Table of Contents

MPC HW 2 - Problem 1

Author: Jonas Wagner Date: 2023-09-29

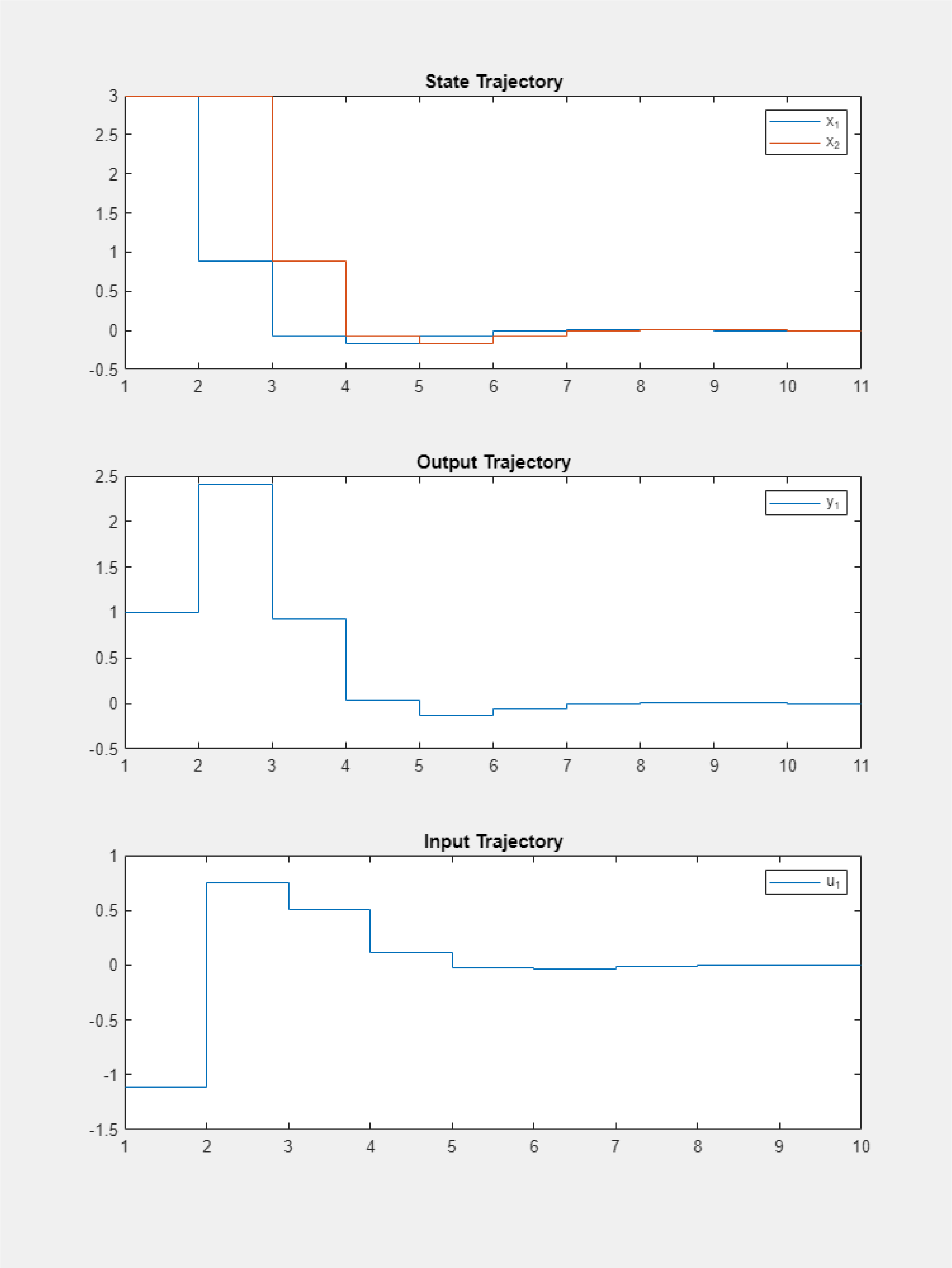
```
clear
close all
subfolder = fileparts(mfilename('fullpath'));
if ~isfolder(strcat(subfolder,filesep,'figs'))
    mkdir(strcat(subfolder,filesep,'figs'));
end

% Problem Information
A = [4/3, -2/3; 1, 0];
B = [1; 0];
C = [-2/3, 1];
D = 0;
dt = 1;
sys = ss(A,B,C,D,dt);

% Size parameters
nx = size(A,1);
nu = size(B,2);

% MPC Parameters
N = 5;
Q = eye(nx);
R = eye(nu);
P = 0;
```

Part 1a

Controller setup

```
cons = @(x,u,s) [];
cons_f = @(x,u,s) [];
controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons, cons_f);

% Simulation
x0 = [3;3]; tf = 10;
[X, U, ~] = run_sim(A,B,controller, x0, tf);
Y = C*X;
```

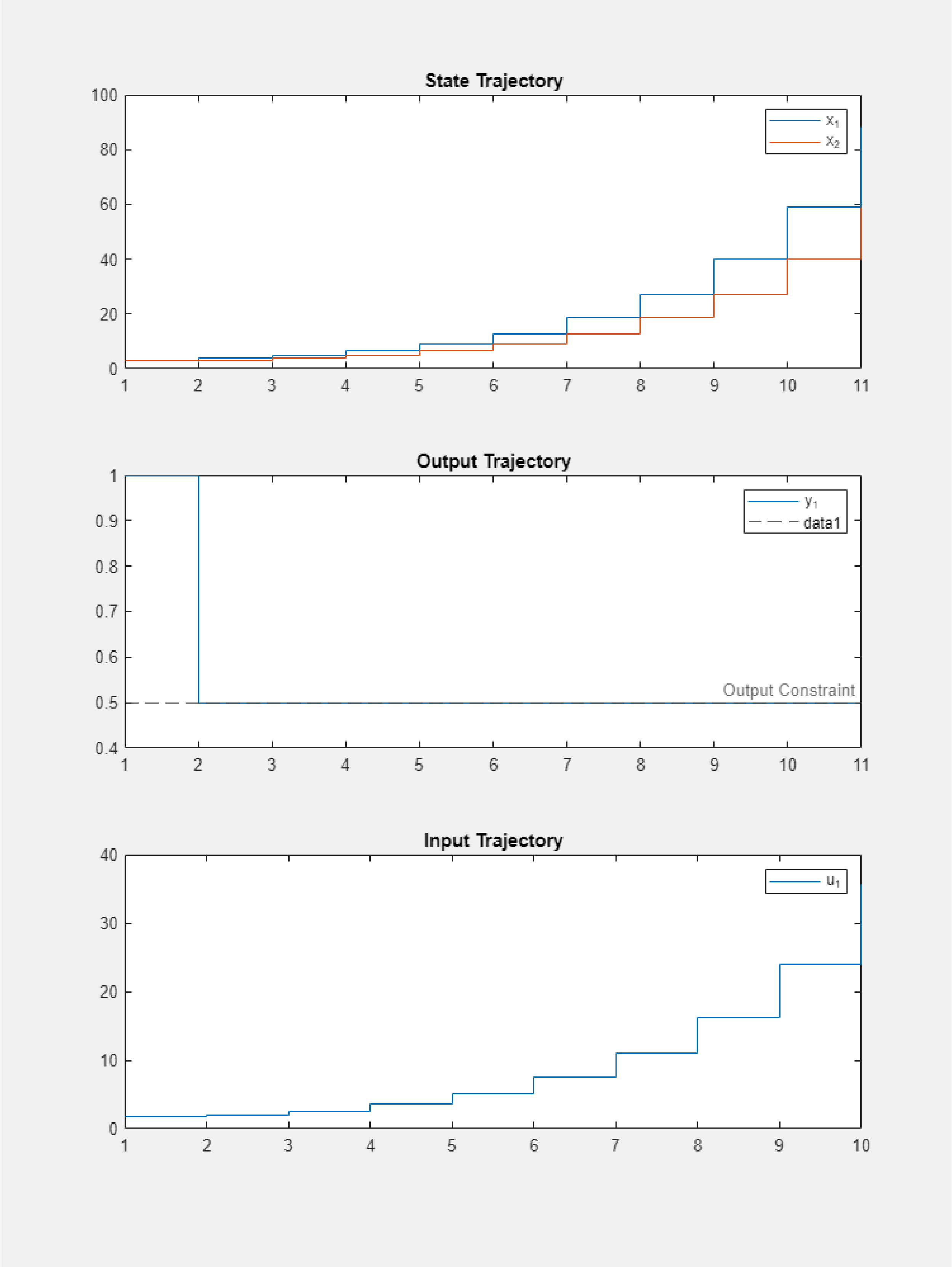
```
% Results
figName = 'pblm1a';
fig = plot_trajectory(X, Y, U);
saveas(fig,[subfolder,filesep,'figs',filesep,figName],'png')

maxOutput = max(Y);
fprintf('Max Output: %f\n',maxOutput);
```


State Trajectory
3
2.5
2
1.5
1
0.5
0
-0.5
1
2
3
4
5
6
7
8
9
10
11
x_1
x_2
Output Trajectory
2.5
2
1.5
1
0.5
0
-0.5
1
2
3
4
5
6
7
8
9
10
11
y_1
Input Trajectory
1
0.5
0
-0.5
-1
-1.5
1
2
3
4
5
6
7
8
9
10
u_1

Part 1b

Controller setup

```
cons = @(x,u,s) (C*x)<=0.5;
cons_f = @(x,u,s) [];
controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons, cons_f);

% Simulation
x0 = [3;3]; tf = 10;
[X, U, ~] = run_sim(A,B,controller, x0, tf);
Y = C*X;

% Results
figName = 'pblm1b';
fig = plot_trajectory(X, Y, U);
subplot(3,1,2); yline(0.5,'--','Output Constraint')
saveas(fig,[subfolder,filesep,'figs',filesep,figName],'png')

% % Changing Q/R
% Q = C'*C;
% R = 0;
% controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons);
% [X, U] = run_sim(A,B,controller, x0, tf);
%
% figName = 'pblm1b_2';
% fig = plot_trajectory(X, Y, U);
% subplot(3,1,2); yline(0.5,'--','Output Constraint')
% saveas(fig,[subfolder,filesep,'figs',filesep,figName],'png')
```


State Trajectory
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
11
x_1
x_2
Output Trajectory
1
0.9
0.8
0.7
0.6
0.5
0.4
1
2
3
4
5
6
7
8
9
10
11
y_1
data1
Output Constraint
Input Trajectory
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
u_1

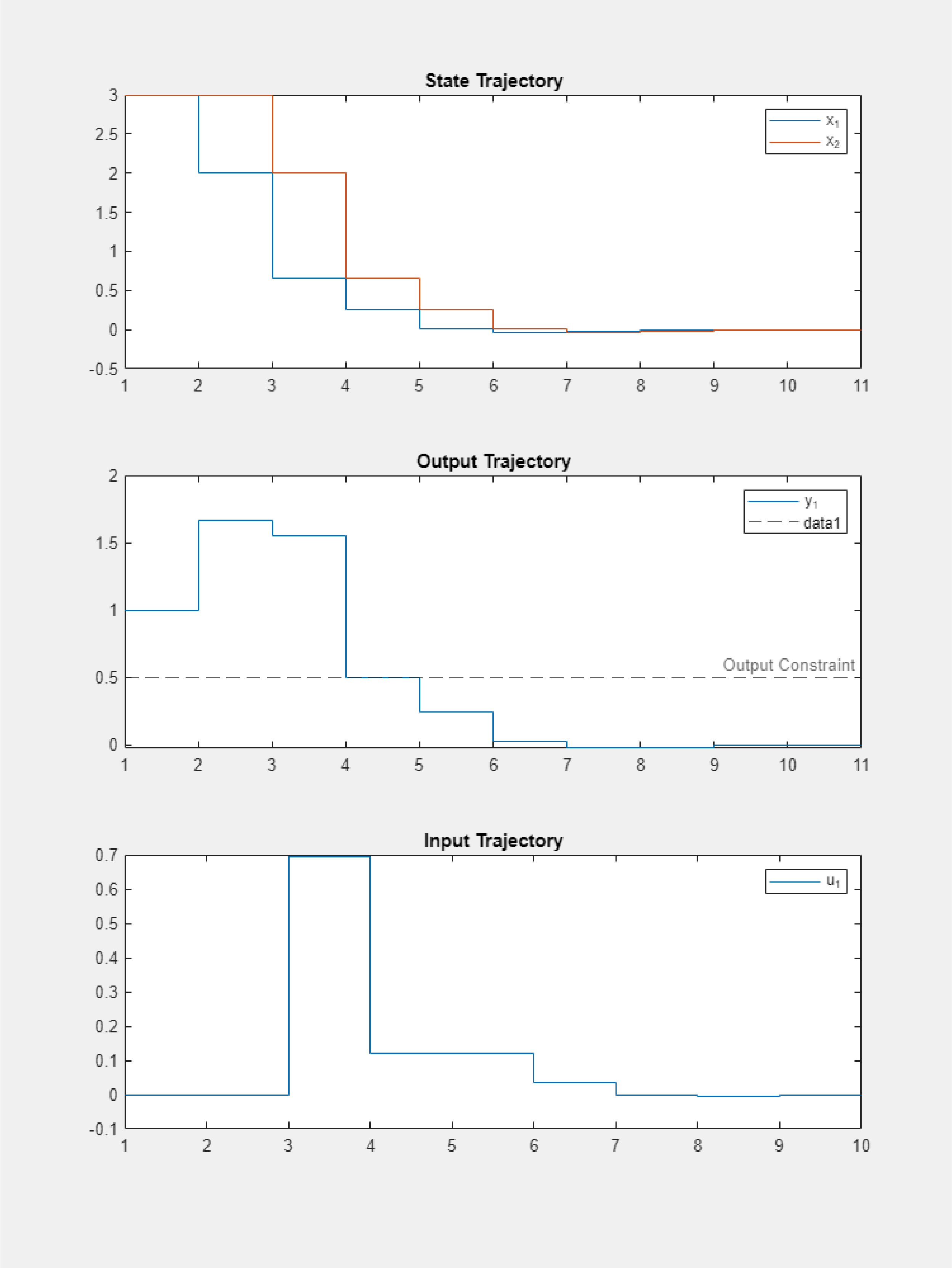
Part 1c

Controller setup

```
cons = @(x,u,s) (C*x)<=0.5;
cons_f = @(x,u,s) x==0;
controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons, cons_f);

% Simulation
x0 = [3;3]; tf = 10;
[X,U, diagnostics_] = run_sim(A,B,controller, x0, tf);
Y = C*X;

% Results
figName = 'pblm1c';
fig = plot_trajectory(X, Y, U);
subplot(3,1,2); yline(0.5,'--','Output Constraint')
saveas(fig,[subfolder,filesep,'figs',filesep,figName],'png')

% U and Errors
for k = 1:tf
    fprintf('U_%d solution: %f\n',k,U(:,k))
    fprintf('Error (k=%d): %s\n',k,yalmiperror(diagnostics_{k}));
end

% %% 1c_2
% % Controller setup
% N = 50;
% controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons, cons_f);
% [X,U, diagnostics_] = run_sim(A,B,controller, x0, tf);
% Y = C*X;
% N = 5
%
% % Results
% figName = 'pblm1c_2';
% fig = plot_trajectory(X, Y, U);
% subplot(3,1,2); yline(0.5,'--','Output Constraint')
% saveas(fig,[subfolder,filesep,'figs',filesep,figName],'png')
%
% % U and Errors
% for k = 1:tf
%     fprintf('U_%d solution: %f\n',k,U(:,k))
%     fprintf('Error (k=%d): %s\n',k,yalmiperror(diagnostics_{k}));
% end
```

```
U_1 solution: 0.000000
Error (k=1): Either infeasible or unbounded
U_2 solution: 0.000000
Error (k=2): Either infeasible or unbounded
U_3 solution: 0.694444
Error (k=3): Successfully solved
U_4 solution: 0.119462
Error (k=4): Successfully solved
```

```
U_5 solution: 0.120159
Error (k=5): Successfully solved
U_6 solution: 0.035475
Error (k=6): Successfully solved
U_7 solution: -0.002816
Error (k=7): Successfully solved
U_8 solution: -0.006824
Error (k=8): Successfully solved
U_9 solution: -0.002636
Error (k=9): Successfully solved
U_10 solution: -0.000120
Error (k=10): Successfully solved
```

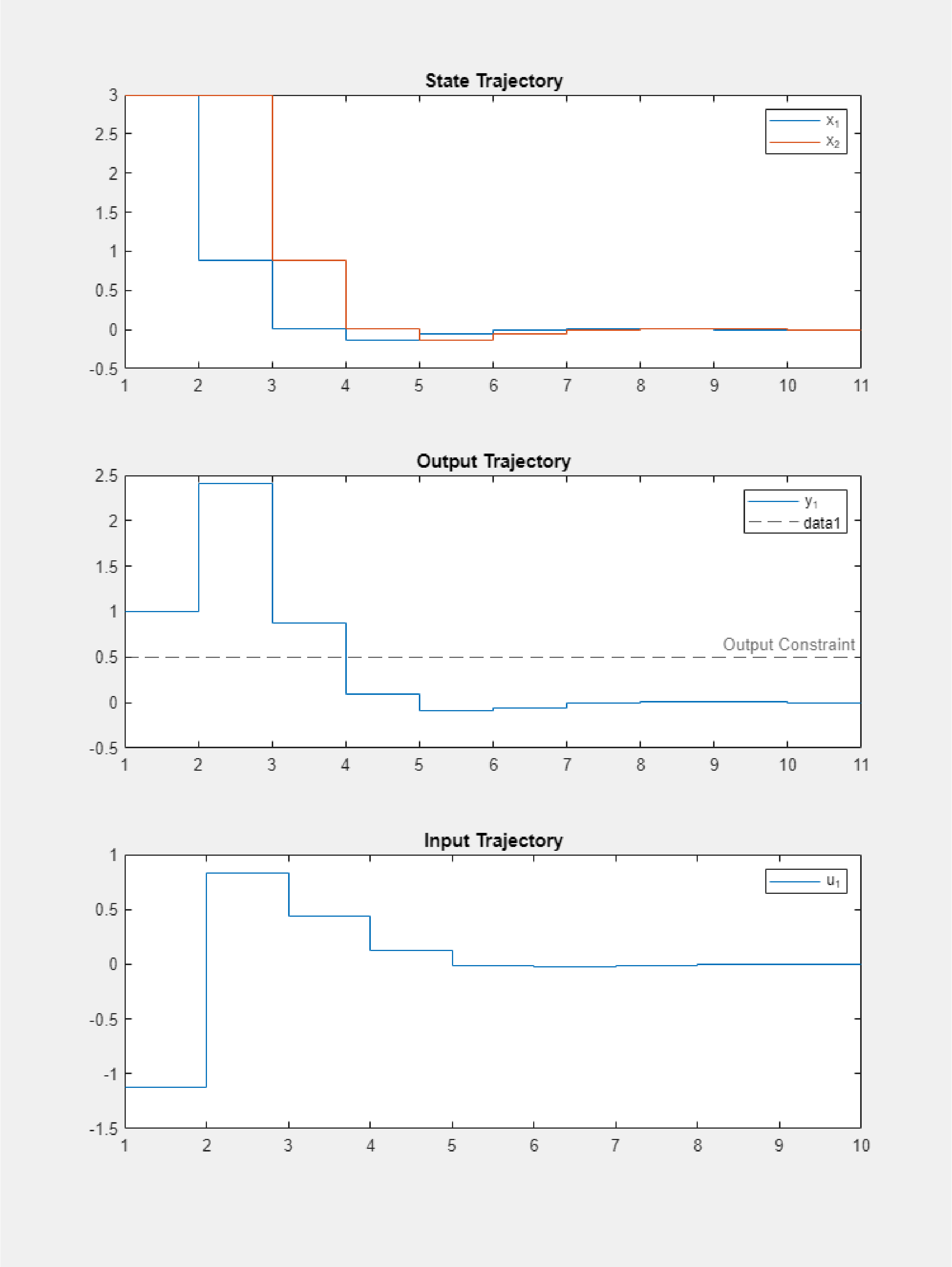

State Trajectory
x_1
x_2
3
2.5
2
1.5
1
0.5
0
-0.5
1
2
3
4
5
6
7
8
9
10
11
Output Trajectory
y_1
data1
Output Constraint
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
11
Input Trajectory
u_1
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
1
2
3
4
5
6
7
8
9
10

Part 1d

Controller setup

```
cons = @(x,u,s) (C*x)<=0.5+s;
cons_f = @(x,u,s) x==0;
controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons, cons_f);

% Simulation
x0 = [3;3]; tf = 10;
[X,U, diagnostics_] = run_sim(A,B,controller, x0, tf);
Y = C*X;

% Results
figName = 'pblm1d';
fig = plot_trajectory(X, Y, U);
subplot(3,1,2); yline(0.5,'--','Output Constraint')
saveas(fig,[subfolder,filesep,'figs',filesep,figName],'png')

% U and Errors
for k = 1:tf
    fprintf('U_%d solution: %f\n',k,U(:,k))
    fprintf('Error (k=%d): %s\n',k,yalmiperror(diagnostics_{k}));
end
```

```
U_1 solution: -1.120675
Error (k=1): Successfully solved
U_2 solution: 0.832654
Error (k=2): Successfully solved
U_3 solution: 0.444066
Error (k=3): Successfully solved
U_4 solution: 0.122051
Error (k=4): Successfully solved
U_5 solution: -0.014490
Error (k=5): Successfully solved
U_6 solution: -0.025641
Error (k=6): Successfully solved
U_7 solution: -0.009292
Error (k=7): Successfully solved
U_8 solution: -0.000176
Error (k=8): Successfully solved
U_9 solution: 0.001376
Error (k=9): Successfully solved
U_10 solution: 0.000648
Error (k=10): Successfully solved
```


State Trajectory
3
2.5
2
1.5
1
0.5
0
-0.5
1
2
3
4
5
6
7
8
9
10
11
x_1
x_2
Output Trajectory
2.5
2
1.5
1
0.5
0
-0.5
1
2
3
4
5
6
7
8
9
10
11
y_1
data1
Output Constraint
Input Trajectory
1
0.5
0
-0.5
-1
-1.5
1
2
3
4
5
6
7
8
9
10
u_1

1d - multiple

```
sigma = 100;
for i = 2:15
    N = i; cons = @(x,u,s) (C*x) <= 0.5 + (1/sigma)*s;
    controller = mpc_yalmip_controller(A, B, P, Q, R, N, cons, cons_f);
    [X,U, diagnostics_] = run_sim(A,B,controller, x0, tf); Y = C*X;
    figName = sprintf('pblm1d_N=%d',i);
    fig(i) = plot_trajectory(X, Y, U);
    subplot(3,1,2); yline(0.5,'--','Output Constraint')
    sgtitle(sprintf('N=%d',i));
    saveas(fig(i),[subfolder,filesep,'figs',filesep,figName],'png')
end
```

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}; %norm(Q*x_{k},2) + norm(R*u_{k},2);
        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,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};
    end
    X = [X_{:}]; U = [U_{:}];
end
```

```
function fig = plot_trajectory(X, Y, U)
    fig = figure(...
        WindowStyle="normal",...
        Position=[0 0 750 1000]);
    hold on; grid on;
    subplot(3,1,1);
    stairs(X')
    title('State Trajectory')
    legend({'x_1','x_2'})
    subplot(3,1,2);
    stairs(Y');
    title('Output Trajectory')
    legend({'y_1'})
    subplot(3,1,3);
    stairs(U');
    title('Input Trajectory')
    legend({'u_1'})
end
```

```
Max Output: 2.408838
```

Published with MATLAB® R2023a