**LAB 5 REPORT**

LITEC

Jordan Cabahug-Almonte

Jacob Montenieri

Kristen Weatherbee

Section 4B

Teng Liu

May 2, 2018

**Table of Contents**

[**Introduction**](#_wn06g184seqi) **3**

[Purpose](#_fzb2qxhic384) 3

[Overview of Accelerometer Feedback Control](#_9dcex4cb9ep0) 3

[**Results, Plots, Analysis of Plots, and Conclusions**](#_8mp5ilyhm2wu) **5**

[Description of Goal Achieved](#_izzl0zxhw4yj) 5

[Verification of Performance Specifications](#_6f2k5nxuq9gm) 5

[Data Analysis](#_qe94tqhjk2fs) 5

[What was Learned](#_qtz2chza38cr) 13

[Problems Encountered and Solutions](#_mt4cp73bgtjq) 13

[Suggested Improvements to HW and SW](#_hsppksn75lz5) 14

[**List of References**](#_hnexg7ssc4np) **15**

[**Academic Integrity**](#_3xuminl9552) **16**

[**Participation**](#_wklrbykek7lp) **17**

# **Introduction**

## **Purpose**

This was fifth lab we have completed through the course. The purpose of this lab was to use a proportional control feedback loop to control the car’s steering and driving motors. Data from the accelerometer was used in this feedback loop to control both of these motors. We had to understand this proportional control loop with the three different gains, understand how changing the gains affected the behavior of the car, and understand how the accelerometer worked and the calibration methods associated with this sensor.

**Objectives**

There were four main objectives for this lab:

1. To understand how the proportional control loop worked and communicated the information from the accelerometer to the drive and steering motors.
2. To understand how changing the different proportional gains affected the results of the driving and steering of the car.
3. To read values from the LCD keypad and SecureCRT, such as the gains.
4. To understand how to set up the accelerometer and calibrate it.

To integrate all of these concepts into our program, we split the code up into three parts: calibrating and reading the accelerometer, a P-loop for the steering, and a P-loop for the driving. We integrated all of these parts and ran many tests on the ramp to find the gains that would give the best results. We plotted this data to find the gains that would get the car to the “flat” point in the shortest time without an undershooting or overshooting.

## **Overview of Accelerometer Feedback Control**

For this lab we were tested on being able to incorporate the Accelerometer into both our program and our circuit board. Implementing most of the wiring from Lab 4, as well as the buzzer from Lab 1, we were tasked with having the accelerometer calibrate itself to a ground value and driving up or down a ramp, depending on user input, and stopping when both the x and y values on the accelerometer matched the calibrated values. For this lab we had to implement the accelerometer to read the x and y values in a constant loop and update our steering and driving based on these values respectively. When the accelerometer reads its values there is a lot of noise in the results which causes the values to be skewed so to solve this we average the values at every 8 inputs and send the average to have a more accurate value.(This isn’t noticeable to us without looking at the code since each read happens every 20 ms.) To calibrate the accelerometer so that it was accurate based on the spot we wanted to test it in it will read in 64 values and average those for our base value and subtract that from our generated values during the trial so that they are based off of the origin plane of our choice. Another part of the accelerometer is the three gains: the kdx, ks, and kdy. These three gains would allow us to change the severity of the pulse width changes for both the steering and driving respectively. After getting the accelerometer to read the right values we had to find what values worked best for all three gains because having a value that is too high would cause the car to keep driving after it should of stopped and having too small of a value for the gains would cause the car to stall and not complete the lab.

# **Results, Plots, Analysis of Plots, and Conclusions**

## **Description of Goal Achieved**

The goal achieved in this lab was to have the car drive either up or down the ramp and have it stop at the top once it detected a flat surface with the accelerometer. The choice of whether the car drove up or down hill was dependant of the user. Using the readings from the accelerometer, the car was able to successfully control both the steering and drive motors using a proportional control loop. Before the car could drive, we had to implement a calibration method that would make the accelerometer more accurate when it reached the “flat” point of the ramp, which is where the car was calibrated (the accelerometer read very low values here). We also needed to make sure that the car did not overshoot this “flat” point or stall in the middle of the ramp. This was achieved by adjusting the three proportional gains and adding in a deadband to obtain the optimal result.

## **Verification of Performance Specifications**

We verified that the performance of the car through several tests before placing on the ramp. To test the steering, we tilted the car 45 degrees to the left and right and the wheels would have to turn in either the same or opposite direction of the tilt. This depended on whether the car was instructed to go uphill or downhill. To test the drive motors, we tilted the car upwards and downwards to see if the car would drive gradually faster as we tilted the car more in either direction.

When testing the car on the ramp, we first tested our calibration method. We calibrated the car at the bottom and top of the ramp. When the car was at these points, we had to check if the accelerometer read very low values. After verifying that our calibration method worked, we placed the car in the middle of the ramp and tested out different proportional gains to see which gains were optimal. If the car did not make it all the way up or down the ramp, the proportional gains had to be increased because the system was overdamped. If the car went past the flat portions of the ramp, the proportional gains had to be decreased because the system was underdamped. These gain values were determined by many tests and by graphing the data read from the accelerometer and the pulse widths of the steering and driving motors. We found that our most optimal gains were a kdy of 3, a kdx of 5, and a ks of 5. These gains created a critically damped system, that caused the car to not sporadically steer or drive and reach the “flat” point without undershooting or overshooting.

## **Data Analysis**

After completing our trials for getting signed off on the lab our next step was gathering data to help show the relationship between the variables used in the lab. The main variables that were used were kdx, ks, kdy, the x and y accelerometer values (after being corrected for the calibration values), and the motor and steering pulse widths. While the relationship is apparent on all of the trials we ran only two of the three trials were actually successful, with the third trial being a failure.

For our first trial, we started on the right side of the ramp and drove down the ramp onto the ground. As shown by our title on the first graph, the kdx value we used was 5, our kdy value was 3, and our ks value was 5. These values proved to be optimal because the car drove at a constant speed and stopped relatively shortly after going off the ramp. Looking at our values we can tell that there is a direct correlation between the Gx and driving pulse width as well as the Gy and steering pulse width. For each case what happens is the independent variable, Gx and Gy, are either increasing or decreasing which means the roll and pitch respectively are increasing or decreasing in value as well. Since this is happening this means that to correct it back to the origin value, like we want to, the pulse width for both groups needs to increase to help fix the problem more quickly, which is exactly what happens. Inversely, if the roll or pitch isn’t severely off from the calibrated position the pulse widths should be closer to their 0 percentage values since less power and turning is needed to get to the desired location. However since we used a deadband in our system to help the car stop more quickly, these values aren’t allowed to go to zero before ending. For example we see this happen in Table 1 in a somewhat sporadic manner (having a lower kdy value caused the car to stall, which will be discussed on the 3rd trial.).

For our second trial, we started on the left side of the ramp and drove up it stopping at the top. This trial is more difficult than the previous one due to the fact that the ramp has drop off at the top so the car has to stop extremely quickly or else it will fall off. Using the same values for kdx, ks, and kdy we were able to get extremely good results for this trial where the car stopped perfectly at the top. As mentioned previously this was due to the deadband that was thrown in because the top of the ramp did have some fluctuation where if a deadband wasn’t implemented the values from the accelerometer could cause the car to keep jerking forward until it would of drove off. Compared to the values in Table 1, Table 2 values for Gx and Gy changed much more gradually and this was because the gain values we used were the bare minimum values that allowed the trial to work which is what we desired since harsher values would cause the car to drive more recklessly. This is further supported by Graph 2 which shows that, despite the big jump in the beginning which is expected, the values change at a much more relaxed manner compared to the jumps in Graph 1.

Finally our last trial was the same as trial 2 but with a kdy value that would cause the car to stall. Since the kdy value is in charge of how hard the motor will drive lowering the value will cause the car to not have as much torque which in turn can cause it to not go up the ramp. Represented in Table three, our motor pulse width percentage was only at 49 percent at the middle of the ramp where the other trials had values for motor pulse width around 60-70 percent. Since the pulse width was too low, this caused the car to stall. This is also shown by Graph 3 where the ending Gy values has a significantly greater magnitude at the end compared to the other trials. This data proves that our best trial was our second one and shows why trials like our last one aren’t capable of working.

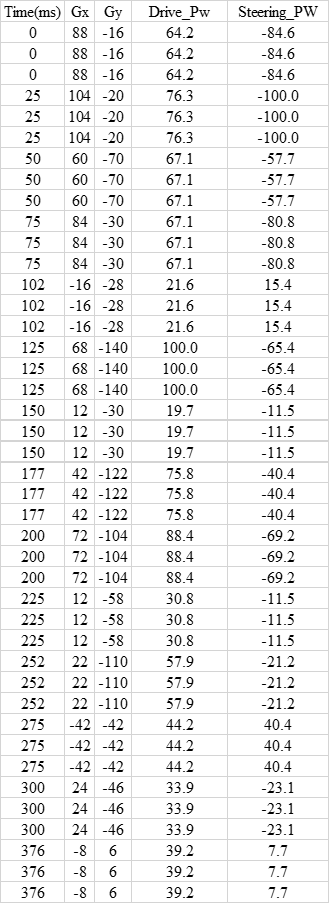
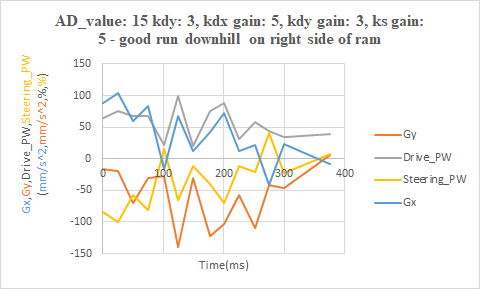


Table 1: Going downhill from right side of ramp



## Graph 1

## 

## Table 2: Going uphill from the left side of the ramp

## 

## 

## Graph 2

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Table 3: Bad run going uphill from the left side of the ramp

## 

## Graph 3

## 

## **What was Learned**

Through this lab the main takeaway was learning how to use the accelerometer as well as furthering our knowledge on proportional gain feedback control. The accelerometer is able to detect data along all three axes. We learned how to use it to detect how much the car tilted forward/back and right/left. This allowed us to calculate the pulse width of the drive motor and the steering based off of how much the car is tilting. In order to read the information from the accelerometer, it used the same I2C communication protocol with minor adjustments. We had to make sure to shift 4 bits to the right to get the information that we wanted.

In this lab we also used three different adjustable proportional gain feedback constants to allow the car to perform as accurate as possible. This showed us how drastic the effects of the proportional gain can really be. If we set our gains too high, this would cause overshoot and the results would not be accurate. If we set the gains too low, the car would either not move at all or move too slow and eventually stop before the desired result.

## **Problems Encountered and Solutions**

The most significant problem we encountered involved getting the car to stop at the top of the slope. It would either keep driving or stop too early. In order to solve this problem, we had to find the correct gain values. We knew if we set the drive gains too high then it would cause overshoot and the car would stop too late. However, if we set the drive gains too low then the car would either stop too early or not go anywhere at all. As for the steering gain, we couldn’t set it too high otherwise it would turn too much and we couldn’t set it too low or else it wouldn’t turn enough. However, we did want a relatively low steering gain so the car would turn smoothly towards the top of the slope. After testing several gain values, we found that the drive feedback gains were best at kdx being set to 6 and kdy being set to 3 while the steering gain worked best at ks being set to 5.

Another problem was run into when testing the car both uphill and downhill. In both cases the car seemed to be very jerky while driving and steering. It was going back and forth between stopping and going and the steering was very inconsistent. To solve this, we set the drive pulse width and steering pulse width to neural outside of the infinite loop in the code rather than inside of it. This made it so that instead of going back to neutral every time the infinite loop went back to the beginning, the car would be set to neutral once and then in the infinite loop the pulse widths would be able to adjust according to the accelerometer values. This stopped the jerking and made the driving/steering more accurate and consistent.

Another issue we were having was solved in two ways. The values on the accelerometer were not accurate enough and this was causing inaccuracy in the car’s performance when it tried to both drive and steer. To solve this, we changed our read\_accels function so that it averaged information from the accelerometer over 8 iterations rather than just 4 therefore averaging 8 different data points to make the values more accurate. The calibration function was updated as well to 64 iterations. We also made a dead-band of -20 to 20 for the y-direction and -10 to 10 for the x-direction. Since the values on the accelerometer were fluctuating, this dead-band was able to solve this problem so that if the values were close to zero while in-between these values then the car would recognize this area as flat and stop.

## **Suggested Improvements to HW and SW**

The main improvement we would like to see would involve the hardware. We would improve the accuracy of the accelerometer. Although we calibrated it using 64 iterations and averaged values when updating it using 8 iterations, that still did not seem to be enough when using the data. The values still fluctuated due to inaccurate readings from the accelerometer. However, if we did more iterations when averaging the values, it would have taken too much time to average so much data.

Another improvement we would make would be the way the driving gain in the y-direction (kdy) was set. In the lab we were instructed to use the potentiometer to set the gain constant value but this caused testing to be a lot slower. We had to spend extra time trying to find the gain value on the potentiometer that we wanted to test out. It could have saved time to choose this gain the same way we chose the other gains by typing in a value on the LCD panel.

# **List of References**

RPI-ECSE. (2017). Laboratory Introduction to Embedded Control Lab Manual v14.8. Troy, NY.

# **Academic Integrity**

All the undersigned hereby acknowledge that all parts of this laboratory exercise and report, other than what was supplied by the course through handouts, code templates and web-based media, have been developed, written, drawn, etc. by the team. The guidelines in the Embedded Control Lab Manual regarding plagiarism and academic integrity have been read, understood, and followed. This applies to all pseudo-code, actual C code, data acquired by the software submitted as part of this report, all plots and tables generated from the data, and any descriptions documenting the work required by the lab procedure. It is understood that any misrepresentations of this policy will result in a failing grade for the course.

The following signatures indicate awareness that the above statements are understood and accurate.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Participation**

Hardware implementation

Wiring \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pin-out sheet \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Circuit Schematic \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Software implementation

Pseudo-code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Debugging \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Data analysis

Data collection \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Graph analysis \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Report development and editing

Creating graphs \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Editing and Formatting

Report \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*The following signatures indicate awareness that the above statements are understood and accurate.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_