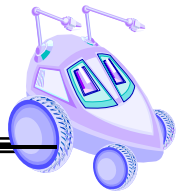

Lab 3 Path (Line) Following

(Demonstration due in class on **Thursday, 4/02/09**)

(Code and Memo due in Angel drop box by midnight on **Friday, 4/03/09**)

Reference: http://roboticsprimer.sourceforge.net/workbook/Locomotion:Exercise2-Path_Following

Purpose: The goal of this lab is to use the robot's line sensor to follow an arbritrary line path.

Equipment: Base Robot (line following sensor)

Line Path (Black on white paper)

Software: Microsoft Visual Studio with C#

Serializer.NET library and firmware

Microsoft Robotics Studio

Bluetooth transmitter

Part I – Run Line Follower Application

1. Download the *Lab3.zip* file from the Lab 3 folder in Angel
2. Open the solution (*Lab3.sln*) in Microsoft Visual Studio
3. Examine the code and comments in *Form1.cs* and try to get some idea of how the program works. The code and comment format for this program is the standard you should follow for your submitted code.
4. Build the solution (F7)
5. Run the application (F5)
6. Turn on the robot
7. Connect to the robot on the correct COM port and place it over one of the line paths
8. This line follower application has a GUI with a line follower user control. This user control the 5 line follower sensors from the robot's perspective (top)
9. The sensor will highlight yellow and show a line when the robot detects one or more lines.

Part II – Line Start

Now that you understand how the line follower works, write a program so that the robot will follow the line path if you start the robot on top of the line (see Figure 1).

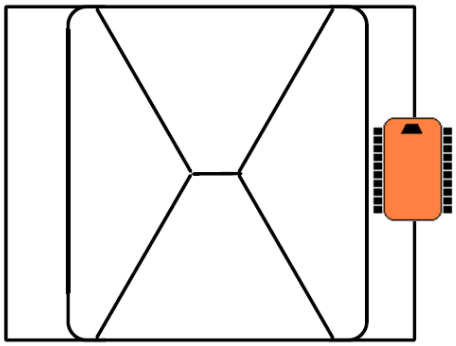

Figure 1: Line Start

Part III – Arbitrary Start Point

Once part II is working correctly, modify your code so that the robot can start at any location and will wander until a line is found and then begin to follow it (see Figure 2).

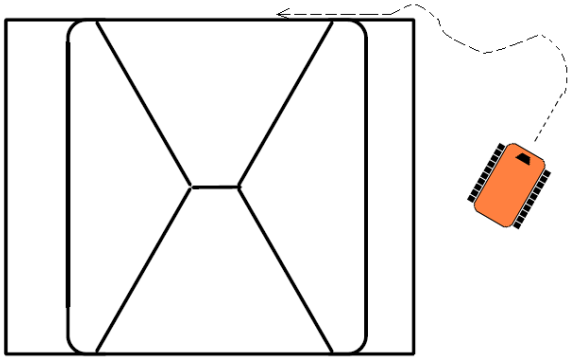

Figure 2: Random Start

Part IV – Making Decisions (Robot States)

Finally, modify your arbitrary start point code to keep track of states such that if the robot encounters a junction of lines or more than one path it will alternate the path that it takes (see Figure 3).

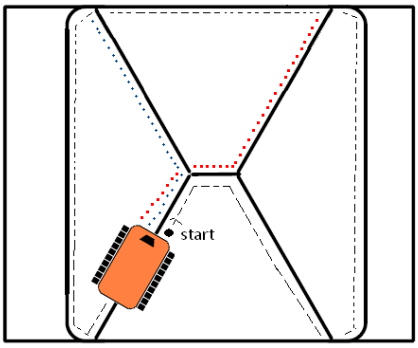

Figure 3: Alternating Paths

Questions to Answer

1. What was the general plan you used to implement the line following behavior?
2. How do you program the robot to remain centered over the line?
3. How does the program respond when the robot loses the line? Or overshoots it?
4. How do you handle it when the robot encounters a meeting point of lines or multiple lines? What is the decision making mechanism?
5. How did you keep track of the robot's states and which line (path) it followed last?
6. How do you get the robot out of a stuck state if it cannot determine which way to move?
7. How do you resolve the issue of the robot oscillating over the line as it follows it?
8. What type of enhancements or improvements could you recommend to improve your program?

Submission Requirements:

You must submit properly commented code for the square, circle and figure eight by midnight on **Friday, 4/3/09**. You must also submit a memo for Lab 1 by midnight on **Friday, 4/3/09**. Please use the following checklist to insure that your memo meets the basic guidelines.

Memo Guidelines

- ✓ Format
  - o Begins with Date, To , From, Subject
  - o Font no larger than 12 point font
  - o Spacing no larger than double space
  - o Includes handwritten initials of both partners at the top of the memo next to the names
  - o Written as a paragraph not bulleted list
  - o No longer than two pages of text
- ✓ Writing
  - o Memo is organized in a logical order
  - o Writing is direct, concise and to the point
  - o Written in first person from lab partners
  - o Correct grammar, no spelling errors
- ✓ Content
  - o Starts with a statement of purpose
  - o Discusses the strategy or pseudocode for implementing the robot paths
  - o Discusses the tests and methods performed
  - o Address any questions posed in the lab
  - o Clear statement of conclusions