



Lab 4 Feedback Control

(Demonstration due in class on Thursday, 4/09/09)

(Code and Memo due in Angel drop box by midnight on Friday, 4/10/09)

Reference: http://roboticsprimer.sourceforge.net/workbook/Feedback_Control

Purpose: The purpose of this lab is to improve robot odometry by using feedback control. This code will use the Serializer library's PID motor controller. You will learn to tune control parameters using trial and error. You will repeat the square motion experiment from Lab 1 and confirm that the error is indeed improved based upon feedback control. Finally, you will use feedback control to improve the wall following behavior implemented in Lab 3.

Equipment: Base Robot (4 infrared sensors)
Walls
Ruler

Software: Microsoft Visual Studio.NET 2008 with C#
Serializer.NET library and firmware
Bluetooth transmitter

Part 1 –Tuning Control Parameters

1. Run Lab4.exe and observe that the code now uses the Serializer libraries' PID motor controller. The GUI has text boxes to input robot distance and angle as well as gains for the proportional, integral, and derivate controllers.
2. Your first task is to tune the robot's PID motor controller to improve the odometry error.
3. Set the robot's forward distance to 12 inches and turn angle to 90°.
4. Set the proportional gain to 1 and the derivative and integral gains to zero.
5. Drive the robot forward and turn 90°.
6. Measure the forward distance with a ruler and estimate the turn angle.
7. Next iteratively adjust the proportional gain to improve robot's accuracy in the forward movement and turn.
8. Once the proportional gain set, adjust the derivative gain to improve the robot's accuracy in the forward movement and turn.
9. Finally, keep the proportional and derivative gain set and adjust the integral gain to improve the robot's accuracy.

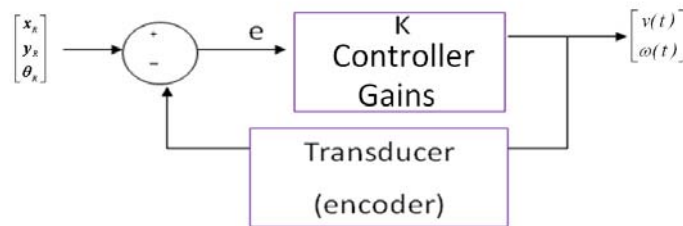


Figure 1: Model for Traxster II PID Motor Control

Part 2 – Improved Odometry (PID Motor Controller)

Using the controller gains found in Part 1, repeat the odometry error experiment performed in Lab 1 using the square motion. Modify the square motion code to use the PID motor controller since it is now possible to explicitly set the length of the sides and turn angle instead of using a timer.

Part 3 – Improved Wall Following (Feedback Control)

1. Improve the Traxster II's wall following behavior by implementing feedback control. The robot must now start at least 10 inches from the wall and move toward the wall and maintain a distance of 4 to 6 inches from the wall for at least 4 feet.
2. The robot should still follow the wall and negotiate obstacles, corners and doorways with minimal contact with walls and obstacles.
3. It is recommended that you start with a proportional controller using error based upon distance from the wall $[K_p * (\text{error input})]$. The gains on the controllers should control motor speed and/or angle.
4. Once the proportional control works at an acceptable level try to incorporate derivative control. $[K_d * d(\text{error input})/dt]$
5. Lastly, if the PD controller works at an acceptable level, try to incorporate integral control to implement a PID controller. $[K_i * \int(\text{error input})dt]$
6. Devise a method to test and confirm that your program works correctly and present the results in your memo

Part 4 – Improved Line Following (Feedback Control)

1. Improve the Traxster II's line following behavior by implementing feedback control. The robot can start on the line and should use a finite state machine to follow the path.
2. A finite state machine would include states such as centered on line (go forward), slight left (turn left), etc. and now with the PID controller you can specify degrees to turn, velocity to go forward or distance to go forward.



3. If you can get the robot to follow a straight line, the next test is to attempt to turn a corner
4. If you can get the robot to turn a corner then try to get the robot to follow a line from an arbitrary start point off the line.
5. Finally, attempt to keep track of states so that the robot follows different paths at a junction of lines.

Questions to Answer

1. What controller gains did you get for the P, PI, PD, PID controller when you tuned it in Part 1?
2. What were the results of the different gains, how did you decide on which one to use?
3. Compare and contrast the robot's accuracy for the 12 inch distance based upon the controller used (P, PI, PD, PID).
4. How does the odometry error for the square motion compare to the error from Lab 1?
5. What plan did you use to implement the feedback controller to improve your wall following and line following behaviors?
6. Compare the results of the improved wall following and line following to the implementations without feedback control.
7. What was the general plan you used to implement the feedback control on the robot?
8. How could you improve the line following and wall following algorithms that you developed in this lab?

Submission Requirements:

You must demonstrate your improved wall following by Thursday, 4/09/09. You must also submit properly commented code and a memo for Lab 4 by midnight on Friday, 4/10/09. Please use the following checklist to insure that your memo meets the basic guidelines.

Memo Guidelines

- ✓ Format
 - Begins with Date, To , From, Subject
 - Font no larger than 12 point font
 - Spacing no larger than double space
 - Includes handwritten initials of both partners at the top of the memo next to the names
 - Written as a paragraph not bulleted list
 - No longer than two pages of text



- ✓ Writing
 - Memo is organized in a logical order
 - Writing is direct, concise and to the point
 - Written in first person from lab partners
 - Correct grammar, no spelling errors
- ✓ Content
 - Starts with a statement of purpose
 - Discusses the strategy or pseudocode for implementing the robot paths
 - Discusses the tests and methods performed
 - States the results including error analysis
 - Answers all questions posed in the lab procedure
 - Clear statement of conclusions