College of Engineering
School of Aeronautics and Astronautics

AAE 36401 Lab
Control Systems Lab

Lab 1 Report
The Cart on a Track

Author:
Tomoki Koike

Supervisor, TA:
A. E. Frazho, Zihao Liang

October 16th, 2020
Purdue University
West Lafayette, Indiana

Introduction

Objective

This lab was a simplified version of the usual AAE364lab course, and we only have conducted the experiment for part (V).

The objective of our experiment was to understand the function of a PID controller and modify the gains in order to control the cart on the track to slide smoothly a certain distance and slide back to the original position with a high precision. The gains of the PID controller, k_p, k_i, and k_d were tuned/modified so that the rise time, maximum overshoot, settling time, and the steady state error are minimized to satisfy our requirements.

Method

We first input the gains that we have obtained from our pre-lab. The Simulink model for the pre-lab was somewhat different from the model (see Appendix for actual Simulink model) used in our experiment so the response was not what we anticipated for. Thus, in order to obtain an ideal response, we were required to tune the PID gains with a method of trial-and-error. While we do so, we had to acknowledge the trade-offs among the gains to reduce the rise time, overshoot, settling time, and steady state error. The trade-off relationships are arranged in the table below.

Table 1: Trade-off effects of PID gains

Closed Loop Response	*Rise Time*	*Overshoot*	*Settling Time*	*Steady State Error*
k_p	decrease	increase	subtle change	decrease
k_i	decrease	increase	increase	zero
k_d	subtle change	decrease	decrease	subtle change

Understanding these relations, we changed the gains in the Simulink model and ran the simulation for 29 seconds to see the response of the cart. Once we have a response that satisfies our requirements of overshoot almost close to 0 and settling time less than 1.5 seconds, we collect the response data expressed as position of the cart as a function of time.

In the Appendix, there is an image showing the setup of the experiment.

Results

Lab-Prep Gains

The experiment performed using the gains obtained in the pre-lab was in the least satisfying. The gains used for this part of the experiment is listed in the table below.

Table 2: Gains obtained from pre-lab

	Gains
k_p	200
k_i	2
k_d	10

The next table indicates the corresponding rise time, percent overshoot, settling time, and steady state error.

Table 3: Response parameters corresponding to the gains from the pre-lab

parameter	*value*
rise time [s]	0.0815
percent overshoot [%]	4.05%
settling time [s]	0.266
steady state error	~0

Final Tuned Gains

The final PID gains are arranged in the table below.

Table 4: Final tuned PID gains from the experiment

	Gains
k_p	40
k_i	3.5
k_d	9.8

From the obtained input data and input square wave data, I used MATLAB's System Identification Toolbox's `tfest()` command to estimate the transfer function (TF) of this system. The TF resulted as the following estimated form with the denominator being an order of 4 (continuous-time identified transfer function).

$$G(s) = \frac{-1.031s^3 + 156.1s^2 - 509.7s + 2.434 \times 10^5}{s^4 + 73.29s^3 + 3294s^2 + 5.602 \times 10^4 s + 2.426 \times 10^5}$$

The parameterization and status of the estimation are the following,

Table 5: Parameters from the estimated transfer function

number of poles	4
number of zeros	3
number of free coefficients	8
fit to estimation of data	98.84%

Then performing `stepinfo(sys)`, we are able to find the corresponding rise time, percent overshoot, settling time, and steady state error. Those values are put together below.

Table 6: Response parameters for the final tuned PID gains in the experiment

parameter	*value*
rise time [s]	0.3810
percent overshoot [%]	0.02% ~ 0
settling time [s]	0.6966
steady state error	~0

Analysis & Discussion

For part (V) of this experiment, the PID gains that I have prepared from the pre-lab has given us a very radical result. The cart moved rapidly to the edge of the track and almost derailed itself. Evidently, the proportional gain was set to an excessively large value which made the cart move in an uncontrollable manner. Thus, there was a lot of space for improvement for the PID controller.

When tuning the PID controller, I first began by lowering the value of the proportional controller so that the cart would not move radically as well as reducing the rise time to an acceptable value. Then, since, we observed that the equilibrium value of the response has a steady state error, we raised the integral gain. After that, we have adjusted the derivative gain to decrease the overshoot and settling time to suffice our controller requirements. The final tuned PID gains gave us the following response plotted in MATLAB (code in Appendix).

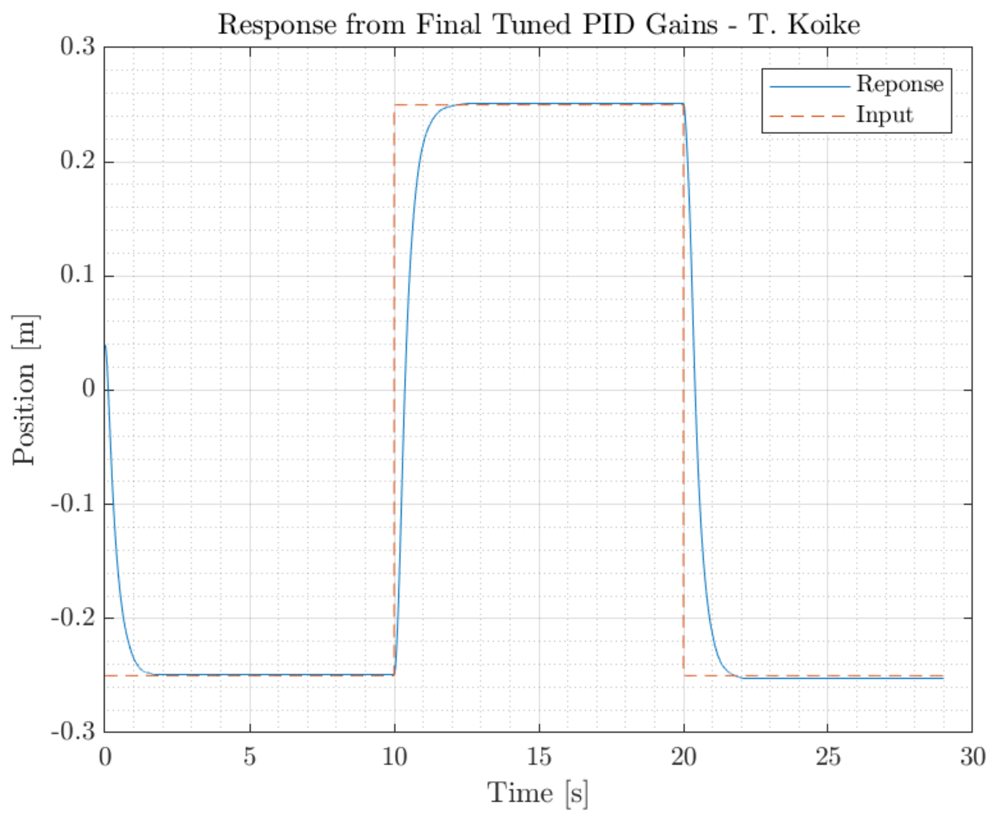


Figure 1: Response from the tuned PID controller

From this we can see that the steady state error is almost zero and the overshoot is minimized to an acceptable degree. The only concern is the rise time that makes the response diverge from the square wave input.

Conclusion & Recommendation

The tuning of the PID controller had some complications from how the system was designed with a saturation causing an integral windup with a high k_i value. Along with this concept, we were introduced to viscous friction and coulomb friction which also added variables to the control system. Conducting this experiment, we were able to satisfy the main objective of understanding how a PID controller works and how to tune it while understanding the effect of each individual gains. And as the results posit, we were successful in obtaining required gains.

As mentioned in the previous paragraph, one limitation to this experiment was the saturation. This pronounced the negative effects of increasing the integral gain value. Also, due to the input voltage of the entire setup, we were limited from increasing the gains beyond 200. Also, there are limitation of how much we can minimize the rise time. However, this would be rather taken for granted because there must be time given for the cart to move smoothly and halt safely.

Initially in the pre-lab I have set the proportional gain to an exceptionally large value; however, from this experiment, I have learned that I was wrong to do such PID design and the balance among the gains was highly important. At the end of this experiment, I was able to adopt the intuition of tuning a PID controller to meet the requirements. In future experiments, I would like to implement this intuition to diagnose what is wrong with a PID controller and how it could be tuned to improve its performance.

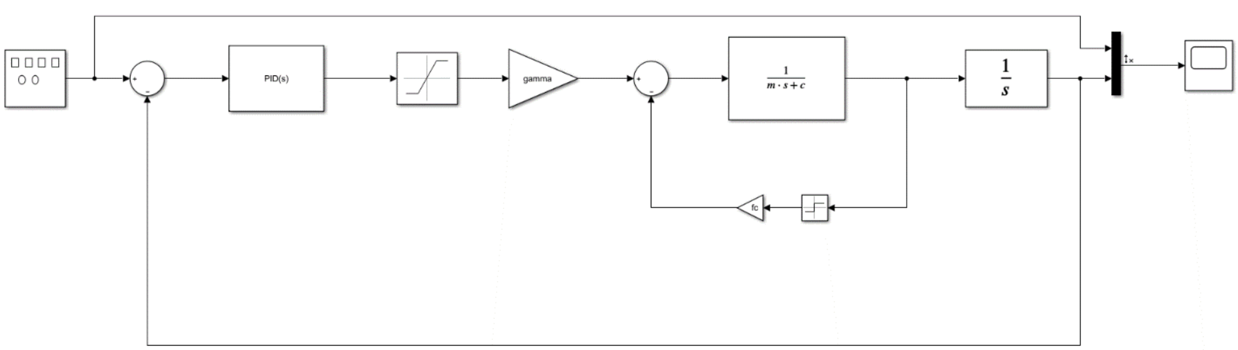
Appendix

Experiment Setup

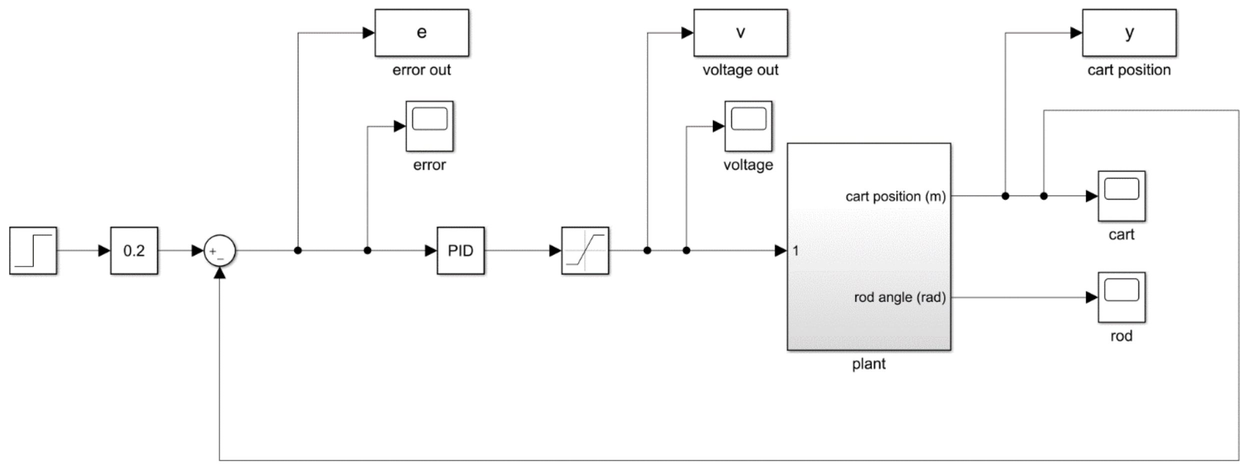
Figure 2: Image of the experiment setup

Simulink Models

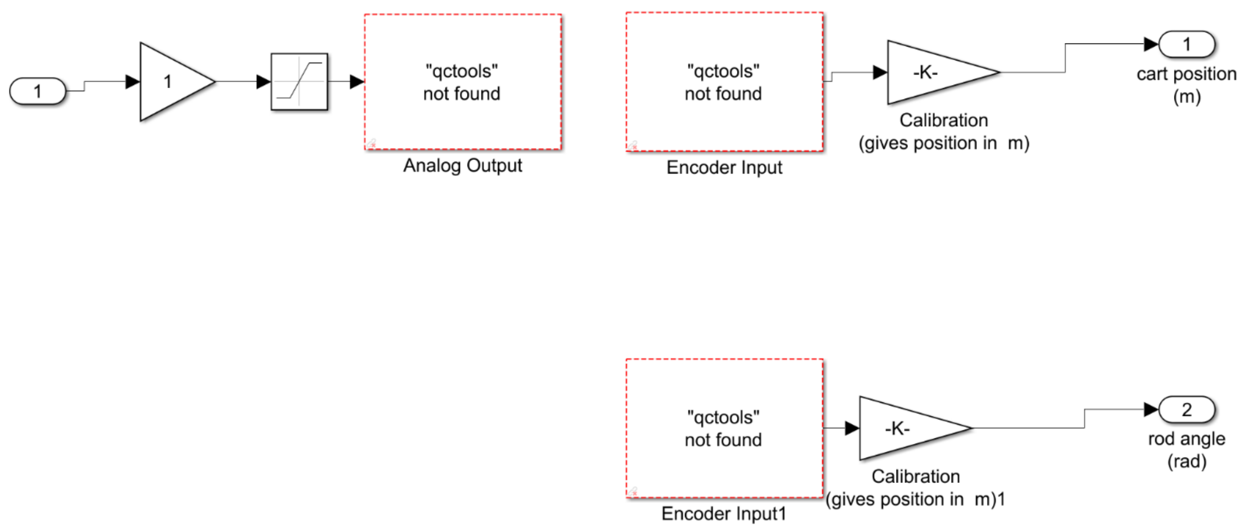
Pre-Lab Model



Lab Model (labcartpid.mdl)



Lab System Plant (labcartpid.mdl)



Raw Data

The data is shortened for convenience (displayed using Python)

	t [s]	y [m]
0	0.002	0.040195
1	0.004	0.040082
2	0.006	0.039991
3	0.008	0.039877
4	0.010	0.039695
...	...	...
14495	28.992	-0.252404
14496	28.994	-0.252404
14497	28.996	-0.252404
14498	28.998	-0.252404
14499	29.000	-0.252404

14500 rows × 2 columns

Python Code

```
# Import necessary modules
import numpy as np
import pandas as pd

# Create dataframe
df = pd.read_excel('aae364lab1_data.xlsx')
# Set column names
df.columns = ['t [s]', 'y [m]']
# Display dataframe
df
```

MATLAB Code

Pre-lab Setup Script

```
% AAE364L LabPrep1 Setting File
% Define the constants of the system
m = 1.0731;
c = 13.1236;
gamma = 1.7235;
fc = 0.2258;

% The rest is done using simulink
```

Lab1 Data Analysis Script

```
% AAE364L Lab1 Data Analysis MATLAB Script
% Tomoki Koike
clear all; close all; clc;
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
% Load the data
load('matlab_koike.mat');
t = y.time;
pos = y.signals.values;
% Plot the results

% The square wave
T = 20;  % period
A = 0.25;  % amplitude
sqr = A * square(2*pi/20 * (t - 10));

fig1 = figure;
plot(t, pos)
title('Response from Final Tuned PID Gains - T. Koike')
xlabel('Time [s]')
ylabel('Position [m]')
grid on; grid minor; box on;
hold on;
plot(t, sqr, '--')
hold off
legend('Reponse', 'Input')
saveas(fig1, 'tuned_PID_response.png');
```

```
% Estimate the transfer function the system to find the rise time, percent
overshoot,
% settling time, and steady state error

data = iddata(pos, sqr, 0.001);
sys = tfest(data, 4);
S =stepinfo(sys);
```