MECH 6V29 - MPC - Homework 3

Table of Contents

Problem 2

2a

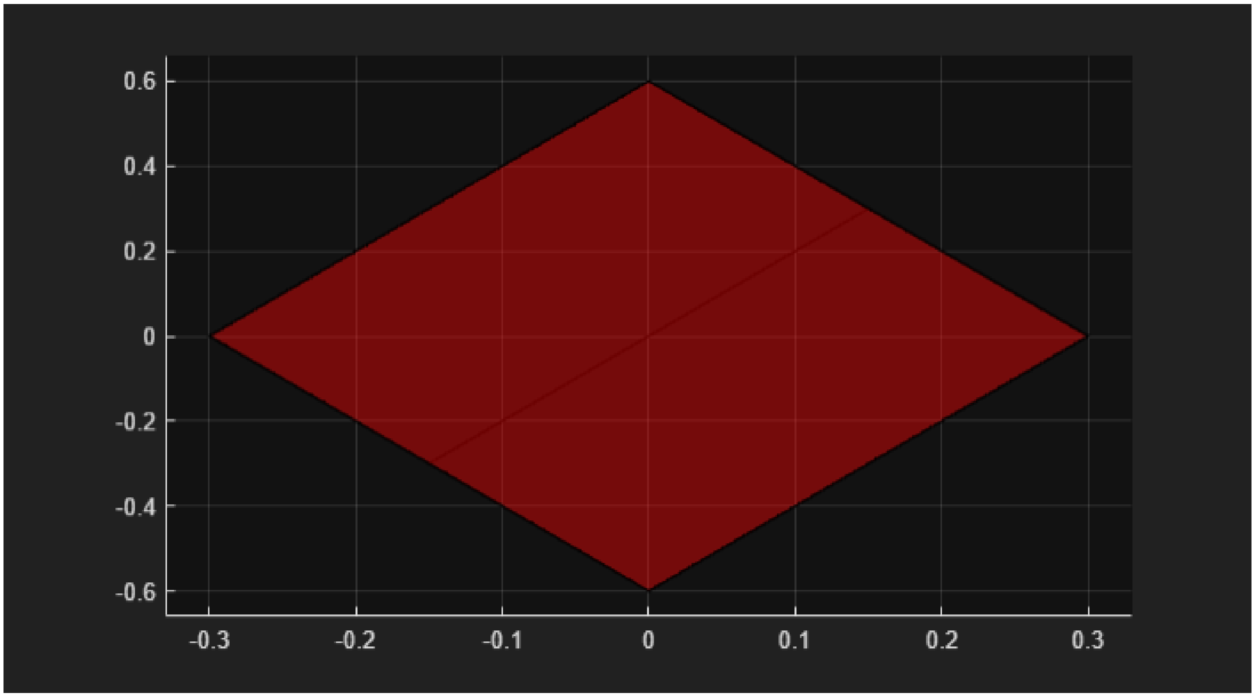
```
A = [1, 1;
    0, 1];
B = [0.5;
    1];
C = eye(3,2);%<--- [1,0;0,1;0,0]
D(3,1) = 1; %<--- [0;0;1]
sys = ss(A,B,C,D,1)

N = 10;

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

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

% sets
X = Polyhedron('A',[eye(nx);-eye(nx)], 'b', ones(2*nx,1));
U = Polyhedron('A', [1;-1], 'b', ones(2*nu,1));
W = B*Polyhedron('A',[1;-1],'b',[0.3;0.3]);
```

```
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.
```

2b - RPI Set

```
A_K = A+B*K;
F = W;
F.minHRep;
fig = figure; hold on;
F.plot
% drawnow
for i = 1:5
    F = F + (A_K)^(i)*W;
    F.minHRep;
    F.plot; alpha(0.1);
    % drawnow
end
hold off
saveas(fig,strcat('figs',filesep,'pblm2b_1','.png'));

% Approx code
epsilon = 1; %<== all epsilon smaller then 1 appear to make it good
F_approx = Approx_RPI(A_K,W,epsilon);

% Plot
fig = figure; hold on
F_approx.plot('color', 'blue'); alpha(1)

F.plot; alpha(0.2);
for i = 1:size(F.V,1)
    plot(F.V(i,1),F.V(i,2),'o');
end
saveas(fig,strcat('figs',filesep,'pblm2b_2','.png'))
```

```
s =
     1


s =

     2
```



2c - tightened state/input sets

```
Z = F_approx;

X_bar = X - Z; X_bar.minHRep;
U_bar = U - K*Z; U_bar.minHRep;
```

2d ----- Setup Controller

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

yalmip('clear'); clear('controller');
u_bar_ = sdpvar(repmat(nu,1,N),ones(1,N));
x_bar_ = sdpvar(repmat(nx,1,N+1),ones(1,N+1));
x_1 = sdpvar(nx,1);
u_1 = sdpvar(nu,1);

constraints = []; objective = 0;
constraints = [constraints,Z.A*(x_bar_{1}-x_1) <= Z.b];
```

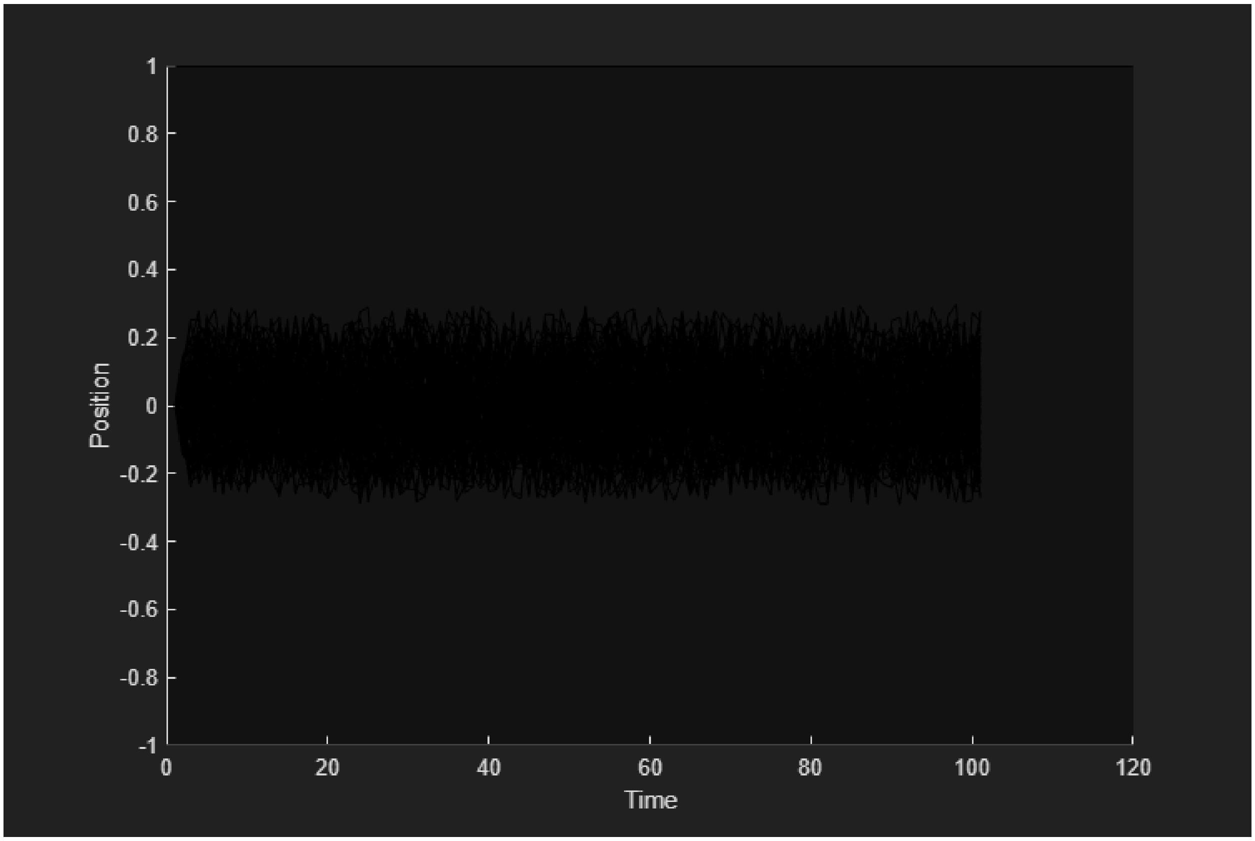
```
% constraints = [constraints, Z.A*x_bar_{1} <= Z.b]; %<-- initial condition
constraint
for k = 1:N
    objective = objective + x_bar_{k}'*Q*x_bar_{k} + u_bar_{k}'*R*u_bar_{k};
    constraints = [constraints, x_bar_{k+1} == A*x_bar_{k} + B*u_bar_{k}];
    constraints = [constraints, X_bar.A*x_bar_{k} <= X_bar.b];
    constraints = [constraints, U_bar.A*u_bar_{k} <= U_bar.b];
end
constraints = [constraints, Z.A*(x_bar_{k+1}+0)<= Z.b];
objective = objective + x_bar_{k+1}'*P*x_bar_{k+1};

constraints = [constraints, u_1 == u_bar_{1} + K*(x_1 - x_bar_{1})];

opts = sdpsettings;
controller = optimizer(constraints,objective,opts,x_1,u_1);
```

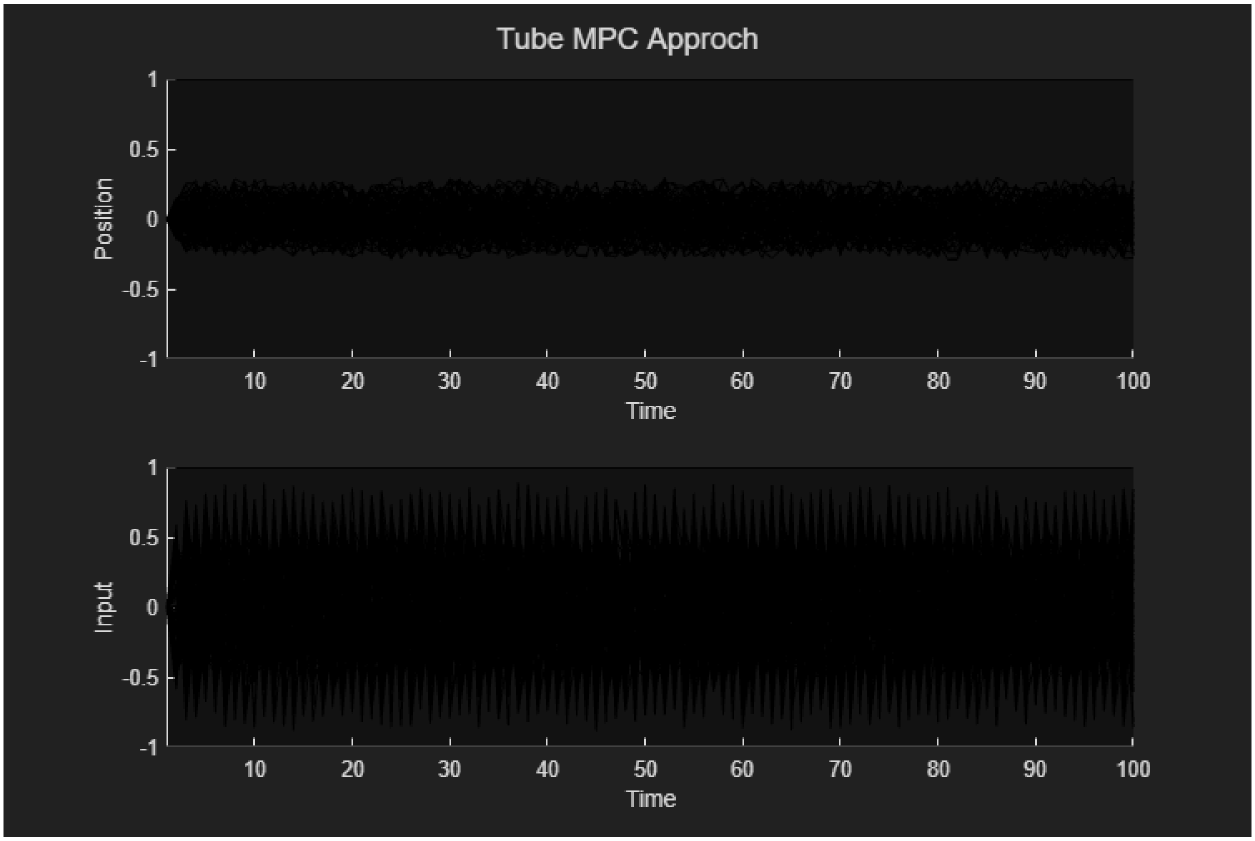
2e ----- Simulation

```
clear X U
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);
end
```

Ploting

```
fig = figure(...
        WindowStyle="normal",...
        Position=[0 0 750 500]);
% States
subplot(2,1,1); hold on;
yline(1,'k'); yline(-1,'k');
ylabel('Position');
xlabel('Time');
xlim([1,100]);
for i = 1:length(X); plot(X{i}(1,:),'k'); end
% Input
subplot(2,1,2); hold on;
yline(1,'k'); yline(-1,'k');
ylabel('Input');
xlabel('Time');
xlim([1,100]);
for i = 1:length(U); plot(U{i}(1,:),'k'); end

% save fig
sgtitle('Tube MPC Approch')
saveas(fig,strcat('figs',filesep,'pblm2e_results','.png'));
```


Position
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
Time


Tube MPC Approch
Position
1
0.5
0
-0.5
-1
10
20
30
40
50
60
70
80
90
100
Time
Input
1
0.5
0
-0.5
-1
10
20
30
40
50
60
70
80
90
100
Time

2f ------- 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 =

   1.4650e+03


J_max =

   1.9730e+03
```

Local functions

function controller = mpc_yalmip_controller(A,B,P,Q,R,N,cons,cons_f) yalmip('clear') nx = size(A,1); nu = size(B,2);

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

   constraints = [];
   objective = 0;
   for k = 1:N
       objective = objective + x_{k}'*Q*x_{k} + u_{k}'*R*u_{k} + s_{k};
       constraints = [constraints, s_{k} >= 0];
       constraints = [constraints, x_{k+1} == A*x_{k} + B*u_{k}];
       constraints = [constraints, cons(x_{k+1},u_{k},s_{k})];
   end
   constraints = [constraints,cons_f(x_{k+1},u_{k},s_{k})];
   objective = objective + x_{k+1}'*P*x_{k+1};

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

```
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

% function fig = plot_trajectory(X, U)
%     fig = figure(...
%         WindowStyle="normal",...
%         Position=[0 0 750 500]);
%     hold on; grid on;
%     subplot(2,1,1);
%     stairs(X')
%     title('State Trajectory')
%     legend({'x_1','x_2'})
%     subplot(2,1,2);
%     stairs(U');
%     title('Input Trajectory')
%     legend({'u_1'})
% end
```

Published with MATLAB® R2023b