Lab 6: Bench-Top Testing of Gondola Hardware and Software (2019S)

Preparation

Reading

Lab Manual

Chapter 7 - Control Algorithms

RPILMS
Gondola Info

Objectives

General

- 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 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 in Lab 3). The **alternate mode** used here will use the ranger to adjust the desired heading. Compass measurements are used to control the gondola fans in order to align the gondola with the desired heading. You will need to modify the code to utilize the ranger input to modify the desired heading within specification.
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

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

3. Read the number pad or terminal to set the initial desired heading. Use the ranger to modify this value during continuous operation. 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* to perform a different task. This is an exercise in migrating code between different systems while demonstrating the similarities between the control routines and programs in general

Traditional Blimp Altitude Control has the thrust fan angle adjusted so that the fans are horizontal (and left in that position) and the thrust is vertical. In this situation, the altitude can be controlled by adjusting the pulse width to the fans. The **alternate control** 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. A PD controller must be implemented for the fan power.

New objectives will be specified every semester that use the 3 fans in different ways or some not at all. Make sure you understand the current goals for your semester. For the Spring 2019 semester the 3 fans will be used to control the heading. The side thrust fans will be vertical and augment the tail fan, spinning in opposite directions so as to not counteract each other.

In addition to the 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 in Lab 3, 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 up to 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. Any value detected greater than that range should revert the control back to the original desired heading.

It is strongly recommended that you develop the control code that maintains a desired heading using PD feedback (ignoring the ranger) before extending the control to implement any offset feature (which will be done open-loop). In the debugging it is helpful to set the feedback gains (P & D) to 0 and then adjust them one at a time to verify correct operation.

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. These are the same type of transceivers used in Labs 4 and 5. The number on the transceiver plugged into the laptop must matches the number on the gondola to insure both use the same channel. **The black transceiver box on the table with the gondola should always be left on the table and *will not work* with the cars.**

Lab Description and Activities

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

Summary - Compass

1. Implement a 2-stage control algorithm: 1) If the gondola rotation speed is less than some nominal value determined by the team, a PD control algorithm will be used to adjust the heading using all 3 fans. 2) If the gondola rotation speed is greater than the same nominal value (as when someone grabs the turntable and gives it a hard spin) the controller will only attempt to reduce the spinning to below the nominal value, again using all fans, and then switch to the PD control mode to correct the heading error.
2. Change pulse width range to match speed controller.
3. Test on turntable with different values of gain constants and rotation speeds that determine the switching point between fast rotation and the slower rotation for correcting the heading error.
4. Record the actual heading vs. time and plot results.

Summary - Ranger

1. **The thrust angle must be determined to set them vertically.** Add code to set the PWM 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. The control interface is left up to the group (LCD, switches, etc.).
2. Demonstrate that the code can be used to manually correct the angle on any gondola.

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 and 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 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. You will be downloading your code onto the C8051 inside the gondola.

Software

Modify your functions from Lab 4 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. Check the **Gondola_info**[1] file on LMS for the latest up-to-date configuration data. 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 for this is:

```
XBR0 = 0x25;
```

Make sure to use the selected feedback control statement from Worksheet 11 (worksheet_11.c) in your code. 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 two stage control algorithm:

Stage 1: Using the compass, detect the rotational velocity of the gondola ($\Delta\theta/\Delta t$). If this magnitude of this value is above a certain threshold, slow the gondola using a proportional control scheme. Once rotational speed is below the threshold, move to Stage 2.

Stage 2: Implement a proportional plus derivative (PD) control algorithm. Hint: in order to determine proper gains, first set the derivative gain to zero, so that it is purely proportional control. Adjust the proportional control until a consistent and reliable oscillation is achieved. Thereafter, introduce the derivative gain and test various values to achieve the desired response. 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 in the **Motivation** section.

The control algorithm should modify the 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.

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. 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 (again check Gondola_info for the latest updates), 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, since the maximum voltage is now ~8000mV). The nominal value for the voltage divider has R1=22kΩ and R2=6.8kΩ, so the voltage read by the ADC is 0.236 x $V_{battery}$ (in other words, the actual battery voltage is 4.235 x ADC voltage or 39.7 x ADC0 value is the voltage in mV).

[1] LMS: Labs → Lab 6 → Gondola_info

Data Acquisition

As a final check of your control calculations you must set a fixed desired heading and manually rotate the gondola a full 360° turn, observing the values of the compass reading, the error calculation, the feedback control and the fan motor PW calculated by your control value. For this you should switch off the fan motors using the 3 switches 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 compass must be calibrated, do so and then repeat this exercise.

When your code is functioning correctly, gather data to plot response curves for your heading 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 terminal screen that is set to copy what is printed to a log file. Then import the file 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 record responses for different gain combinations.

For this semester, three response plots are required (listed below). It is suggested to disable the ranger control when capturing these responses.

1. "Optimal" – Gains which allow the gondola to achieve the desired heading relatively quickly and hold steady (minimal oscillation) once aligned properly.
2. "Underdamped" – Gains which roughly follow the desired heading but are unable to eliminate oscillations. Responses may be unstable.
3. "Overdamped" – Gains which cause a slow approach of the gondola towards the desired heading with little oscillations. Responses may be unstable.

The following settings are provided **as suggestions** to help you gain a feel for how the system responds. It isn't necessary to plot their responses while experimenting. You may also look at other combinations to verify trends while determining values for the 3 required plots above. Discussion on how you determined gains for your plotted responses will be considered when the report is graded.

With 3 fans there is a lot of power available to control the rotation accurately. You may wish to disable the 2-stage control algorithm and compare responses for just the PD feedback.

Without differential gain ()

Case 1:	kp =	0.1,	kd = 0
Case 2:	kp =	4,	kd = 0

With differential gain ()

Case 3:	kp =	0.1,	kd = 3
Case 4:	kp =	0.5,	kd = 25
Case 5:	kp =	3,	kd = 25

Case 6:	kp =	3,	kd =	60
Case 7:	kp =	8,	kd =	25
Case 8:	kp =	8,	kd =	60

Proportional and derivative gains will need to be adjusted to optimize the response. Adding the 2 thrust fans instead of using the tail fan alone significantly changes the control system characteristics. Some of the gains listed in the 8 cases above, particularly the kp values, may be too large to produce an overdamped 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. Explain how the control to stop the gondola spinning works.
2. Show the TA the heading vs. time graphs for your optimal settings. Good performance is indicated by a short rise time, low over shoot and short settling time.
3. 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).
4. Demonstrate the ability for the gondola to quickly reduce an excessive rotational velocity (Stage 1 control)
5. 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.
6. 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 a single screen shot as an example and attach it to your lab notebook. Desired heading 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 (ranger disabled). 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 - Gondola Report

You will be submitting a single design report detailing the operation of your *Gondola*. The report covers the development of the Lab 6 controller. Some of this report may be written while working on the lab. Once Lab 6 is functioning, the report can effectively be completed with the response plots. This should include 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 Grading *Gondola Report Guidelines*. A rubric for the report follows.

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.

Comment on the performance (heading and ranger values) with both high gains and low gains. For each set of gains, print the actual and desired headings. Use the capture command within the terminal emulator to save these values in a file. Open the file as an Excel or MATLAB CSV file and graph the pulse-width modulated as signals over an interval of 15 seconds. Title all graphs to clearly show the parameters and gains used.

Lab 6 Gondola Report

Group names: ______________________ ______________________ ______________________

The following list is a suggestion of material that would logically be included in the report. It is not intended as a direct guide. You have done the experiments, so you have the best idea of what you needed to complete those projects and what you learned during the process. This means that the following information may be incomplete based on your experience with the project. If you feel part of what you did during the laboratory is pertinent, include that information.

	(suggested # of pages, but you may go higher)		
Introduction	(<1pg)	**5**	
Purpose/Objectives	_____	______	
Overview of gondola feedback control	_____	______	*Sum:* ______
Schematics	(1pg)	**15**	
Content (Chips, Devices, Buses, Slaves, Master, Switches, Passive components, etc.)	_____	______	
Layout/Format	_____	______	*Sum:* ______
Results, Analysis & Conclusions	(3-4pg)	**30**	
Description of gondola performance	_____	______	
Verification of performance to specifications	_____	______	
Analysis of plots from data	_____	______	
What was Learned	_____	______	
Problems Encountered & Solution	_____	______	*Sum:* ______
Performance Plots	(?pg)	**35**	
Logical layout of data	_____	______	
Labeled axes with units	_____	______	
Presentation of plots	_____	______	*Sum:* ______
Well-formatted Code (Submitted to LMS)	0[2]	**10**	
(Verify that uploaded C code is commented)			
Format (indenting, correct font & point size)	_____	______	
Fully Commented with Names	_____	______	*Sum:* ______
Formatting & Neatness		**5**	
Consistent Page Numbering thru report	_____	______	
Cover Sheet (section # & side, grading TA)	_____	______	
Spelling & Grammar	_____	______	
Required: Academic Integrity and Division of Labor page - **signed** (See the provided template form)	_____	______	*Sum:* ______
Lateness (unexcused)			
-20% per School Day	-20% x _____	______	*Sum:* ______
Total		**100**	***Total Points:***__________

NOTE: No report grades will be given without uploading softcopies of the .c file to LMS. Report must be signed.

Be sure to read "(Bad) LITEC_Report_examp_graphs" under Lab 6 on LMS to avoid common mistakes when plotting data. Axes must be scaled and units specified. These plots are being used to show bad plotting formats (independent of the actual data).

[2] Code should not be included in the report but is instead submitted to LMS. Use last initial of members in the .c file name (ex. 2B_HLR_lab6.c. for a team in section 2, side B with last names Hamlet, Lear, and Romeo). Only one team member should upload the file but it must contain all 3 member's name in the header comments. Everyone on the team must sign the hardcopy of the report using the Academic Integrity form given below.

LITEC Gondola Report Guidelines

The gondola lab report for LITEC covers only the material in Lab 6. This rubric (GradingGondolaRpt-student) on LMS in the Laboratories & Worksheets section under Course Materials, lists most of the items to be included, but the list is not necessarily exhaustive. **It is important to note that some of the report can be written before finishing Lab 6**. About half of the written portion (excluding the plots and code listings) deals with describing the control strategies used to achieve your final results and how well they worked. This can greatly reduce the time crunch at the end of the semester.

The report should include detailed descriptions of the final goal: the feedback systems on the gondola involving the compass and ranger in correcting heading errors (adjusted with the ranger). The discussion should explain how the PWM pulse-width calculations are made based on the errors and feedback gains (proportional, derivative, and integral, if appropriate). With respect to response plots (described below), analyze the various plots and justify their characteristics for the sets of gains used. It is up to you to decide how to document and justify your results through carefully chosen plots and clear narrative in your report.

2019 Spring: Three plots are required showing a range of responses (overdamped, underdamped, oscillating, or spinning) and at least one that demonstrates an "optimized" response.

Reports must contain:

1) Wiring diagram[3] of the Lab 6 gondola (with LCD display & keypad, motor connections [4], and ADC input from battery, as well as any other necessary connections). You didn't wire it but you have all the information on how it was wired. Note: I2C devices and motors in the gondolas are wired the same way as for the cars.
2) Program listings (.c file) for gondola (Lab 6) program is only uploaded on LMS
   Program listings **must be formatted as follows**:
   Make sure proper indenting is used consistently throughout
   Include an appropriate prolog (programmer names, section & side, date, brief description, etc.)
   Line comments and block comments should be used liberally with detailed description of what the code is doing or how it works
4) Clearly labeled and captioned plots for data acquired during lab, with scaled axes & units
   Time plots from gondola showing heading angle as it corrects itself for several different values of P and D heading gains (Follow the cases given in the lab procedure). Use normalized motor pulse-width (-100% to +100%).
   Desired heading and actual heading tracking vs. time when ranger distance value is used to input a desired heading value, if this was working and data was collected.
   {Optional} Normalized fan pulse-width (y-axis) vs. ranger distance (x-axis) for a set of height gains (Kp = 3, Kd = 30) as the ranger distance varies from the neutral position to a maximum or minimum distance, and back to the neutral. Do this with a slow and fast change in distance.

[3] Diagrams should either be generated using appropriate drafting software or *very neatly* hand-drawn. A free version of PSpice is available on campus for drawing circuits. Check LMS for a link to the download.

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) ____
____

Software implementation: ____
(pseudo-code & code) ____
____

Data analysis (if relevant): ____
____
____

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

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

____

____

____

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