Simulation and Execution of Learning Methods and Algorithms of an Automated Lawnmower
TJHSST Senior Research Project Proposal
Computer Systems Lab 2009-2010

Jeff Hobson
Period 3

October 29, 2009

Abstract

The purpose this project sets out to achieve is to develop efficient technology in the field of automation, and to combine this technology with the day-to-day task of mowing one's lawn. Automated Lawnmowers, or ALMs, are already in circulation, but these function on the property that if the ALM moves in random directions consistently during most of the day, then the entire lawn will stay trim (Husqvarna's AutoMower uses this process—see `http://www.automower.us/` for more information). The project centers around the idea that the mowing of a lawn can be done more efficiently, such as when it is done manually. Identifying cut grass versus uncut grass, dividing the lawn into sections to be completed at separate times, and avoiding obstacles are just three of the many ways humans work more efficiently than their automated counterparts. This project uses the computer languages C# (in implementation) and Java used in conjunction with the Processing Development Environment (in simulation) in order to test and evaluate the performance of an ALM as it grows to learn its environment and work more efficiently.

1 Introduction

Mundane tasks involving manual labor have been greatly replaced by automated processes in today's day and age. Though automated lawnmowers (ALMs) have been created before, none are efficient enough to completely replace manual labor and complete the job in the same (or less) amount of time. Most operate under a technique similar to that of the Roomba, an automated vacuum cleaner.

1.1 Problem Statement and Purpose

The purpose of this project is to create a more efficient ALM that can learn to adapt to any environment and become even more efficient and spend less time mowing with each run. The unique nature of each individual's lawn makes this a difficult task to accomplish. Different techniques will be tested in order to maximize efficiency and minimize time spent mowing.

2 Background

This project will involve numerous algorithms and learning methods. SLAM—Simultaneous Localization and Mapping—can be used to help create a virtual environment identical to the real one the robot traverses. This topic is to be implemented by another student and used in conjunction with this project. A thesis paper from a Virginia Polytechnic Institute graduate was released to the public titled *Navigation and Control of an Autonomous Vehicle*. This paper has provided and continues to provide invaluable information about different learning techniques to be considered when creating an ALM.

3 Computer Language

Since this project incorporates both the simulation and the implementation of an efficient ALM, two computer languages must be used to achieve full results. The Java Runtime Environment (JRE) will be used to create a simulation of the ALM in order to provide a graphical representation of the robot as it makes its way around a lawn. A variation of C, either Arduino or C#, must be used when programming the physical robot. Arduino is a microcontroller that allows a user to write a slight variation of C in order to achieve functionality.

3.1 Software

The Processing Development Environment (PDE) will be used when creating the simulation. This application imports custom Java commands specifically designed for facilitating graphics. Processing may also be incorporated into the physical robot's programming as it is fully compatible with Arduino code (in fact, both use the same development environment).

4 Algorithms and Methods

The ALM will learn to adapt to its environment based on a number of factors. The most prominent is SLAM. The robot will primarily use the virtual map it creates in order to determine where to move next. Initially, the robot will wander aimlessly around the yard, gathering data for its map. Once it has created a sufficient map, it will proceed to use the following methods to cut down runtime and maximize efficiency:

- Notice nearby obstacles and be aware of their proximity.
- Identify cut grass from uncut grass and avoid "backtracking," mowing over areas that have already been cut.
- Determine the width of the lawn at various points, so as to avoid moving into sections for which the entry point has already been mowed.
- Divide the lawn into sections as needed and complete entire sections rather than move about the entire lawn, possibly driving over cut grass unwillingly.
- Convert circular and elliptical obstacles into rectangular ones by cutting a border around them. This makes it easier for the ALM to move around them later.

5 Testing and Results

The scope of research for this project is narrow. The topics of SLAM and the different ways to mow a lawn are crucially fundamental in this project, but most methods an ALM is capable of, a human is also capable of, and are therefore easily researched through observation and logic.

5.1 Testing

The program will be tested based upon three factors of efficiency:

- Time taken to mow entire lawn
- Percent coverage
- Amount of backtracking allowed

The robot will be timed as it mows the lawn on several simulated days to be sure the robot learns to better itself. Once the robot is able to determine the difference between cut and uncut grass, both backtracking (how much uncut grass was driven over more than once) and percent coverage (based upon how much uncut grass was missed) will be recorded by the robot and easily accessible to be sure the robot is making forward progress.

5.2 Expected Results

The robot is expected to start out with poor efficiency during its first few runs on a new lawn. Since the purpose of these runs is only to generate a map for the future, they are relatively unimportant on the large scheme of the project. The ALM should provide data that suggests a unique mixture of all or most of the methods will maximize the efficiency while minimizing the time spent on the lawn.