Table of Contents


```
%
 ##########################################################################
% TUHH :: Institute for Control Systems :: Control Lab
%
 ##########################################################################
% Experiment CSTD2: Magnetic Levitation Plant
%
% Copyright Herbert Werner and Hamburg University of Technology, 2014
%
 ##########################################################################
% This file is to be completed by the student.
% The completed version is to be published using
%   publish('cstd2_design.m','pdf')
% and submitted as a pdf-file at least one week prior to the scheduled
 date
% for the experiment
%
% !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% !!!  The gaps in the code are denoted by TODO !!!
% !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%
% HINT 1:
% if you want to find out more about a certain command, just type
% 'help command' into the matlab window
% HINT 2:
% use section evaluations (Ctrl+Enter) to run the code within a single
% section

%----------------------------
% v.0.9 - 13-11-2014
% by Michael Heuer
%----------------------------
% Last modified on 25-11-2014
% by Julian Theis
```

```
% ---------------------------

clear all; clc; close all
```

I. Load and scale the plant

In the first step we load the plant which was identified in the previous task. After that the system matrices are extrected and the number of state, inputs and outputs are stored, cause we need them later.

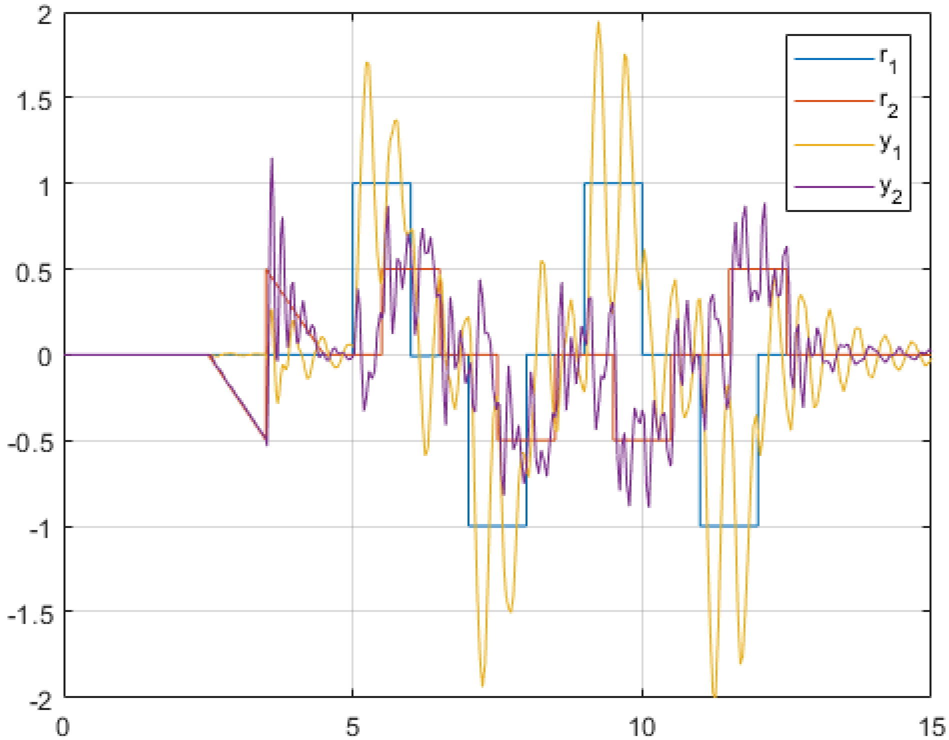
```
load models.mat

% Choose the model for the controller design
sys = sys_noise_2; % TODO

% Extract the relevant matrices
[A,B,C,D] = ssdata(sys);

% Extract the system dimensions
n  = size(A,1);
ni = size(B,2);
no = size(C,1);
```

I.b Design of a Prefilter for Reference Tracking

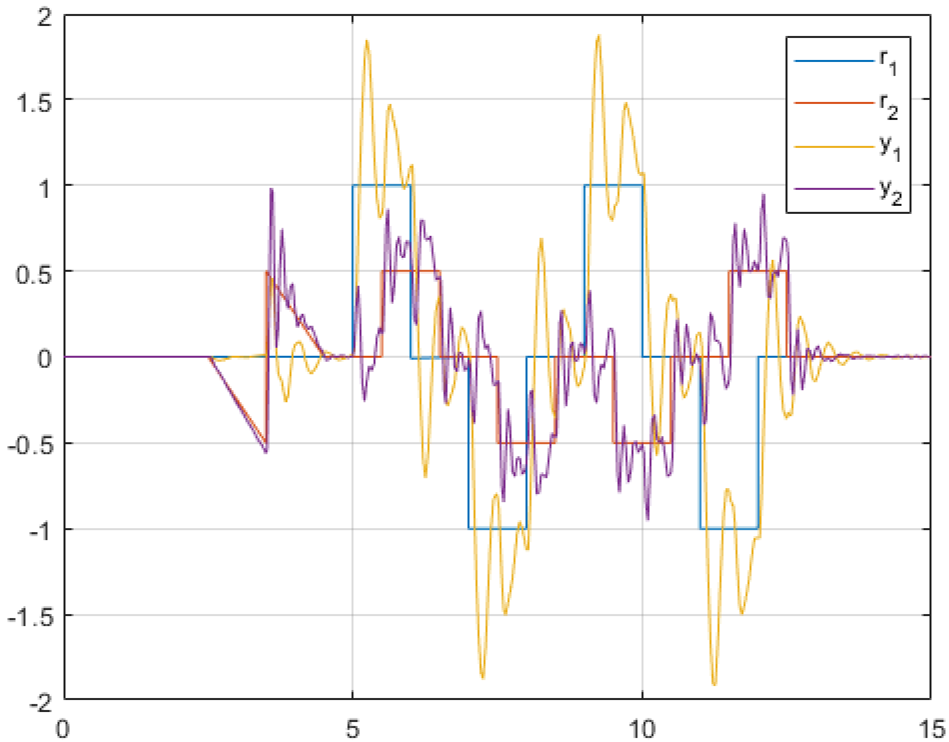
```
V = -(inv(C*inv(A)*B)); % TODO
```

I.c Simulation of the Feed Forward Design

```
sys = sys_noise_2; % TODO: Synthesis model is simulation model
sim('cstd2_sim_ff');

figure(1);
t = data(:,1);
plot(t, data(:,2), t, data(:,3), t, data(:,4), t, data(:,5));
legend({'r_1','r_2','y_1','y_2'});
grid('on');

% Simulation with an other plant

sys =sys_prbs_1; % TODO: Simulation model gains are sligtly different
sim('cstd2_sim_ff');

figure(2);
t = data(:,1);
plot(t, data(:,2), t, data(:,3), t, data(:,4), t, data(:,5));
legend({'r_1','r_2','y_1','y_2'});
grid('on');
```



2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15
r_1
r_2
y_1
y_2



2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15
r_1
r_2
y_1
y_2

II.a Design of the observer

For the linear quadratic regulator, it is important to have access to the states, which are not measured in general. For that reason we have to estimate them using an Luenberg observer.

```
Q_obsv =B*B';% TODOblkdiag(10,0.1,100,0.1);%
R_obsv =blkdiag(0.1,10); % TODO

L = -lqr(A',C',Q_obsv,R_obsv)';% TODO

% Build observer system
A_obsv = A+L*C;% TODO
B_obsv = [B -L];
C_obsv = eye(n);% TODO
D_obsv = zeros(n,ni+no);% TODO
```

II.b Analysis of the observer

```
disp('Eigenvalues of the observer are: ');
damp(A_obsv);
```

```
Eigenvalues of the observer are:


         Pole                  Damping        Frequency        Time Constant

                                            (rad/TimeUnit)      (TimeUnit)


 -7.04e+00 + 3.41e+01i        2.02e-01        3.48e+01           1.42e-01

 -7.04e+00 - 3.41e+01i        2.02e-01        3.48e+01           1.42e-01

 -3.02e+01 + 3.25e+01i        6.82e-01        4.44e+01           3.31e-02

 -3.02e+01 - 3.25e+01i        6.82e-01        4.44e+01           3.31e-02
```

III.a Design of the controller

In the next step the optimal state feeback gains are calculated.

```
Q = C'*C; % TODO
R = blkdiag(1,1);% TODO
F = -lqr(A,B,Q,R) % TODO

% Calculate Prefilter for Reference Tracking

V = -inv(C*inv(A+B*F)*B) % TODO

A_cl=[A, B*F;-L*C,A+B*F+L*C];
```

```
B_cl=[B;B];
C_cl=[C,zeros(2,4)];
D_cl=0;

sys_cl = ss(A_cl,B_cl,C_cl,D_cl)% TODO
```

```
F =

   -1.0650    1.0783   -3.1481    4.0618
   -1.6879   -0.8995    3.5984    0.6182


V =

   -1.1128   -0.2051
    0.2823    1.8357


sys_cl =

  A = 
             x1        x2        x3        x4        x5        x6        x7
   x1    -4.226      34.4     12.27    -56.76     3.061    -9.015     28.52
   x2    -12.69     8.965      37.8    -28.13    0.6537    -6.449     20.98
   x3     15.82    -23.31     4.785    -24.55     7.289   -0.6908   -0.4767
   x4      6.26      2.56     2.489    -18.08      14.7    -8.699     23.09
   x5     4.066     54.84     50.09    -21.77    -5.231    -29.46    -9.296
   x6    0.5009     26.62     21.98    -9.642    -12.53     -24.1     36.81
   x7     1.713     31.65      27.9    -12.17      21.4    -55.64    -23.59
   x8   -0.8933     -19.5    -16.93     7.393     21.85     13.36     42.51
 
             x8
   x1    -24.87
   x2     -15.4
   x3    -12.88
   x4    -42.27
   x5    -59.86
   x6    -33.89
   x7    -25.26
   x8    -67.74
 
  B = 
            u1      u2
   x1   -6.469   2.268
   x2    -4.13   2.218
   x3    -2.78  -2.564
   x4   -10.04  -2.371
   x5   -6.469   2.268
   x6    -4.13   2.218
   x7    -2.78  -2.564
   x8   -10.04  -2.371
 
  C = 
```

```
              x1           x2           x3           x4           x5           x6
   x7
  y1   -0.05749     -1.648     -1.402     0.6134            0            0
    0
  y2      3.218     -6.416   0.003077     0.2261            0            0
    0

              x8
  y1           0
  y2           0

 D =
       u1   u2
  y1    0    0
  y2    0    0

Continuous-time state-space model.
```

III.b Analysis of the observer

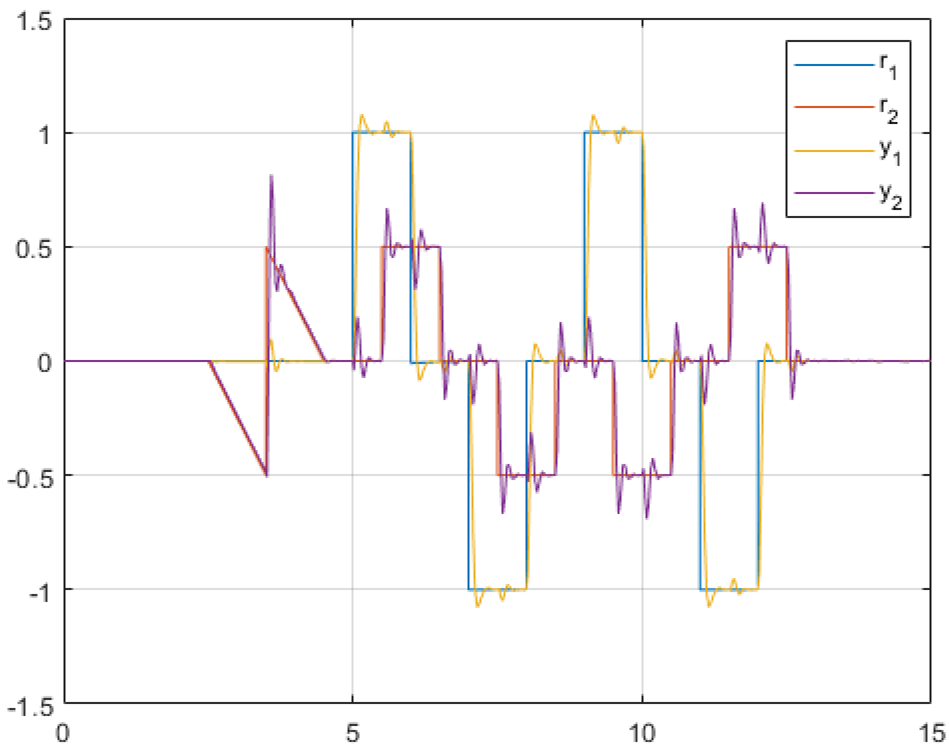
```
disp('Eigenvalues of the closed loop are: ');
damp(sys_cl);
```

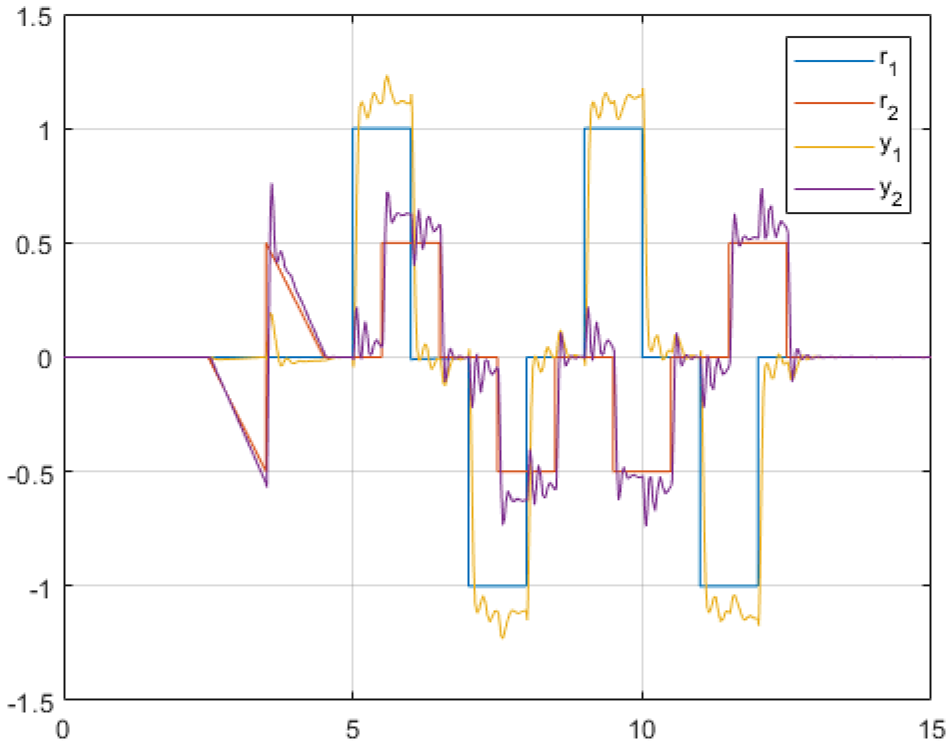
```
Eigenvalues of the closed loop are:


         Pole              Damping       Frequency      Time Constant

                                       (rad/seconds)     (seconds)



 -1.55e+01 + 2.01e+01i     6.10e-01       2.54e+01         6.46e-02

 -1.55e+01 - 2.01e+01i     6.10e-01       2.54e+01         6.46e-02

 -7.04e+00 + 3.41e+01i     2.02e-01       3.48e+01         1.42e-01

 -7.04e+00 - 3.41e+01i     2.02e-01       3.48e+01         1.42e-01

 -1.19e+01 + 3.63e+01i     3.11e-01       3.82e+01         8.42e-02

 -1.19e+01 - 3.63e+01i     3.11e-01       3.82e+01         8.42e-02

 -3.02e+01 + 3.25e+01i     6.82e-01       4.44e+01         3.31e-02

 -3.02e+01 - 3.25e+01i     6.82e-01       4.44e+01         3.31e-02
```

III.c Simulation

```
sys = sys_noise_2;% TODO: Synthesis model is simulation model
```

```
sim('cstd2_sim_lqg');

figure(3);
t = data(:,1);
plot(t, data(:,2), t, data(:,3), t, data(:,4), t, data(:,5));
legend({'r_1','r_2','y_1','y_2'});
grid('on');

% Simulation with an other plant

sys =sys_prbs_1; % TODO: Simulation model gains are sligtly different
sim('cstd2_sim_lqg');

figure(4);
t = data(:,1);
plot(t, data(:,2), t, data(:,3), t, data(:,4), t, data(:,5));
legend({'r_1','r_2','y_1','y_2'});
grid('on');
```





IV.a Design of the Controller with integral action

The problem of the previous design is the steady controll offset. To cope that it is important to add an integrator to the controller.

```
% Build augmented plant
A_aug = [A zeros(n,ni); -C zeros(ni)];% TODO
B_aug = [B;zeros(ni)];% TODO
C_aug = [C, zeros(ni)];% TODO
D_aug = D;  % TODO

% Tuning Parameter
Q_C =  C'*C; % TODO

Q_aug = [Q_C, zeros(n,ni);zeros(ni,n),[150,0;0,150]];% TODO
R_aug = blkdiag(0.5,0.1);% TODO

F_aug = -lqr(A_aug,B_aug,Q_aug,R_aug); % TODO

F =  F_aug(:,1:n); % TODO
Fi = F_aug(:,n+1:end);% TODO

A_cl_int=[A+B*F, -B*F, B*Fi; zeros(n), A+L*C, zeros(n,ni); -C,
 zeros(ni,n), zeros(2,2)];
```

```
B_cl_int= [zeros(n,ni);zeros(n,ni);eye(2)];
C_cl_int=[C, zeros(2,n), zeros(2)];
D_cl_int = D;
sys_cl_int = ss(A_cl_int, B_cl_int, C_cl_int, D_cl_int);% TODO
```

IV.b Analysis of the new Design

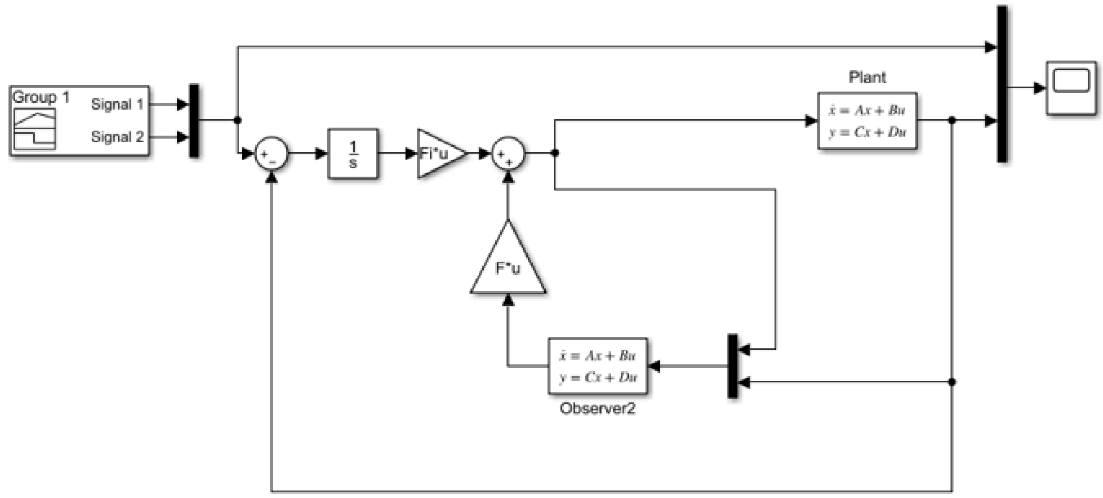
```
disp('Eigenvalues of the closed loop: ');
damp(sys_cl_int);
```

Eigenvalues of the closed loop:

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-1.03e+01	1.00e+00	1.03e+01	9.73e-02
-1.17e+01	1.00e+00	1.17e+01	8.54e-02
-1.89e+01 + 2.46e+01i	6.09e-01	3.11e+01	5.28e-02
-1.89e+01 - 2.46e+01i	6.09e-01	3.11e+01	5.28e-02
-7.04e+00 + 3.41e+01i	2.02e-01	3.48e+01	1.42e-01
-7.04e+00 - 3.41e+01i	2.02e-01	3.48e+01	1.42e-01
-3.02e+01 + 3.25e+01i	6.82e-01	4.44e+01	3.31e-02
-3.02e+01 - 3.25e+01i	6.82e-01	4.44e+01	3.31e-02
-3.00e+01 + 4.28e+01i	5.74e-01	5.22e+01	3.34e-02
-3.00e+01 - 4.28e+01i	5.74e-01	5.22e+01	3.34e-02

TODO: Complete the Simulink Model

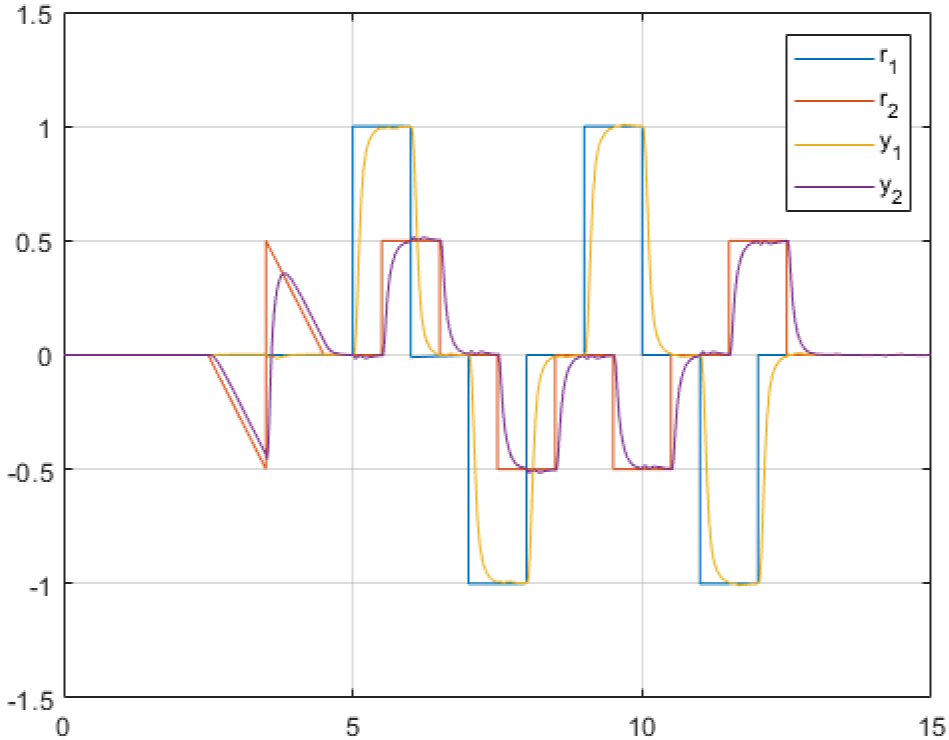
```
open('cstd2_sim_lqg_int');
```

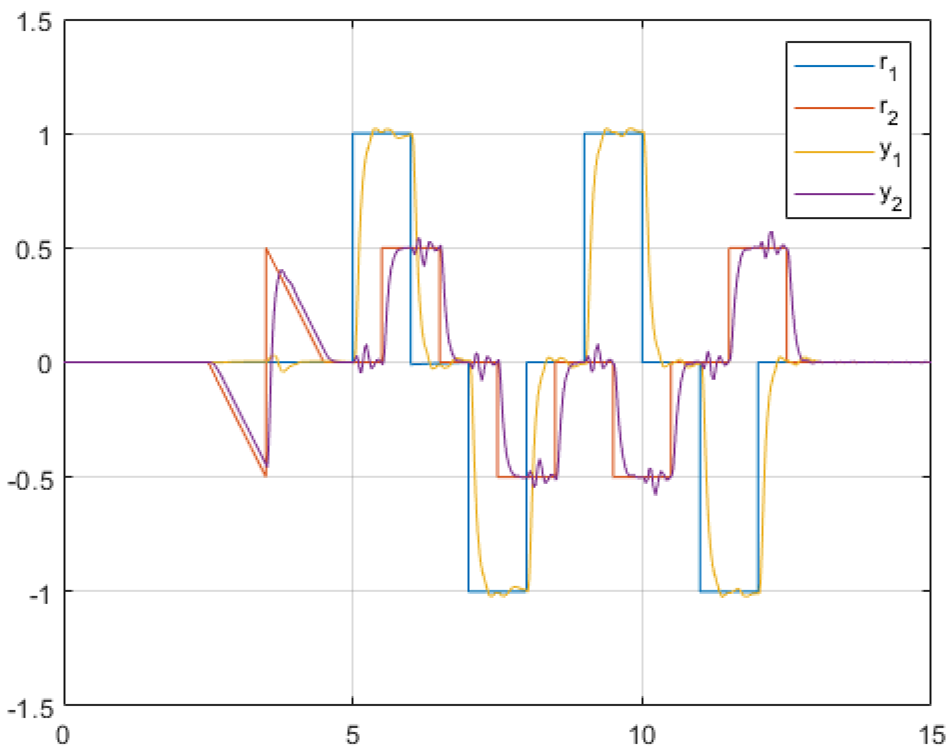


IV.c Simulation of the new design

```
sys = sys_noise_2;% TODO: Synthesis model is simulation model
sim('cstd2_sim_lqg_int');
data_s1 = data;

figure(3);
t = data(:,1);
plot(t, data(:,2), t, data(:,3), t, data(:,4), t, data(:,5));
legend({'r_1','r_2','y_1','y_2'});
grid('on');

% Simulation with an other plant
sys = sys_prbs_1; % TODO: Simulation model gains are sligtly different
sim('cstd2_sim_lqg_int');
data_s2 = data;

figure(4);
t = data(:,1);
plot(t, data(:,2), t, data(:,3), t, data(:,4), t, data(:,5));
legend({'r_1','r_2','y_1','y_2'});
grid('on');
```

```
Warning: Model '<a href="matlab:open_system
('cstd2_sim_lqg_int')">cstd2_sim_lqg_int</a>' is using a default value
 of 0.3
for maximum step size. You can disable this diagnostic by setting <a
href="matlab:configset.internal.open('cstd2_sim_lqg_int','SolverPrmCheckMsg');">Au
solver parameter selection</a> to 'none'
Warning: Model '<a href="matlab:open_system
('cstd2_sim_lqg_int')">cstd2_sim_lqg_int</a>' is using a default value
 of 0.3
for maximum step size. You can disable this diagnostic by setting <a
href="matlab:configset.internal.open('cstd2_sim_lqg_int','SolverPrmCheckMsg');">Au
solver parameter selection</a> to 'none'
```


r1
r2
y1
y2
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15


r1
r2
y1
y2
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15

Published with MATLAB® R2019b