Lab 4 Line Following: PI Control

Reading: *Introduction to AI Robotics (Sec. 4.3), Lec. 3-1*

(Demonstration due in class on Thursday)

(Code and Memo due in Angel drop box by midnight on **Thursday**)

Read this entire lab procedure before coming to lab.

****

Purpose: The purpose of this lab is to implement a *line following* behavior on the Traxster II by using feedback control. The line following sensor will be used to detect the line and the robot should use proportional – integral (PI) control to follow the line for at least one lap of the track. Finally, the *line following* behavior should then be integrated as the top layer onto the subsumption architecture implemented in Lab 2.

Equipment:
Base Robot
4 IR Sensors
3 sonar on servo turret
keypad, speech module, LCD display
obstacles
line path

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

****

LAB PROCEDURE

****

Part 1 – Line Following

A finite state machine will be used to implement *line following* behavior on the Traxster II robot. The robot should start over the line and keep track of how many of the five sensors are activated. The goal should be to keep 4 to 5 of the sensors activated as the robot traverses the line. The robot states should include *on line_go forward*, *slight left*, *sharp left*, *slight right*, and *sharp right*. The PI controller would be used to determine the angle or heading for the turn and possibly the angular or linear velocity based upon the robot's current state.

The first step is to design the proportional (P) controller, where the error is the difference between the desired number of activated line sensors and the actual number of activated line sensors $[K_p * (error\ input)]$. Tune the gain on the P controller until the robot can follow a straight line at an acceptable level. Recall that the integral (I) controller should correct for offset error and should be triggered when the accumulated error reaches a pre-specified threshold and then modify the robot's heading or turn speed $[K_i * \int (error\ input)dt]$. In order to implement the I controller, it is necessary to keep track of the robot's error after each sensor update and then sum. Next, tune the integral controller until the robot follows a straight line at an acceptable level. Finally, devise a method to test that the *line following* behavior works correctly and completely traverses a circular path (see Figure 1).

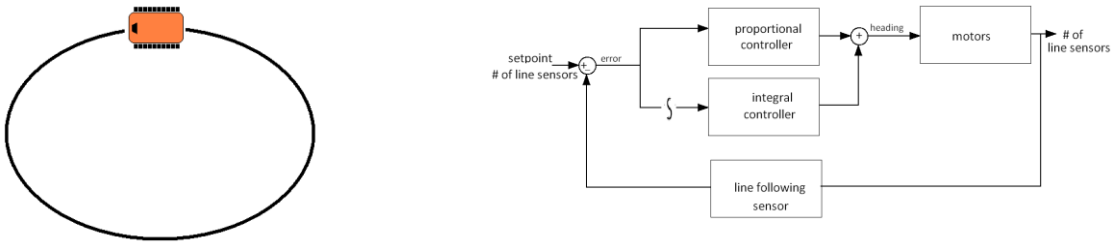

Figure 1 Line Following PI Controller

.

Part 2 – Layer 2 – Subsumption Architecture

Now modify the obstacle avoidance program created in Lab 2 so that there is a *line follow* layer. The robot should wander safely through the environment until a line path is detected. The robot should then abandon the *wander* behavior and follow the path. If the path is lost, the robot should return to the wandering behavior.

Demonstration:

The demonstration of the program for lab 4 will be performed in two parts. In the first part, the robot will start over the line and after input from the keypad or GUI the robot should start to follow the line and maintain contact as long as possible. There should be a mechanism to show the robot's current state and heading based upon output from the controller. In the second part, the robot should start in the world and wander while avoiding obstacles. Once the robot encounters a line, the robot should follow the path as long as possible. If the line is lost, the robot should return to the wander with obstacle avoidance behavior. As always, there should be a mechanism to identify the robot's state and/or behavior and heading.

Program:

The program should be properly commented and modular with each new behavior representing a new function call. The design of the subsumption architecture should be evident from the program layout. You should use the GUI, keypad, LCD and speech module as needed to illustrate robot state, input and output data.

Memo:

The following list provides the basic guidelines for writing a technical memorandum.

- ✓ Format
  - Begins with Date, To , From, Subject
  - Font no larger than 12 point font
  - Spacing no larger than double space
  - Written as a paragraph not bulleted list
  - No longer than three 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 (may include a flow chart)
- Discusses the tests and methods performed
- States the results including error analysis
- Shows data tables with error analysis and required plots or graphs
- Answers all questions posed in the lab procedure
- Clear statement of conclusions

Questions to Answer in the Memo:

1. What does the subsumption architecture with the integrated *line following* behavior diagram look like?
2. Did you use any inhibition and/or suppression to integrate the *line following* behavior in the existing architecture?
3. Did the robot exhibit any overshoot, oscillating or offset error?
4. What were the results of the different P and I controller gains? How did you decide which one to use?
5. Did the addition of the integral controller improve the performance and decrease the offset error?
6. What was the pseudocode and flow chart for the program design?
7. What were the statistics for the robot's ability to successfully traverse the path? How long? How many times? How many laps? Repeatability? Accuracy?
8. How could you improve the *line following* behavior?

Grading Rubric:

The lab is worth a total of 30 points and is graded by the following rubric.

Points	Demonstration	Code	Memo

10	Excellent work, the robot performs exactly as required	Properly commented, easy to follow with modular components	Follows all guidelines and answers all questions posed
7.5	Performs most of the functionality with minor failures	Partial comments and/or not modular with objects	Does not answer some questions and/or has spelling, grammatical, content errors
5	Performs some of the functionality but with major failures or parts missing	No comments, not modular, not easy to follow	Multiple grammatical, format, content, spelling errors, questions not answered
0	Meets none of the design specifications or not submitted	Not submitted	Not submitted

Submission Requirements:

You must submit the lab memo and code by midnight on *Thursday*. You must also submit a memo for Lab 4 by midnight on **Thursday**.