

RENSSELAER POLYTECHNIC INSTITUTE

Laboratory 5 Report

Yilu Zhou, Z Jin, Zhongqi Liu

Section 02, Side 9B, 10B, 11B

supervised by

Dr. Jeffrey BRAUNSTEIN

graded by

Syed NAQVI

July 12, 2019

1 INTRODUCTION

1.1 PURPOSE & OBJECTIVES

The purpose of the Lab 5 is:

1. Constructing circuits and writing program base on Lab 4 programs and circuits, to control the car going straight up or down the ramp and stop by the end of the ramp, using the data from 2-axis accelerometer.
2. Showing all necessary data on PuTTY for further study and analyzing the close-loop control.
3. Seeing how the closed-loop feedback can control the direction and orientation of the car without outside interference.

1.2 OVERVIEW OF ACCELEROMETER FEEDBACK CONTROL

The general control method of the car for this lab is closed-loop proportional control. By collecting real-time data from the accelerometer, the micro-controller can know the tilting of the car on both axis, and therefore, decide the next step. Commanding the car moving forward or backward, turning right or left, the micro-controller can control the car to move straight up or down the ramp depending on switch, and stop at the end of the ramp.

1. Collecting data from both axis of the accelerometer. Since the raw data is very noisy, 4 readings were averaged and left shift 4 bits to make it usable and converted to gx and gy for further application. Example code for collecting y-axis data:

```
AVG_GX += ((ACCEL_DATA[1] << 8) >> 4);
```

```
AVG_GY += ((ACCEL_DATA[3] << 8) >> 4);
```

2. Base on data collected, the PWM of steering servo motor and drive motor are controlled proportionally to run or stop and turn to desired direction, the end of the ramp, in this case. Here is an Example code for Servo and Drive motor:

```
STEERING_SERVO_PW = STEERING_SERVO_NEUT + KS * GX;  
DRIVE_MOTOR_PW = DRIVE_MOTOR_NEUT +/- (KDY * GY) - (KDX * abs(GX));
```

As long as either of \boxed{GX} or \boxed{GY} is not zero, the micro-controller would run the motor forward or backward. As long as \boxed{GX} is not zero, micro-controller would turn the steering toward the bottom of the ramp. And when both of \boxed{GX} and \boxed{GY} are zero, the micro-controller would stop the car.

2 RESULTS, PLOTS, ANALYSIS OF PLOTS, & CONCLUSIONS

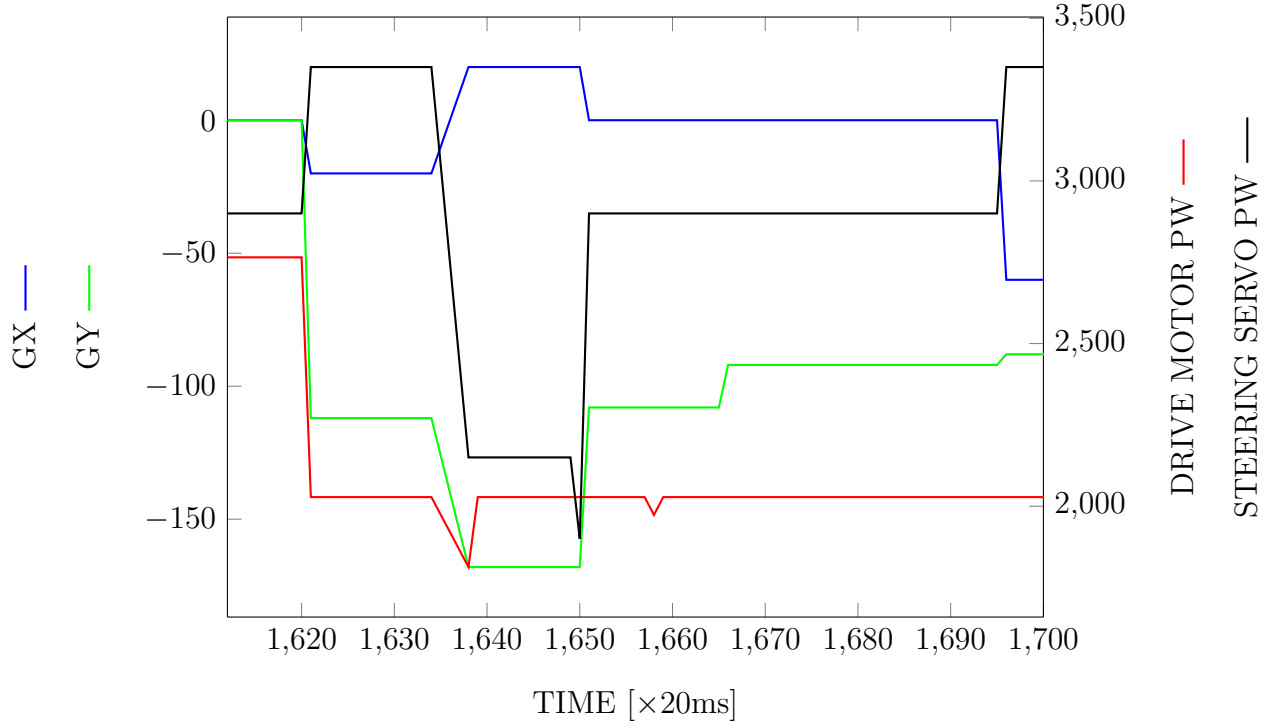
2.1 VERIFICATION

The car can perform exactly as we expected. The car is capable of moving down the ramp every time we tested. However, in the first several tests, the car cannot stop as we expected. The problem is solved and the method will be mentioned later in section 2.3. Also, the car is not capable of running up the ramp without external help due to the weak power of the drive motor. This problem only happens when the slope is too small because the `DRIVE_MOTOR_PW` depends on the slope of the ramp. However, the car could stall parallel to the slope of the ramp and it will not stall when the slope is large enough to set the `DRIVE_MOTOR_PW` to `DRIVE_MOTOR_MIN`. Therefore, if the car has enough power in the drive motor, it will finish the tasks exactly to the lab guidance.

When we placed the car on the ramp, we first place the car with front of the car down the ramp to minimize uncertainties. When we finished typing the gain, the car began to go down the ramp smoothly. As there was no need to change the pulse width of the steering servo, no change in direction happened when the car is moving. After the car reaches the bottom of the ramp, we change the status of slide switch 1. When it changed, the car should changed from going down the ramp to going up the ramp. However, because the car is currently horizontal to the ground at the bottom of the ramp, the car will not move unless the student pushed the car back to the ramp. After doing so, the car was capable to move up the ramp. When the car is at about the half of the ramp, the car stalled because the power of the drive motor was not enough to push the car the ramp when the slope is large. One of us pushed the car up the ramp and the car was capable to stop at the top of the ramp.

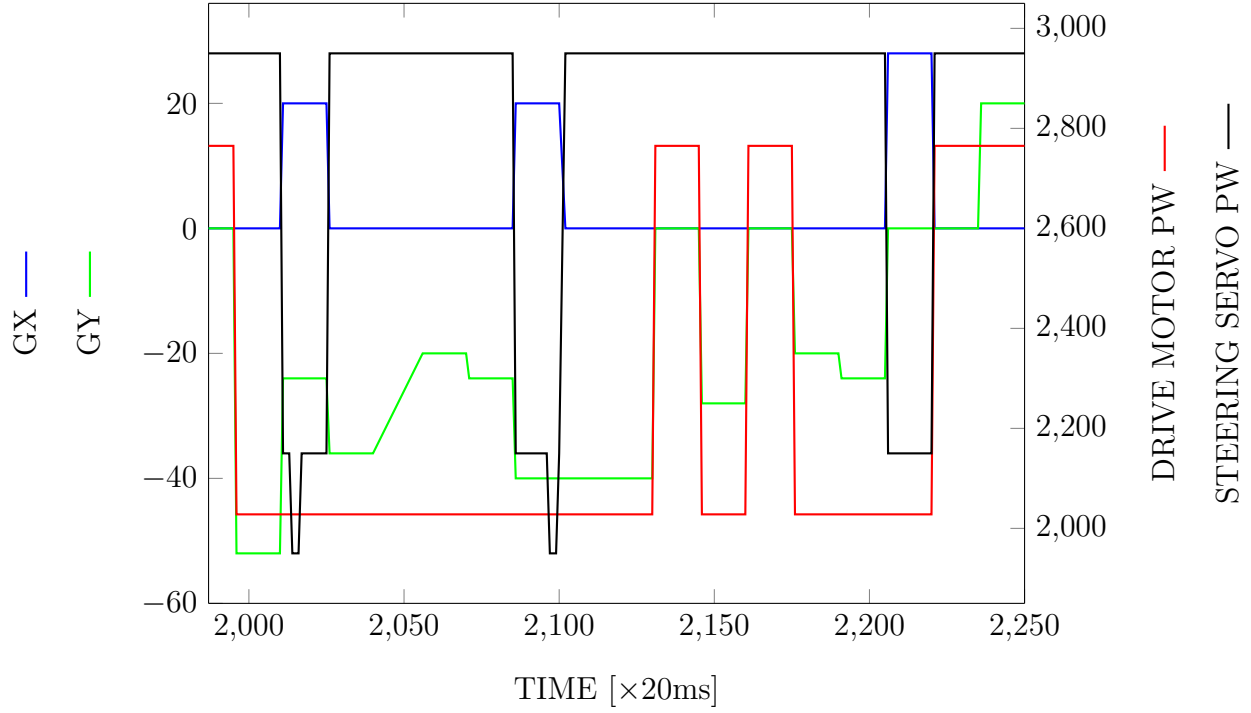
2.2 PLOTS

2.2.1 GO DOWN THE RAMP, $KS = 50$, $KDX = 50$, $KDY = 50$



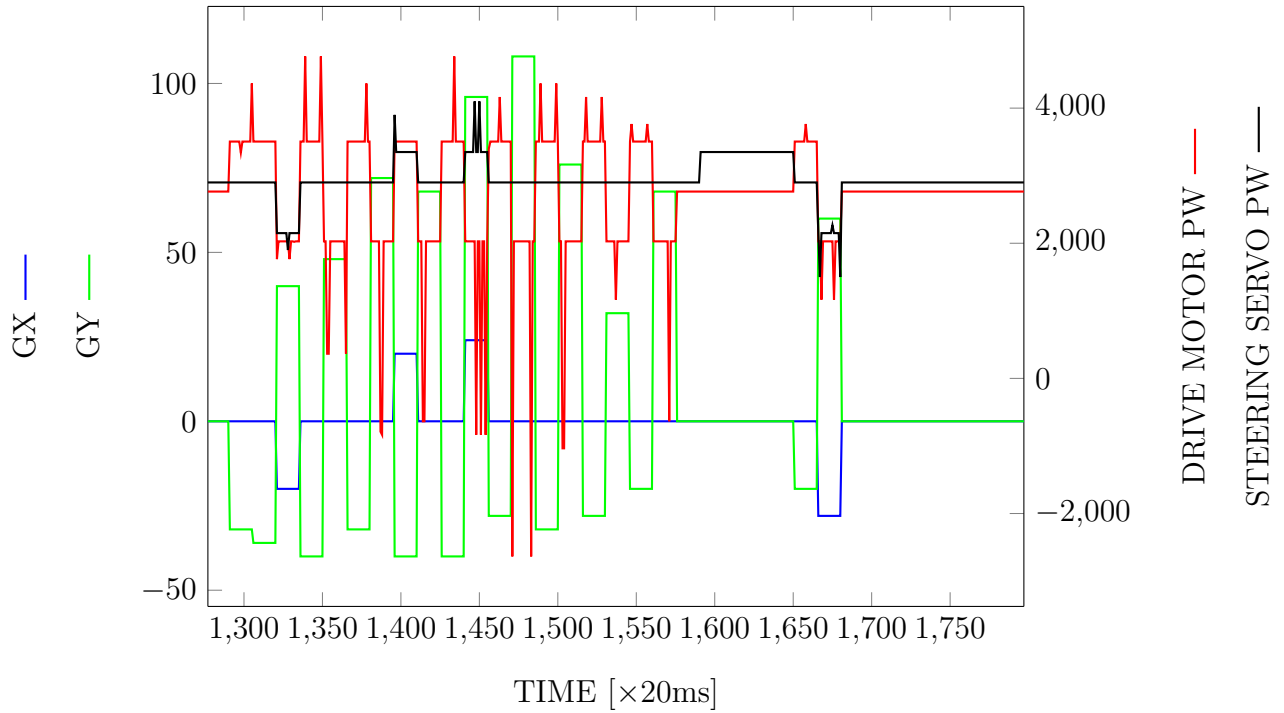
This plot shows the situation when the car goes down the ramp. Because the front of the car is pointing down the ramp, \boxed{GY} is negative as soon as the first data is read from the accelerometer. We place the car near the top of the ramp. Therefore, the \boxed{GY} is not smallest at the beginning of the plots. When the car passed through the middle of the ramp, the \boxed{GY} is smallest and therefore the car will move the fastest during the middle of the route. In this first experiment, we place the car nearly parallel to the slope of the ramp. Therefore, \boxed{GX} is close to 0 throughout the whole experiment.

2.2.2 GO UP THE RAMP, $KS = 50$, $KDX = 50$, $KDY = 50$



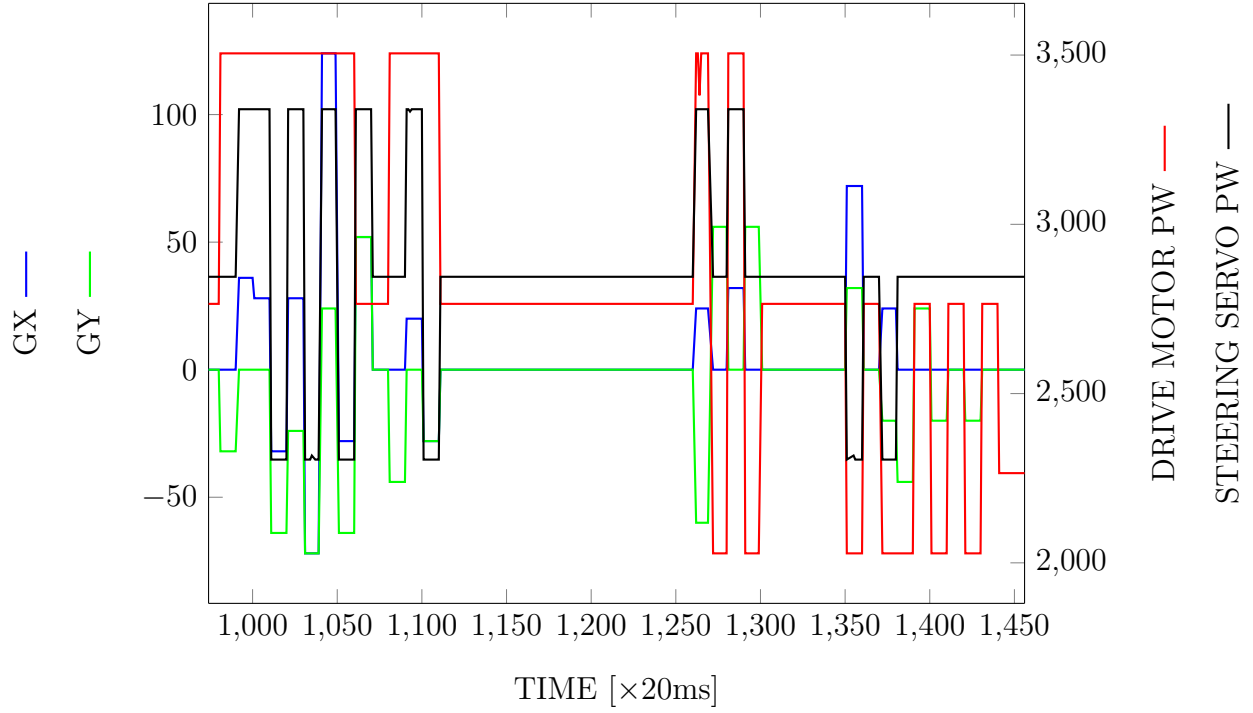
In this experiment, the car is still placing down the ramp. Therefore, the GY is still negative. However, the $DRIVE_MOTOR_PW$ is below $DRIVE_MOTOR_NEUT$ for the reverse motion. GX is nearly 0 because the car is parallel to the slope of the ramp. Near the end of the route, because the ramp is sensitive to the pressure, the car is heading to positive slope. Therefore, GY is positive near the end of the route.

2.2.3 GO DOWN THE RAMP, $KS = 50$, $KDX = 50$, $KDY = 50$



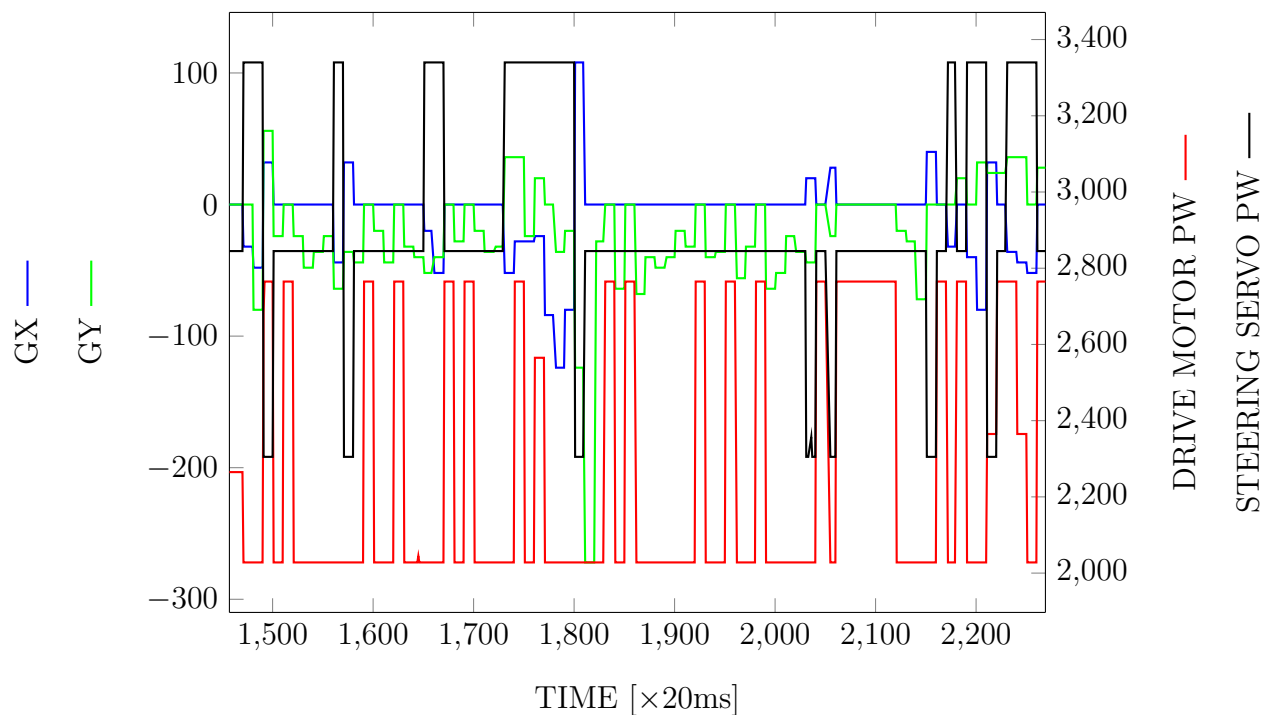
This plot shows the situation when the car was on the ramp and moving horizontal across the ramp. Because the table is not perfectly horizontal, the GX and GY are not equals to zero. The car will not try to move up the ramp nor to move down the ramp at first. However, after moving for some distance, the car will choose to move upward. This situation is confusing to us at first. However, after we plot the $DRIVE_MOTOR_PW$ and $STEERING_SERVO_PW$, the situation is clear that some pulse width data is not processed and it will exceed the limitation set by $DRIVE_MOTOR_MAX/MIN$ and $STEERING_SERVO_LEFT/RIGHT$. The correct result is shown by plot 1 and 2.

2.2.4 GO DOWN THE RAMP, $KS = 75$, $KDX = 75$, $KDY = 75$



From this plot, we can see that the period which the car process the `DRIVE_MOTOR_PW` and `STEERING_SERVO_PW` is about 20ms from the interval the PW changes from max value to min value.

2.2.5 GO UP THE RAMP, $KS = 75$, $KDX = 75$, $KDY = 75$



In this plot, GY is always negative and $DRIVE_MOTOR_PW$ is below $DRIVE_MOTOR_NEUT$. Therefore, the car is reversing up to the ramp about all the time.

2.3 PROBLEMS ENCOUNTERED & SOLUTIONS

During our testing, we had 2 problems and solved all of them in the end.

2.3.1 PROBLEMS

1. The car did not stop when it reach the bottom or top of the ramp, it moved slowly instead.
2. The car did not move backward when the slide switch was off.
3. Sometimes when the car is placing on the flat ground, it will move forward or reverse which the car should stop moving.
4. When the car successfully moved down the ramp, it cannot stop.

2.3.2 SOLUTIONS

1. The reading from accelerometer is so accurate that it detected the smallest tilting on the ground where should be considered flat. The solution is to include an offset on the data to create an area that should be considered the car is on the ground.
2. `DRIVE_MOTOR_PW` and `DRIVE_MOTOR_MAX/MIN` in the code are unsigned int, heritaged from Lab 4 codes. When calculating using a `KDX` or `KDY` that is too large, the result of `DRIVE_MOTOR_PW` would be a negative integer. And if `DRIVE_MOTOR_PW` is signed integer, neither of `DRIVE_MOTOR_MAX` or `DRIVE_MOTOR_MIN` could be unsigned int since `DRIVE_MOTOR_PW` should be limited between the minimum and the maximum and unsigned int comparing with signed int cause error. The solution is changing the data type of `DRIVE_MOTOR_PW` and `DRIVE_MOTOR_MAX/MIN` in the code to signed int.
3. Before moving up the ramp, we check the `GX` and `GY` and ensure that the accelerometer is horizontal on the EVB. Also, because the accelerometer is very sensitive

to distribution, we treat the value of `GX` and `GY` as zero when the absolute value is below 16.

4. We try to increase the interval the car reading the data from the accelerometer. Before changing the interval, the acceleration caused by the rotation of drive motor will be read and it will leads to positive feedback and the car will not stop. After changing, the useless data will be overwritten before using it to change the `DRIVE_MOTOR_PW` and `STEERING_SERVO_PW`.

2.4 SUGGESTED IMPROVEMENTS TO HW & SW

The Lab 5 is linked to the codes from the Lab4. Therefore, these codes should not be deleted. This message was shown at the beginning of the guide for LAb 5. Therefore, some of the students might have already deleted them. This message should be shown earlier in the Lab 3.3 guide to lower the probability of deleting the file.

3 ACADEMIC INTEGRITY CERTIFICATION

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.

4 PARTICIPATION

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. Participation (this is only a template; make changes as appropriate or necessary) The following individual members of the team were responsible for (give percentages of involvement)

1. Hardware implementation:

- Zhongqi Liu 30%
- Yilu Zhou 40%
- Z Jin 30%

2. Software implementation:

- Zhongqi Liu 20%
- Yilu Zhou 40%
- Z Jin 40%

3. Data Analysis:

- Zhongqi Liu 30%
- Yilu Zhou 50%
- Z Jin 20%

4. Report Development & Editing:

- | | |
|---------------|-----|
| • Zhongqi Liu | 25% |
| • Yilu Zhou | 50% |
| • Z Jin | 25% |

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

-
-
-