Coverage Efficiency in Autonomous Robots With Emphasis on Simultaneous Localization and Mapping Algorithms

Mo Lu P.4

Abstract

Today, automated systems have supplemented humans in previously labor-intensive tasks. Automated lawnmowers are an example of these systems, but the currently available technology in automated lawnmowing is inefficient and primitive. This project will propose and implement an alternate method to automated lawnmowing, known as Simultaneous Localization and Mapping.

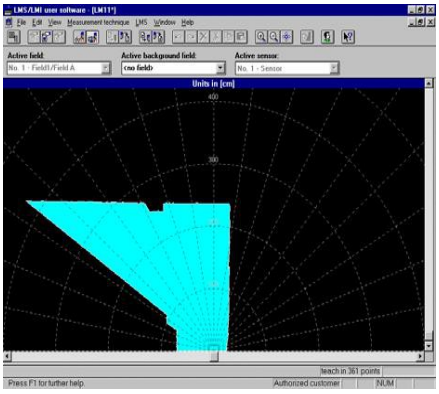
Background

Current approaches to commercial robotic lawnmowing operate under the idea that if a lawmower is constantly mowing the lawn, then the lawn stays constantly mowed. This is done by a series of random cuts and turns, which if given enough time, theoretically could cover an entire unmowed lawn. This method is horrifically inefficient in terms of time and energy, when backtracking is taken into consideration. This project proposes a different approach to this method: use of mapping techniques to recognize landmarks, avoid obstacles, and navigate an environment This method consists of three parts: 1) Use of a constantly updating laser scanner to recognize obstacles, 2) Creation of obstacle map using the laser data, and 3) Processing that obstacle map for runtime efficiency. Success is determined by how effectively the robot avoids the obstacles, how quickly it runs through the lawn, and how accurate the created obstacle map is.

Procedures, Methods, and Results

The parts of the simulation have been adapted for use with the LMS Rangefinder. Most testing and coding has been focused on making the rangefinder successfully scan the environment it has been placed in. This has involved code translation from Python to C++, necessary because the rangefinder has no support for Python platform. While the robot is not self-propelling at this stage, it is able to scan and map out the immediate environment, if the chassis is hand turned. Interestingly enough, base software that came with the rangefinder completely eliminates the need for matrices, and simply relays back the scanned information in a graphics based environment.

Simulation Environment



Actual Environment