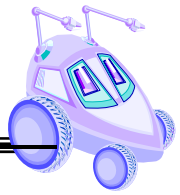


Lab 1 Locomotion and Odometry

(Demonstration due in class on Thursday, 3/19/09)

(Code and Memo due in Angel drop box by midnight on Thursday, 3/19/09)

Reference: http://roboticsprimer.sourceforge.net/workbook/Locomotion:Exercise1-Locomotion_and_Odometry

Read this entire lab procedure and complete Part 1 before coming to lab.

Purpose: The purpose of this lab is to confirm that you have all of the necessary software installed on your computer to connect to and program the Traxster II. The secondary goal is to get the Traxster II moving and to examine problems with raw odometry for pose estimation.

Equipment: Base Robot (Buzzer, 2 pushbuttons, 2 LEDs)
Masking Tape
Ruler

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

Part 1 – Software Installation

Microsoft Visual Studio with C#

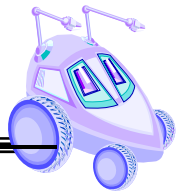
If your laptop did not come with Microsoft Visual Studio, please follow these directions.

1. Click Start -> Control Panel
2. Click "Run Advertised Programs"
3. Select Microsoft Visual Studio 2008 Professional
4. Click Run
5. Click Next through all of the default options

Serializer library

Click the following link to download the Serializer library

http://www.roboticsconnection.com/multimedia/libraries/Installer_15.msi



Bluetooth Transmitter

You will connect to the Traxster II through a Bluetooth Serial Port. If your laptop did not come with Bluetooth please follow these directions.

1. Log into your computer as *localmgr*
2. Download and run the HP Integrated Wireless driver installer (sp38145.zip), in the Angel folder *(if you have a different computer search the web for the driver)*
3. When prompted, log off and log back on
4. When you log back on, the Bluetooth icon should appear in the lower right hand corner of the screen. Double-click this icon to run "Initial Bluetooth Configuration Wizard".
5. When you reach "Bluetooth Services" make sure that "Bluetooth Serial" is checked.
6. Complete the configuration wizard and click "cancel" to exit setup when finished.

Part 2 – Connect to the Traxster II

1. Turn on the robot
2. Confirm that the power light is on for the Bluetooth Serializer V1.0
3. Right click the Bluetooth icon in the tray on the bottom right hand of the screen
4. Click Explore "Explore My Bluetooth Places"
5. Click "Entire Bluetooth Neighborhood"
6. Find eb100. If you are in the classroom full of robots there will be many of them! Each Bluetooth's receiver has a serial number and it should be on a label on the robot.
7. Right click properties on each eb100 until you find the one with your serial number (i.e. Device address: 00:0c:84:3f:e9)
8. On the Authorization tab check Bluetooth Serial Port
9. Click OK
10. Right click Pair on the icon (the pass code is "0000")
11. Double Click on the eb100 icon and it should display "A# serial port on eb100 not connected"
12. Double click the icon and it should now show that you are connected to the robot
13. Right click on the icon and the General tab will show which COM port the Bluetooth Serial port for the robot is connected on. You will need this number for your code!

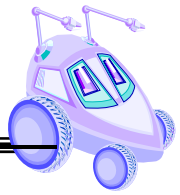
14. Go back to My Bluetooth Places and rename your eb100 icon to the name of your robot, i.e. “Bart”, by clicking on the name. This will save you time in the future when trying to connect to the robot.
15. My Bluetooth Places should now show “eb100 <RobotName> Connected: COM#”
16. The robot should now be connected and ready to use.

Part 3 – Differential Drive and IO Demo

1. Create a folder for all of your course programs
2. Download the *Lab1.exe* file from the Lab 1 folder in Angel
3. Run the application
4. Turn on the robot
5. Make sure that the COM PORT on the GUI is the same as the Bluetooth serial port where the robot is connected. After confirmation of connection, experiment with the Locomotion and Odometry GUI and make sure you understand how to move the robot and check digital IO status. (Make sure the robot is clear of tables and legs!!)

Part 4 – Differential Drive and IO Code

6. Create a folder for all of your course programs
7. Download the *Lab1.zip* file from the Lab 1 folder in Angel
8. Open the solution (*Lab1.sln*) in Microsoft Visual Studio
9. 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.
10. Expand 'References' in the Solution Explorer, and ensure that *“RoboticsConnection.Serializer”* is properly resolved. This means it is listed under References, and that it has a white box icon next to it. If it has a yellow icon, then it isn't resolved. To resolve it, try double clicking on the 'RoboticsConnection.Serializer' namespace. If that doesn't work, then simply right click on “RoboticsConnection.Serialize”' namespace, then 'Remove'. Now, right click on 'References', and then select 'Add Reference'. An Add Reference window pops up. Make sure you're under the .NET tab, then scroll down to 'SerializerLib', and select it. It should now be in the References section correctly



11. We will walk through the code for this program together in class
12. Build the solution (F7)
13. Run the application (F5)

Part 5 – Square Path

Write a program to move the robot in a square path with sides between 2 and 3 feet (see Figure 1).

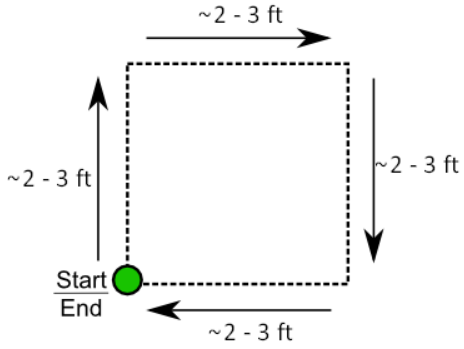


Figure 1: Square Robot Motion

Part 6 – Odometry Error

Now that we have code to control the robot, let's measure how well odometry performs. As you may recall from Part 1, the robot relies on odometry to determine how far it has traveled, and how far it has turned. If odometry readings are perfect, then the robot should make a perfect square. For this part we will measure the amount of odometry error.

1. Place masking tape on the ground where the robot will start.
2. Run the *"Square"* program and place masking tape at each of the other three corners.
3. Move the robot back to the start point, run the square code, and place a marker where the robot turns and ends.
4. Repeat this 3 at least 5 times and each time measure the distance between each pair of points (the correct corner and the actual) and compute an average. This is the average error.

Part 6 – More Paths

1. Write a *"Circle"* program to move the robot in a circle with a diameter between 2 and 3 feet (see Figure 2).

2. Write a *"FigureEight"* program to move the robot in a figure eight using two circles (see Figure 3).

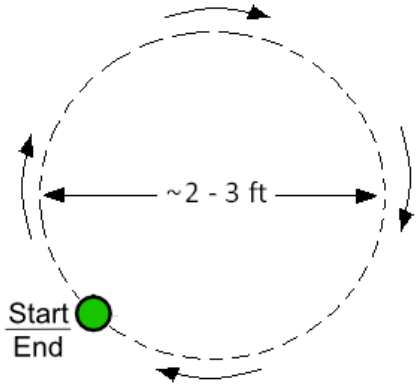


Figure 2: Circle Robot Motion

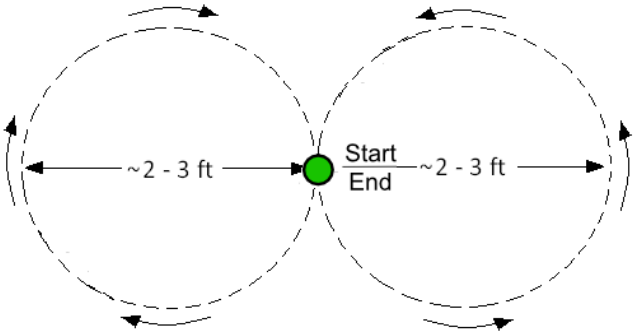


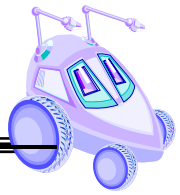
Figure 3: Figure Eight Robot Motion

Part 7 – Digital I/O responses

1. Modify all three programs so that the robot emits a warning sound through the buzzer before it moves.
2. Modify the *"Square"* program so that the robot moves in a square when a pushbutton is pressed and turns on one LED while moving.
3. Modify the *"Circle"* program so that the robot moves in a circle when a different pushbutton is pressed and turns on a different LED while moving.
4. Modify the *"Figure Eight"* program, so that the robot moves in a figure eight when two pushbuttons are pressed and turns on two LED while moving.

Questions to Answer

1. What are some sources of the odometry error?
2. How could you correct for this error?
3. How could you improve the three motion (*Square, Circle, FigureEight*) programs?



Submission Requirements:

You must submit properly commented code for the square, circle and figure eight by midnight on Thursday, 3/19/09. You must also submit a memo for Lab 1 by midnight on Thursday, 3/19/09. 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 States the results including error analysis
  - o Answers all questions posed in the lab procedure
  - o Clear statement of conclusions