Table of Contents


```
%
 ###########################################################################
% TUHH :: Institute for Control Systems :: Control Lab
%
 ###########################################################################
% Experiment CSTD1: Identification and Control of a Torsional 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('CSTD1_LQGdesign','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 XXX !!!
% !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%
% 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.1.0 - 30-10-2017
% by Antonio Mendez G
%----------------------------
```

```
close all
clear all
clc
```

1 LQG Design

```
%
% Design an LQG controller
%
% Load the identified system
load IdentifiedSystem

[A,B,C,D] = ssdata(sys);

%Adjust the C matrix to have only the third output as the feedback
 signal
C3 = C(3,:);
```

1.1 Observer

Determine the Q and R matrices for the state estimation

```
Q_lqe = B*B';
R_lqe = 10;

% Calculate the optimal state estimator gain L with the lqr() command
L = -lqr(A',C3',Q_lqe,R_lqe)';
```

1.2 State Feedback

Determine the Q and R matrices for the state feedback

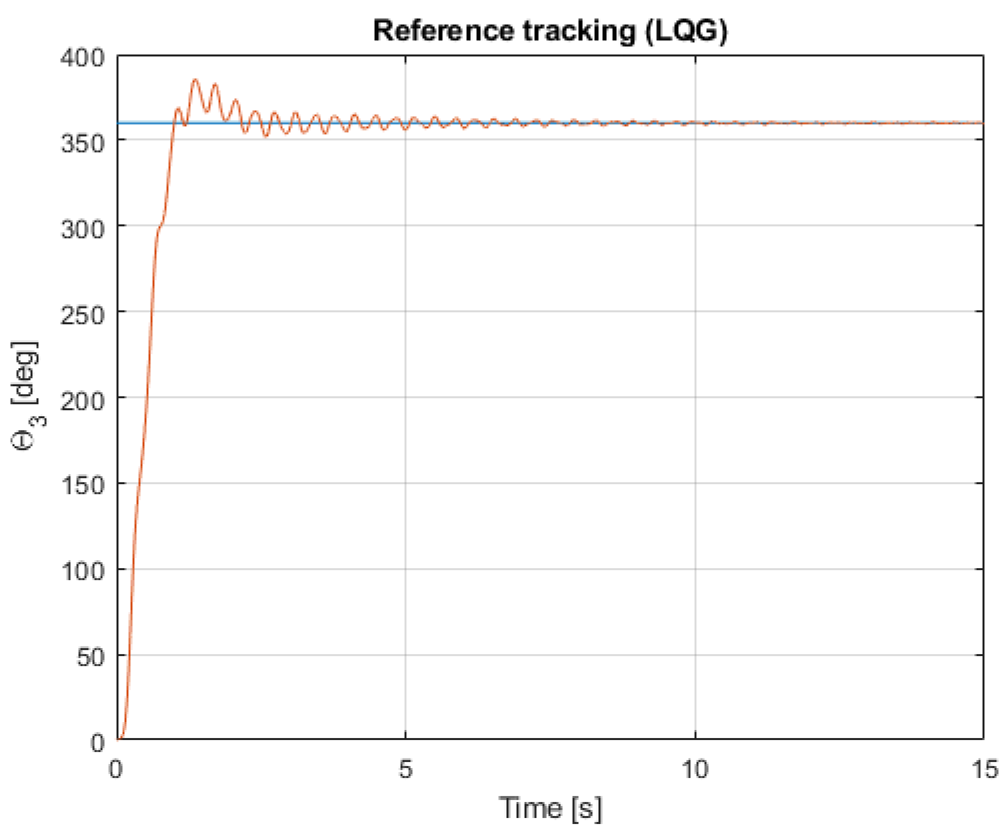
```
Q_lqr = C'*C;
R_lqr = 100;

% Calculate the optimal state feedback gain F with the lqr() command
F = -lqr(A,B,Q_lqr,R_lqr);

% Calculate the prefilter gain.
pre_v = -1/(C3*inv(A+B*F)*B);
```

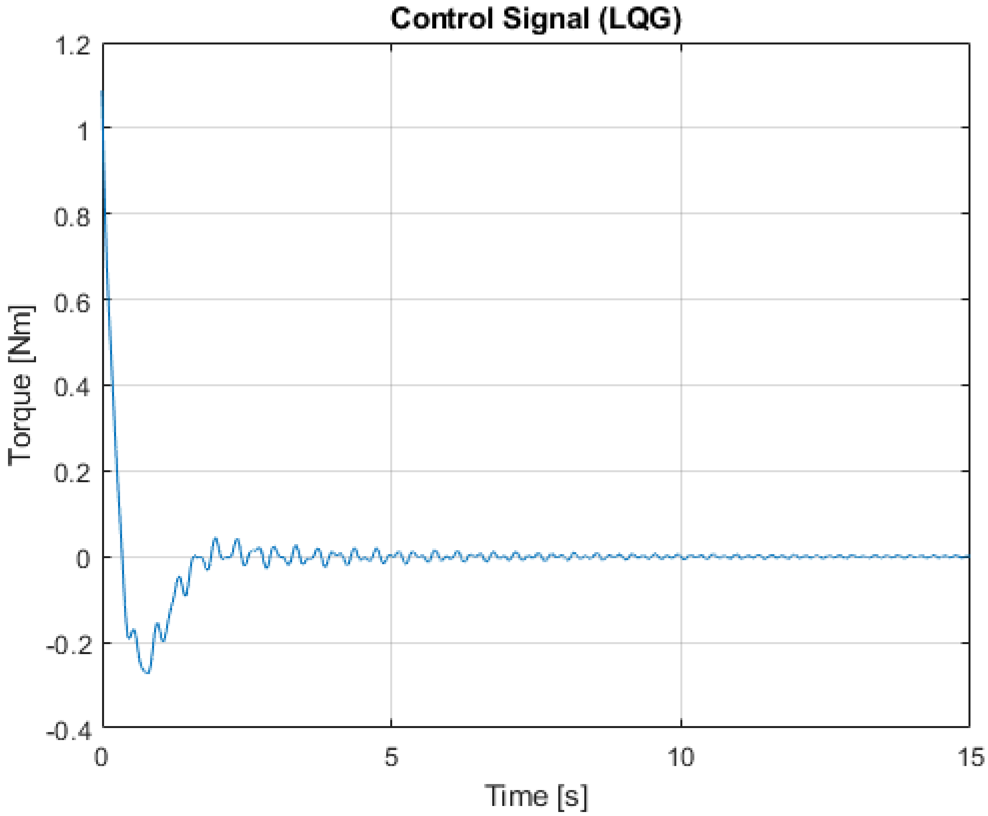
1.3 Simulation

Step response of the closed loop system

```
LQG_activation = 1; %Switch to select standard LQG control loop
dist_on = 0;        %disturbances are off


Ki = 0;             %Necessary to run the simulation
```

```
F2 = zeros(1,6);    %Necessary to run the simulation

figure
sim('CSTD1_ClosedLoopLQG');
plot(tout, sim_reft3);
hold on
plot(tout, sim_Theta3);
title('Reference tracking (LQG)');
ylabel('\Theta_3 [deg]')
xlabel('Time [s]')
grid on

% Control input
figure
plot(tout, sim_Torque);
title('Control Signal (LQG)');
ylabel('Torque [Nm]')
xlabel('Time [s]')
grid on
```

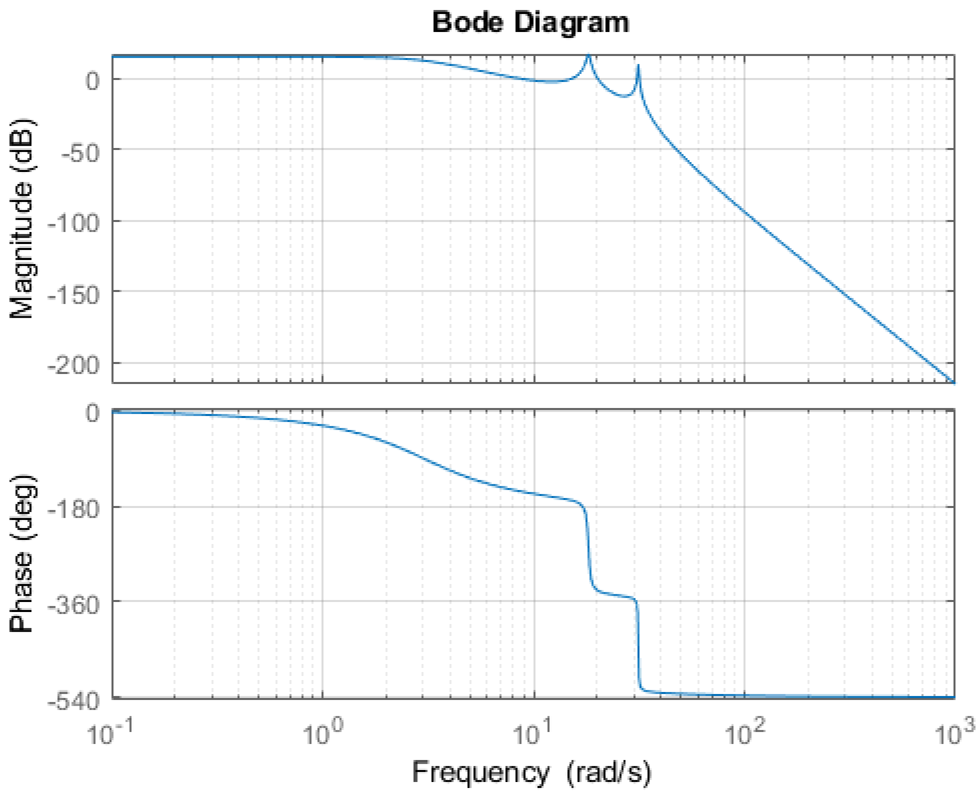




1.4 Frequency Responses

Bode diagram of the closed-loop system You can ignore the observer, since it is not controllable

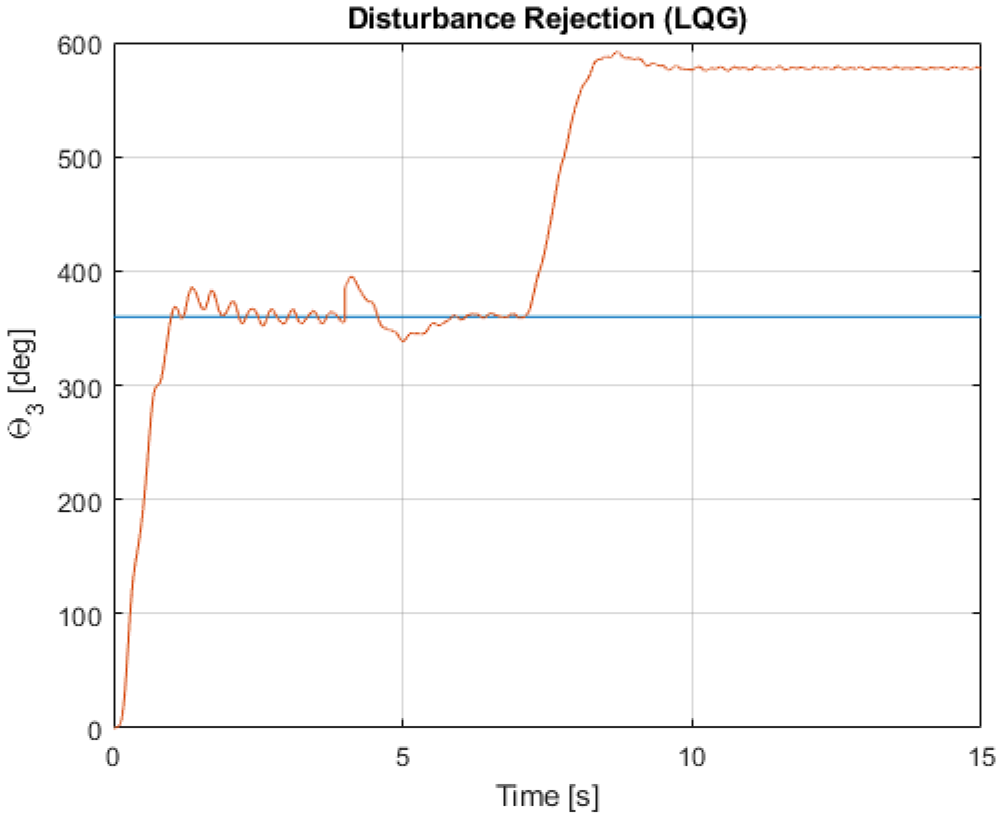
```
r2y = minreal(ss(A+B*F,B,C3,0));

bode(r2y)
grid on
```



1.5 Disturbance Rejection

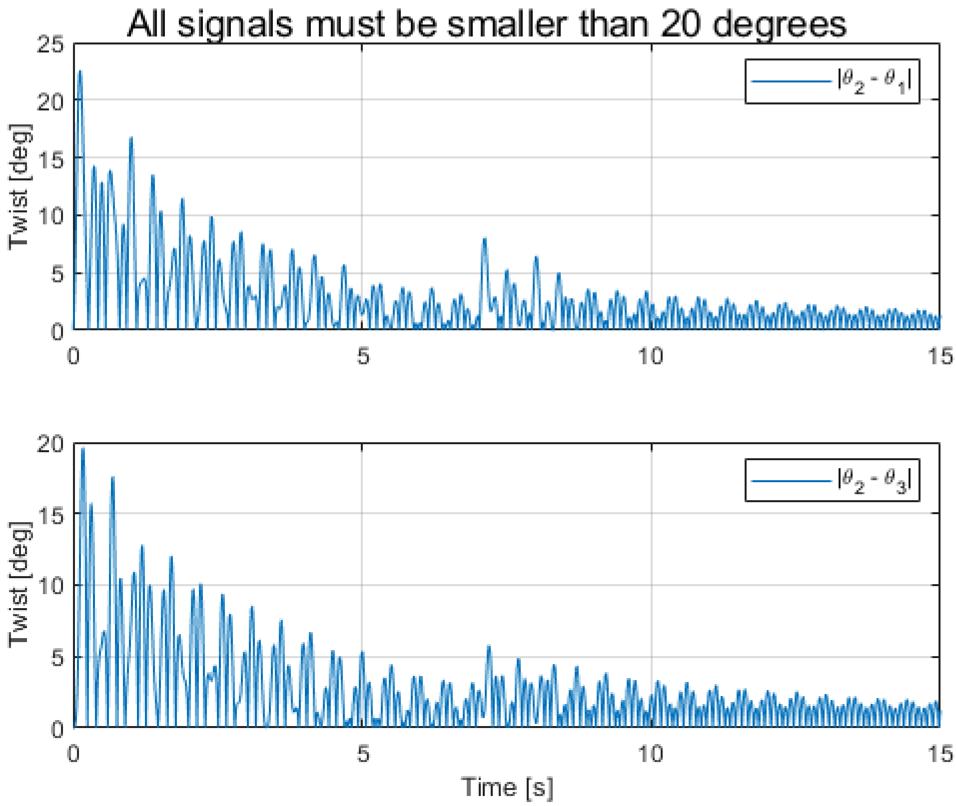
Step response of the closed loop system under disturbances

```
LQG_activation = 1; %Switch to select standard LQG control loop
dist_on = 1;        %disturbances are on

figure
sim('CSTD1_ClosedLoopLQG');
plot(tout, sim_reft3);
hold on
plot(tout, sim_Theta3);
title('Disturbance Rejection (LQG)');
ylabel('\Theta_3 [deg]')
xlabel('Time [s]')
grid on
```



1.6 Torsional Springs Safety

```
%
% The plot of the twist angle of the torsional springs
% is shown. The absolute twist value should be always
% smaller than  20 degrees.
figure()

suptitle('All signals must be smaller than 20 degrees')

subplot(2,1,1)
plot(tout, abs(sim_Theta2-sim_Theta1));
grid on
ylabel('Twist [deg]')
legend('|\theta_2 - \theta_1|')

subplot(2,1,2)
plot(tout, abs(sim_Theta2-sim_Theta3d));
grid on
xlabel('Time [s]')
legend('|\theta_2 - \theta_3|')
ylabel('Twist [deg]')
```



2 Integral Action

Design an LQG controller with integral action

```
clc
close all

% Load the identified system
load IdentifiedSystem

[A,B,C,D] = ssdata(sys);

%Adjust the C matrix to have only the third output as the feedback
 signal
C3 =C(3,:);
```

2.1 Augmented System

Compute the augmented matrices

```
Aaug = [A zeros(6,1);-C3 0];
Baug = [B;0];
```

2.2 State Feedback with Integral Action

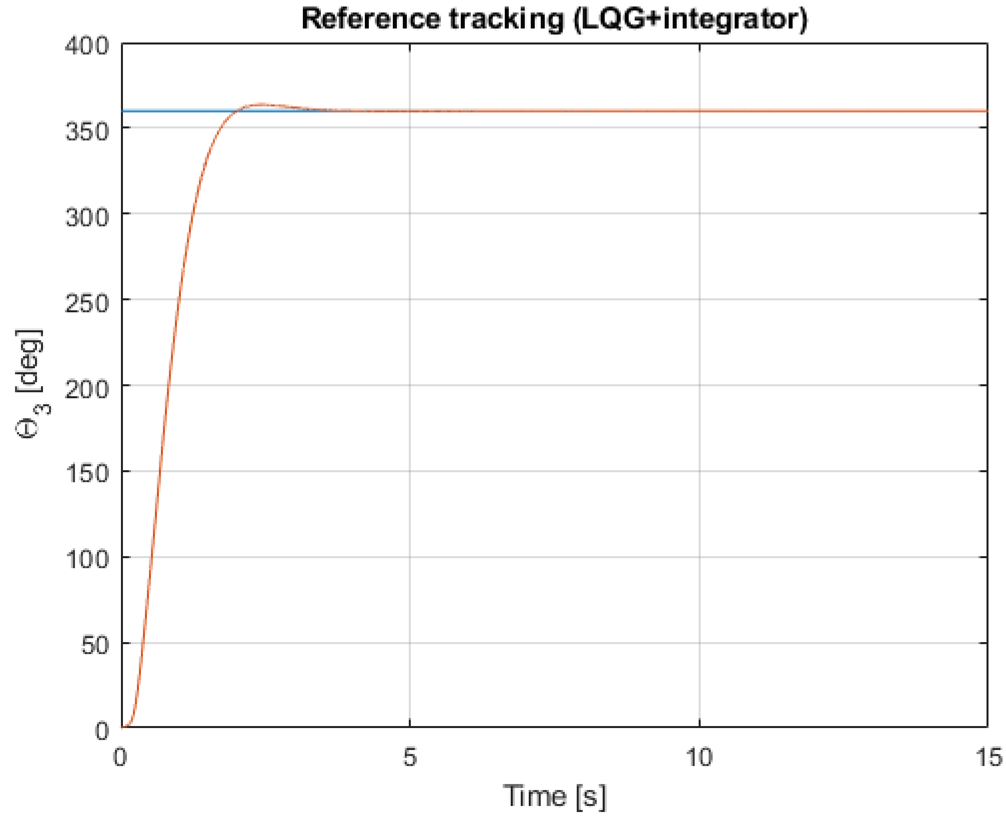
Determine the Q and R matrices for the state feedback with integrator

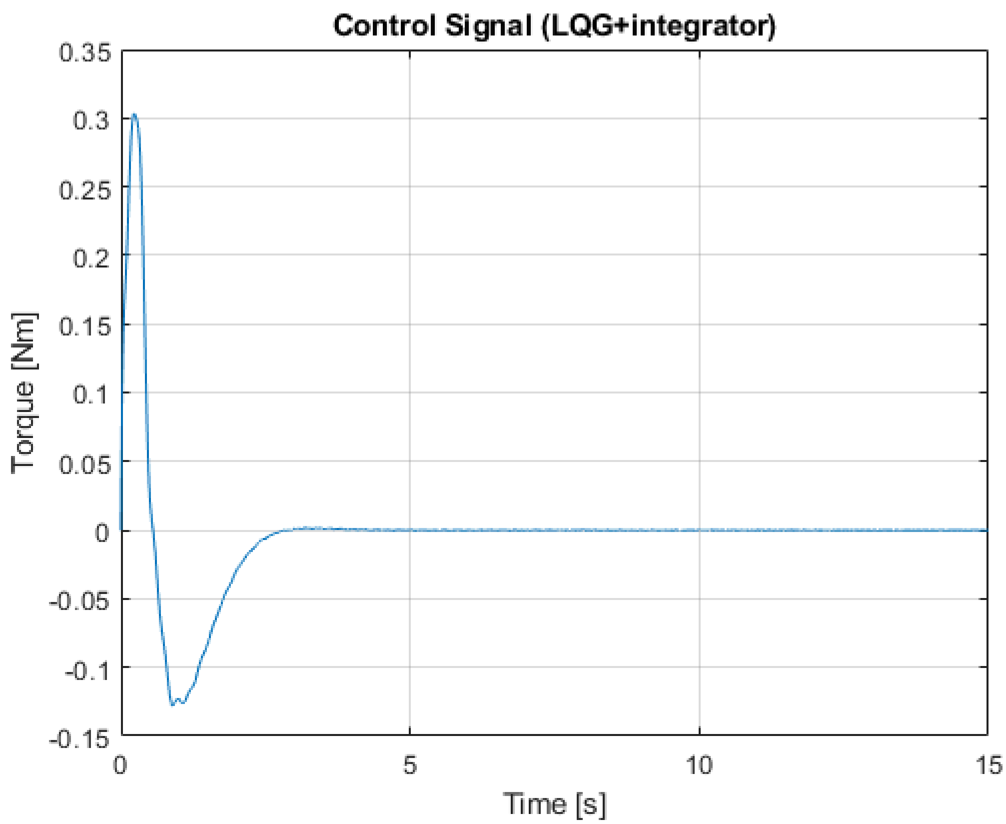
```
Qaug = blkdiag(1,0.1,0.1,0.1,0.1,0.1,10);
Raug = 10;

% Compute the state feedback gain with integrator
Fi = -lqr(Aaug,Baug,Qaug,Raug);

% Separate the gains
Ki = Fi(7); %Gain for the integrator
F2 = Fi(1:6); %Gain for the states of the plant
```

2.3 Simulation

Step response of the closed loop system

```
LQG_activation = 0; %Switch to select the LQG control loop with
 integrator
dist_on = 0;        %disturbances are off

pre_v = 0;          %Necessary to run the simulation
F = zeros(1,6);     %Necessary to run the simulation

figure
sim('CSTD1_ClosedLoopLQG');
plot(tout, sim_reft3);
hold on
plot(tout, sim_Theta3);
title('Reference tracking (LQG+integrator)');
ylabel('\Theta_3 [deg]')
xlabel('Time [s]')
grid on

% Control input
figure
plot(tout, sim_Torque);
title('Control Signal (LQG+integrator)');
ylabel('Torque [Nm]')
xlabel('Time [s]')
grid on
```


Reference tracking (LQG+integrator)
400
350
300
250
200
150
100
50
0
Θ3 [deg]
0
5
10
15
Time [s]

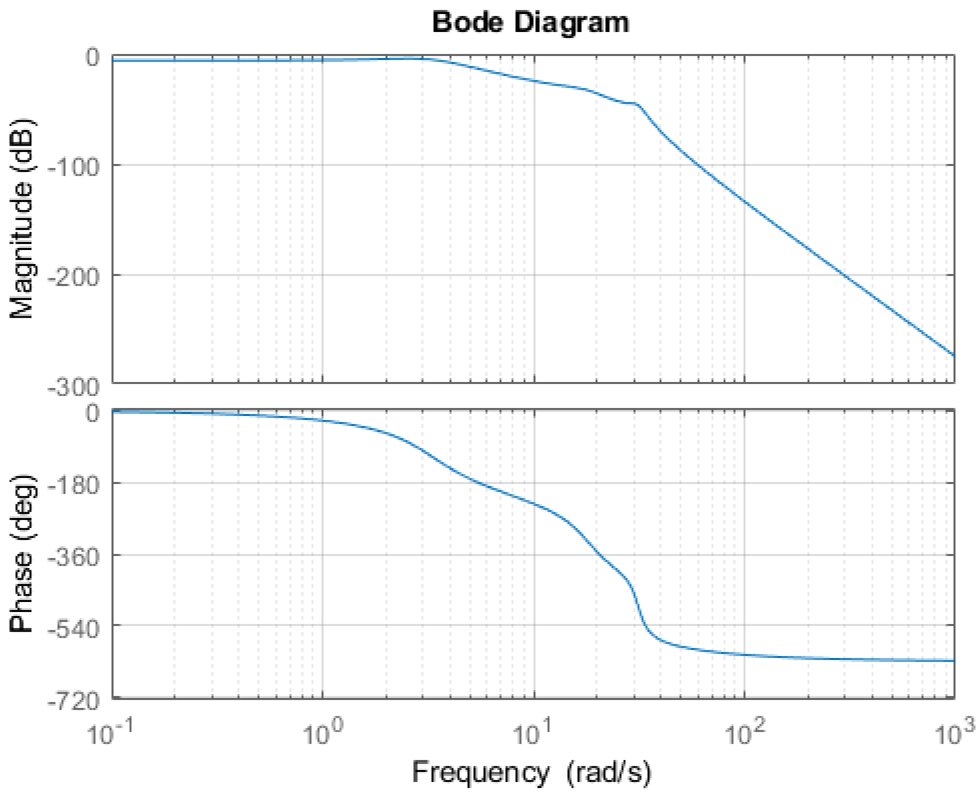

Control Signal (LQG+integrator)
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
Torque [Nm]
0
5
10
15
Time [s]

2.4 Frequency Responses

Bode diagram of the closed-loop system You can ignore the observer, since it is not controllable Consider both, the state feedback and the integrator

```
integrator = Ki;

SYS1 = ss(Aaug+Baug*Fi,[zeros(6,1);1],[C3 0],0);
SYS2 = 1;
r2y = feedback(SYS1,SYS2)

bode(r2y)
grid on
```

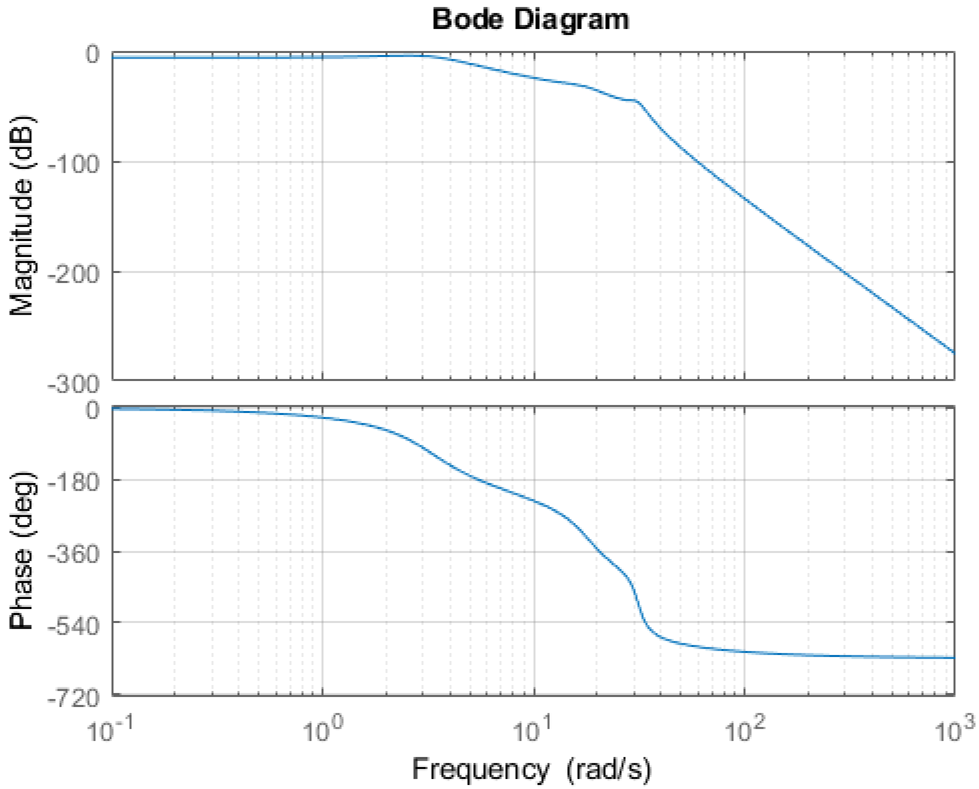
```
r2y =

  A =
             x1       x2       x3       x4       x5       x6       x7
   x1         0        1        0        0        0        0        0
   x2    -596.3   -28.06      566   -4.563   -88.61   -7.712    145.8
   x3         0        0        0        1        0        0        0
   x4     335.8        0   -653.3   0.2162    317.5        0        0
   x5         0        0        0        0        0        1        0
   x6         0        0    355.9        0   -355.9     0.16        0
   x7         0        0        0        0       -2        0        0

  B =
       u1
   x1   0
   x2   0
   x3   0
   x4   0
   x5   0
   x6   0
   x7   1

  C =
       x1  x2  x3  x4  x5  x6  x7
   y1   0   0   0   0   1   0   0

  D =
       u1
   y1   0

Continuous-time state-space model.
```

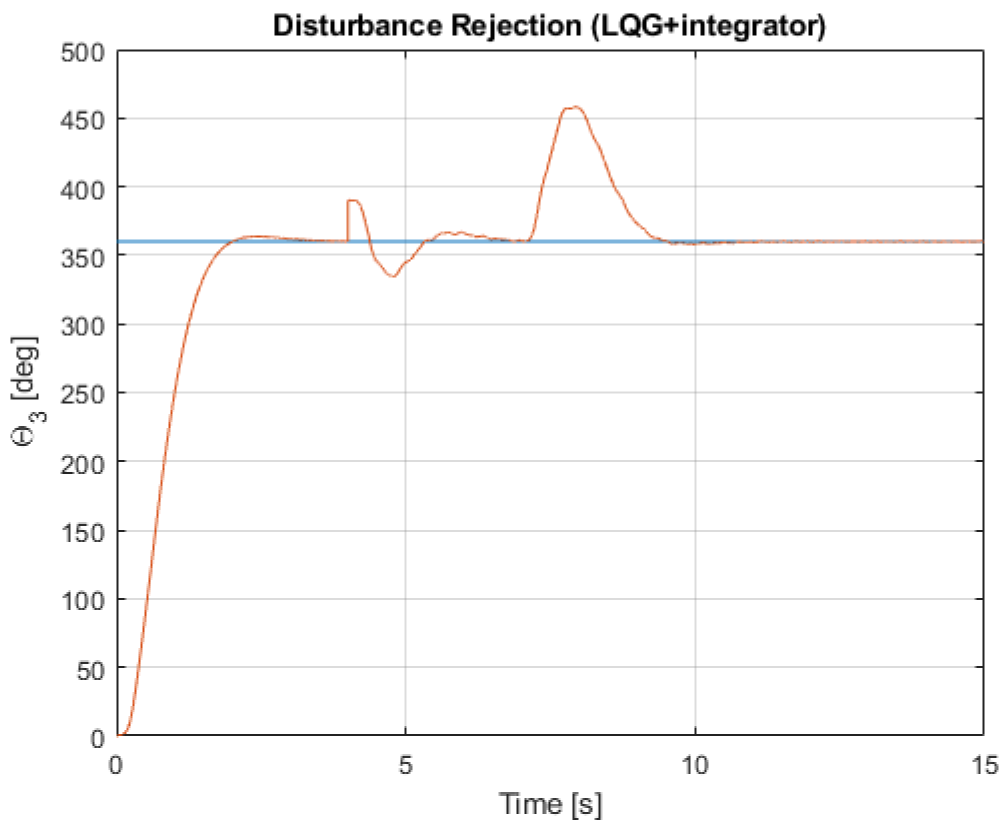


2.5 Disturbance Rejection

Step response of the closed loop system under disturbances

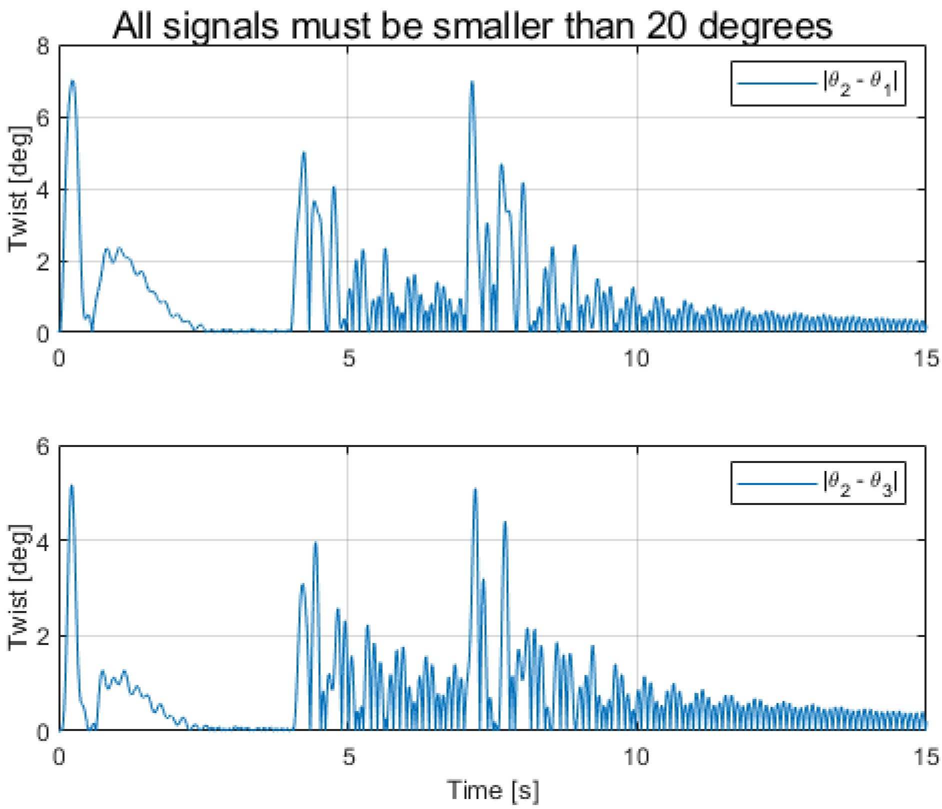
```
LQG_activation = 0; %Switch to select the LQG control loop with
 integrator
dist_on = 1;        %disturbances are on

figure()
sim('CSTD1_ClosedLoopLQG');
plot(tout, sim_reft3);
hold on
plot(tout, sim_Theta3);
title('Disturbance Rejection (LQG+integrator)');
ylabel('\Theta_3 [deg]')
xlabel('Time [s]')
grid on
```


Bode Diagram
Magnitude (dB)
0
-100
-200
-300
Phase (deg)
0
-180
-360
-540
-720
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)


Disturbance Rejection (LQG+integrator)
Θ_3 [deg]
500
450
400
350
300
250
200
150
100
50
0
0
5
10
15
Time [s]

2.6 Torsional Springs Safety

```
%
% The plot of the twist angle of the torsional springs
% is shown. The absolute twist value should be always
% smaller than  20 degrees.
figure()

suptitle('All signals must be smaller than 20 degrees')

subplot(2,1,1)
plot(tout, abs(sim_Theta2-sim_Theta1));
grid on
ylabel('Twist [deg]')
legend('|\theta_2 - \theta_1|')

subplot(2,1,2)
plot(tout, abs(sim_Theta2-sim_Theta3d));
grid on
xlabel('Time [s]')
legend('|\theta_2 - \theta_3|')
ylabel('Twist [deg]')
```



Published with MATLAB® R2019b