Laboratory 6 Report

Embedded Control, Rensselaer Polytechnic Institute, Summer 2019
Daniel Alegria, Ankur Singh, and Martin Xu
Section 1, LITEC Side B
6-27-2019

Introduction

Purpose/Objectives

The main focus behind this lab was for the team to gain more experience with applying the concepts learned throughout Embedded Control by having to write code in order to control a blimp gondola mounted on a turntable. Through this lab, the team learned about the limitations of proportional control, and how derivative control can help overcome these limitations, as well as reviewed how to read data from SMBus devices, and control motors through pulsewidth changes.

Overview of Gondola Feedback Control

The gondola works by implementing a closed-loop feedback system that inputs readings from a compass into a PD (proportional/derivative) equation to set motor pulsewidth. After the desired heading is defined, the error—the difference between the desired and actual heading—is calculated. The error is then multiplied by a proportional gain to set the pulsewidth of the fans; a larger error means a larger pulsewidth, and thus a faster fan speed and faster turning. A negative error will run the fan in the opposite direction.

However, because the gondola turntable has little friction, the gondola has a lot of momentum, and the fans are quite powerful, there is a large chance that the gondola will overshoot the desired heading as it attempts to turn towards it, and either continuously sway around the desired heading, or complete the circle and loop back, just to overshoot again.

This problem can be solved with adding a derivative part to the equation. The previous error is recorded, and as the difference between errors becomes smaller, the fans will also slow down accordingly. This makes the gondola approach the desired heading more cautiously, limiting overshoot and sway.

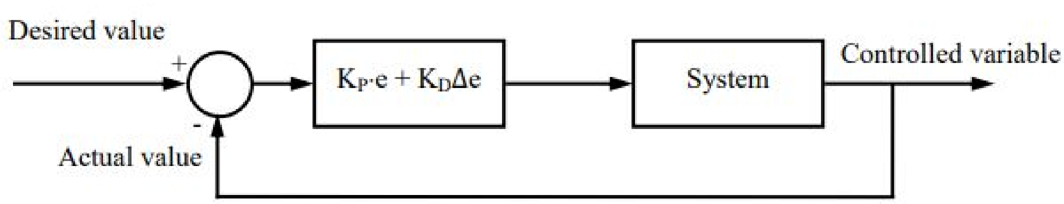


Figure 1: PD control diagram, Source: Litec Manual

An illustration of the control loop is given above, in Figure 1. Desired value represents the desired heading, and the actual value represents the actual heading, or the current heading at some time. The feedback system reads the desired value, and uses the proportional gain and derivative gain equations (displayed in the first box in the figure) to bring the system to the desired value/desired heading.

Results, Analysis, & Conclusions

Description of Gondola Performance

Proportional Gain: 10
Derivative Gain: 20
Desired Heading: 0°
Heading Increment: +120°

Stage 1:
Initial error: 50°
The gondola ran all three fans, with the left fan in reverse, and passed 0° after 1 second. It overshot by 5°, and swayed around 0° 2 times before settling after 5 seconds.

Stage 2:
Initial error: 120°
The gondola ran the two side fans, with the left fan in reverse, and passed 120° after 2 seconds. It overshot by 2°, and swayed around 120° 4 times before settling after 15 seconds.

Stage 3:
Initial error: 120°
The gondola ran just the rudder fan, and passed 240° after 2 seconds. It overshot by 50°, and swayed around 240° 2 times before settling after 10 seconds.

Verification of Performance to Specifications

The required specifications are as follows:

When a run/stop switch is off, the user can:

>input a proportional gain value, a derivative gain value, and a desired heading with a keypad.

>adjust the tilt of the side fans through the laptop keyboard.

>choose with a switch whether the 2nd and 3rd stages of the run add or subtract 120° to the desired heading.

When the run/stop switch is on, the gondola:

>turns toward the desired heading with all three fans: the rudder, and the two side fans.

>20 seconds after the run started, the gondola will add or subtract 120° (depending on the predetermined switch) to the desired heading, and turn towards that direction with just the two side fans.

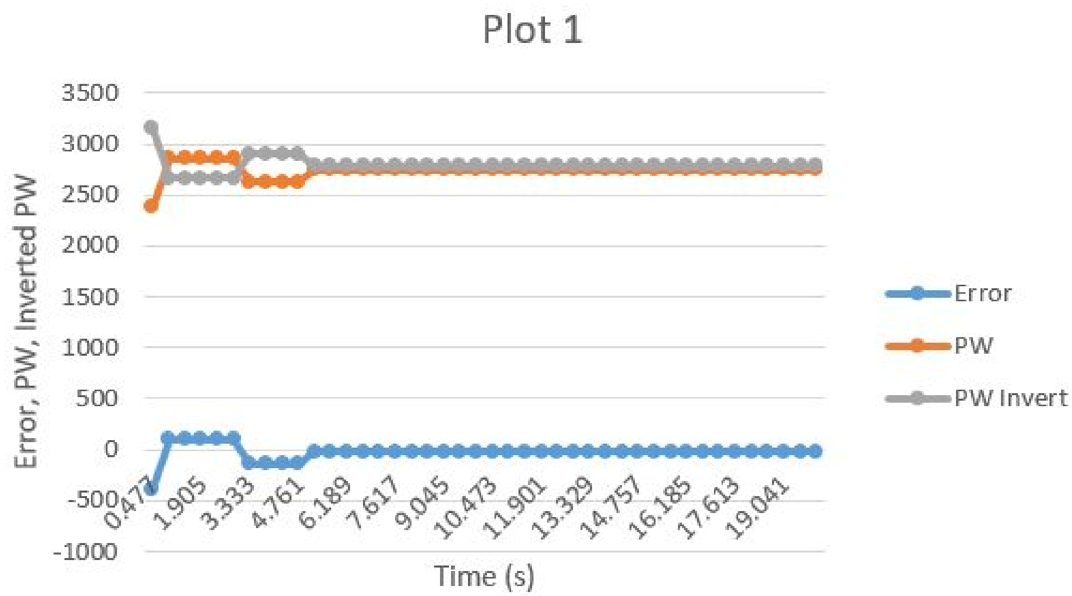
>another 20 seconds later, the gondola will again add or subtract another 120° to the new desired heading, and turn towards it with just the rudder fan.

In testing, the team successfully inputted a proportional gain of 10, a derivative gain of 200, as well as a desired heading of 0°, through the keypad. The tilt of the side fans was also successfully set to be perpendicular to the floor. The heading increment switch was set to +120°.

When the run/stop switch was turned on, the gondola immediately turned to 0° with all three fans on. When the gondola was turned to face away from 0°, the farther the gondola was away from 0°, the harder the fans ran. However, when the gondola was released, it turned back to the desired heading with little overshoot; it swayed around 0° perhaps 2 or 3 times before settling. The time from start to settling was around 5 seconds.

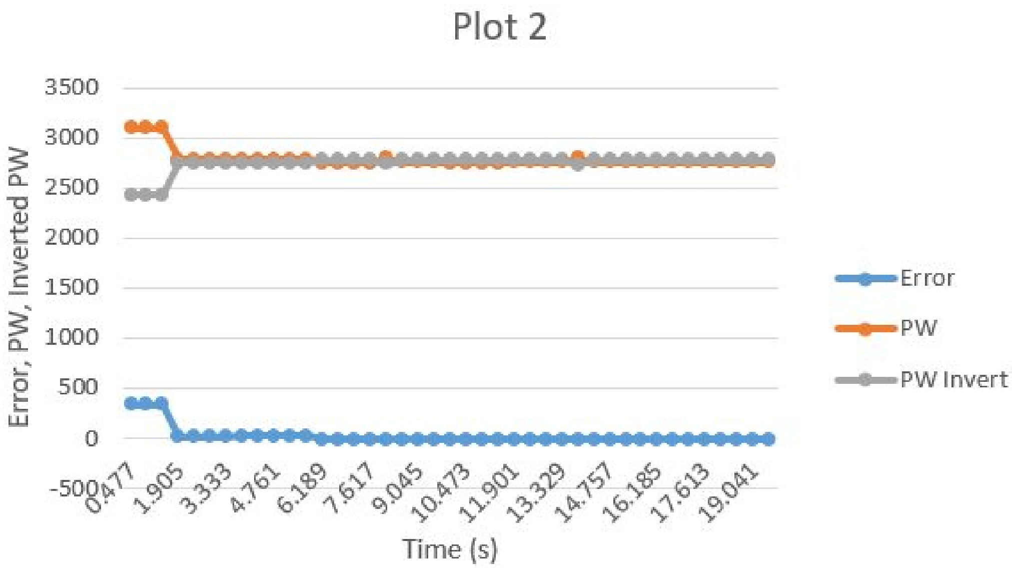
The 2nd and 3rd stages changed the desired heading to 120° and 240°, respectively, as expected. The gondola also faced the correct direction using the correct fans in both stages.

Analysis of Plots from Data



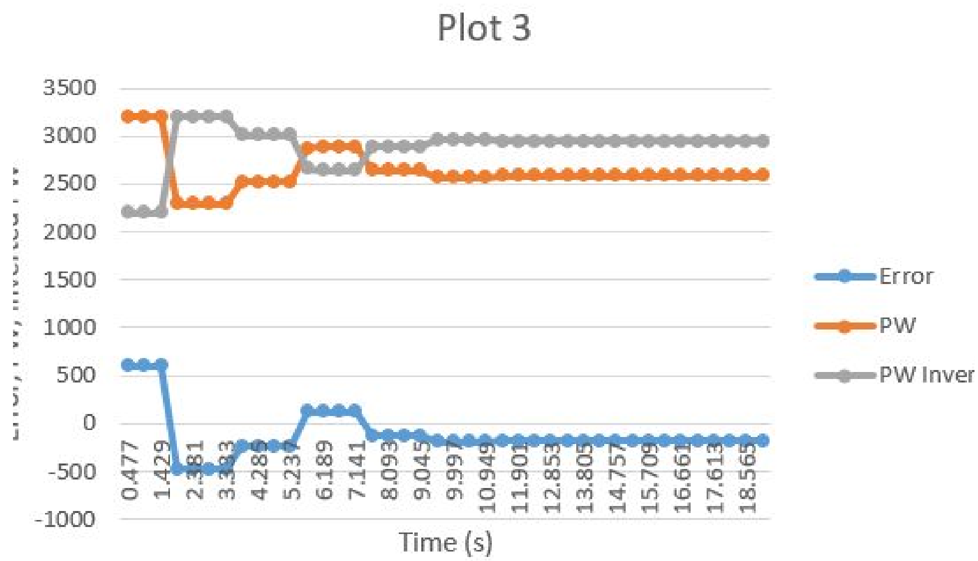
Plot 1: Proportional Gain = 10, Derivative Gain = 20, Desired Heading = 0°

This is the first stage of run 1; all three fans are running, with the left fan running in the opposite direction of the right and rudder fans (using PW Invert instead of PW). The gondola started off about 50° from the desired heading of 0°, and thus all three fans ran at full throttle. After swaying around 0° twice, the gondola steadied out—error reached 0, and the pulsewidths reached their neutral state.



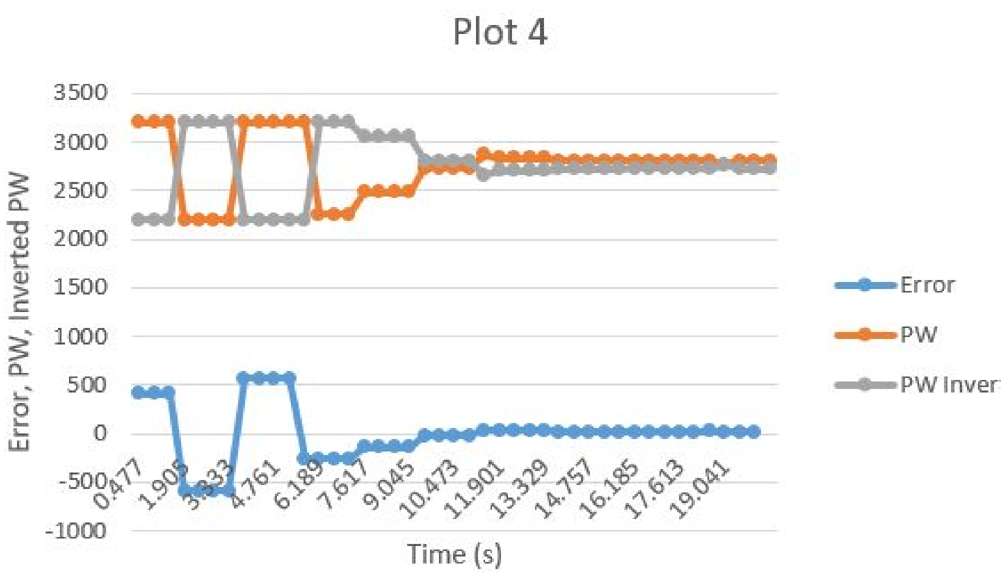
Plot 2: Proportional Gain = 10, Derivative Gain = 20, Desired Heading = 120°

This is the second stage of run 1. Because the switch was set to increment desired heading by 120°, desired heading changed to 120°, from the initial of 0°. Only the two side fans are running. Compared to the first stage, the gondola steadied out sooner, with much smaller swaying, possibly due to the two side fans giving just enough power to position the gondola correctly; having all three fans running gave the gondola too much momentum, causing it to overshoot a little more.



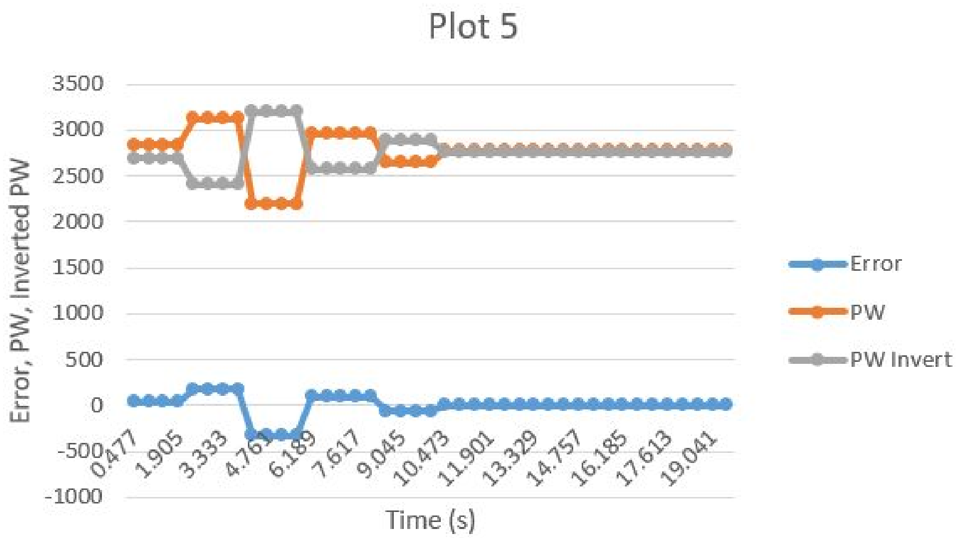
Plot 3: Proportional Gain = 10, Derivative Gain = 20, Desired Heading = 240°

This is the third stage of run 1, with only the rudder fan running. This stage showed the most oscillation about the desired heading, possibly due to the rudder fan not being powerful enough to counteract the momentum of the gondola. This lack in power can also be seen in the fact that the error does not reach 0, and that the pulsewidths do not converge to their neutral state; when the gondola was close to the desired heading, but not on it, the single fan was not enough to move it all the way.



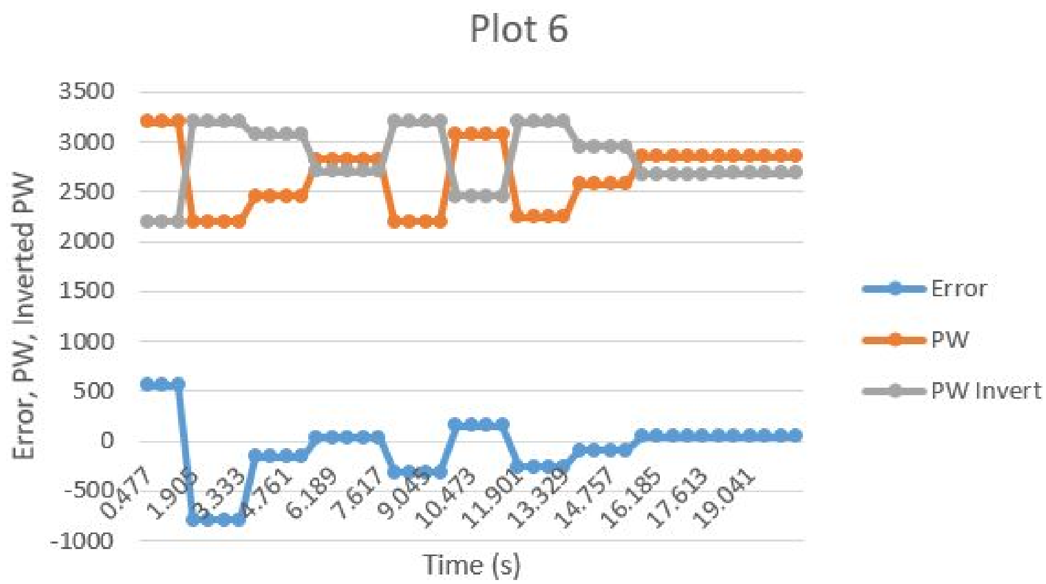
Plot 4: Proportional Gain = 20, Derivative Gain = 40, Desired Heading = 0°

This is the first stage of run 2, sporting higher gain values than run 1. These higher gain values are the primary cause of the relatively longer time it takes for the error to approach zero. Here the three fans were running, and due to the combination of all three, there are many oscillations about the desired heading before it points accurately.



Plot 5: Proportional Gain = 20, Derivative Gain = 40, Desired Heading = 120°

The second stage of run 2 was with just the side fans running, and the lack of a rudder fan made the gondola rotate at a lesser speed which impacted its ability to point in the desired direction. The two fans converge to the desired direction about as fast as all 3 fans; however, it did that with fewer oscillations. At the beginning of the graph, the error is close to 0, but goes up again due to the high thrust of the gondola at the beginning of the run, an anomaly that doesn't appear again.



Plot 6: Proportional Gain = 20, Derivative Gain = 40, Desired Heading = 240°

The final part of run 2 is just the rudder adjusting to the new desired direction, 240 degrees away from the original heading. Because it is just the rudder turning, the lack of two side thrusters that act as stabilizers gave the rudder coarser control. This lengthened the time it took to get centered because whenever the rudder inverts its spin, the gondola keeps spinning due to the single fan not being strong enough to counteract momentum.

What Was Learned

By completing this lab, the team was able to learn a lot. Not only did completing this lab assist the team in gaining more intellectual knowledge, but by completing this lab the team was also able to learn and discover more about their own abilities. Completing this lab gave the team more confidence in themselves and made them realize that they too have the ability to solve complex engineering problems. Before this lab and even before this class the team never thought they could use code to control such complex systems. Though after doing this lab and completing the lab by the application of the concepts taught in class, the team was able to discover more about their abilities and gain more confidence in their own engineering abilities and intellectual potential.

The team also gained more in-depth knowledge and experience with proportional control, derivative control and how closed loop feedback systems operate. The team was able to gain more insights behind how systems which operate based off some input data are coded and designed in terms of software operation.

Problems Encountered & Solutions

One problem the team first encountered when doing this lab was actually understanding the concept of derivative gain. The team was unsure how this concept can be applied to help meet the requirements of the lab. On top of that the team did not have a proper understanding of this concept. The team did read the Lab Manual and paid attention during lecture, but still could not grasp the concepts. The team solved this problem by using the help of the teaching assistants. Teaching Assistants Syed Naqvi, Weston Fong and Neelanga Thelasingha were very helpful as the team went over the concepts with them and with their help were able to grasp a firm understanding. The Teaching Assistants helped the team understand these concepts by drawing out diagram and explaining step by step what proportional and derivative control were accomplishing in the labs and how their application was necessary and useful. The team was very thankful for this, as without a firm understanding of these concepts it would not be feasible to implement the correct code in the time frame given to complete this lab.

Another problem the team encountered was that not all of the fans turned on one of the gondolas. The pulsewidth equations worked, as the rudder fan worked, and turned towards the desired heading quickly and accurately. However, the left fan propellor was loose, and the right fan was disconnected.

After fixing these hardware errors, the right fan still did not turn, despite being sent the same pulsewidth as the other two fans. Simply moving to another gondola solved this problem.

Other than these two problems, there weren't any other major issues. The P/D pulsewidth equations, along with the inputted gains, worked surprisingly well the first time the gondola was run. This can be attributed to the gondola simulation application, which helped to keep trial-and-error times down.

Academic Integrity Certification *(this part is required exactly as stated)*

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)

Hardware implementation:
(wiring & pin-out sheet)

Daniel Alegria 100 (pin out)
Ankur Singh
Martin Xu

Software implementation:
(pseudo-code & code)

Daniel Alegria 45
Ankur Singh 10
Martin Xu 45

Data analysis (if relevant):

Daniel Alegria 33
Ankur Singh 33
Martin Xu 33

Report development & editing*:
(schematic, diagrams & plots)

Daniel Alegria 33
Ankur Singh 33
Martin Xu 33

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

Daniel Alegria
Ankur Singh
Martin Xu

*Note, notebook keeping and report development/formatting do not constitute an engineering contribution toward successful laboratory completion.