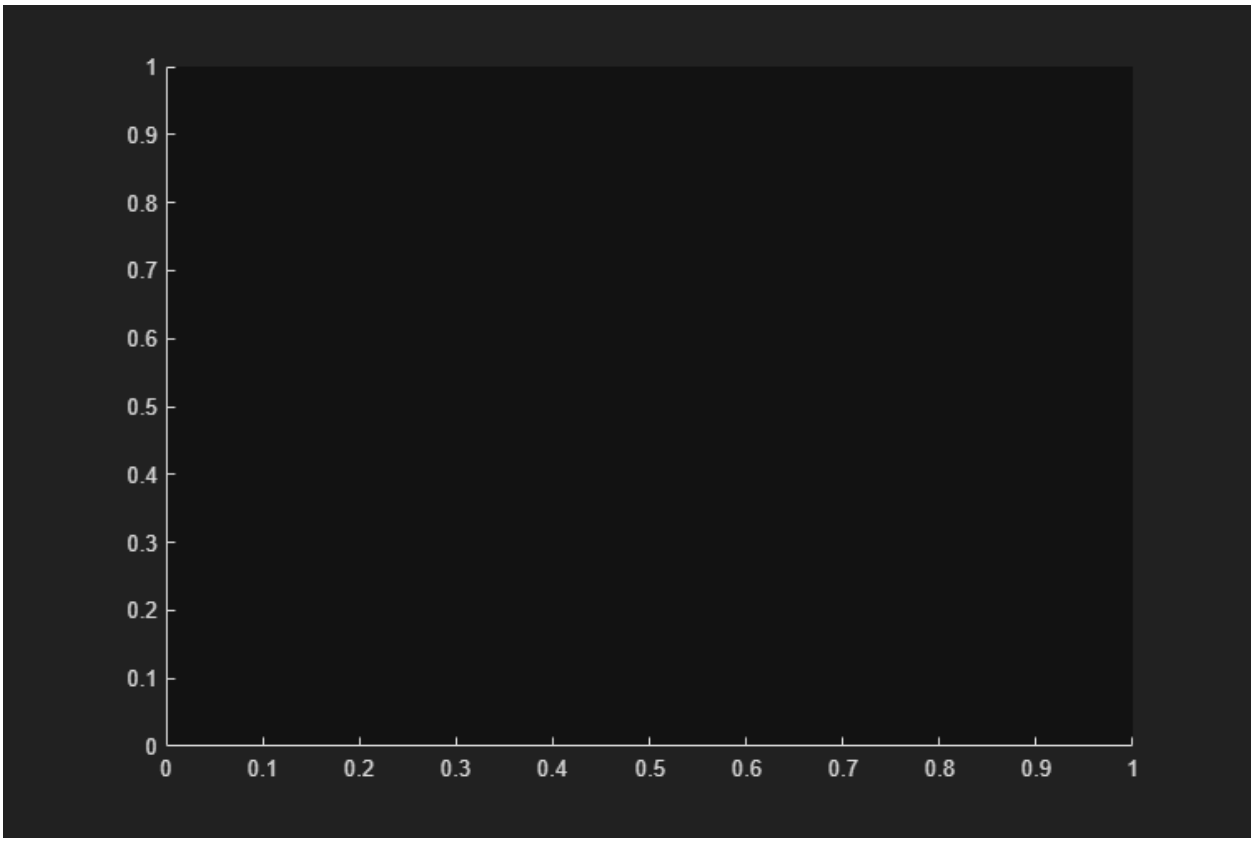
```
J_mean =

   NaN


J_max =

   NaN
```

```
J_mean =

  238.7968


J_max =

  357.6807
```

```
end
```

```
sys =

  A =
       x1  x2
   x1   1   1
   x2   0   1

  B =
        u1
   x1  0.5
   x2    1
```

```
  C =
       x1  x2
   y1   1   0
   y2   0   1
   y3   0   0

  D =
       u1
   y1   0
   y2   0
   y3   1

Sample time: 1 seconds
Discrete-time state-space model.

K =

   -1.0000   -1.5000
```




# Local functions

```
function [X,U,diagnostics_] = run_sim(A,B,V,controller,x0, tf)

    X_{tf+1} = []; U_{tf} = []; diagnostics_{tf} = [];
```

```
    X_{1} = x0;
    for k = 1:tf
        [U_{k},diagnostics_{k}] = controller{X_{k}};
        X_{k+1} = A*X_{k} + B*U_{k} + B*V{k};
    end
    X = [X_{:}]; U = [U_{:}];
end
```

*Published with MATLAB® R2023b*