Lab 6: Bench-Top Testing of Gondola 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 Lab 4 to the *Blimp gondola*. The *Blimp Gondola* mounted on the turntable in the classroom replicates all the flying and control characteristics of the actual blimp. 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 *Gondola* 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. The wireless radio frequency (RF) link permits communication between the *Gondola* and your laptop. Use the LCD display and number pad or SecureCRT terminal 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, if it is used.

Software

1. Transfer the integrated code used on a *Smart Car* from Lab 4 and apply it to the gondola of the *Blimp*. **Traditionally** (but no longer) the measurements from the ranger is used to control the thrust fans and hence the altitude (analogous to the drive motor on the *Smart Car*). The **enhanced mode introduced here** will use the ranger to adjust the desired heading. As normal, the measurement from the compass with the desired heading controls the spin direction of the tail fan and hence the yaw (analogous to the steering servo on the *Smart Car*). You will need to modify the code to utilize the ranger input to modify the desired heading.
2. Implement proportional and derivative (PD) 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, or by using either the keypad and LCD display or

SecureCRT terminal. The lab requires gains to be entered with menu prompts to either the LCD or SecureCRT, however using SecureCRT is probably more flexible and easier for operators to use.

3. Read the number pad or SecureCRT terminal to set the initial desired heading, but at some point the ranger should also be used to modify the desired heading while the program is running. Raising or lowering a hand over the ranger around a nominal neutral height will change the desired heading of the gondola.

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 *Gondola*. 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 actual heading. This is compared it to a desired heading and the control algorithm sets a pulse width for the tail fan based on the error. 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 feedback 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 number pad on the LCD display/keypad module or keyboard input on SecureCRT.

Normally, 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 values, 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.

Traditionally for altitude control the thrust fan angle is adjusted so that the fans are horizontal and the thrust is vertical, and left there. In this situation, the altitude can be controlled just by adjusting the pulse width 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 (neutral pulse width). 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 may happen in Lab 7 using the actual *Blimp*.)

The **enhanced control to be used here** instead rotates the thrust fans to be vertical – the same orientation as the tail fan. In this orientation they will triple the force available to rotate the gondola on the turntable and provide a much stronger and faster correction to heading errors indicated by the compass. The 3 fans will be driven equally, but remember that the left and right fans must spin in opposite directions to augment the tail fan. Code must be added that sets and fixes the angle of the thrust motors to vertical and implement a PD controller for the thrust power.

In addition to the 3 fans used for steering, the ranger code must be changed to permit modification of the desired heading. As in the speed control on the car driven on the paper, choose a nominal height (~50 cm) that doesn't change the initial heading value entered when the program starts. Raising a hand above this nominal height will increase the desired heading by up to 180° and lowing a hand decreases it by 180°. A reasonable change of ±40 cm should modify the desired heading by ±180°, but you may experiment to determine more suitable values if desired.

The gondola has a radio frequency link to allow 2-way communication for data between the *Gondola* and your laptop. This may be used to send the desired heading and gain values to the *Gondola*, and also allows telemetry data to be sent from the autonomous *Gondola* to your laptop. The RF (radio frequency) link appears as another serial port, but requires a new laptop driver. The Gondola Info link on LMS provides the details as does the **Drivers for RF link** section in the **Installing_SiLabs-SDCC-Drivers** manual also on the LMS front page. The number on the transceiver plugged into the laptop must matches the number on the gondola to insure both use the same channel.

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 6 (and 7, if applicable). 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 PD 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 and plot results.

Summary - Ranger Pair

1. Add code to set PMW signal for thrust angle. Code must allow the user to adjust the pulse width to achieve sideways thrust (fans in vertical position). Since every gondola will have a different thrust angle when the controller is in the neutral position, your code should include a routine that allows you to manually adjust the angle so the fans are vertical when initializing everything.
2. Modify the code reading the ranger data to adjust the desired heading angle.
3. Demonstrate that the code has the correct type of response by moving your hand above the range sensor and printing out the desired heading.

Summary - Joint effort

1. Use the LCD display and number pad or RF link to set initial desired heading, heading proportional gain constant, and heading derivative gain constant. The nominal altitude that doesn't adjust the initial heading can be fixed but should be ~50 cm for Lab 6.
2. Use one combined printf() statement that includes desired heading, actual heading, ranger reading, heading angle adjustment, thrust pulse width, and battery voltage. The modified desired heading and may be printed less frequently to save time and reduce clutter. This should be printed in columns (separated by commas) to allow later processing and plotting using Excel or MATLAB.

Hardware

Gondolas placed on turntables are located in the LITEC studio. These will be used to emulate actual *Blimps*. A ranger sensor, a compass sensor, C8051, LCD display/keypad, 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 would face downwards. You will be downloading your code onto the C8051 inside 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 pulse-width 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 P0.4, P0.5, P0.6 and P0.7. The crossbar setting is then

```
XBR0 = 0x25;
```

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 and sideways oriented thrust fans. Limit your pulse width 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 desired and actual heading that will be plotted later using Microsoft Excel or MATLAB.

For the ranger, set the nominal distance to 50 cm for this lab. Use deviations from this nominal distance to adjust the desired heading as stated at the bottom of page 2. The control algorithm should modify the duty cycle of a pulse-width modulated (PWM) signal sent to all 3 fans. Limit your pulse width signals to between 2000 and 3500, as was done in the previous labs Remember that the left and right thrust fans must spin in opposite directions. If the right fan is set to NEUTRAL + PW, the left should be set to NEUTRAL – PW.

Once everything is set up and verified, test several combinations 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 if we move to the *Blimp* in Lab 7). For the mock-up, this optional code should demonstrate that the thrust motor speed change with the distance reading when fans are held horizontally for altitude control.

Since, in addition to implementing the thrust motor speed control, the thrust angle must also be adjusted it is suggested that a short start-up function be written that allows manual adjustment of the angle by pressing 2 keys, either on the terminal or keypad, that rotate the shaft clockwise or counterclockwise. A 3rd key is pressed when the angle is set correctly. This requires using an additional capture compare module, CCM, to send a PWM signal to the thrust angle servo. The code is simple because the PW doesn't need to change automatically while running. It is set once during initialization. Since every gondola will have a different default angle setting, create a function that allows the user to adjust the pulse width to the thrust angle servo to achieve sideways thrust. A code similar to the steering calibration of the car is appropriate.

Lastly, obtain A/D conversion on Port 1 Pin 3, which provides the voltage of the battery on the *Blimp*. The value should display at least a tenth of a Volt with correct scaling, but since the LCD doesn't handle float numbers easily, it is suggested values be displayed in mV (4 digits).

Data Acquisition

As a final check of your control calculations you must manually rotate the gondola a full 360° turn and observe the values of the compass reading, the error calculation, the feedback control and the tail fan motor PW calculated by your control value. For this you should switch off the fan motors using the 3 switched on the side of the gondola since you will only be observing the values. For simplicity set the derivative gain kd to zero. This exercise will validate two items: the accuracy of the electronic compass (you may want to compare values with a mechanical compass) and the correctness of your control calculations. With improper use of long integer calculations in C it is possible that the control values will be incorrectly calculated for certain angles and this exercise will allow you to detect and fix those errors much more quickly. Starting at the desired heading angle, the control should be 0 and increase (or decrease, depending on the rotation direction) until the error reaches 180°. Passing this angle should flip the error and the control value as you continue on back to the desired heading. If any anomalies occur in any values the way the program calculates the results should be checked. If the electronic must be calibrated, do so and then repeat this exercise.

When your code is functioning correctly, gather data to plot response curves for your steering control. You should plot actual heading (y-coordinate) vs. time (x-coordinate). In order to save the data, you will need to print the heading to the SecureCRT screen and then copy the output to a plotting utility, such as Excel or MATLAB. Read the **Terminal Emulator Program** section in the **Installing_SiLabs-SDCC-Drivers** manual on LMS for more details. 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

The new exercise is to explore the improved response for correcting heading error when the tail fan is augmented with both thrust fans that have been rotated 90° to be horizontal. In this mode the 2 thrust fans must be turning in opposite directions to effectively triple the power for turning the gondola on the turntable. You may optionally compare the speed of the response for this modified system to the original by turning off both thruster fans with the 2 slide-switches labeled "LEFT" and "RIGHT" on the side of the gondola. If the fans have been turned off, in order to turn them back on they must once again remain in a neutral position for 1 second before adjusting their speed. This is usually done by first turning the slide-switches back on and then rotating the gondola to point in the

direction of the desired heading for 1 second, which should output the neutral pulse width to the controllers.

Proportional and derivative gains will need to be adjusted to optimize the response. The combination of adding the 2 thrust fans and switching to higher voltage LiPo batteries in the gondola significantly changes the control system characteristics. The gains listed in the 8 cases above, particularly the kp values, may be to large to produce an over damped response. If this is the case you may substitute smaller values to produce the full range of system responses from heavily over damped, through critically damped and under damped, to outright unstable.

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 heading 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. For the graphs in 4. show responses to both forced changes in the gondola's position as the desired heading remains fixed (regulator control) and when the desired heading changes with ranger distance (tracking control).
6. Demonstrate how the orientation of thrust motors' axis changes with changes in pulse width of the PWM signal to thrust angle servo.
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 to the terminal screen in the following format similar to

```
Des. Heading   Actual: Ranger - Heading - PW - Batt. Voltage (mV)
      xxxx,           xxxx,      xxxx,   xxxx,    xxxx
      xxxx,           xxxx,      xxxx,   xxxx,    xxxx
      ....,           ....,      ....,   ....,    ....
      xxxx,           xxxx,      xxxx,   xxxx,    xxxx
```

   Capture and print the screen and attach it to your lab notebook. Note that the "Altitude" 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 decide how to present the data for plotting, since in some cases desired heading doesn't change. PW is the pulse width count of the thrust fans, used to indicate the force of the fans. Graphs should subtract out the neutral count from this value when plotting thrust vs. heading. You may want to plot the controller gains at the beginning of a run to keep the data complete.

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 notebooks 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 control work in one and ranger adjustments to the desired heading 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 or MATLAB file and graph the pulse-width 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 *Smart Car and Gondola* (except Lab 7, if applicable). The report covers everything after the Game Report (Labs 3 – 6). Much of this report may be written while working on Labs 5 and 6. Once Lab 6 is functioning, the report can effectively be completed, except for the response plots that should be added as an appendix. The appendix should include the plots and a brief discussion of response curves. Detailed information about writing the final report can be found in the sections *Embedded Control Design Report format* and *Final Design Report Guidelines*. LMS has a rubric for the report in the Lab 6 section 2.6 under *Lecture & Lab Modules*.