Rensselaer Polytechnic Institute

Laboratory 6 Report

Yilu Zhou, Z Jin, Zhongqi Liu

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

supervised by

Dr. Jeffrey Braunstein

graded by

Syed Naqvi

June 28, 2019

1 INTRODUCTION

1.1 PURPOSE & OBJECTIVES

1. Learning the differences between the proportional and derivative (PD) control of smart car and that of the gondola.

2. Implement PD control based on the measurements from the compass. Read from the keypad to set the initial-desired heading and `KD` (derivative gain) and `KP` (proportional gain).

3. Making the gondola go to the desired heading at a reasonable speed while minimizing the damping. Using the direction from compass to modify this value during continuous operation by using closed-loop feedback.

1.2 OVERVIEW OF GONDOLA FEEDBACK CONTROL

The general control method in this lab for gondola is PD control. First, the program will ask the users to input the `KP` (proportional gain), `KD` (derivative gain) and the `TARGET_HEADING` (final heading of the gondola) from the keypad on the gondola.

`KP` is in charge of controlling the speed for the Gondola to move to `TARGET_HEADING` , If `KP` is large, the Gondola will move to the `TARGET_HEADING` fast. However the larger the `KP` , the larger the chance that Gondola will go over `TARGET_HEADING` . Since the turntable does not have a lot of friction, the Gondola will not be stopped when it reaches the `TARGET_HEADING` . Therefore, the Gondola will perform an oscillating motion around the desired heading, and if the speed is too big, completing more than an half-circle motion, the gondola will lose control and just keep spinning.

The solution to this problem is to use the derivative control, which is only designed to slow down the current motion. By setting the proper `KD` , we can slow down the Gondola when it is reaching the `TARGET_HEADING` , and therefore making it stop at the `TARGET_HEADING` without passing the limit. Here is an example of our PD control method for the tail fan:

```
TAIL_FAN_PW = NEUT_PW + KP * (CURRENT_HEADING_ERROR) + KD * (
   CURRENT_HEADING_ERROR - PREVIOUS_HEADING_ERROR);
```

2 RESULTS, ANALYSIS & CONCLUSIONS

2.1 DESCRIPTION OF GONDOLA PERFORMANCE

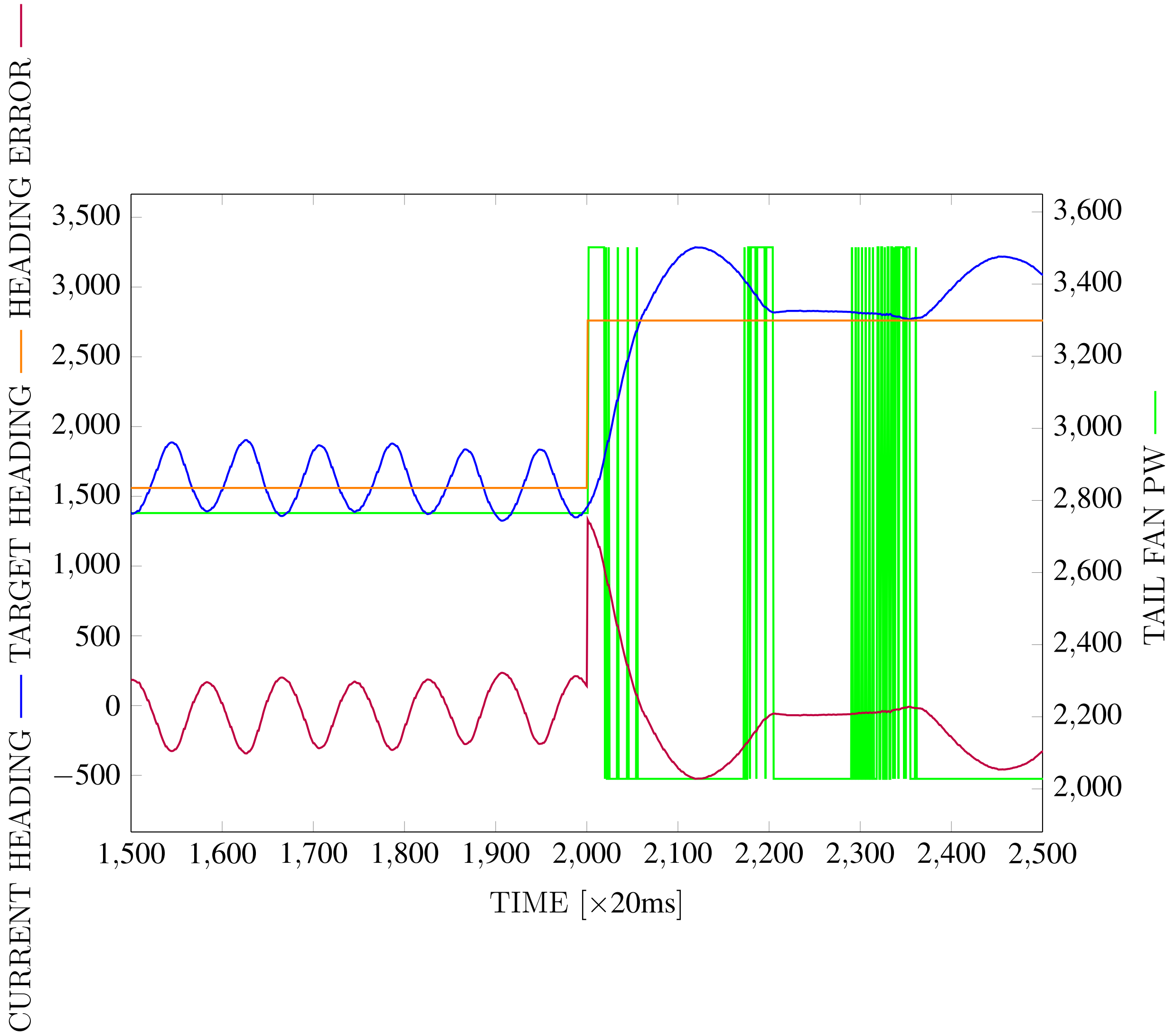
The gondola show performs under damp when `KP` is too big or `KD` is not big enough, performs over damp when `KP` is not big enough or `KD` is too big and performs critical damp when `KP` and `KD` are suitable values.

2.2 VERIFICATION OF PERFORMANCE TO SPECIFICATIONS

The gondola performed exactly as we expected. If we set the `KP` too large and `KD` too small, gondola will perform under damp motion and start spinning. If we set the `KP` too small and `KD` too large. Gondola will perform over damp motion which will take it a long time to reach the `TARGET_HEADING` . If we set the `KP` and `KD` to the proper value, gondola will perform suitable damp motion which is the optimal solution.

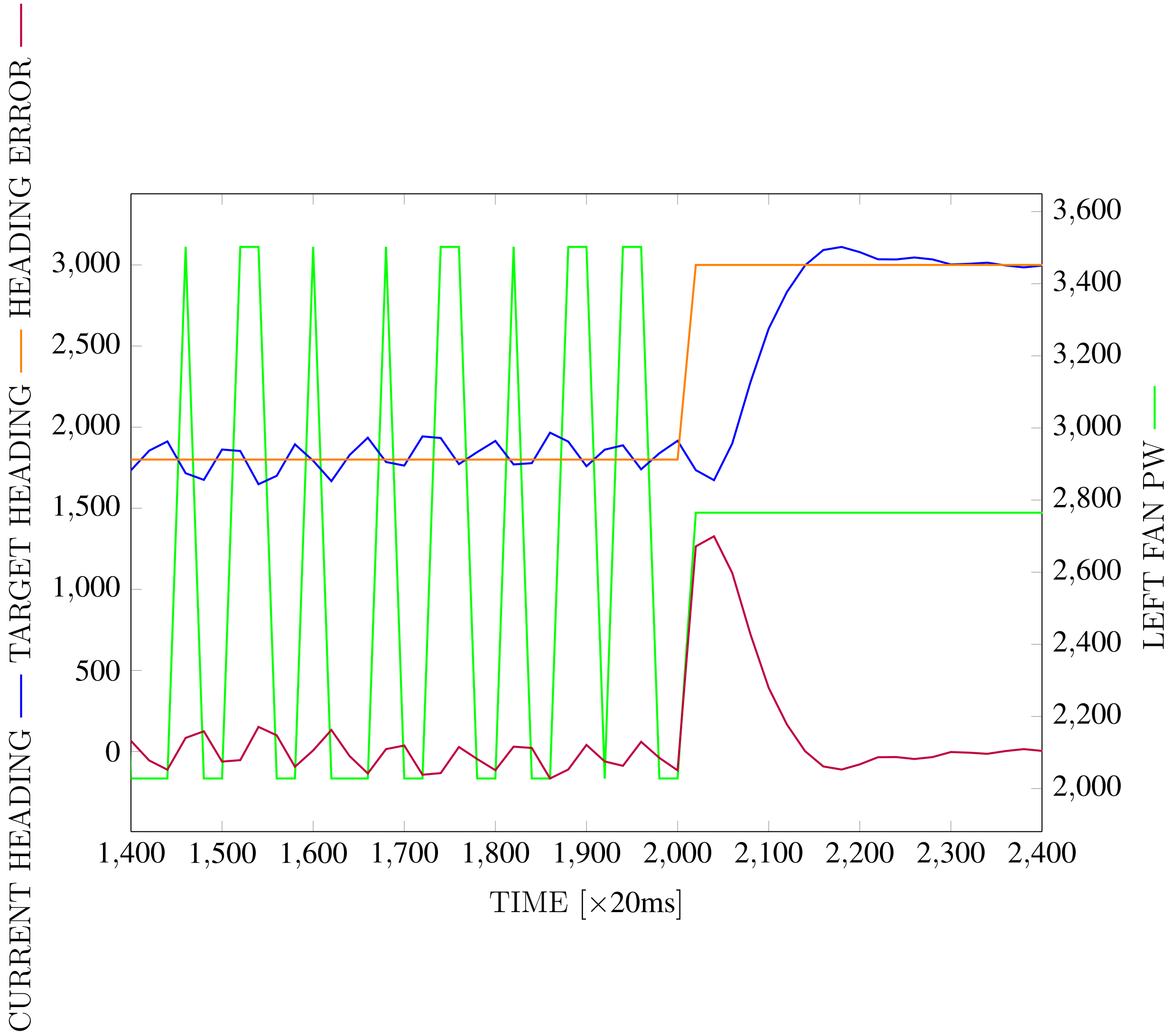
2.3 ANALYSIS OF PLOTS FROM DATA

2.3.1 OVER DAMPED (`KP = 2000` , `KD = 30000` , `STAGE` FROM 2 TO 3)



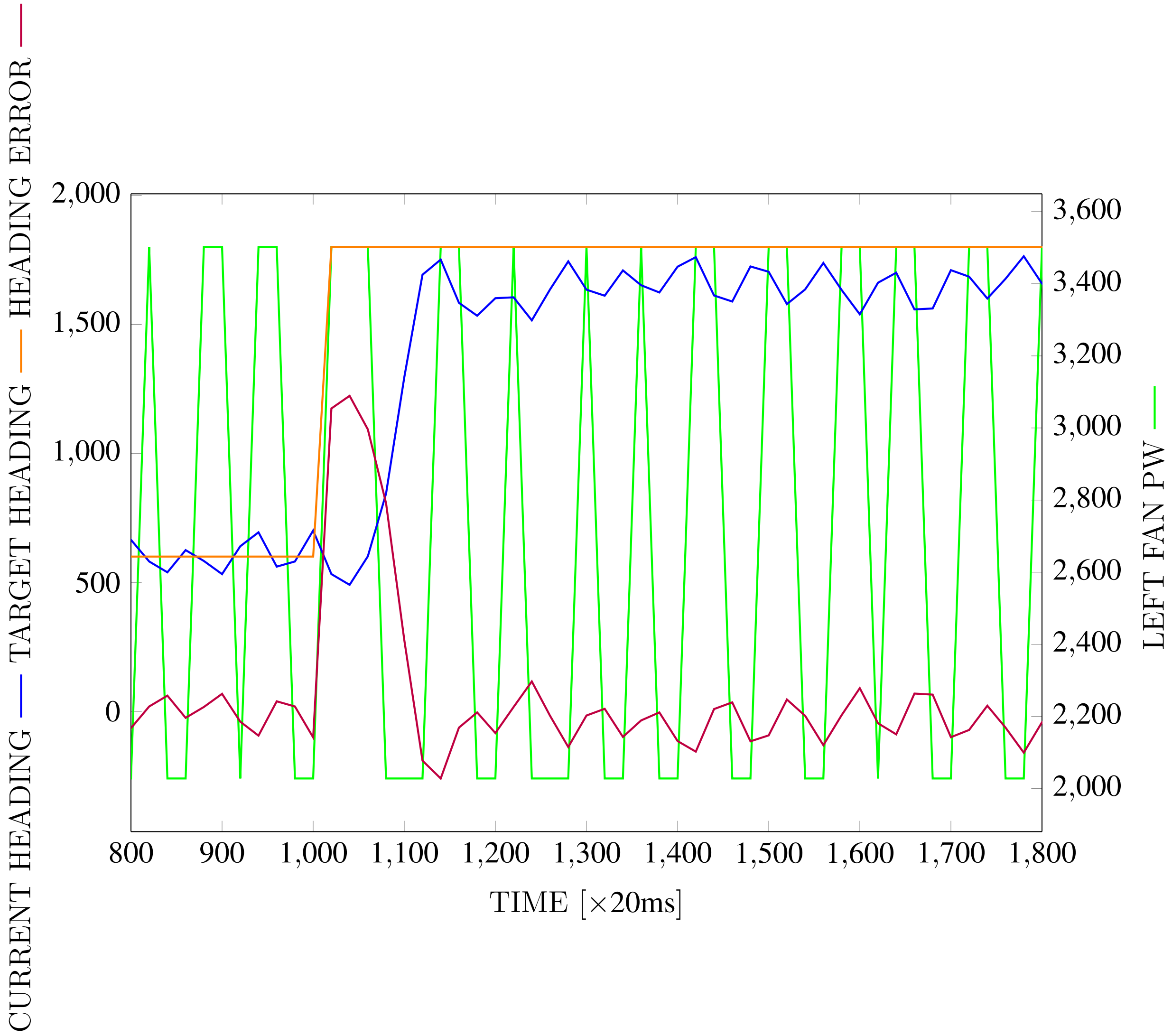
This is an example of an over damping of the gondola. When the `TARGET_HEADING` changes from previous heading to another heading, `TAIL_FAN_FAN` changes from `NEUT_PW` to another value. However, because the `KP = 1000` and `KD = 30000` , the gains lead to an over dump. Therefore, the `CURRENT_HEADING` exceeds the `TARGET_HEADING` and the gondola first passes through the designated heading and then the tail fan reverses to fix the heading.

2.3.2 OPTIMIZED (`KP = 1000` , `KD = 30000` , `STAGE` FROM 2 TO 3)



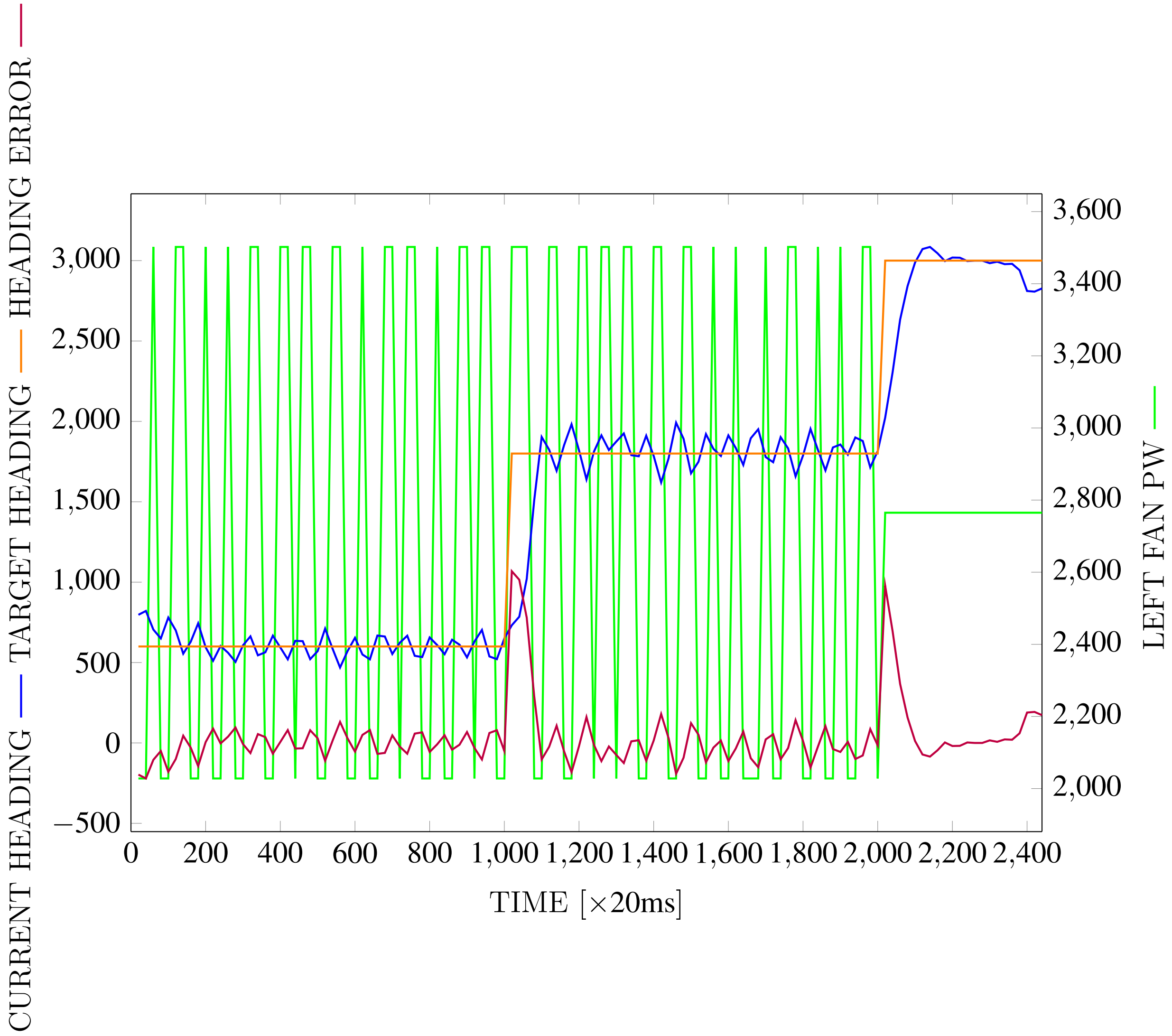
This plot is an example of an suitable damping. In this plot, the `CURRENT_HEADING` is about the same as the `TARGET_HEADING` after the heading was changed. This means that the `KP` and `KD` is suitable so that the damping is about zero.

2.3.3 UNDER DAMPED (`KP = 500` , `KD = 30000` , `STAGE` FROM 1 TO 2)



This plot is an example of an under damping. In this plot, the `CURRENT_HEADING` is below the `TARGET_HEADING` after the heading was changed. This means that the `KP` is too small for the gondola to get to the suitable direction.

2.3.4 ANOTHER PLOT (`KP = 500` , `KD = 30000` , `STAGE` FROM 1 TO 3)



This plot is an example of a full angle transition.

2.4 WHAT WAS LEARNED

In this Lab, a type of data called long was used to calculate pulse width. This types of data is able to store a larger range of integer comparing to the data types learned before, which is from -2^{16} to 2^{16}. The corporation between KD and KP is another core subject in this lab. In the function used to calculate the pulse width. The codes are here:

```
TEMP_PW = NEUT_PW - (long)KP * CURRENT_HEADING_ERROR - (long)KD * (
    CURRENT_HEADING_ERROR - PREVIOUS_HEADING_ERROR);
```

KP is used to control the range of rotation and KD is to control the error of rotation to prevent from overturning.

We also learnt that the importance of using simulation. At the beginning of the Lab, we forgot that there was a web page to show some simulation for different KP and KD. We then encountered large problems when trying to find the correct gains. After we found the web page, we were capable to find the corresponding gains without actually typing them from the keypad, which really saves some time for us.

2.5 PROBLEMS ENCOUNTERED & SOLUTION

Several problems were encountered and we solved all of them at the end of the Lab:

2.5.1 PROBLEMS ENCOUNTERED

1. The gondola rotated so violently, that it did not stop after meeting the target heading. Instead, it turned over the limit of 180° and continued rotating.

2. After the online simulation, a new pair of KP and KD was used in reality testing, but result is significantly different.

3. During testing, one side of the fan run without commands now and then, and stops working when running the code.

4. During testing, the gondola restarts sometime after running the code, following with fan running slow or stopping or dim keypad LCD screen.

2.5.2 SOLUTION

1. `KP` is too large while the `KD` is too small. `KD` did not limit the rotation speed or error when moving near the target heading. The solution was to find another pair of appropriate of `KP` and `KD` , and the simulator online was used to find them.

2. In simulation, the right side fan and the tail fan was set to share a same value of PW, while the left fan has another, that

   ```
   PW_LEFT = 2 * PW_NEUT - PW_RIGHT/TAIL;
   ```

   In the coding, however, PW of the tail fan was set to be same as the left side fan, which means now,

   ```
   PW_LEFT/TAIL = 2 * PW_NEUT - PW_RIGHT;
   ```

   Simulation that setting LEFT and TAIL fans share equal PW was done. It provided evidence to this mistake and helped correct the code.

3. It was certain that the gondola has some mechanic problems according to TA's respond. A new gondola was used to test from then.

4. After providing feedbacks to TA, who provided a new fully charged battery, it was certain that the gondola met an electricity problem.

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.

- Yilu Zhou
-
-