Object Recognition and Path Smoothing Robot, Phase 5

Design Specification

University of Washington Tacoma, School of Engineering and Technology

Authors:
Ammon Dodson ammon0@uw.edu
Alex Marlow alexmarlow117@gmail.com
Jake McKenzie jake314@uw.edu

Distribution: Matthew Tolentino
Document state: draft
Modified: January 18, 2019

Contents

1 Revision History

- v0.1.0: Initial specifications for the document itself.
- v0.2.0: Additional information was added about the Robot Operating System (ROS).
- v0.2.1: Ethical concerns were added to the document.
- v0.2.2: Created a diagram for the overall system architecture.
- v0.3.0: Additions to the introduction. Rewrite of the requirements section with enumeration. Additions to the System Architecture and System Design section including hardware architecture and software architecture with higher level and lower level diagrams with text describing these diagrams and text for these sections generally.
- v0.4.0: Made the requirements table. Clarified some requirements. Improved page formatting. Added some to the design section. Use case diagram created for the introduction and general mistakes addressed.

2 Introduction

2.1 Overview

This design specification describes the architectures and design framework that will aid and abet the production of the ORPS-Robot version 0.4 – The Object Recognition and Path Smoothing Robot. The ORPS-Robot will be a platform for validating the research of Michael McCourt and a scheme for exploring object recognition via OpenCV with Robot Operating System, which is a powerful framework for writing robot software. There will be a demonstration of Simultaneous Localization and Mapping (SLAM). Together this will demonstrate a "finder robot" with applications in search and rescue and threat detection.

Additionally, there will be beacon triangulation and/or GPS to fuse additional location information into the SLAM or finder functionality. Typically, feedback is used when some amount of uncertainty exists within a system and the environment for which that system operates. To implement feedback this requires analyzing the underlying dynamics of whatever system you plan to look at and the implementation of communications, computing and software to accomplish some task. The project aims at investigating the dynamics of a semi-autonomous robot with the Object Recognition and Path Smoothing Robot by delivering a platform that is some interplay of communications, computing and software to Michael McCourt to aid in his research in control.

The networked control system to be explored in this project is commonly used in telerobotics. Telerobotics is still in the experimental phase of research and has been theorized for active shooter detection, a way for doctors to diagnose patients in disaster areas and telecommute from home.

2.2 Network Control

Network 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 Network 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 network control 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 communication delays are mitigated. This specification outlines the development of a controller-robot system intended to demonstrate the McCourt filter.

2.3 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 act based on the current environment.

2.4 Scope

This specification covers the following:

- Any and all hardware modifications to the ORPS-Robot.
- 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 input output transformation.
- An overview of implementing SLAM on the ORPS-Robot.
- Integration of OpenCV into the ORPS-Robot.

3 Requirements

In this section we will delve through the minimum requirements of this project to translate the needs of our client Michael McCourt into precise targets, establish metrics for a successful product and support design trade-off decisions. For this revision of the design specification we will articulate the marginally acceptable target values, without any of the ideal target values for our stretch goals. **INSERT TABLE HERE**

3.1 Network Control

R1. **There will be some ROS capable mobile platform.**
This robot must be ROS capable, mobile, and remotely controllable. The mobile platform must be capable of processing the R-transform for incoming and outgoing signals. All the hardware and software modules of the mobile platform must fit within the constraints of that platform, including such things as power, weight, and processing speed.

R2. **There must be a base station interfaced with a USB game controller.**
The Human controller we have available currently is a USB game controller. This controller will have to be connected to some base station that can connect to the ORPS-Robot wirelessly. The base station is a laptop which will run the graphical user interface for the user, route communication traffic, perform transformations and public commands for the ORPS-Robot.

R3. **The base station must communicate with the robot through a delayed link.**
In order to demonstrate the McCourt input-output transformation the robot must be remotely controllable. Control signals must be routed through a delayed communication medium. Ideally, or as a second stage, the human controller's feedback information should also be routed through the delayed medium.

R4. **The Robot Should be wirelessly controllable.**
To fully demonstrate the usefulness of the McCourt transform we should model a real-world situation where the robot is out of sight and controllable solely from the base station.

R5. **The robot must be able to report its position in space.**
To fully demonstrate the McCourt input-output transformation the full control loop should be routed through it as shown in Figure 2. This will require the position feedback to be in the form of a simple numerical array such as an x-y coordinate.

R6. **There must be a means of recording and comparing the planned and actual path the robot follows.**
To demonstrate the McCourt input-output transformation we need to be able to compare the actual and planned path and have some measure of how closely they align. We should be able to make multiple test runs with and without the filter functioning and compare the course fidelity in aggregate.

3.2 Autonomous Control

4 System Architecture

4.1 Hardware Architecture

4.2 Software Architecture

5 System Design

5.1 Hardware Design

5.1.1 Objectives

5.1.2 Constraints

5.1.3 Composition

Xbox 360 Controller

Marvelmind Indoor Navigation System

Raspberry Pi Model B

OpenCR

DYNAMIXEL Actuator System

Li-Po Battery

LDS-01 LiDAR

5.1.4 Interface

Xbox 360 Controller

Marvelmind Indoor Navigation System

Raspberry Pi 3 Model B

OpenCR

DYNAMIXAL Actuator System

Li-Po Battery

5.2 Software Design

5.2.1 Objectives

5.2.2 Constraints

5.2.3 Compositions

5.2.4 Uses and Interactions

5.2.5 Interface

5.2.6 Resources

5.2.7 Base Station Software

Robot Operating System (ROS)

Graphical User Interface (GUI)

5.2.8 R-Transformation

5.2.9 M-Transformation

5.2.10 Human Interface Design

6 Ethical Considerations

7 References

8 Errata