Grid Walker
Robot Come Home

Lab Assignment

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


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 NXT Resources

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

- public void travel(float distance)
  - Moves the NXT robot a specific distance. A positive distance causes forward motion; negative distance moves backward.
- public void rotate(int angle)
  - Rotates the NXT robot through a specific angle; Rotates left if angle is positive, right if negative, Returns when angle is reached.
- public void setSpeed(int speed)
  - Sets speed of both motors in degrees/sec.
- public void stop()
  - Stops the NXT robot.

A Pilot object can be constructed as follows:

```java
public Pilot(float wheelDiameter, float trackWidth, Motor leftMotor, Motor rightMotor, boolean reverse)
```

When true the robot will move forward when moving in reverse.
Diameter of the drive wheel in centimeters.
Width of the drive axle in centimeters.
The Pilot class requires that the robot use two motors for propulsion. On the NXT robot these motors are servos-motors with a built-in angle sensor. The important point is that unlike the standard motors on the old RCX robot, the motors on the NXT robot can be told to rotate X degrees and they will do so. The Pilot class uses this capability to be able to move the robot a given distance. The Pilot class can be found in the leJOS NXT `lejos.navigation` package. You should read the leJOS NXT documentations on the Pilot class as a starting point for the lab.

Design

*Application Architecture:*

![Application Architecture Diagram](image)

**Figure 1: 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 contain a Pilot object
- 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 Pilot class needs.

Implementation of the GridWalker class must take into account that the Pilot class and the GridWalker use different coordinate systems. The Pilot class uses an absolute coordinate system, the GridWalker system uses a coordinate system based upon the size of the grid cells.

- Pilot 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)