Object Recognition and Path Smoothing Robot

Jake McKenzie

[jake314@uw.edu](mailto:jake314@uw.edu)

Alex Marlow

[alexmarlow117@gmail.com](mailto:alexmarlow117@gmail.com)

Ammon Dodson

[ammon0@uw.edu](mailto:ammon0@uw.edu)

Prepared: 10/17/2018 Rev: 0.1

|  |  |  |
| --- | --- | --- |
| **Revision History** | | |
| v0.1 | 10/17/2018 | Initial Specification |
|  |  |  |

Table of Contents

[1. Introduction 3](#_Toc527578747)

[Remote Control 3](#_Toc527578748)

[Autonomous Control 3](#_Toc527578749)

[2. Requirements 4](#_Toc527578750)

[3. System Architecture 4](#_Toc527578751)

[3.1. Scope 4](#_Toc527578752)

[3.2. Base Station 5](#_Toc527578753)

[3.3. TurtleBot3 Burger 5](#_Toc527578754)

[3.4. The McCourt Filter 5](#_Toc527578755)

[3.5. Game Controller 5](#_Toc527578756)

[4. System Design 5](#_Toc527578757)

[4.1. Hardware Design 5](#_Toc527578758)

[4.1.1. Objectives 5](#_Toc527578759)

[4.1.2. Constraints 5](#_Toc527578760)

[4.1.3. Composition 5](#_Toc527578761)

[4.1.4. Uses and Interactions 5](#_Toc527578762)

[4.1.5. Interface 5](#_Toc527578763)

[4.1.6. Resources 5](#_Toc527578764)

[4.1.7. Details 5](#_Toc527578765)

[4.2. Software Design 5](#_Toc527578766)

[4.2.1. Objectives 5](#_Toc527578767)

[4.2.2. Constraints 5](#_Toc527578768)

[4.2.3. Composition 5](#_Toc527578769)

[4.2.4. Uses and Interactions 5](#_Toc527578770)

[4.2.5. Interface 5](#_Toc527578771)

[4.2.6. Resources 5](#_Toc527578772)

[4.2.7. Details 5](#_Toc527578773)

[4.3. Human Interface Design 5](#_Toc527578774)

[4.3.1. Functionality from the User’s Perspective 5](#_Toc527578775)

[4.3.2. Interface Objectives 6](#_Toc527578776)

[4.3.3. Interface Constraints 6](#_Toc527578777)

[4.3.4. Use Cases 6](#_Toc527578778)

[5. Ethical Considerations 6](#_Toc527578779)

[6. References 6](#_Toc527578780)

[7. Errata 6](#_Toc527578781)

# Introduction

The Object Recognition and Path Smoothing Robot is systems project with multiple objectives, they are, in sequence:

* Implement for verification Dr. McCourt’s path smoothing algorithm.
* Demonstrate Simultaneous Localization and Mapping (SLAM)
* Implement a “finder robot” by integrating machine vision with the SLAM functionality

Additional objectives may include:

* Using beacon triangulation and/or GPS to fuse additional location information into the SLAM or finder functionality
* Implement functionality to allow the system to report back on mapped spaces or found objects

## Remote Control

Remote control systems have many useful applications; The typical use case involves using robots to interact with an environment that is too hazardous for a person. Any such remote control involves some delay of both outgoing control signals and incoming sensor data. In some situations, this delay may impair the intended function of the remote system.

Dr. McCourt has developed a set of filters intended to be placed in such a delayed, closed-loop control system. These filters apply a mathematical transformation on both incoming and outgoing loop signals such that timing delays are mitigated.

## Autonomous Control

There may also be use cases for robots in hazardous environments where direct human control is impossible. Such a robot must be able to autonomously navigate and interact with an unknown space. SLAM is a fundamental technology for such autonomous activity, allowing the robot to navigate. Additionally, such an autonomous machine must be able to sense and recognize an objective before being able to interact with it. Computer vision is another fundamental technology for sensing a real-world environment. Sensing a condition is a necessary first step for being able to take action based on the current environment.

# Requirements

This robot will use the TurtleBot3 Burger as a base. All the hardware and software modules of the remote component must fit within the constraints of that platform.

In order to demonstrate the McCourt filter the robot must be remotely controllable. Control signals must be routed through a delayed communication medium. Ideally, the human controller’s feedback data should also be routed through the delayed medium.

In autonomous mode the robot should be able to completely explore and map a real-world environment. Potentially it should also be able to identify, locate, and report on the location or condition of some object in the environment.

# System Architecture

## Scope

This specification covers the following:

* Any and all hardware modifications to the TurtleBot3 Burger
* Software installed on the burger bot insofar as it deviates from the stock installation
* Base station control software setup insofar as it deviates from stock installation
* Any necessary techniques for integrating a Human Interface to the base station installation
* Any methods used to implement a communication delay between the robot and its base station
* Implementation of the McCourt filter
* An overview of implementing SLAM on the TurtleBot3 Burger
* Integration of OpenCV into the TurtleBot3 Burger

## Base Station

## TurtleBot3 Burger

## The McCourt Filter

## Game Controller

# System Design

## Hardware Design

Our hardware is largely preset

### Objectives

### Constraints

* Raspberry Pi 3 (32-bit ARM Cortex-M7)
* OpenCR (32-bit ARM Cortex-M7)

### Composition

### Uses and Interactions

### Interface

### Resources

### Details

## Software Design

### Objectives

### Constraints

### Composition

### Uses and Interactions

### Interface

### Resources

### Details

## Human Interface Design

### Functionality from the User’s Perspective

When in remote control mode the robot can be controlled by a consumer game controller.

### Interface Objectives

### Interface Constraints

### Use Cases

# Ethical Considerations

Everyone loves this project. It will be really good for America.

# References

1. Turtlebot3 Manual: <http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>
2. Robot Operating System: <http://www.ros.org/>
3. Open Source Computer Vision Library (OpenCV): <https://opencv.org/>

# Errata

There is no previous version of this document.