Grid Walker
Robot Come Home

Lab Assignment

The problem posed by this lab is to have the robot navigate to a given location on a occupancy grid (a discrete grid representing an environment) and then to be able to retrace its steps home. The detailed assignment is given on the following page:

http://cronus.mcs.alma.edu/LMICSE/LabMaterials/AlgoComp/Lab3/AlgCoL3.htm

Strategy

The basic strategy that will be used in this is as follows: Move the robot randomly one grid cell at a time until it reaches the specified destination. Save the direction of each move on a stack of moves. After reaching the destination pop the directions off the stack and moving one grid cell in the opposite direction until arriving at home.

leJOS Resources

The leJOS RotationNavigator class provides the basic navigational capabilities that are needed to navigate the robot. Important RotationNavigator methods are:

- public void `gotoPoint(float x, float y)`
  - Rotates the RCX robot towards the target point and moves the required distance
- public float `getX()`
  - Returns the current x coordinate of the RCX.
- public float `getY()`
  - Returns the current y coordinate of the RCX
- public float `getAngle()`
  - Returns the current angle the RCX robot is facing
- public `stop()`
  - Halts the RCX and computes a new x and y coordinates.

A RotationNavigator object can be constructed as follows:

```java
public RotationNavigator(float wheelDia, float AxleWidth, float ratio)
```

Diameter of the drive wheel in centimeters.

Width of the drive axle in centimeters.

Ratio of angle sensor rotations to axle rotations.
The RotationNavigator requires that the robot use two motors for propulsion and that an angle sensor be used with each drive wheel. The angle sensor measures the rotations of the wheels and they must measure positive values when the robot moves forward. Their use enables the RotationNavigator object to determine how far to move the robot. Figure 1 shows the robot configured with two angle sensors. The distance it moves the robot is measured in centimeters. The RotationNavigator class can be found in the leJOS josx.robotics package. You should read the leJOS documentations on the RotationNavigator class as a starting point for the lab.

Figure 1: Robot with angle sensors.
Design

*Application Architecture:*

![Diagram of Application Architecture]

*GridWalker Class:*

The GridWalker Class should be designed to meet the following criteria. It should contain:

- a constructor that specifies the grid size and the starting point of the robot
- a goto method that takes as a parameter a grid location and causes the robot to move from its current grid position to that location
- convenience methods north, east, south, and west that move the robot one cell in the respective direction.

In addition:
- Define GridWalker to extend RotationNavigator
- Have instance variables for
  - initial grid position (x and y)
  - current grid position (x and y)
  - the number of columns and rows in the grid
  - the size of a grid cell (assume it is square)
- The constructor is passed this information, plus the information the RotationNavigator class needs
Implementation of the GridWalker class must take into account that the RotationNavigator class and the GridWalker use different coordinate systems. The RotationNavigator uses an absolute coordinate system, the GridWalker system uses a coordinate system based upon the size of the grid cells.

- RotationNavigator assumes the robot starts at 0,0 with an orientation of 0 degrees
- GridWalker allows the programmer to specify any cell as the starting location, but still assumes orientation of 0 degrees

Reference

McNally, Myles. Walking the Grid: Robotics in CS 2. LMICSE Workshop, June 14 - 17, 2005, Alma College, MI. (PowerPoint Presentation)