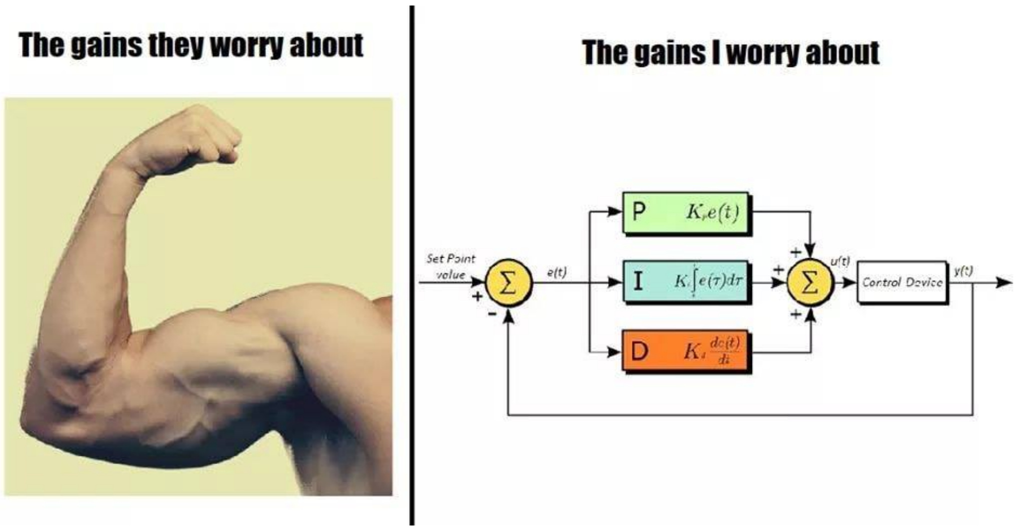
Lab 5 Report

034406: Advanced Control and Automation Lab.



Danny Apsel [redacted]

August 2024

Contents

1 Experiment Objective **1**

2 Results **2**

2.1 Controller Design . . . 2

2.2 Block diagrams . . . 4

2.3 Graphs . . . 6

2.3.1 Bode & polar plots . . . 6

2.3.2 Experiment . . . 8

2.3.3 Simulations . . . 11

3 Conclusion **14**

4 MATLAB **15**

1 Experiment Objective

The Linear model of the experimental setup from lab 4 with a servo controller in closed loop, is depicted in Figure 1

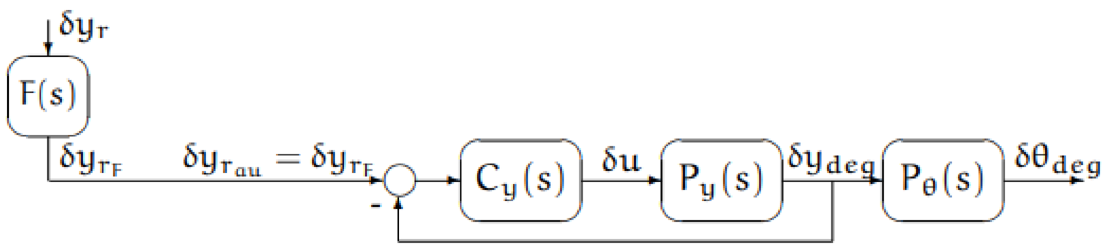
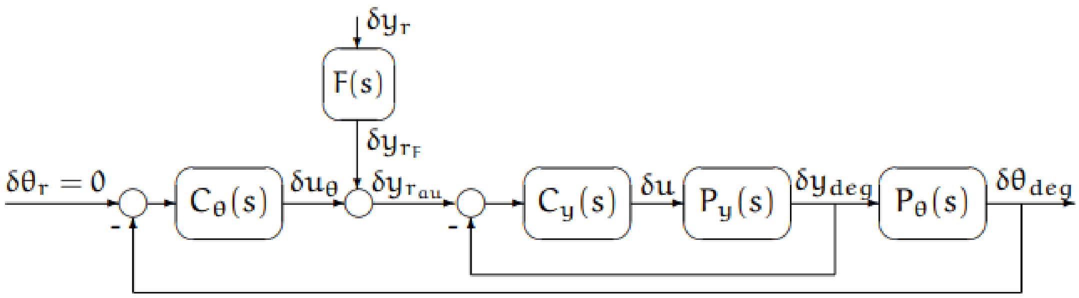
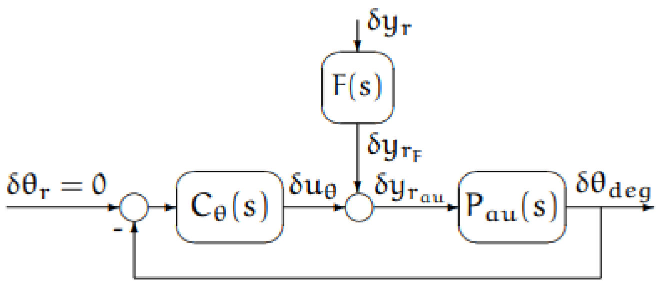


Figure 1: Servo control loop of arm angle with prefilter F(s)



(a) System with two control loops



(b) Dampening loop with P_{au}

Figure 2: Control loop for the block diagram in Figure 1

The transfer functions $P_y(s)$ and P_θ have been found in Lab 4:

$$P_\theta(s) = \frac{-0.01277s^2}{0.01487s^2 + 0.003268s + 0.5732}$$
$$P_y(s) = \frac{0.6189s^2 + 0.136s + 23.85}{9.086 \cdot 10^{-6}s^4 + 1.213 \cdot 10^{-4}s^3 + 0.0014s^2 + 0.0044s}$$

The controller $C_y(s)$ is the first servo controller from Lab 3.

$$C_y(s) = \frac{0.3951s^2 + 3.041s + 3.225}{s(10s + 250.8)}$$

P_{au} is defined as:

$$P_{au}(s) = P_\theta \frac{P_y C_y}{1 + P_y C_y}$$

Design a controller for the system in Figure 1 so that:

- The steady-state error for a step reference command in the arm angle is 0,
- The high pendulum oscillations (see the response of the pendulum angle to the step reference command in the arm angle in Lab 4) is restrained.

The value of the step command in the arm angle is given by $\pm 30°$. The feedback loop for the system described in Figure 1 is given in Figure 2(a). Pay attention to the signal $\delta y_{r_{au}}$, which is defined as the input signal to the summing point of the servo control loop: According to Figure 1 it is given by $\delta y_{r_{au}} = \delta y_{r_F}$, and according to Figure 2 this signal is given by $\delta y_{r_{au}} = \delta y_{r_F} + \delta u_\theta$.

2 Results

2.1 Controller Design

We want to design a controller for the pendulum angle $\theta(t)$ with the following requirements:

- The gain of the Closed Loop system in Figure 2(b) is no more than 3 [dB]
- The Controller gain is less than 0 [dB] for frequencies above 200 [rad/s]
- The controller slope is at least -20 [dB/s]

For the first requirement, we will design a controller of the form:

$$C_1(s) = -\frac{1}{C_{lead}}$$

$$C_{lead}(s) = \frac{\sqrt{\alpha}s + w}{s + \omega\sqrt{\alpha}}$$

Where ω is the gain crossover frequency that is the closest to the critical point which causes the peak on the Bode plot, as seen in Figure ??. We got $\omega = 4.3906 \left[\frac{rad}{s}\right]$. α was chosen after lots of tests with different numbers, we ultimately decided on $\alpha = 4.928$.
For the third requirement, we will add a 1st order Low Pass Filter (LPF), for frequencies above $\omega_{cutoff} = 200 \left[\frac{rad}{s}\right]$

$$C_{LPF}(s) = \frac{1}{\tau s + 1} = \frac{1}{\frac{1}{200}s + 1}$$

Our Pendulum angle controller $C_\theta(s)$ is:

$$C_\theta(s) = -\frac{1}{C_{lead}} \cdot C_{LPF} = -\frac{s + 4.3906\sqrt{4.928}}{\sqrt{4.928}s + 4.3906} \cdot \frac{1}{0.005s + 1} = \frac{-s - 9.7468}{0.0111s^2 + 2.2419s + 4.3906}$$

We will now verify the requirements using Bode & Nichols plots.

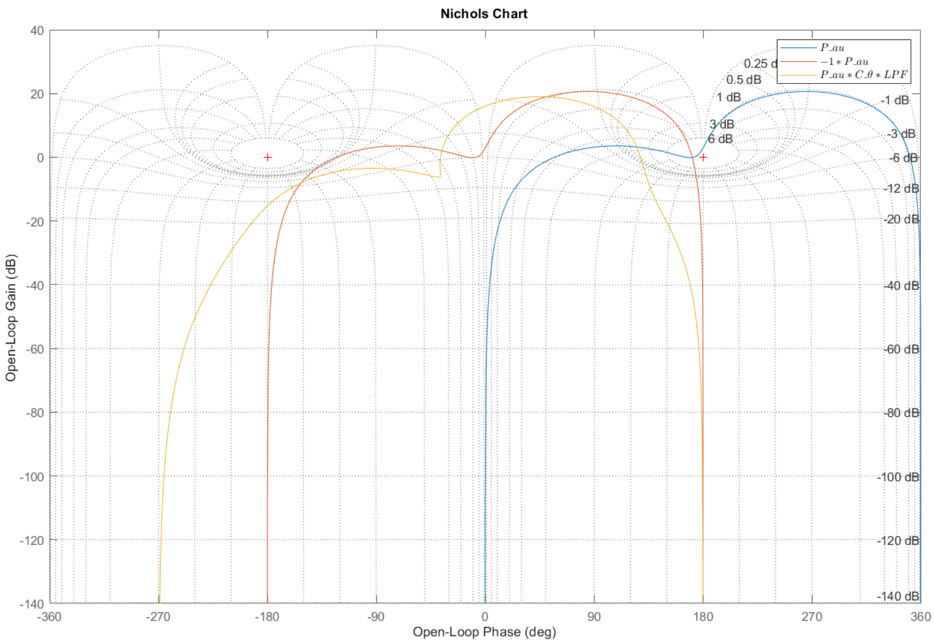


Figure 3: Nychols plot of Open loop vs Closed loop (controlled) system

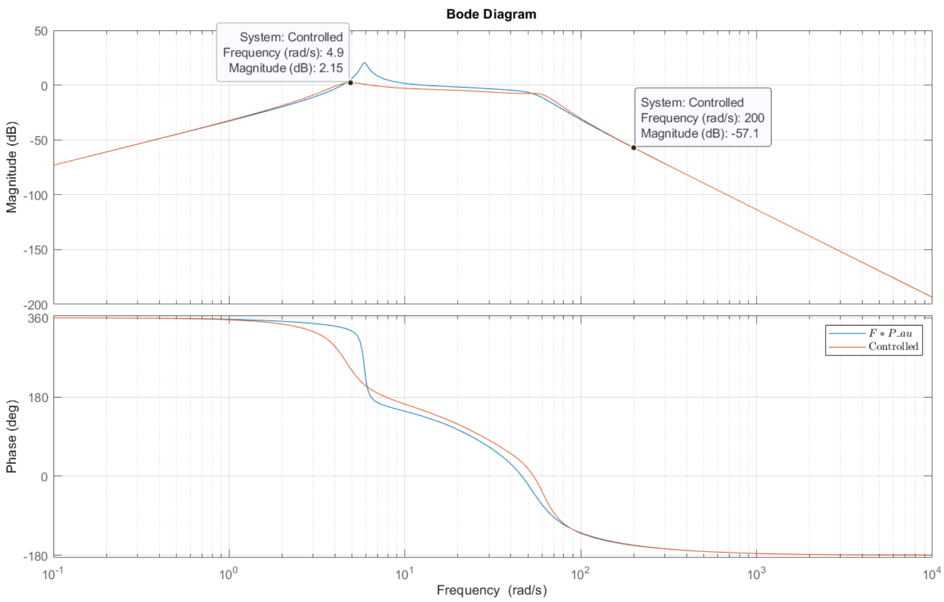


Figure 4: Bode of Open loop vs Closed loop (controlled) system

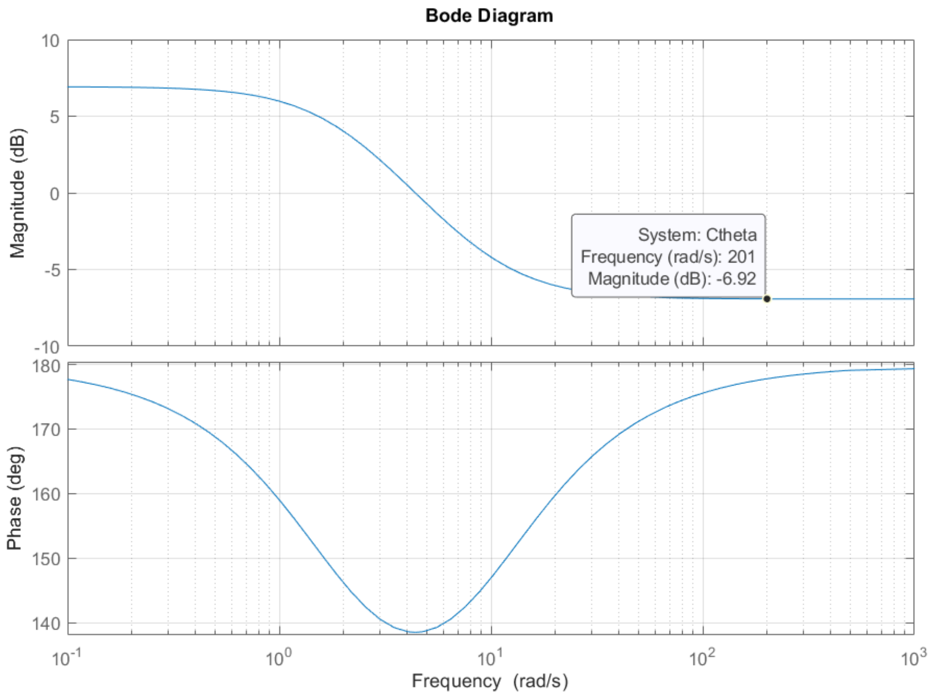


Figure 5: Bode of Controller C_θ

As we can see in both Figure 3 and 4, The requirement of the gain being lower than 3 [dB] is satisfied. The slope of -20 [dB/s] is also satisfied. In Figure 5, we can see that for frequencies above 200 [rad/s], the gain is less than 0 [dB].

2.2 Block diagrams

The updated block diagram of the nonlinear system, with the pendulum controller C_θ is:

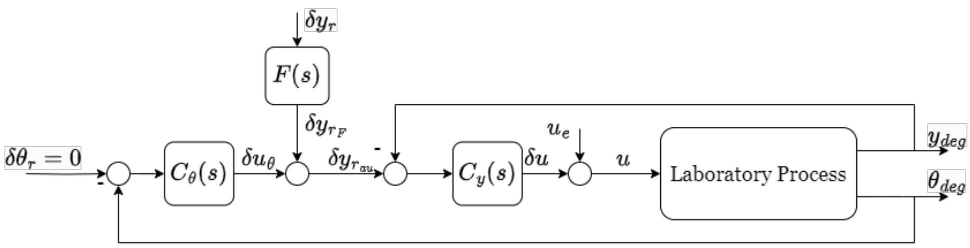


Figure 6: NonLinear block diagram from lab 4

The SIMULINK schematics for both Linear & NonLinear systems can be seen in the figures below. The input signals are the reference signals used in the experiment, one without the pendulum controller ($C_\theta = 0$) and one with the pendulum controller C_θ

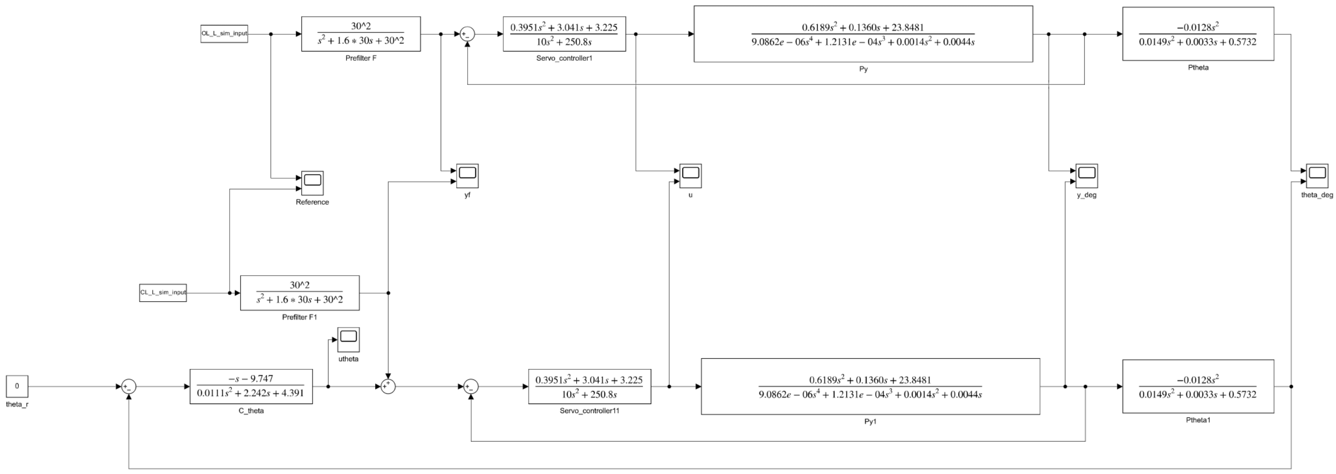


Figure 7: SIMULINK schematic for Linear system

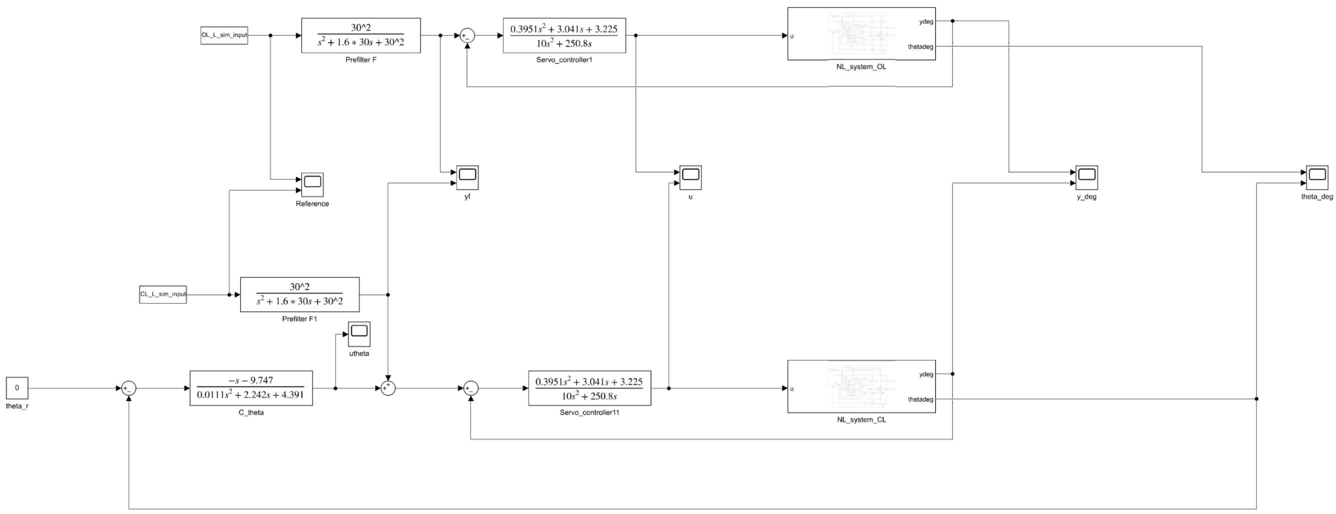


Figure 8: SIMULINK schematic for NonLinear system

The SIMULINK experiment schematic can be seen below.

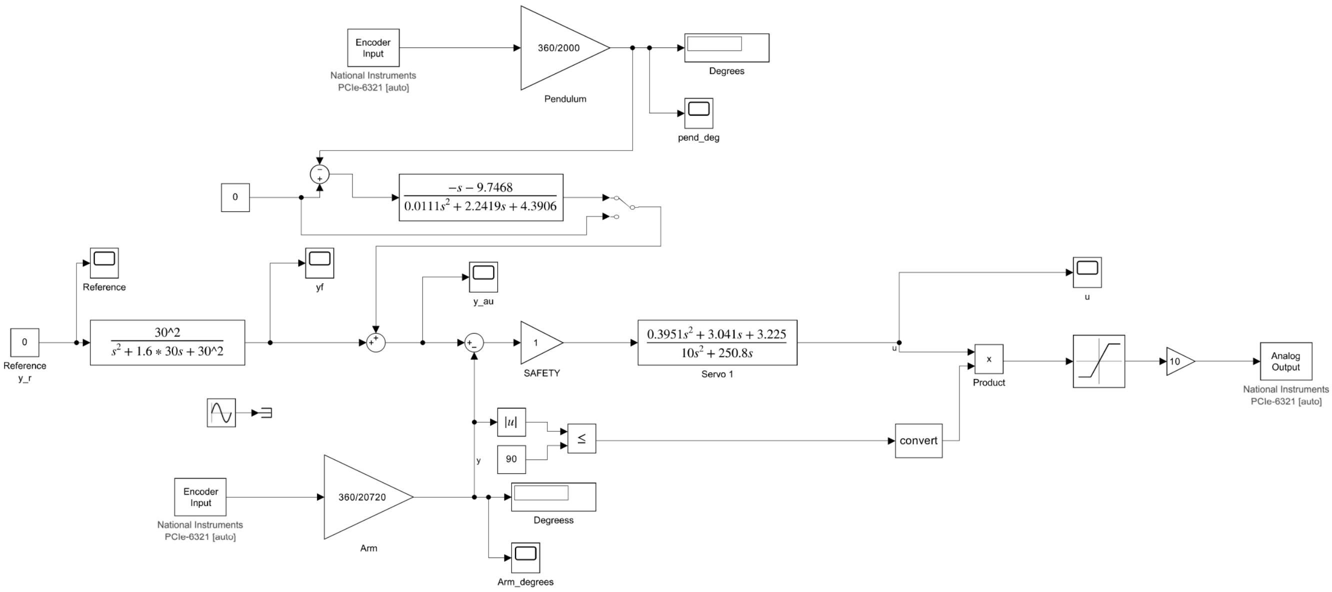


Figure 9: SIMULINK schematic for the experiment

2.3 Graphs

2.3.1 Bode & polar plots

The Bode plot of the transfer function from δy_r to $\delta\theta_{deg}$ in open loop (with $C_\theta(s) = 0$) and in closed loop (with designed $C_\theta(s)$) can be seen in Figure 10

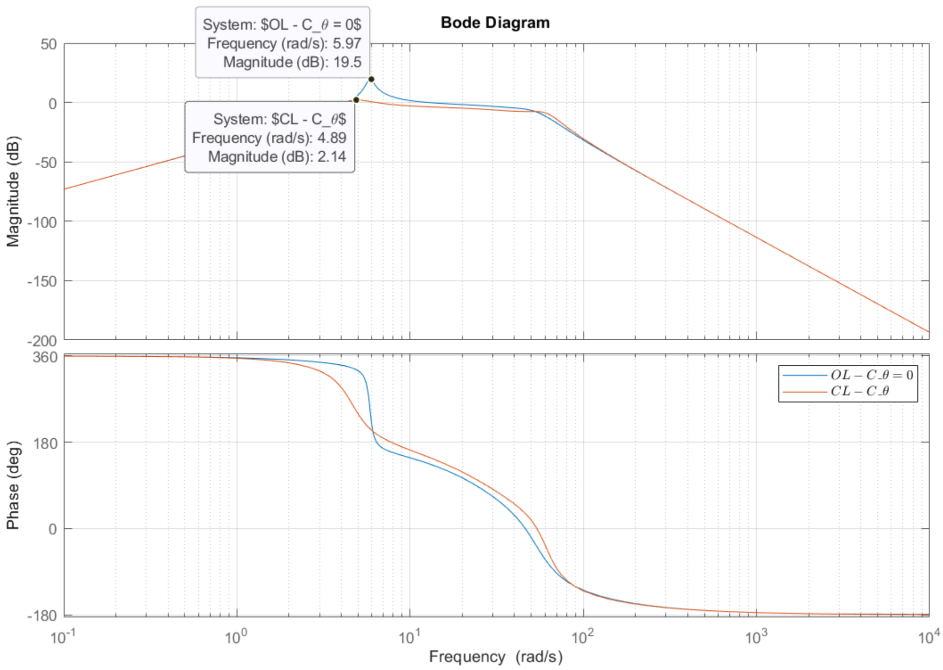


Figure 10: Bode plot from of transfer function δy_r to θ_{deg}

The poles of the transfer function from δy_r to $\delta\theta_{deg}$ in open loop (with $C_\theta(s) = 0$) and in closed loop (with designed $C_\theta(s)$) can be seen in Figures 11 and 12

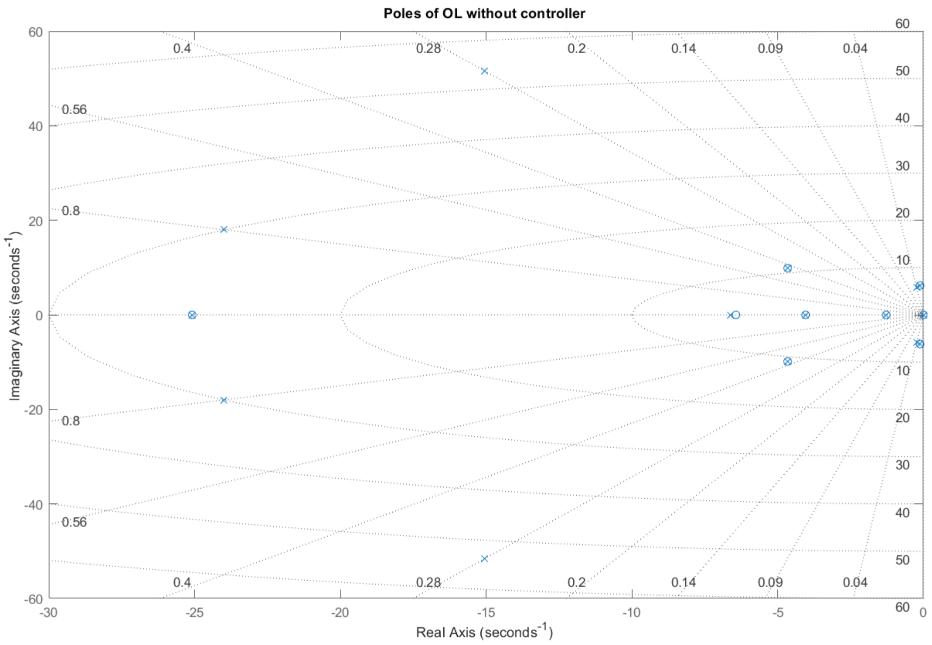


Figure 11: Poles of the transfer function δy_r to θ_{deg}, in open loop ($C_\theta(s) = 0$)

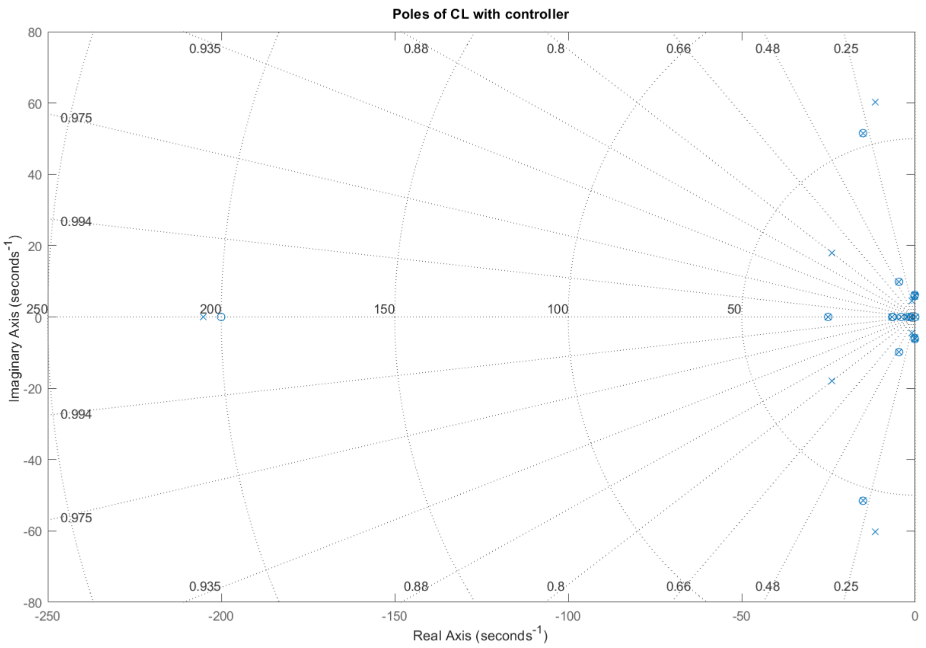


Figure 12: Poles of the transfer function δy_r to θ_{deg}, in closed loop (with designed $C_\theta(s)$)

The Bode plot of the transfer function from δy_r to δy_{deg} in open loop (with $C_\theta(s) = 0$) and in closed loop (with designed $C_\theta(s)$) can be seen in Figure 13

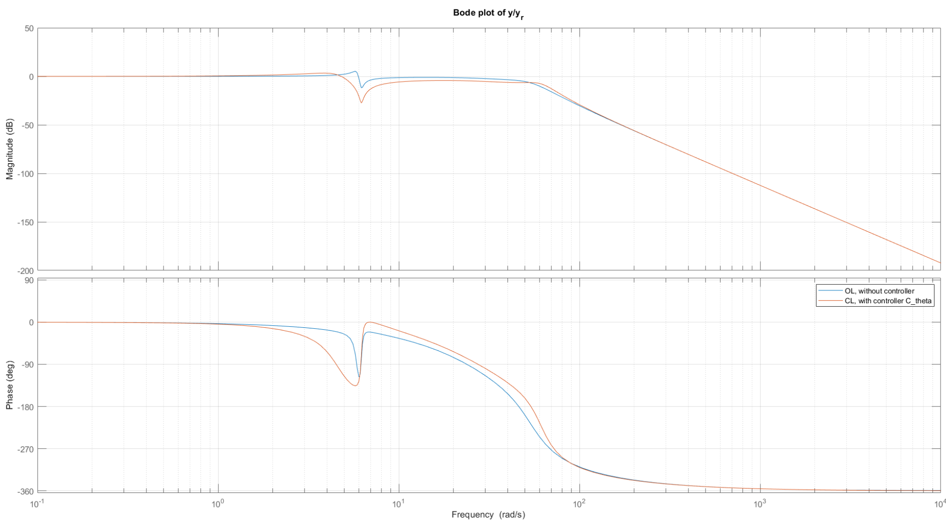


Figure 13: Poles of the transfer function δy_r to θ_{deg}, in closed loop (with designed $C_\theta(s)$)

2.3.2 Experiment

The comparison graphs between the original & the filtered reference signal, arm angle, pendulum angle and control signals of the two controllers can be seen in the figures below:

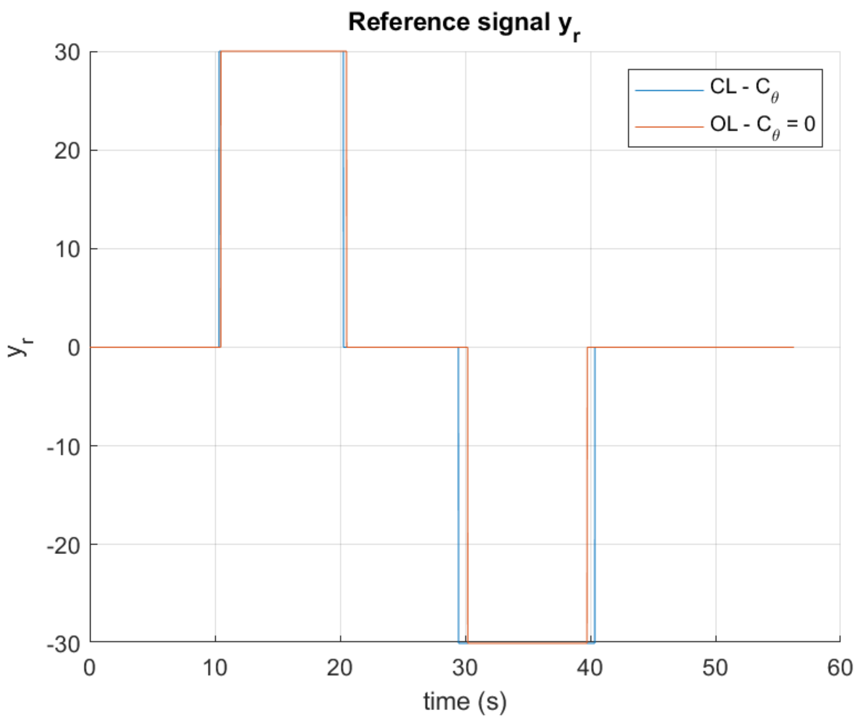
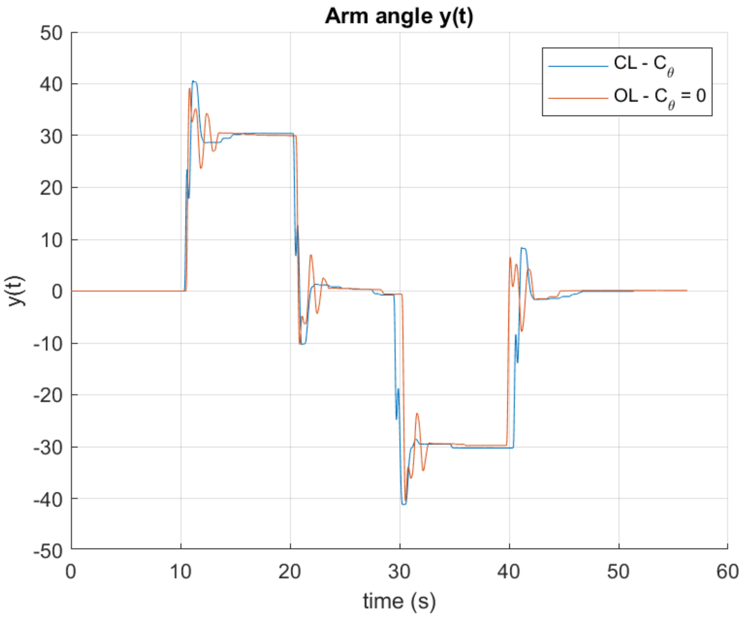
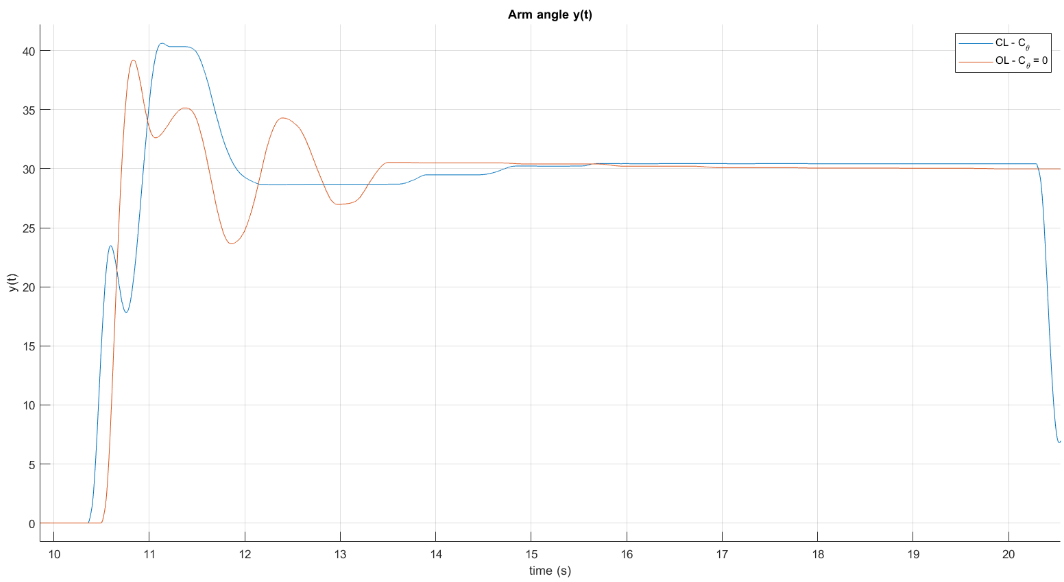


Figure 14: Reference signal y_r of the open-loop & closed-loop system

We tried to set the steps of $\pm30°$ to every 10 seconds, for both experiments.

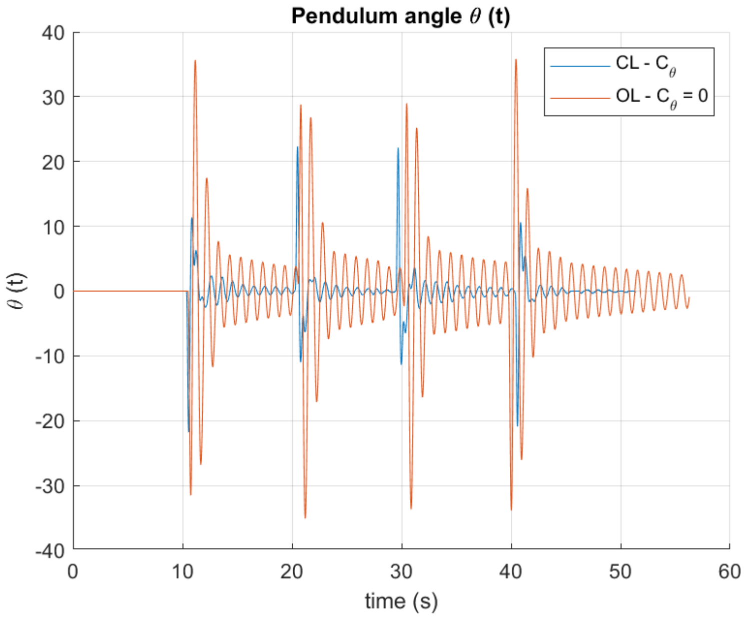


(a) Full view

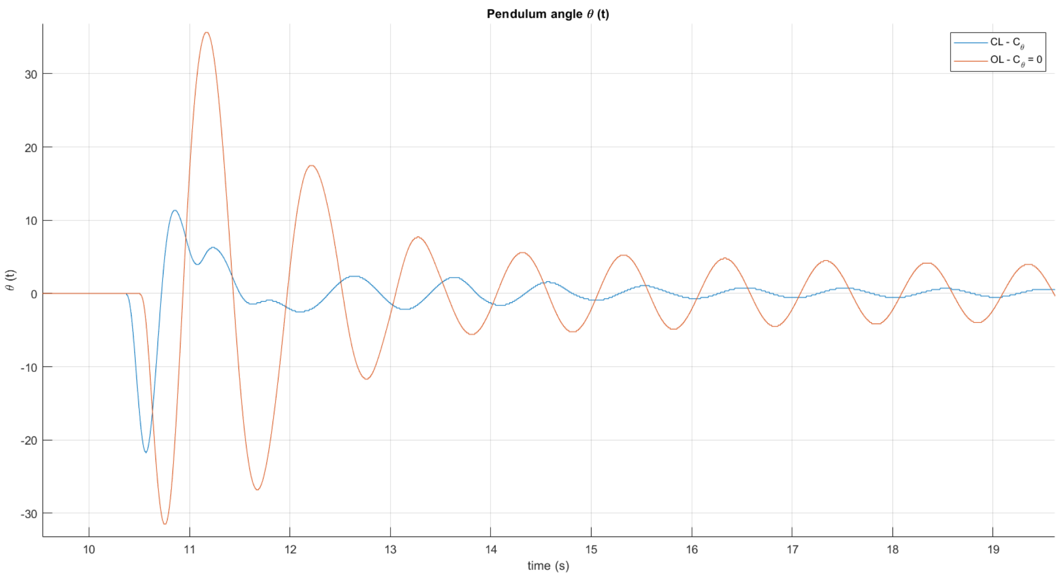


(b) Zoomed to a single response

Figure 15: arm angle y of the open-loop & closed-loop system



(a) Full view



(b) Zoomed to a single response

Figure 16: pendulum angle θ of the open-loop & closed-loop system

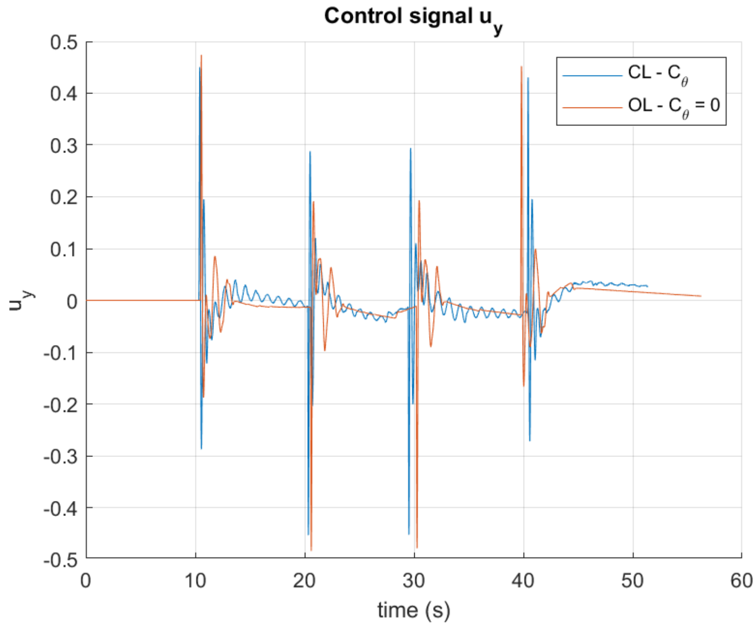


Figure 17: control signal u_y of the open-loop & closed-loop system

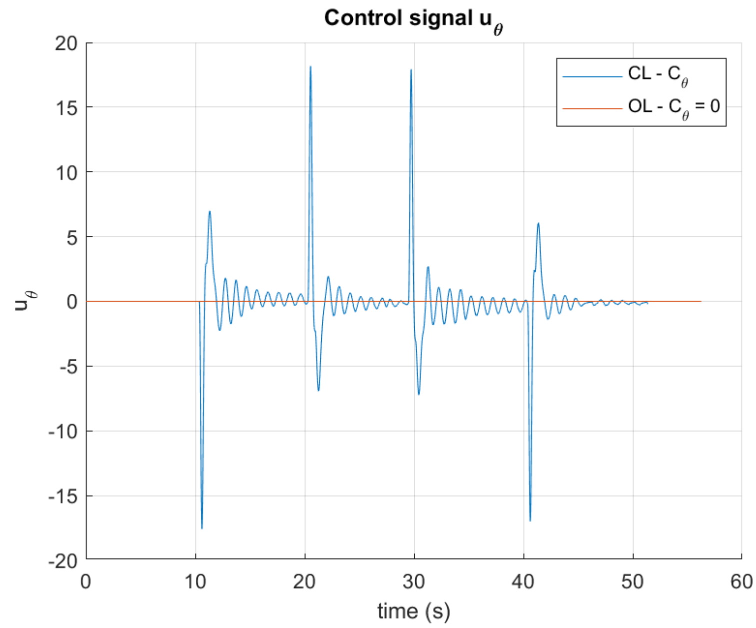
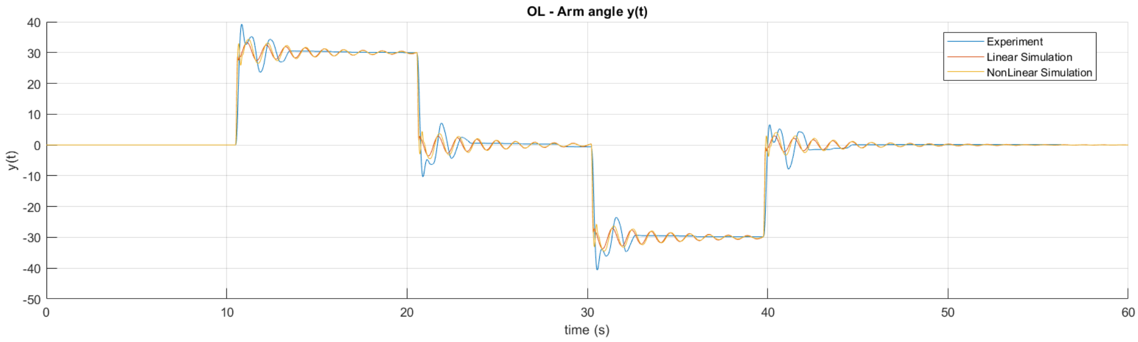


Figure 18: control signal u_y of the open-loop & closed-loop system

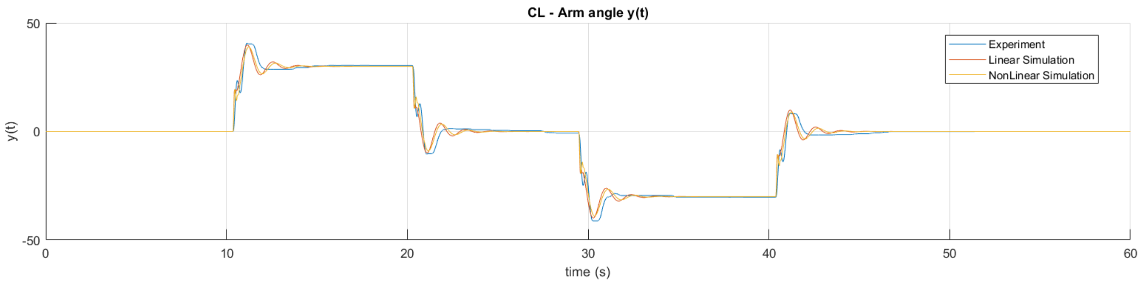
We can see in Figure 16 that our pendulum controller works because the pendulum oscillations have been dampened. Note that in Figure 18, the open-loop signal is 0, as it is non-existent.

2.3.3 Simulations

The experiment signals of both open-loop ($C_\theta(s) = 0$) & closed-loop (with designed $C_\theta(s)$) have been compared with a linear & non-linear simulation on SIMULINK. (the Schematics can be seen in Figure 7 & 8, respectively). The input signals are the reference signals that can be seen in Figure 14. Both Linear & non-linear simulations have been compared with the experiment values using the RMSE method. The values can be seen in the table 1

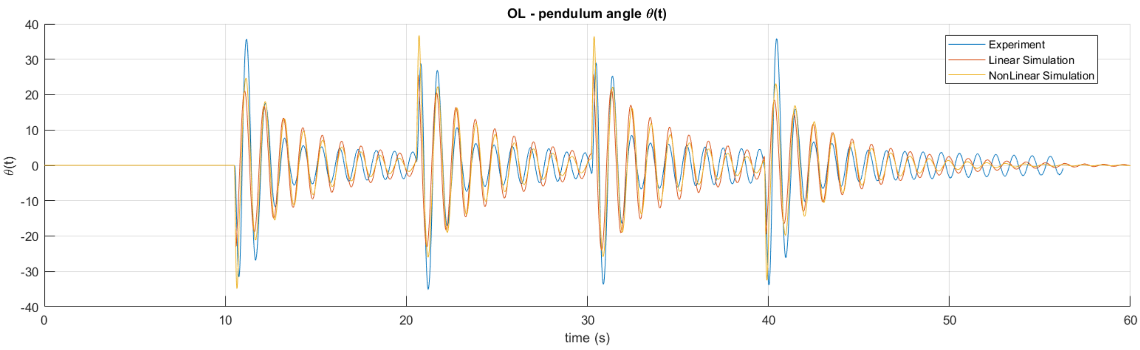


(a) Open-loop

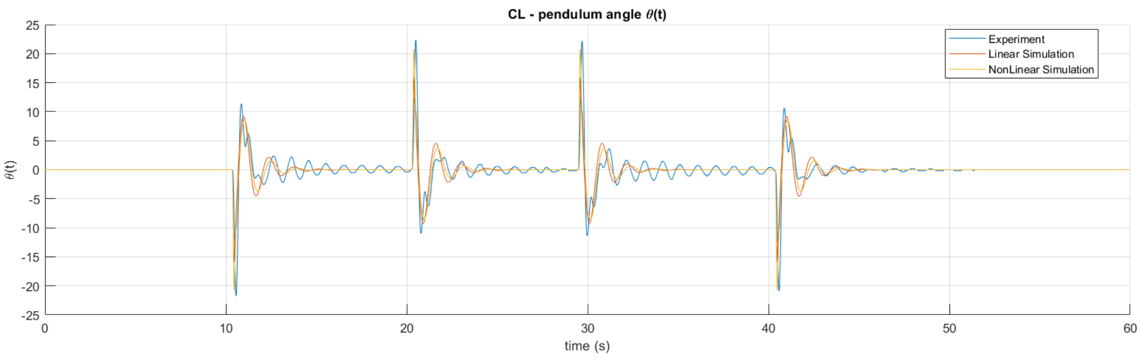


(b) Closed loop

Figure 19: arm angle y_{deg} of the open-loop & closed-loop system

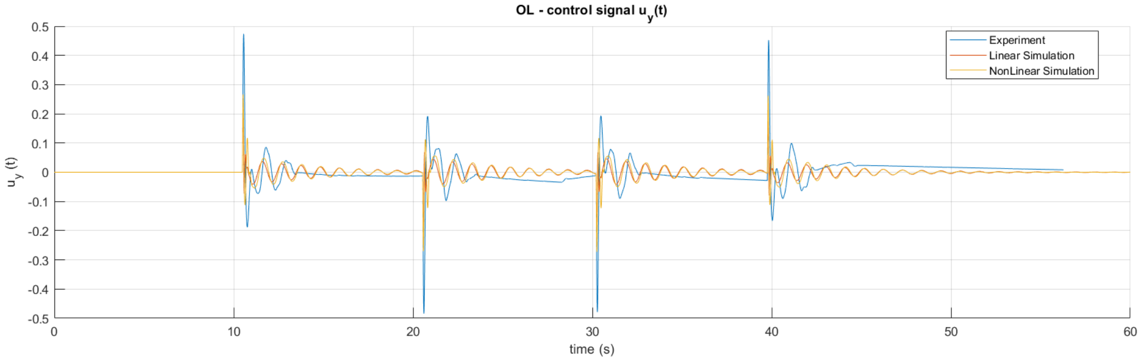


(a) Open-loop

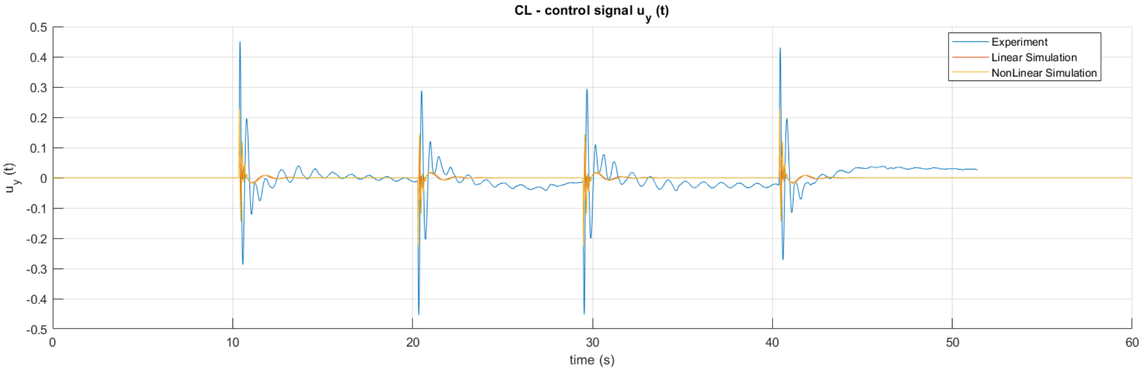


(b) Closed loop

Figure 20: pendulum angle θ_{deg} of the open-loop & closed-loop system



(a) Open-loop



(b) Closed loop

Figure 21: control signal u_y of the open-loop & closed-loop system

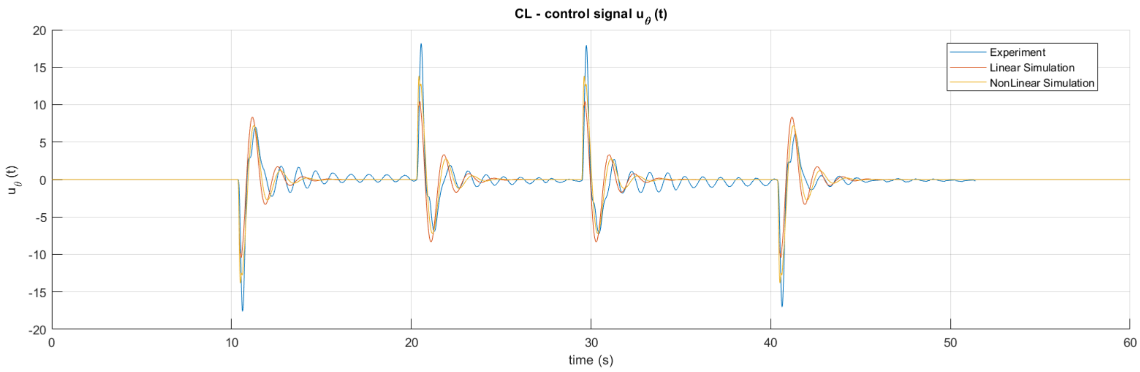


Figure 22: control signal u_θ of the closed-loop system

Angle	**Loop**	**Simulation**	**RMSE**
Arm y_{deg}	Open	Linear	2.4442%
		Non-linear	2.4209%
	Closed	Linear	2.24%
		Non-linear	1.8044%
Pendulum θ_{deg}	Open	Linear	5.9224%
		Non-linear	4.1747%
	Closed	Linear	1.9599%
		Non-linear	1.7698%
Control u_y	Open	Linear	0.0453%
		Non-linear	0.0455%
	Closed	Linear	0.0507%
		Non-linear	0.0526%
Control u_θ	Closed	Linear	1.5354%
		Non-linear	1.1131%

Table 1: RMSE values of the simulations vs experiment signals

In all cases, the non-linear simulation provides a more accurate simulation. This is to be expected since our experiment is part of a non-linear system. However, this accuracy is not improved significantly. Since we are near our equilibrium point, and since the non-linear approximation took significantly more time and is a lot more complex to run, the linear approximation is preferred. The pendulum angle θ_{deg} has slightly higher RMSE values than the other signals, meaning that, although the controller did manage to dampen the pendulum oscillations,our experiment didn't fully follow the simulation signals.

3 Conclusion

In this lab, we successfully designed and implemented a controller to manage the pendulum angle $\theta(t)$ oscillations within the servo control system. The controller met the design requirements, effectively reducing the steady-state error to zero and significantly damping the high pendulum oscillations observed in previous labs. The performance of the controller was verified through both Bode and Nichols plots, which demonstrated compliance with the specified gain and phase margins. The comparison between the linear and non-linear system simulations with experimental data showed that while the non-linear model provided slightly more accurate results, the linear model was sufficiently accurate for practical purposes, offering a simpler and more computationally efficient solution. This validates the effectiveness of our control design and its application in similar control systems.

4 MATLAB

```
clear; close all; clc
load("Controller1.mat")
load("Py.mat")
load("Ptheta.mat")

% Rename Variables
Ptheta = P_theta1; Cy = controller1; Py = P_y1;
s=tf('s');
F = 30^2 / (s^2+1.6*30*s+30^2); % Prefilter

% Parameters
kc = 0.015;
Km = 24.3E-3; % [Nm/A]
Kb = Km;
Ra = 2.08; %[Ohm]
la = 264E-6; %[H]
Ku = 12; % [v]
Nr = 5.18;
g = 9.81; % [m/s^2]
L = 0.35; % [m]         %arm length
r = 0.26; % [m]         %pendulum length
M = 0.051; % [kg]       %pendulum mass
J = 1.4094e-6; % From lab report 2 page 10
f = 2.424e-6; % From lab report 2 page 10
c = 5.446324869703380e-04; % From lab report 4

Pau = Ptheta * (Py*Cy / (1+Py*Cy));

marg = allmargin(Pau);
Wcp = marg.PMFrequency(1);

%Design controller
phi_m = 41.5;
alpha = (1+sind(phi_m)) / (1-sind(phi_m));
% alpha = 3;
Clead = (sqrt(alpha)*s + Wcp) / (s+sqrt(alpha)*Wcp);
Ctheta = -1/Clead;

tau = 1/200;
lpf = 1 / (tau*s+1); % Low Pass Filter
Controller_Theta = Ctheta*lpf;

open_loop = Pau * Controller_Theta;
closed_loop = (F*Pau) / (1+Controller_Theta*Pau);

%% Q1
figure; hold on
nichols(Pau); nichols(-Pau); nichols(open_loop)
grid on
legend('$P_{au}$','$-1*P_{au}$', '$P_{au}*C_{\theta}*LPF$','Interpreter','latex')

figure; hold on
bode(F*Pau); bode(closed_loop);
grid on
legend('$F*P_{au}$', 'Controlled','Interpreter','latex')

figure; hold on
bode(open_loop)
grid on
legend('$L = P_{au}*C_{\theta}$','Interpreter','latex')

%% Q2

Q2_OL = (F*Pau) / (1 + (Pau*Controller_Theta*0));
Q2_CL = (F*Pau) / (1 + (Pau*Controller_Theta) );

figure; hold on
bode(Q2_OL) ; bode(Q2_CL)
grid on
legend('$OL - C_{\theta} = 0$', '$CL - C_{\theta}$','Interpreter','Latex')
```

```matlab
%% Q3

figure; hold on; pzmap(Q2_OL); grid on; title('Poles of OL without controller');
figure; hold on; pzmap(Q2_CL); grid on; title('Poles of CL with controller');

%% Q4

y_to_yr_OL = F*Pau / Ptheta;
y_to_yr_CL = (closed_loop) / Ptheta;

figure; hold on;
bode(y_to_yr_OL); bode(y_to_yr_CL);
grid on
title('Bode plot of y/y_r')
legend('OL, without controller' , 'CL, with controller C_{theta}');

%% Q5
close all;

% Experiment signals
CL = load('data_closed_0.mat');      % WITH CONTROLLER
CL_time = CL.reference_0.time;
CL_yr = CL.reference_0.signals.values;  %Before prefilter
CL_yf = CL.yf_0.signals.values;
CL_yau = CL.y_au_0.signals.values;
CL_uy = CL.u_0.signals.values;
CL_utheta = CL_yau - CL_yf;          %We forgot to save u_{theta}, but we know that utheta
    +yf = yau
CL_ydeg = CL.armdegrees_0.signals.values;
CL_penddeg = CL.pend_deg_0.signals.values;
CL_L_sim_input = [CL_time CL_yr];

OL = load('data_open_2.mat');        % WITHOUT CONTROLLER
OL_time = OL.reference_2.time;
OL_yr = OL.reference_2.signals.values;  %before prefilter
OL_yf = OL.yf_2.signals.values;
OL_yau = OL.y_au_2.signals.values;
OL_uy = OL.u_2.signals.values;
OL_utheta = OL_yau - OL_yf;          %We forgot to save u_{theta}, but we know that utheta
    +yf = yau
OL_ydeg = OL.armdegrees_2.signals.values;
OL_penddeg = OL.pend_deg_2.signals.values;
OL_L_sim_input = [OL_time OL_yr];

figure; hold on; grid on;
plot(CL_time,CL_penddeg); plot(OL_time,OL_penddeg); legend('CL - C_{\theta}','OL - C_{\
    theta} = 0');
title('Pendulum angle \theta (t)'); xlabel('time (s)'); ylabel('\theta (t)'); hold off

figure; hold on; grid on;
plot(CL_time,CL_ydeg); plot(OL_time,OL_ydeg); legend('CL - C_{\theta}','OL - C_{\theta} =
     0');
title('Arm angle y(t)'); xlabel('time (s)'); ylabel('y(t)'); hold off

figure; hold on; grid on;
plot(CL_time,CL_utheta); plot(OL_time,OL_utheta); legend('CL - C_{\theta}','OL - C_{\
    theta} = 0');
title('Control signal u_{\theta}'); xlabel('time (s)'); ylabel('u_\theta'); hold off

figure; hold on; grid on;
plot(CL_time,CL_uy); plot(OL_time,OL_uy); legend('CL - C_{\theta}','OL - C_{\theta} = 0')
    ;
title('Control signal u_y'); xlabel('time (s)'); ylabel('u_y'); hold off

figure; hold on; grid on;
plot(CL_time,CL_yr); plot(OL_time,OL_yr); legend('CL - C_{\theta}','OL - C_{\theta} = 0')
    ;
title('Reference signal y_r'); xlabel('time (s)'); ylabel('y_r'); hold off

saveas(figure(1),'theta.png'); saveas(figure(2),'y.png'); saveas(figure(3),'utheta.png');
     saveas(figure(4),'uy.png'); saveas(figure(5),'yref.png');

%% Linear + NonLinear simulation
```

```
Lsim = load('L_simulation2.mat');
Lsim_time = Lsim.out.tout;
Lsim_ydeg_OL = Lsim.out.L_sim_ydeg.signals(1).values;
Lsim_ydeg_CL = Lsim.out.L_sim_ydeg.signals(2).values;
Lsim_penddeg_OL = Lsim.out.L_sim_thetadeg.signals(1).values;
Lsim_penddeg_CL = Lsim.out.L_sim_thetadeg.signals(2).values;
Lsim_utheta_OL = Lsim.out.L_sim_utheta.signals(1).values;
%Lsim_utheta_CL =: 0 by default
Lsim_u_OL = Lsim.out.L_sim_u.signals(1).values;
Lsim_u_CL = Lsim.out.L_sim_u.signals(2).values;

NLsim = load('NL_simulation3.mat');
NLsim_time = NLsim.out.tout;
NLsim_ydeg_OL = NLsim.out.NL_sys_ydeg.signals(1).values;
NLsim_ydeg_CL = NLsim.out.NL_sys_ydeg.signals(2).values;
NLsim_penddeg_OL = NLsim.out.NL_sys_thetadeg.signals(1).values;
NLsim_penddeg_CL = NLsim.out.NL_sys_thetadeg.signals(2).values;
NLsim_utheta_OL = NLsim.out.NL_sys_utheta.signals(1).values;
%NLsim_utheta_CL == 0 by default
NLsim_u_OL = NLsim.out.NL_sys_u.signals(1).values;
NLsim_u_CL = NLsim.out.NL_sys_u.signals(2).values;


figure; hold on; grid on;
plot(OL_time,OL_ydeg); plot(Lsim_time,Lsim_ydeg_OL); plot(NLsim_time,NLsim_ydeg_OL); legend('Experiment','Linear Simulation','NonLinear Simulation', 'location','best');
title('OL - Arm angle y(t)'); xlabel('time (s)'); ylabel('y(t)'); hold off;

figure; hold on; grid on;
plot(CL_time,CL_ydeg); plot(Lsim_time,Lsim_ydeg_CL); plot(NLsim_time,NLsim_ydeg_CL); legend('Experiment','Linear Simulation','NonLinear Simulation', 'location','best');
title('CL - Arm angle y(t)'); xlabel('time (s)'); ylabel('y(t)'); hold off;

figure; hold on; grid on;
plot(OL_time,OL_penddeg); plot(Lsim_time,Lsim_penddeg_OL); plot(NLsim_time, NLsim_penddeg_OL); legend('Experiment','Linear Simulation','NonLinear Simulation','location','best');
title('OL - pendulum angle \theta(t)'); xlabel('time (s)'); ylabel('\theta(t)'); hold off;

figure; hold on; grid on;
plot(CL_time,CL_penddeg); plot(Lsim_time,Lsim_penddeg_CL); plot(NLsim_time, NLsim_penddeg_CL); legend('Experiment','Linear Simulation','NonLinear Simulation', 'location','best');
title('CL - pendulum angle \theta(t)'); xlabel('time (s)'); ylabel('\theta(t)'); hold off;

figure; hold on; grid on;
plot(OL_time,OL_uy); plot(Lsim_time,Lsim_u_OL); plot(NLsim_time,NLsim_u_OL); legend('Experiment','Linear Simulation','NonLinear Simulation', 'location','best');
title('OL - control signal u_y(t)'); xlabel('time (s)'); ylabel('u_y (t)'); hold off;

figure; hold on; grid on;
plot(CL_time,CL_uy); plot(Lsim_time,Lsim_u_CL); plot(NLsim_time,NLsim_u_CL); legend('Experiment','Linear Simulation','NonLinear Simulation', 'location','best');
title('CL - control signal u_y (t)'); xlabel('time (s)'); ylabel('u_y (t)'); hold off;

figure; hold on; grid on;
plot(CL_time,CL_utheta); plot(Lsim_time,Lsim_utheta_OL); plot(NLsim_time,NLsim_utheta_OL); legend('Experiment','Linear Simulation','NonLinear Simulation', 'location','best');
title('CL - control signal u_\theta (t)'); xlabel('time (s)'); ylabel('u_\theta (t)'); hold off;


%RMSE between experiment & L/NL simulation
disp('OL - RMSE of the Lsim and the experimental arm angle y(t):')
disp(rmse(OL_ydeg, Lsim_ydeg_OL(1:length(OL_ydeg))))

disp('OL - RMSE of the NLsim and the experimental arm angle y(t):')
disp(rmse(OL_ydeg, NLsim_ydeg_OL(1:length(OL_ydeg))))

disp('CL - RMSE of the Lsim and the experimental arm angle y(t):')
```

```
disp(rmse(CL_ydeg, Lsim_ydeg_CL(1:length(CL_ydeg))))

disp('CL - RMSE of the NLsim and the experimental arm angle y(t):')
disp(rmse(CL_ydeg, NLsim_ydeg_CL(1:length(CL_ydeg))))



disp('OL - RMSE of the Lsim and the experimental pendulum angle theta(t):')
disp(rmse(OL_penddeg, Lsim_penddeg_OL(1:length(OL_penddeg))))

disp('OL - RMSE of the NLsim and the experimental pendulum angle theta(t):')
disp(rmse(OL_penddeg, NLsim_penddeg_OL(1:length(OL_penddeg))))

disp('CL - RMSE of the Lsim and the experimental pendulum angle theta(t):')
disp(rmse(CL_penddeg, Lsim_penddeg_CL(1:length(CL_penddeg))))

disp('CL - RMSE of the NLsim and the experimental pendulum angle theta(t):')
disp(rmse(CL_penddeg, NLsim_penddeg_CL(1:length(CL_penddeg))))


disp('OL - RMSE of the Lsim and the experimental control signal u:')
disp(rmse(OL_uy, Lsim_u_OL(1:length(OL_uy))))

disp('OL - RMSE of the NLsim and the experimental control signal u:')
disp(rmse(OL_uy, NLsim_u_OL(1:length(OL_uy))))

disp('CL - RMSE of the Lsim and the experimental control signal u:')
disp(rmse(CL_uy, Lsim_u_CL(1:length(CL_uy))))

disp('CL - RMSE of the NLsim and the experimental control signal u:')
disp(rmse(CL_uy, NLsim_u_CL(1:length(CL_uy))))


disp('CL - RMSE of the Lsim and the experimental control signal utheta:')
disp(rmse(CL_utheta, Lsim_utheta_OL(1:length(CL_utheta))))

disp('CL - RMSE of the NLsim and the experimental control signal utheta:')
disp(rmse(CL_utheta, NLsim_utheta_OL(1:length(CL_utheta))))
```