Mobile Robots Lab Report

¡Noah Harvey

November 30, 2014

Abstract

The robot localization problem is a key problem faced in modern day mobile robots. This problem involves obtaining “clean” data about the robot’s environment, determining the current location of the robot, and deciding on an optimal travel path towards a desired location. The robot receives information from its environment via robotic sensors (wheel encoders and sonar range finders). The robot then uses the sensor data and information about its previous location to determine its current location.

Mobile robot simulation software is a useful tool for designing and testing localization algorithms. Aria is a C++ SDK used for developing and testing mobile robot software. MobileSim is a simulation software for testing mobile robot software which uses Aria. Both Aria and MobileSim were used to develop and test mobile robot software.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Dead Reckoning

In order for a robot to move effectively about an environment it must have information about its current motion state (position, velocity, heading, etc.). For example for a two wheeled robot that attempts to move from one position in a room to another needs to keep track of its current heading, position, speed, and acceleration.

Dead reckoning is a solution to this problem that we will explore. Dead reckoning is the determining of the location of an object based on a known motion model and current state information. Once the current state of an object is known then its motion model can be used to predict future motion states.

A two wheeled robot with a caster (or free wheel used for balance) uses the following odometry model:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\mathrm{w}} & \frac{1}{\mathrm{w}} \end{bmatrix} \begin{bmatrix} s_R \\ s_L \end{bmatrix} = \begin{bmatrix} S \\ \theta \end{bmatrix} \tag{1}$$

Where S and θ respectively are the robot distance travelled and heading of the robot. s_L and s_R are the distances traveled by each wheel (left and right respectively).

While Equation 1 is useful it is difficult to use in determining the motion state of a robot. To simplify our analysis we will assume the following:

- The robot's motion model is considered on a time differential Δt that is small with respect to the total time the robot performs its motion.
- The robot's heading θ is constant on time differential Δt while translating. That is the robot always travels in a straight line.

From these assumptions we have a motion model for the robot:

$$\xi = \xi_0 + \begin{bmatrix} S\cos\theta \\ S\sin\theta \\ \theta \end{bmatrix} \tag{2}$$

1.2 Forward Kinematics

1.3 Wheel Encoders

1.4 Sonar Range Finder

2 Methods

2.1 Square Path Robot

2.2 Sonar Measure Robot

2.3 S-Curve Robot

3 Results

4 Conclusion