CS Capstone Problem Statement

9 October 2017

Robotic Wheelchair Data Collection and Analysis

Prepared for

Oregon State University

Matthew William Shuman ________ *Signature* ________ *Date*

Prepared by

Group 4

Project Chiron

Marie Bomber ________ *Signature* ________ *Date*

Aaron Leondar ________ *Signature* ________ *Date*

Hadi Rahal-Arabi ________ *Signature* ________ *Date*

Abstract

For people that have extreme physical disabilities such as ALS, or for people who are paraplegic or even quadriplegic, the use of wheelchairs is necessary for even basic maneuverability. However, even with a wheelchair it is still marginally difficult to operate one while afflicted by these extremely serious conditions. Therefore, having a wheelchair able to perform tasks autonomously, such as moving from point to point, would help those kinds of people tremendously. This project involves collecting data from users using a wheelchair to determine the wheelchair users' proficiency in driving it. The ways that the users' proficiency will be tested are how cleanly they are able to make tight turns, slow down, and in general maneuver around a tight space in a limited amount of time. The end result of this project will be an interface for researchers to generate and analyze trial data on the robotic wheelchairs.

CONTENTS

1 Problem Definition

Project Chiron, an OSU venture, has developed a prototype kit for allowing those with extreme disabilities to interact with Permobil wheelchairs. While the software and hardware is existing, Project Chiron lacks an interface for data collection and analysis. Our project aims to allow for simple recording of trial data, as well as organizing the data for analysis. Furthermore, there is little documentation on the custom kit that can be mounted on wheelchairs, so it is necessary to document the applications and usage instructions of the kit as we interact with it.

The prototype hardware has several sets of sensors that have the capability of constantly collecting data. It will be necessary to develop an understanding of the value of the data collected from each sensor; the data may need to be truncated, or collected at set intervals.

Presenting the data in a usable format for analysis provides an entirely different challenge. While data collection will require some understanding of what each sensor is measuring, data analysis and formatting will require an intimate understanding of the purpose of each measurement, as well as the implications of those measurements.

2 Proposed Solution

Currently, the core solution can be broken into three parts, each part representing a layer in a user application that will allow for smooth interaction between a tester and the data a user test creates. Languages and libraries chosen were chosen due to their scalability and frequency of use, allowing the project to continue even after the project is completed and extended as client needs change.

1) **ROS bag transfer**
   During a user test, a large amount of data will be generated from the LIDAR, wheel data and other sensors. This information will be gathered into a ROS bag for storage and must be organized logically and filtered so only necessary data is kept to keep the storage size down. In an ideal scenario, all user tests will be stored within a 8Gb thumb drive for ease of transfer.
   Components of this stage will include connecting to the wheelchair using it's wireless network, meaningful error messages if a connection cannot be made and receiving the ROS bag data. All of this information will be filtered to only the relevant data (to be determined once a connection is established and sample ROS bag data can be obtained) and passed to the main storage module of the solution.
   This component will be created primarily with ROS and python and will pass information to the storage phase.

2) **Storage/Core Control**
   Once data has been collected, it will be passed to the main storage of the program. At this stage, the completed ROS bag data will be linked with the entered user data (adding a unique ID should one not already exist) and stored in the project database.
   This component will also be responsible for managing the fetch and retrieval of user data according to client needs and will provide the backbone of communication between the user interface and core data. Controls for this stage will include fetching and organizing results, auto populating form data, control of user ID's and deletion of said ID's as the client needs.

Like the previous component, this stage will use python as it's primary language, though SQL will define the database structure.

3) **User Interface**

The final component of this project will be a user interface that a tester will use to initiate the data collection, fill out any information that cannot be auto-populated and interface with the data as needed. This interface may be a web application, a graphical user interface, or both as time allows, with preference given to a web interface in order to increase portability. This stage will have different views for different user types, including a researcher/developer view (to be used by the client and project members), tester view (if a third party is needed to run a test) and any other levels of usability needed over the course of the research.

Currently, three main pages will need to be designed for this component. First, front page/login (password protected) will be needed to control access to the user data and allow for navigation between all other pages. Second a "run new test" page will be used to allow the tester to add a new user or add a test for an existing user and initiate data collection on the wheelchair. Lastly a "view results" page will allow the researcher to view the results for a specific user or for all users depending on the clients needs.

For this stage, python (likely through the pyjamas library) will be used to define the user interface, with Node.js to fill in any gaps that python cannot provided for a web interface.

3 Performance Metrics

The explicit deliverables and understandings of the project will be:

1) Robot Operating System Understand and Data Collection Organization

Understand how messages are formed and how ROSbags collect messages. Know enough about ROS to keep each ROSbag small enough that all ROSbags (approximately 120 bags) can fit on an 8GB flash drive.

2) Institutional Review Board Compliance

Anonymize IDs of participants so a failed trial or potentially embarrassing data cannot be traced back to the ID of that participant. The trials must be grouped by participant to show improvement, but the participant cannot be known.

3) Practical GUI

Make an application that is able to start and stop trial runs with just a button click. There should be few text fields that information will be entered, and auto-populates with the current time and date. The interface should look good and be able to be run by someone with very little experience.

4) Documentation of the System

Every piece of hardware that that did not come off the shelf should be documented so that the system could be recreated with little difficulty. All software used for the project will be documented.