Lab 5: Bench-Top Testing of Blimp Hardware and Software

Preparation

Reading

Lab Manual

Chapter 7 - Control Algorithms

RPILMS

Gondola Info

Objectives

General

1. Port the heading (steering) and altitude (speed) control systems developed in the previous lab to the *Blimp gondola*. *Blimp* flying time is limited so control code is tested using a gondola mounted on a turntable in the classroom. The turntable has a fan mounted to simulate the tail fan of the *Blimp*. The gondola is mounted upside down in the studio just to allow variable ranger readings.

Hardware

1. Recognize the similarity between the thrust fans and tail fan on the *Blimp* to the car drive motor and steering servo, respectively.
2. Become familiar with the layout and design of the gondola and the sensors placed within it.
3. Use the LCD display and number pad to set 1) desired heading, 2) proportional and derivative gain for the heading control loop, and 3) set the proportional and derivative gain for the altitude control loop.

Software

1. Transfer the integrated code used on a *Smart Car* from Lab 4 and apply it to gondola of the *Blimp*. Use the measurements from the ranger to control the thrust fans and hence the altitude (analogous to the drive motor on the *Smart Car*). Use the measurement from the compass to control the direction of the tail fan and hence the yaw (analogous to the steering servo on the *Smart Car*). You would need to modify the code to utilize inputs.
2. Implement proportional and derivative (P&D) control based on the measurements from the ranger and compass. Note that you may change control gain constants by altering DIP switch states after the code had been downloaded, but using the keypad and LCD display is much easier and is **required** for this lab.
3. Read the number pad to set the desired heading and gain constants.

Motivation

Now that the integrated software has been tested on the *Smart Car*, we will make slight modifications to the code so that the code can be run on the *Blimp*. The simplest *Blimp* control is Hover Code, where the embedded controller causes the *Blimp* to maintain a constant altitude and a constant heading. The electronic compass is used to read the heading, it is compared to a desired heading and the control algorithm sets a pulse width for the tail fan. The *Blimp* uses a speed control module to control the tail fan. That module uses PWM signals that are only slightly different than those for steering servo of the car. In addition the *Blimp* requires proportional plus derivative control, P&D. Therefore, with respect to the heading control, code developed to steer the car based on the compass reading can be ported to the gondola on a turntable with only two changes, the PWM range and the control algorithm. The desired heading and the gain constants are set using the LCD display and the number pad.

Concurrent with the need to control the heading of the *Blimp*, is the need to control the altitude. The *Blimp* is propelled by two thrust fans. The thrust fans have two control features, the thrust angle setting and the thrust power setting. The fans are mounted on a shaft that can be rotated by a servo to change the thrust angle.

The simplest altitude control is to set the thrust fan angle so that the fans are horizontal and the thrust is vertical, and leave it there. In this situation, the altitude can be controlled just by adjusting the power to the fans. If the *Blimp* is near the floor, the fan power should be set to maximum. If the *Blimp* is at the desired distance from the floor, the power level should be set low. And if the *Blimp* is too far from the floor, the thrust should be reversed. To do this automatically, a P&D controller for altitude is required for the *Blimp*, but it can't be fully tested using the gondola on the turntable. A more complete test will happen in Lab 6 using the actual *Blimp*. Nonetheless, this lab requires the development of the P&D controller. Therefore, with respect to altitude control, the code developed to control the car speed can be ported to the gondola on a turntable with basically two changes: add code that sets the angle of the thrust motors and implement a P&D controller for the thrust power. Once again, the control gains are to be set using the LCD display and the number pad.

The gondola has a radio frequency link to allow telemetry data to be sent from the autonomous *Blimp* to your laptop. This is used to record the system response data. The RF (radio frequency) link appears as another serial port, but requires a new driver. The Gondola Info link on LMS provides the details.

Lab Description and Activities

To the extent possible, the ranger pair and the compass pair in a team for Labs 3 and 4 should reverse their roles in Lab 5 and 6. That is, the earlier compass pair will now be a ranger pair, and vice versa. But beyond this specialization, you will be confined to work as a team.

Summary - Compass Pair

1. Implement P&D control algorithm.
2. Change pulse width range to match speed controller.
3. Test on turntable with different values of gain constants.
4. Record the actual heading vs. time, plot results.

Summary - Ranger Pair

1. Add code to set PMW signal for thrust angle. Code must allow the user to adjust the pulsewidth to achieve up/down thrust.
2. Implement P&D control algorithm for thrust power
3. Demonstrate that the code has the correct type of response by moving your hand above the range sensor.

Summary - Joint effort

1. Use the LCD display and number pad to set desired heading, heading proportional gain constant, heading derivative gain constant, altitude proportions gain constant, and altitude derivative gain constant. Desired altitude can be fixed but should be 50 cm for Lab 5 and 150 cm for Lab 6.
2. One combined printf statement that include desired heading, actual heading, desired altitude, and actual altitude. This should be printed in columns to allow process using Excel or MATLAB.

Hardware

A gondola placed on a turntable is located in the LITEC studio. This will be used to emulate an actual *Blimp*. A ranger sensor, a compass sensor, C8051, LCD display with number pad, and all the necessary wiring is already installed within the gondola. Familiarize yourself with the placement of the hardware components inside the gondola. One of the major differences between the *Blimp* and this mock-up is that the ranger is still facing upwards while on the *Blimp* it will be face downwards. You will be downloading your code onto the C8051 within the gondola.

Software

Although pairs were allowed to choose a capture compare module (CCM) in Labs 3 and 4, the CCMs used by the ranger and compass are preset. They will be specified by the instructor in class. Modify your functions so that the pulsewidth modulated signals based on the ranger and compass readings are output to the appropriate CCM.

Since the hardware in the gondola has already been implemented, the Port pin connections are fixed. The gondola uses 4 Capture/Compare Modules with CEX outputs on Port pins P1.0, P1.1, P1.2 and P1.3. The crossbar setting is then

```
XBR0 = 0x27;
```

For the compass, allow the user to set a desired heading at the start of your program. The code uses that desired heading and the measurements from the compass to implement a proportional plus derivative control algorithm. At first, set the derivative gain to zero, so that it is purely proportional control. The control algorithm should modify the duty cycle of a PWM signal sent to the tail fan. Limit your pulsewidth signals to be between 2000 and 3500 pulses, as the actuator is now a fan powered by a speed control module.

Thereafter implement a proportional and derivative control algorithm. Test different values for the proportional and derivative gains. Print the values of actual heading that will be plotted later using Microsoft Excel or MATLAB.

For the ranger, set the desired altitude to 50 cm for this lab, and 150 cm for Lab 6. Use the desired altitude and the actual altitude measurement from the ranger to implement a proportional plus derivative control algorithm. The control algorithm should modify the duty cycle of a pulse-width modulated (PWM) signal sent to the thrust fans. Limit your pulsewidth signals to between 2000 and 3500, as was done in the previous labs.

Test one or two values for the proportional and derivative gains. The response of the altitude correction through the control algorithm cannot be fully tested with the mock-up of the *Blimp*. This portion of the code will be fully tested when we move to the *Blimp* in Lab 6. For the mock-up, the code should demonstrate that the thrust motor speed change with the distance reading.

In addition to implementing the thrust motor speed control, the thrust angle must also be controlled. This requires using an additional capture compare module, CCM, to send PWM signal to the thrust angle servo. The code is simple because the PW doesn't need to change automatically. It is set once. Create a function that allows the user to adjust the pulse width to the thrust angle servo to achieve up/down thrust. A code similar to the steering calibration of the car is appropriate.

Lastly, obtain A/D conversion on Port 1 Pin 5, which provides the voltage of the battery on the *Blimp*. Print it in the format shown in the next page.

Data Acquisition

When your code is functioning correctly, gather data to plot response curves for your steering control. You should plot error (y-coordinate) vs. time (x-coordinate). In order to save the data, you will need to print the error to the SecureCRT screen and then copy the output to a plotting utility, such as MatLab or Excel. You need to obtain several curves for different gain combinations. The following settings are required, however you also must also find an 'optimal' setting. You should also look at other combinations to verify trends. Testing only the following represents a minimum effort and that will be considered when the second report is graded.

Without differential gain

Case 1:	kp =	0.1,	kd =	0
Case 2:	kp =	5,	kd =	0

With differential gain

Case 3:	kp =	0.1,	kd =	10
Case 4:	kp =	0.5,	kd =	70
Case 5:	kp =	3,	kd =	70
Case 6:	kp =	3,	kd =	180
Case 7:	kp =	12,	kd =	70
Case 8:	kp =	12,	kd =	180

Lab Check-Off: Demonstration and Verification

1. Complete the entries in your lab notebook and present it to your TA.
3. Explain to the TA how P&D control works on the gondola.
4. Show the TA the error vs. time graphs for the eight required cases and your optimal settings. Good performance is indicated by a short rise time, low over shoot and short settling time.
5. Demonstrate how the orientation of thrust motors' axis changes with changes in pulse width of the PWM signal to thrust angle servo.
6. Set a desired value for the altitude. Show that varying the distance to an object causes the thrust fans to operate in forward or reverse direction depending on the distance.
7. Your TA may ask you to explain how sections of the C code you developed for this exercise works. To do this, you will need to understand the entire system.
8. Print the output in the in the HyperTerminal screen in the following format

```
Actual: heading - Altitude - Battery Voltage
         xxxx       xxx            xxx
         xxxx       xxx            xxx
         ....       ...            ...
         xxxx       xxx            xxx
```

   Capture and print the screen and attach it to your lab notebook. Note that the "distance" is the reading of the Ultrasonic ranger which might be random (depending on the objects between the turntable and the ceiling of the room). Desired heading and desired altitude should be printed, but it is your option to print these just once at the beginning of the data run or as separate columns.

Writing Assignment - Lab Notebook

It is recommended that the team still use two lab notebooks to allow entry of discussions, tests and results. But only one lab notebook will have to be submitted later. All relevant code, schematics, test results and comments must at some point be transferred into the official team notebook. Flexibility will be allowed here. If a team can justify submitting two lab notebook rather than one, it will be allowed. In this case, it will be necessary to provide the grading TA a roadmap of which activities are included in which lab notebook. The separation should be logical, heading work in one and thrust work in the other for example. Both must have the full code and schematic.

Enter the full schematic of the circuitry. You must show the pins (and hence which CCM) that are connected to the components on the gondola. You know enough to create the schematics even though you don't do the wiring. The inputs use the SMBus, the outputs use the PCA. The power

modules and the thrust angle servo are black boxes with three wires; input, power and ground. The DIP switch is connected to Port 3.

Print and attach the full code. Describe in your lab notebook any modifications you made to the code from Lab 4 to allow it to function on the gondola. Make comments about the performance (heading and altitude) with both high gains and low gains.

For each set of gains, print the actual and desired headings. Use the capture command within HyperTerminal to record these values. Copy and paste the values into an Excel file and graph the pulsewidth modulated as signals over an interval of 15 seconds. Title these graphs and place them in your lab notebook. Also place the data sets used to generate the graphs.

Writing Assignment - Final Design Report

You will be submitting a single design report detailing the entire operation of your *Blimp* except Lab 6. This will be submitted following the completion of Lab 5. While you are working on this report, you are required to complete Lab 6 and then create an addendum to the Lab 5 blimp report. The addendum includes the results of Lab 6. Detailed information about writing the final report can be found in the sections *Embedded Control Design Report format* and *Final Design Report Guidelines*.