RichardVannoy.info

Lynxmotion.com 4-Wheel Drive Rover Kit

Part 4; Source Code

- Part 1: The First Steps - Assembling the documentation and deciding on the robot capabilities. (This page)
- Part 2: The Mechanics - Assembling the Rover Body
- Part 3: The Sensors - Mounting and Testing the Sensors
- Part 4: The Source Code - Implementing the Main Algorithm

The Basic Algorithm

As the block diagram is used to keep me on track about the hardware required, the pseudocode is the tool to build, refine and define the behaviors required for the robot. Here is a simplified version of what the robot will be doing:

Step 1: Run a scan with the IR and Sonar Sensors to determine the best direction to travel next.

Step 2: If all (or most) sensors show no room to go forward, make a U-Turn and go to Step 1, otherwise record the best direction and go to Step 3.

Step 3: If the robot is already headed in the preferred direction, do nothing with motor control and go to Step 5, otherwise go to Step 4.

Step 4: Turn the robot to the preferred direction.

Step 5: Move the robot forward a safe distance.

Step 6: Go to Step 1.

Steps 1 through 6 will repeat until the robot is shut down. Several comments:

1. I specifically purchased a Sabertooth 2X10 motor controller because it will take speed signals from the microprocessor, then continue applying the same command/speed until a different command is sent. This should allow the robot to scan almost continually.
2. The pseudocode above is a first draft. I know from experience that a number of things will come up that will need to be handled, and some of the steps shown are too simple and need to be expanded. For example, I have already see that Step 1 above needs, or will soon need, more detail, such as...
   Step 1.a.: Loop from -75 degrees to +75 degrees in seven 25 degree steps. (To cover the angles -75, -50, -25, 0, +25, +50, +75 degrees)
   Step 1.b.: At each of the angles in Step 1.a., take a Ping sonar reading. Record it. Take an IR reading and record it.
   Step 1.c.: When all seven readings are taken, deterine the best direction to proceed.
   ...and so on...

Notice that Step 1.c. requires some thought and an algorithm. Exactly how do I determine the best direction to proceed???

I know that the IR Sensor will return seven readings from 10 to 80 centimeters. Obviously I need to go in the direction least likely to result in a collision. I also need a panic or U-Turn mode in case the robot gets stuck in a tight spot.

Here are a few IR Sensor strategies that come to mind... I'll need to pick one and see how it works.

- Pick the direction of the greatest distance detected and go in that direction. This seems simplest to code, but what if several or many sensors detect 80 centimeters?
- Average each set of two side-by-side readings, giving six directional distances. Proceed in the direction of the six with the highest average.
- Average each set of three side-by-side readings as in the previous step.
- There are probably more strategies, but that should be enough for now.

I haven't mentioned the Ping Sensor yet, but the same rules and problems apply to the sonar readings. How do I decide the best direction based on the sonar distances? If the sonar and the IR agree, all is well, but what if they disagree? It seems like the sensor that has the nearest obstacle should have priority, but this needs to be thought out and the algorithm needs to be developed.

The Biggest Unknown

At this point, the biggest unknown seems to be the performance of the IR and sonar sensors when working together. There will no doubt be some targets detected by one but not the other. Which will generally be more reliable in the environment expected? The expected environment for now will be a classroom that will have plenty of thin chrome chair legs, some office chairs with black wheels and bases (both POOR reflectors!), some electronic workbench table legs and a number of students. There will need to be a number of tests on each of the expected obstacles to ensure that one or both sensors have a reliable detection range and rate. For now, the plan is to take numerous static readings (robot immobile, reading different obstacles at a known range, and comparing results). At this point all I can do is be aware of possible problems here when the testing of the sensors begins.

Please email me at RoboticsProfessor@gmail.com if you have any questions or comments.