12-1-2015

ROBOTIC WHEELCHAIR PLATFORM WITH AUTONOMOUS MOBILITY

Pragya Neupane

ROBOTIC WHEELCHAIR PLATFORM WITH

AUTONOMOUS MOBILITY

by

PRAGYA NEUPANE

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2015

ACKNOWLEDGMENTS

I would like to thank Dr. Christopher D. McMurrough, who has been an exemplary mentor and professor for me throughout the project over the last two semesters. This project would not have come to life without his support and guidance. I would also like to thank the College of Engineering for providing the tools and facilities necessary for the project.

I would like to thank my team, Team SNAPH, for working together and proving that together we can accomplish anything. Hector Ortiz, Nathan Hill, Ahad Virani, and Sulabh Siwakoti; you all have helped make this project the interesting project that it is today.

Finally, I would like to thank my family and friends for their blessings and constant encouragement during my academic career. I would not be the person I am today without your hard work and love.

January 20, 2016

ABSTRACT

ROBOTIC WHEELCHAIR PLATFORM WITH

AUTONOMOUS MOBILITY

Pragya Neupane, B. S. Computer Science Engineering

The University of Texas at Arlington, 2016

Faculty Mentor: Christopher D. McMurrough

An automated robotic wheelchair can allow for increased efficiency in terms of motion. This work details how through the use of computing, a wheelchair can be teleoperated to produce a system that can obviate the need for manual wheelchairs that can be difficult to operate. At this stage, the system is composed of various components and subsystems interacting to deliver precise control and navigation. By sending and receiving signals between the wheelchair platform and the computer, this system is able to communicate with the wheelchair platform and pass on instructions. The robotic wheelchair platform is able to record movement information, and can use this feedback to monitor movement, and attempt to improve any inaccuracy. Overall, the system works to maximize wheelchair efficiency of motion.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Purpose

This paper describes the process Team SNAPH underwent in designing and building a robotic wheelchair platform capable of receiving instructions via an external device such as a laptop and performing actions based on those instructions. It is the result of my learning as the "programming lead", overseeing the software required to operate the platform.

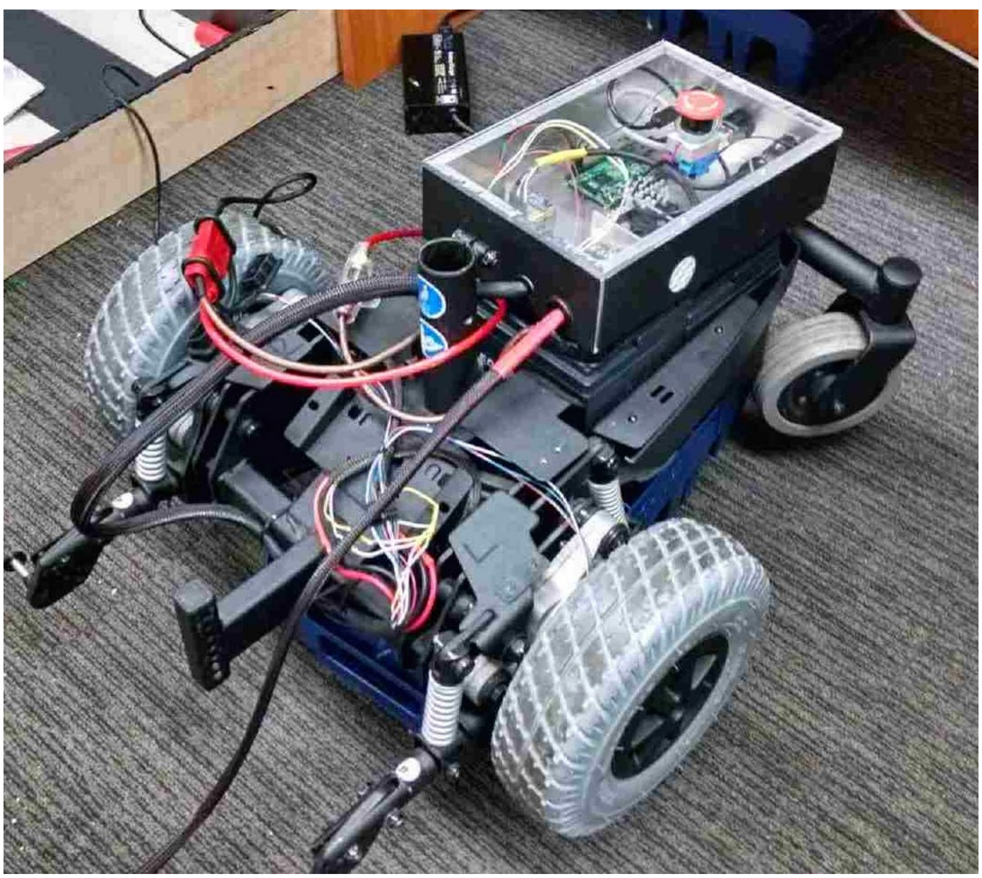

Figure 1.1: Robotic Wheelchair Platform Prototype

1.2 Motivation

As a team we realized that not much has been accomplished in the case of wheelchairs with autonomous mobility. This is what inspired us to take on the robotic wheelchair platform project. The robotic wheelchair platform is able to perform movements automatically. It does not require that the user continuously control the wheelchair platform. For me personally, the primary motivation for being a part of this project was the opportunity to challenge my software skills as well as gain hands-on hardware experience. I had the chance of managing the ZeroMQ messaging library, which is a library that allows for distributed messaging. It is a free, cross-platform, asynchronous distributed messaging library used to monitor the state and control the vehicle via updates that happen multiple times per second. ZeroMQ, since it is free, was very helpful for our team as we were on a limited budget. Also it was uncertain which programming language future teams would use, and so ZeroMQ was the best choice as it supports multiple languages such as Java, C, and Python. We had three programs running concurrently, one on the laptop, and two on the raspberry pi. These programs were able to communicate with each other and function accordingly with the help of ZeroMQ.

The final goal for this project was to design a robotic wheelchair that will compete in the highly competitive Intelligent Ground Vehicle Competition (IGVC). This means that it is long-term project that calls for the addition of several exciting features in the future. There are many possibilities for further improving this robotic wheelchair platform. One such potential is making it commercially available to wheelchair users, allowing them to utilize the autonomous capabilities in a wheelchair.

1.3 Scope

The goal of this project was to plan, design, and build a wheelchair platform that can carry out certain functions upon user instruction. In order to do that successfully, three things were essential; the ability to move, an emergency stop (e-stop) and a feedback control mechanism. Our primary motive was to get the wheelchair platform to move upon receiving instructions. It was crucial to calculate those movements in order to make sure that they were accurate to what was specified in the instructions. The details of how that was done will be explained later in the paper. It was equally important to incorporate an emergency stop (e-stop) to avert harm and existing hazards. This was implemented in two ways: manually, by needing to directly press on the e-stop button; and automatically, by cutting power via computerized instruction. Lastly, it was essential to have a feedback control mechanism that can regularly examine the distance traveled, position, and angle with respect to the ground, in order to make changes that are used to improve its output efficiency.

1.4 Approach

Two different development approaches were carried out to complete the project. In the first half of the project, the team decided to follow the waterfall model. But in the second half, we found it more efficient to follow agile development methodology. The waterfall-based approach is a sequential design process, whereas agile methodology is based on incremental planning. While the first model is carried out in five levels (requirements, design, implementation, verification, and maintenance), the latter follows no such pattern, allowing room for continuous improvement. The main reason for this shift to agile methodology was the flexibility of rapid and flexible response to change it permitted.

CHAPTER 2

ANALYSIS

2.1 Requirements

In proceeding with the project, the team found it practical to lay out the requirements first. They began doing so by first holding a meeting with the sponsor and getting a set of requirements. These requirements were then divided into four different categories; customer, packaging, performance, and safety requirements. The team was very specific as to what the requirements were and made strict guidelines to stick to the requirements. This helped avoid future feature creep such as obstacle avoidance, camera and camera vision.

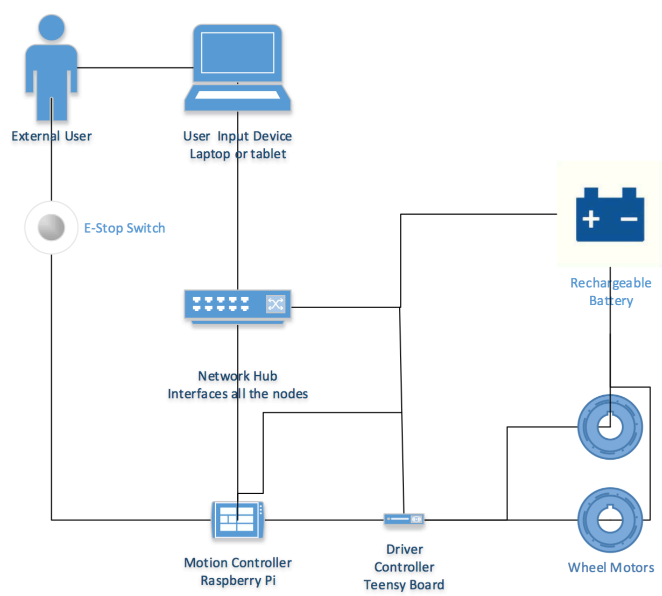

Figure 2.1: Vehicle Circuit Diagram

2.1.1. Customer Requirements

These were requirements specified by the project sponsor, Dr. Christopher D. McMurrough. They refer to the functionalities of the system and were essential to mark the completion of the project. The primary requirement was that the system should allow users to instruct the wheelchair to travel from its current position to a desired position. The movement must be continuous, and the wheelchair must not stray from its track. The platform must also be capable of accurately rotating a certain angle based on user instruction. In addition, the user must be able to stop operation of the wheelchair in order to prevent/avoid any possible damage, or harm. It must also regularly evaluate system performance with the goal of providing the user with improved performance. With respect to one of the many rules of the Intelligent Ground Vehicle Competition, the system is able to carry a payload of at least 20lb whose shape and size is approximately that of an 18" x 8" x 8" cinder block.

2.1.2. Packaging Requirements

This section covers all the packaging requirements of the robotic wheelchair platform. The wheelchair platform must come assembled, and act as a base for future teams to add features satisfying the competition requirements. All documents related to the platform should be accessible to these teams to help them understand the implementation of the platform. A user manual must be provided in the form of a printed document with instructions to operate the wheelchair platform. The user manual should be in the English language. Overall, it must act as a guide to future teams that will be adding features to the platform. Also, in compliance to the IGVC rule the platform should be no greater than 7 feet long, 4 feet wide and 6 feet tall. Lastly, in order to prevent the wires and any other components from being exposed to the surroundings and causing a safety hazard, the components of the wheelchair platform must be encased in a plastic housing.

2.1.3. Performance Requirements

This section details the performance requirements for the wheelchair platform. The requirements in this section provide detailed specifications of user interaction with the wheelchair platform and measurements placed on system performance. Primarily, the wheelchair platform must maintain a minimum speed of one mile per hour and must not exceed a speed of five miles per hour. As for the battery life, the motors should operate on a total of 24V drawing from two 12V batteries. These batteries should allow a constant drive time of one hour on flat ground and a continuous 44-foot travel distance. While performing these movements if there is a need to halt operation, the emergency stop must force the vehicle to stop within 0.5 seconds of the request. The wheelchair platform should contain a Micro-PC with a minimum speed of 700MHz quad-core processor and 512MB of RAM. It must also be able to determine a specified degree to which it turns when interpreting user directions.

2.1.4. Safety Requirements

This section details the safety requirements for the wheelchair platform. These requirements were designed to help ensure the safety of operators and maintainers of the wheelchair platform, to protect the wheelchair platform itself, and to avoid creating hazards that could lead to property and equipment damage. The mechanical e-stop button located on the center rear of the vehicle when pressed should cause the power to the motor to break. This must be a red button for easy recognition. Also as a safety precaution the motor must not be allowed to actuate under any condition while the vehicle is stalled. To reduce the risk of electric shock, all electrical connections must be safely packaged and properly grounded. Finally, the product must not contain sharp edges, to minimize the danger of injury to operators and maintainers.

2.2 Risk Analysis

All projects come with some risks associated with them. It is the role of the team to identify possible risks and create an action plan to tackle and manage those risks through planning.

2.2.1. Risk Identification and Control

This section outlines the risks that may impact the testing process of the system. The following table is a listing of identified test risks along with their relative impacts, severity and strategies to reduce their impacts on the test process.

Table 1.1: Risk Analysis Table

Risk ID	Risk	Impact	Severity	Mitigation Strategy
R1	Component failure due to physical damage	Major vehicle components may be damaged and need to be replaced	High	Visual inspection of vehicle before and after use to note signs of damage. Responsible use and safe storage of vehicle.
R2	Component failure due to faults in electrical subsystem.	Major vehicle components may be damaged and need to be replaced.	High	Ensure major components are fuse protected and periodically inspect electrical connections for signs of damage and degradation
R3	Bad Batteries	System may not function or function at reduced performance	High	Periodic testing of batteries and proper battery handling and storage
R4	Failure on subsystem integration	System may not function as intended	Medium	Individual subsystems will be tested and shown to be able contribute toward system operation prior to integration when such testing is possible
R5	Environment and test induced failures	Components and system may fail due to environment or test conditions	Medium	Anticipate systems intended use and test environment. Build the system and components to be able to withstand test conditions and beyond
R6	Connection failure	System may not function or function improperly	Low	Periodically inspect connections and have spare connections at hand if possible.

CHAPTER 3

DESIGN

This section describes the overall design of the robotic wheelchair platform. It details how information flows within its system by focusing on the architectural layers and the subsystems of these layers.

3.1 Architecture Design

The architecture design depicts the high-level overview of the platform. It is categorized into 4 major layers. These layers are User Interface layer, Processing layer, Network layer, and actuation layer.

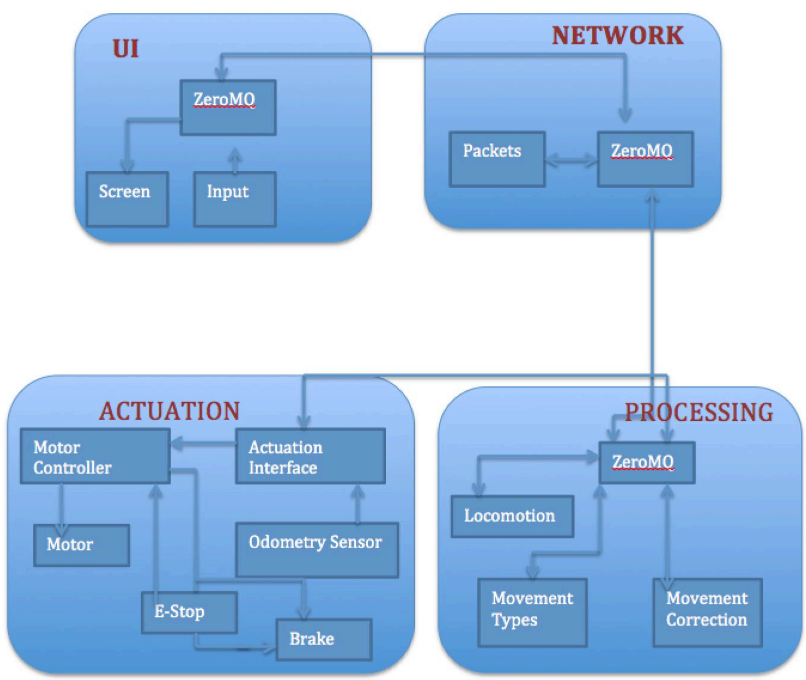

Figure 3.1: Architecture Overview

3.1.1. User Interface Layer

The user interface layer is responsible for providing a system that allows for the intake of user input. It controls the appearance and sends any processing task to the next layer down, the network layer. This layer does not directly communicate with the processing and actuation layers. User input is passed as commands and transferred by the user interface layer to the network layer. Commands follow a specific format as required by the system.

3.1.2. Network Layer

The network layer is the second layer of our architectural design. It is the layer that communicates with the user interface layer and processing layer making use of ZeroMQ libraries. The network layer consists of packets and the ZeroMQ interface. The user interface layer and the processing layer communicate through the network layer by sending information in the form of packets. The network layer conveys those instructions to the processing layer by relaying packets. The guideline used to define this layer was drawn from the necessity to have a medium in order to convey information from the user to the processor, and vice versa.

The network interface receives commands from the user interface. It then sends those data to the packets subsystem in order to create instructions for the processing layer. The network interface serves as an intermediary between the user interface layer and the processing layer.

The packets subsystem was used to create and send packets to the network interface. A packet consists of a packet number, protocol, destination address, and originating address. The packets subsystem is responsible to process information received from the user interface layer, and sending those packets to the processing layer.

3.1.3. Processing Layer

The purpose of the processing layer is to receive packets of instructions from the network layer and translate them into machine-readable code. This layer calculates the position of the platform and handles linear as well as angular movement. The processing layer receives voltage and other movement related information recorded by the odometry sensor and uses it to correct movement defects.

The processing layer receives and processes translated instructions to travel from one point to another, cover a certain distance, or turn a specified angle correctly and efficiently. It also receives movement data collected by the odometry sensor, which is used to correct any defects in the platform's movement.

3.1.4. Actuation Layer

The actuation layer is the bottom layer of the design. It is the layer that interfaces directly with the hardware of the vehicle. This layer provides the step between the instruction and application of the processor running the vehicle, and the actual movement that the vehicle executes. There is a motor controller that receives the instructions from the processor. This controller takes the instructions and transforms them into voltages that determine the speed and direction of the spinning wheels. It sends the voltage to the motors, which create movement. This layer also consists of the emergency stop, which interfaces with the controller and the brake of the vehicle. Activating the emergency stop notifies the brake to engage, and instructs the motor controller to cut power to the motor. The last subsystem is the odometry sensor. This continuously receives movement data from the spinning wheels and sends it back to the processor for the feedback system. It allows the processor to know how fast the wheels are spinning, which permits real-time corrections to the movement. The communication in this layer is established by interfacing directly with the Micro PC.

The actuation layer is composed of components such as the actuation interface, motor controller, motor, emergency stop, brake, and odometry sensor. Their roles are mentioned here. The actuation interface receives the data and directs them to the motor controller to be used for motion. It also receives data from the odometry sensor and passes them back to the processing layer to be used by the Micro PC. The actuation interface handles the communication with processing layer and the interface provided there. It acts as a broker, sending data received from other subsystems to the right destination.

The motor controller receives data from the processor, passed through the actuation interface, and converts them into voltages. It then sends those data to the motor to control movements. These data are formatted in a way such that the controller can decode the instructions and then convert them into voltage. The controller is a dual-channel device that can interface with two motors at the same time. It sends each motor a unique signal of voltage that instructs it on how to move.

The motor takes the voltage commands from the controller and creates movement. It turns the wheels in speeds and directions based on the input from the controller. There are two independently controlled motors that take voltage inputs and turn them into movement. They spin forward and backward, with speed controlled by the voltages received from the controller.

The Emergency stop is connected to the motor controller and the brake with a closed circuit. This circuit is engaged by a 'high' signal. When the Emergency stop is engaged, that signal is cut, and the motor controller and the brake recognize the change in voltage and execute the proper actions. The Emergency stop switch has to be manually dis-engaged by the user.

The brake is an electronic switch triggered by voltage and is used to stop all movement in the wheels. The switch is connected to the motor controller and the E-Stop. It is on the circuit of those devices, and responds to the change in the voltage from 'high' to 'low'. When this change is detected, the brake engages, stopping all rotation by the wheels.

The odometry sensor constantly gathers rotational data from the wheel to determine the speed of the vehicle. The data acquired from this sensor are sent to the actuation interface and on to the processor as feedback. The odometry sensor is attached to the motor. It constantly monitors the rotation of the wheel. The data collected are sent to the processor by way of the actuation interface. The data are sent without modification, so it is up to the processor to decode what is sent.

3.2 Detailed Design

The subsystems for each of the layers in the section above are broken down here to give a detailed anatomy of the system as a whole.

3.2.1. User Interface Layer Modules

The user interface layer sends and receives data to and from the network layer. The user interface layer does not directly communicate with the processing and actuation layers. It uses the screen subsystem, a module that provides users with a platform where they can enter their desired instructions. And to accept the input received through the screen, it uses the input subsystem.

3.2.2. Network Layer Modules

The network layer sends and receives data to and from the UI and processing layer. The network layer does not directly communicate with the actuation layer. The subsystems of this layer will not communicate with any external resources. The packets subsystem is used to create packets and send those packets to the network interface. A packet consists of a packet number, protocol, destination address, and originating address. OMQ is a multiplatform intelligent socket library for messaging. Its libraries are used to establish communication like that of a client and server.

3.2.3. Processing Layer Modules

The primary responsibility of the locomotion module in the processing layer is to receive movement directions from ZeroMQ and simply perform the desired movement. What kind of movement is executed is defined with the help of the movement types module. Once the movement is completed, the movement correction module receives movement data collected by the odometry sensor which is used to correct any defects in the platform's movement.

3.2.4. Actuation Layer Modules

The actuation interface receives data sent from the processing interface and directs that data to the motor controller. It also receives data from the odometry sensor and passes them back to the processing layer to be used by the raspberry pi. The raspberry pi is a microcomputer that consists of a program that receives messages from the laptop and sends those messages to the motor controller that in turn controls the motor. The actuation layer modules are responsible for motor control directions and data transfer via odometry sensor and processing layer to facilitate the Mini PC.

CHAPTER 4

IMPLEMENTATION

Planning and design are core processes in any system design. Our team spent a considerable amount of time brainstorming and preparing documents that helped to construct the system from the ground up. The robotic wheelchair platform project relies equally on the knowledge of both hardware and software. The majority of the team members, including myself, lacked any prior hardware experience, making this project a challenge and a great learning experience at the same time.

As part of the teamwork, we divided roles among team members based on individual strengths and weaknesses. These roles included project manager, programming lead, document master, procurement manager, and recorder. I took on the task of the programming lead. A programming lead is responsible for overseeing code required to run the system. As a programming lead I was also involved in managing the ZeroMQ messaging library, which allows for cross-platform communication. In addition to the assigned roles, all documents were divided among team members and compiled at last to produce the final product.

4.1 The Platform

The platform was salvaged from an old wheelchair donated by our sponsor Dr. Christopher D. McMurrough. Much of the circuit had to be built from scratch and attached to their respective ports. We removed the seat from the wheelchair and opened and cleaned the remaining area. We replaced the motor as well as the batteries with two new batteries as the old ones had worn out. A casing was created to house components such as the raspberry pi, and the wires were tied in place. The wheels were neither changed nor moved from their original location. Also a round image with black and white parts was attached to the inside of each front wheel allowing the encoders to keep track of the rotations. This information is essential to determine speed and the distance travelled by the platform.

4.2 Movement Initiation

The platform relies on user input to receive commands. Based on the commands issued, it performs the appropriate action. Commands are supplied in a specific format such as the distance is specified in terms of rotations per minute and speed in miles per hour. Each command is translated to machine-readable code. The motor controller is responsible for whether or not any current is sent. It is therefore also responsible for initiating the E-stop mechanism by ceasing the flow of current.

CHAPTER 5

TESTING

Since the beginning of the project, our team understood the importance of testing the system. We realized that it was highly important to test the system extensively to make sure it was dependable.

5.1 Approach

This section describes in detail how Team SNAPH carried out the testing approach and maintained records of the tests.

5.1.1 Overall Test Strategy

The Team utilized both black-box and white-box testing to ensure that the system met the requirements defined in the SRS. black-box testing is the testing of the external structures like the functionality of any system and white-box testing is the testing of the internal structures or workings of any system. The tests were documented utilizing the following items:

- Test ID
- Test Type (unit, component, singular, etc.)
- Test Date
- Results and outputs
- Pass or Fail
- Comments, details, and plans to correct errors, and severity of errors.

5.1.2 Hardware and Software Configurations

A number of hardware and software components were tested to ensure that they were reliable enough to handle their tasks. Then plans to ensure that the defects in the hardware were minimal were carried out. The following items were used to test both configurations:

- Test Name/Test ID
- Test Type Hardware/Software
- Fixed/Ignored
- Results
- Pass or Fail
- Comments, details, contingency plans and severity of errors.

5.1.3 Testing Metrics

The success of the prototype was measured through our test cases and their priority in the requirements list. A successful prototype must ensure that all high priority requirements are fulfilled and the acceptance criteria are met. This warrants that the prototype is acceptable to the end user and all stakeholders, before the product is considered for further production.

5.2 Features Tested

The robotic wheelchair is a long-term project. This means that all functionalities implemented by Team SNAPH were only for the robotic wheelchair platform. Here we discuss how those functionalities were tested.

5.2.1 Instruction-Based Movement

The team sent input commands and tested whether the platform performed those commands. In doing so the team made sure that the movement was continuous, and the wheelchair did not stray from its track. A message was received after every action was successfully completed.

5.2.2 Feedback Control Mechanism

To ensure this function the team collected information about how the system was functioning with the help of the encoder.

5.2.3 Angular Movement (spin)

Similar to the instruction based movement feature, input commands were sent for angular movement and it was tested whether these commands were performed. A message was received after each angular movement was successfully completed.

5.2.4 Payload

The team put a load of 20lb block on the wheelchair base, and observed whether it was able to carry the load. The same was tested for when the wheelchair was in motion.

5.2.5 Vehicle Speed

The team tested the speed of the wheelchair platform based on the number of rotations performed by the tires and setting a limit on the maximum current that can be drawn.

5.2.6 Vehicle Emergency Stop

The team ensured the functionality of this feature by instructing the platform to move, and pressing the e-stop button to stop the motion. The e-stop was tested both manually and automatically by supplying the appropriate instruction.

CHAPTER 6

RESULTS AND CONCLUSION

6.1 Results

The robotic wheelchair platform has succeeded in carrying out all of team SNAPH's high-priority requirements. It is capable of performing both linear and angular movements without halting. It is also safe to operate due to the emergency stop feature. However, there is a slight error in the measure of the distance. As a result there is a minor difference in the distance instructed to travel and the actual distance travelled. One of our top priorities was to have the platform maintain a constant speed while moving. We were able to achieve this feature on smooth surfaces. The wheelchair platform utilizes a 'heartbeat' feature which is a message sent by the laptop to the raspberry pi. The raspberry pi in turn sends a blank message each time it receives a heartbeat. This helps to find whether the laptop and the raspberry pi are able to communicate to each other or not.

6.2 Conclusion

As a team, we are gratified that the platform was able to meet all its high priority requirements. Given the short amount of time, it was important to stay on track with the original plan and design. It was the application of both the waterfall model and the agile model that allowed us to do that. We were given a budget of $800 that we used very carefully, since some of the parts of the wheelchair platform such as the battery and the motor were quite expensive. The robotic wheelchair platform called for the incorporation of software into hardware allowing me to receive valuable hands on experience in robotic projects. Working with ZeroMQ helped me oversee the structure of messaging and information flow between the various programs. I was exposed to multiple messaging patterns and common practices. Lastly, based on my experience with the ZeroMQ messaging system, I am confident that I will be able to work with any messaging system in the future.

APPENDIX A

SYSTEM REQUIREMENTS SPECIFICATION

1. Product Concept

This section describes the purpose, use, and intended user audience for the Wheelchair Platform. The Wheelchair Platform is an electronic wheelchair base that has the ability to move a certain distance specified by the user. Users of the Wheelchair Platform shall be able to direct the device to move a certain distance at a specified speed.

1.1 Purpose and Use

The Wheelchair Platform is a base product that shall be further developed in order to compete in the Intelligent Ground Vehicle Competition. The Wheelchair Platform shall be able to take directions from the user, and move a specified distance. For example, the user may request that the wheelchair base move seven feet forward, then stop at a forty-five degree angle. The features of the Wheelchair Platform shall increase as future teams increase its functionality.

1.2 Intended Audience

The intended audiences of the Wheelchair Platform are future Senior Design students who shall continue to develop the product. The audience of the finished product shall be the judges at the Intelligent Ground Vehicle Competition. This product shall not be commercialized, as only one shall be designed in order to compete in a competition.

2. Product Description and Functional Overview

This section contains an overview of the description and features of the Intelligent Ground Vehicle. The most important features, functions, and operations are described here, from the perspective of the customer and the maintainers. It also discusses the user interactions, specific input/output data flows and the user interface in detail.

2.1 Features and Functions

The primary functions and features of the Vehicle are described as follows:

- The Vehicle shall move in forward and backward directions, with the ability to turn in place to any heading that is requested.
- It shall be take input of headings and distances, and execute the commands to position itself within a fixed radius of the target locations.
- It shall execute its movement with a specific speed and the ability to turn in an arc, through the use of controllers and drivers,
- It shall take inputs from a user, or other devices on its network, and understand the commands to execute the path.
- It shall have an emergency stop switch that shall stop all the motors.

2.2 Product Interfaces

The main user interface shall leverage a laptop or a computer tablet with a screen and keyboard. The screen shall provide graphic user interfaces to see feedback and the input data that is being sent to the vehicle. The keyboard shall provide the necessary means for input, so that the user can control the vehicle in any way necessary.

Another important interface is the e-stop switch that shall be used in the case of any emergency. E-stop shall be a simple button or trigger switch so that the command shall be easy and immediate.

The vehicle shall also interface with a network of other computers that shall be not be implemented in this design. It must be able to receive commands from that network that it can execute without any problem. There shall be a main network hub, which shall allow all the single nodes to interface with one another; the extent of this design shall be one of the nodes in that network.

3. Customer Requirements

This section details the hardware and software requirements set forth by the customer, as well as the development team to further the functionality of the system. Since this project is the first phase of a long-term project, the requirements listed here are strictly related to this phase of the project. The requirements below are ranked on a scale of 1to 5, with 1 being the highest priority, and 5 being the lowest priority.

3.1 Instruction Based Movement

3.1.1 Description: The system shall allow users to instruct the wheelchair to travel from its current position to a desired position. The movement must be continuous, and the wheelchair must not stray from its track.

3.1.2 Source: Dr. Christopher McMurrough

3.1.3 Constraints: Path must be free of objects.

3.1.4 Standards: None

3.1.5 Priority: Critical

3.2 Feedback Control Mechanism

3.2.1 Description: The system performance shall be evaluated regularly, providing the user with improved performance.

3.2.2 Source: Dr. Christopher McMurrough

3.2.3 Constraints: None

3.2.4 Standards: None

3.2.5 Priority: High

3.3 Debugging Mode

3.3.1 Description: The debugging mode shall show information of all vehicle hardware input to the system. The debugging mode shall allow the user to manually adjust individual vehicle hardware components.

3.3.2 Source: Dr. Christopher McMurrough

3.3.3 Constraints: Some vehicle components are interdependent and can't be individually manipulated.

3.3.4 Standards: None

3.3.5 Priority: High

3.4 Angular movement (spin)

3.4.1 Description: The system shall accurately rotate a certain angle based on user instruction.

3.4.2 Source: Dr. Christopher McMurrough

3.4.3 Constraints: Spin path must be free of obstacles

3.4.4 Standards: None

3.4.5 Priority: Critical

3.5 Payload

3.5.1 Description: The system shall carry a payload of at least 20lb.

3.5.2 Source: Dr. Christopher McMurrough

3.5.3 Constraints: None

3.5.4 Standards: The shape and size of the payload is approximately that of an 18" x 8" x 8" cinder block.

3.5.5 Priority: Critical

4. Packaging Requirements

This section covers all the packaging requirements of the first phase of the wheelchair robot that shall later be competing in the IGVC competition. The wheelchair platform shall come assembled, and shall act as a base for future teams to add features satisfying the competition requirements. All documents related to the platform shall be provided to these teams to help them understand the implementation of the platform, and speed up their addition process.

4.1 Full User Manual

4.1.1 Description: A user manual shall be provided with instructions to operate the wheelchair platform. It shall be a printed document.

4.1.2 Source: Team SNAPH

4.1.3 Constraints: None

4.1.4 Standards: The user manual shall be in English language

4.1.5 Priority: Low

4.2 Platform Size

4.2.1 Description: The platform shall be less than 7 feet long, 4 feet wide and 6 feet tall.

4.2.2 Source: IGVC

4.2.3 Constraints: None

4.2.4 Standards: None

4.2.5 Priority: Critical

4.3 Platform Housing

4.3.1 Description: The components of the wheelchair platform shall be encased into a plastic housing for security.

4.3.2 Source: IGVC

4.3.3 Constraints: None

4.3.4 Standards: None

4.3.5 Priority: High

5. Performance Requirements

This section details the Performance Requirements for the Wheelchair Platform. The requirements in this section provide detailed specifications of user interaction with the Wheelchair Platform and measurements placed on system performance.

5.1 Vehicle Speed

5.1.1 Description: The Wheelchair Platform must maintain a minimum speed of one mile per hour. The Wheelchair Platform must not exceed a speed of five miles per hour.

5.1.2 Source: IGVC

5.1.3 Constraints: Terrain, slope

5.1.4 Standards: None

5.1.5 Priority: High

5.2 Vehicle Battery

5.2.1 Description: The Wheelchair Platform battery must pass the threshold operation time of one hour.

5.2.2 Source: IGVC

5.2.3 Constraints: Speed, payload

5.2.4 Standards: Onboard electronics

5.2.5 Priority: High

5.3 Vehicle Payload

5.3.1 Description: The Wheelchair Platform shall carry a 20-pound payload.

5.3.2 Source: IGVC

5.3.3 Constraints: None

5.3.4 Standards: 18” x 8” x 8” cinder block

5.3.5 Priority: High

5.4 Distance Traveled

5.4.1 Description: The Wheelchair Platform shall travel a continuous 44-foot distance.

5.4.2 Source: IGVC

5.4.3 Constraints: Speed, Payload

5.4.4 Standards: None

5.4.5 Priority: High

5.5 Vehicle Emergency Stop

5.5.1 Description: The Wheelchair Platform emergency stop shall force the vehicle to stop within 0.5 seconds of the request.

5.5.2 Source: IGVC

5.5.3 Constraints: None

5.5.4 Standards: None

5.5.5 Priority: High

5.6 Micro-PC

5.6.1 Description: The Wheelchair Platform shall contain a Micro-PC with a minimum speed of 700MHz quad-core processor and 512MB of RAM.

5.6.2 Source: IGVC

5.6.3 Constraints: None

5.6.4 Standards: None

5.6.5 Priority: Moderate

5.7 Directional Capabilities

5.7.1 Description: The Wheelchair Platform shall determine a specified degree at which it faces when interpreting user direction.

5.7.3 Constraints: None

5.7.4 Standards: None

5.7.5 Priority: High

6. Safety Requirements

This section details the safety requirements for the wheelchair platform. These requirements shall help ensure the safety of operators and maintainers of the wheelchair platform, to protect the wheelchair platform itself, and to avoid creating hazards that could lead to property and equipment damage.

6.1 Emergency Stop

6.1.1 Description: A mechanical emergency stop button must be located on the center rear of the vehicle and should break power to the motor when pressed.

6.1.2 Source: Dr. Christopher McMurrough

6.1.3 Constraints: None

6.1.4 Standards: None

6.1.5 Priority: Critical

6.2 Kill Motor on Stall

6.2.1 Description: The motor should not actuate when the vehicle is stalled.

6.2.2 Source: Dr. Christopher McMurrough

6.2.3 Constraints: Implementation of this requirement shall depend on accurate stall detection.

6.2.4 Standards: None

6.2.5 Priority: High

6.3 Electrical Packaging

6.3.1 Description: All electrical connections shall be safely packaged to reduce the risk of electric shock to operators and maintainers.

6.3.2 Source: Dr. Christopher McMurrough

6.3.3 Constraints: None

6.3.4 Standards: None

6.3.5 Priority: High

6.4 Electrical Grounding

6.4.1 Description: All electrical connections shall be properly grounded to reduce the risk of electric shock to operators and maintainers.

6.4.2 Source: Dr. Christopher McMurrough

6.4.3 Constraints: None

6.4.4 Standards: None

6.4.5 Priority: High

6.5 Safe Edges

6.5.1 Description: The product shall not contain sharp edges to reduce the risk of cuts to operators and maintainers.

6.5.2 Source: Dr. Christopher McMurrough

6.5.3 Constraints: None

6.5.4 Standards: None

6.5.5 Priority: Low

6.6 Maximum Current Draw

6.6.1 Description: The current draw shall not exceed what the power supply can provide.

6.6.2 Source: Dr. Christopher McMurrough

6.6.3 Constraints: None

6.6.4 Standards: None

6.6.5 Priority: Critical

6.7 Fuse Protection

6.7.1 Description: The power supply shall be fused.

6.7.2 Source: Dr. Christopher McMurrough

6.7.3 Constraints: None

6.7.4 Standards: None

6.7.5 Priority: High

7. Maintenance and Support Requirements

This section outlines the maintenance and support requirements for the wheelchair platform upon final product delivery. Due to the incremental nature of the development of the unmanned ground vehicle platform, these requirements shall play a vital role in ensuring that future teams can thoroughly understand and successfully build upon the platform.

7.1 Source Code Documentation

7.1.1 Description: All code, algorithms, interfaces, and API shall be properly commented.

7.1.2 Source: Team SNAPH

7.1.3 Constraints: None

7.1.4 Standards: None

7.1.5 Priority: Low

7.2 User Manual

7.2.1 Description: The team shall provide a user manual detailing the capabilities of the wheelchair platform, details about operation instructions, and access of major platform components such as battery, sensors, and onboard computers.

7.2.2 Source: Team SNAPH

7.2.3 Constraints: None

7.2.4 Standards: None

7.2.5 Priority: Moderate

7.3 Documentation Availability

7.3.1 Description: Final versions of document deliverables shall be made available to Dr. McMurrough so that future teams involved with the platform can access them.

7.3.2 Source: Dr. Christopher McMurrough

7.3.3 Constraints: None

7.3.4 Standards: None

7.3.5 Priority: Moderate

7.4 Defect Documentation

7.4.1 Description: Team SNAPH shall provide a document enumerating latent defects in the platform.

7.4.2 Source: Team SNAPH

7.4.3 Constraints: None

7.4.4 Standards: None

7.4.5 Priority: Moderate

7.5 ROS Robotic Standards

7.5.1 Description: The Wheelchair Platform application shall leverage the ROS operating system.

7.5.2 Source: Dr. Christopher McMurrough

7.5.3 Constraints: None

7.5.4 Standards: ROS

7.5.5 Priority: High

8. Other Requirements

This section shall detail all other requirements that have not been set forth, and that do not identify with the other categories of this document. The requirements shall be presented with details about source, priority, etc.

8.1 Coding Language

8.1.1 Description: The programs written for the controllers of the vehicle shall all be in the same language, and that common language shall be C++

8.1.2 Source: Team SNAPH

8.1.3 Constraints: Coding Language

8.1.4 Standards: C++

8.1.5 Priority: High

8.2 Command Packets

8.2.1 Description: The packets used for sending commands between the nodes shall all conform to the same rules, standards, and format, so that any functionality added after the completion of this phase can be done with ease.

8.2.2 Source: Dr. Christopher McMurrough

8.2.3 Constraints: None

8.2.4 Standards: Format of the packets sent and received by network nodes

8.2.5 Priority: High

8.3 IGVC Rules and Regulations

8.3.1 Description: Every aspect of this vehicle shall conform to the rules of the Intelligent Ground Vehicle Competition, ensuring the ability to compete when the product is complete.

8.3.2 Source: IGVC

8.3.3 Constraints: Any changes in IGVC Rules and Regulations

8.3.4 Standards: IGVC

8.3.5 Priority: Critical

8.4 Performance to win the Competition

8.4.1 Description: The product and all controllers and interfaces shall perform in such a way that the vehicle shall have the ability to outperform other competitors and succeed in the competition.

8.4.2 Source: Dr. Christopher McMurrough

8.4.3 Constraints: None

8.4.4 Standards: None

8.4.5 Priority: Low

9. Acceptance Criteria

The primary stakeholders of the project agree that the Wheelchair Platform shall be accepted as a completed product once the system being designed met the following criteria. The sponsor has agreed to verify each criterion by following the included verification procedures.

9.1 The system shall use a mobile vehicular platform.

9.1.1 Requirement(s) addressed: 3.1

9.1.2 Verification Procedure: The sponsor shall verify that the product may be movable to a certain distance and meets sponsor aesthetic standards.

9.2 The vehicle system shall be connected to the microcontroller.

9.2.1 Requirement(s) addressed: 8.1

9.2.2 Verification Procedure: For wireless connectivity, Bluetooth recognition of the microcontrollers shall be apparent on the mobile device's display. A visual inspection by the sponsor may verify the wireless connection or a hardwired connection.

9.3 The microcontroller shall control the vehicle hardware interfaced with the system.

9.3.1 Requirement(s) addressed: 8.3, 8.4

9.3.2 Verification Procedure: The sponsor shall verify that the system can control all interfaced vehicle hardware by adjusting individual component settings within the ‘debug’ GUI.

9.4 The system application shall have a debugging mode.

9.4.1 Description: The debugging mode shall show information of all vehicle hardware input to the system. The debugging mode shall allow the user to manually adjust individual vehicle hardware components.

9.4.2 Requirement(s) addressed: 3.3

9.4.3 Verification Procedure: When the user touches the individual parts on the debug menu, it should show the information about that individual component. When the user picks the functionality that is specific to that hardware, that functionality should be performed by the respective hardware part.

9.5 Electrical wiring shall be insulated.

9.5.1 Requirement(s) addressed: 6.3, 6.4

9.5.2 Verification Procedure: The sponsor shall verify that performing a visual inspection of the prototype installation insulates wiring

APPENDIX B

ARCHITECTURE DESIGN SPECIFICATION

1. Brief Introduction

The Wheelchair Platform is an electronic wheelchair base that has the ability to move a certain distance specified by the user. Users of the Wheelchair Platform will be able to direct the device to move a certain distance at a specified speed. The device will contain a feedback control mechanism, which consistently examines the distance traveled and angular position. The design will also incorporate an emergency stop (e-stop) to avert harm or to reduce existing hazards. A Raspberry Pi 2 configured with the Robotics Operating System (ROS) will govern the movement of the Wheelchair Platform.

Network Layer

The Network Layer is the second layer of our architectural design. It is the layer that will communicate with the User Interface Layer and Processing Layer using the Robot Operating System (ROS). The Network Layer consists of packets and the ROS interface. The User Interface Layer and the Processing Layer will communicate through the Network Layer by sending information in the form of packets. The Network Layer will accept instructions in the form of packets from the User Interface Layer, and then convey those instructions to the Processing Layer by relaying packets. The guideline used to define this layer was drawn from the necessity to have a medium in order to convey information from the User to the Processor, and vice versa.

1.1 Network Interface

General Description: The Network Interface will use ROS in order to receive the data sent from the User Interface. It will then send that data to the Packets Subsystem in order to create instructions for the Processing Layer.

Assumptions: The interfaces will have a good connection and there will be minimal noise introduced to those connections.

Responsibilities: The Network Interface will serve as an intermediary between the User Interface Layer and the Processing Layer.

1.2 Packets Subsystem

General Description: The Packets Subsystem will be used to create packets and send those packets to the Network Interface. A packet consists of a packet number, protocol, destination address, and originating address.

Assumptions: The interfaces will have a good connection and there will be minimal noise introduced to those connections.

Responsibilities: The Packets Subsystem is responsible to process information received from the User Interface Layer, and sent to the Processing Layer.

The different layers are described below:-

2. Layers

2.1 Processing Layer

The purpose of the processing layer is to receive packets of instructions from the network layer and translate them into machine-readable code. This layer will calculate the position of the platform and handle movements from one point to another point and angular spin. The processing layer will be executed with the help of the ROS platform. The processing layer will also receive voltage and other movement related information recorded by the odometry sensor. This information will be used to correct any movement defects.

2.1.1 Locomotion

General Description: This subsystem will function by receiving translated instructions from ROS. It will receive instructions to travel from one point to another, or to cover a certain distance.

Assumptions: All user instructions are translated to become machine-readable.

Responsibilities: This subsystem is responsible for executing movement instructions correctly and efficiently.

2.1.2 Movement Types

General Description: This subsystem will function by receiving translated instructions from ROS. It will receive instructions to travel at a certain direction, or turn a specified angle.

Assumptions: All user instructions are translated to become machine-readable.

Responsibilities: This subsystem is responsible for executing movement instructions correctly and efficiently.

2.1.3 Movement Correction

General Description: This subsystem will function by receiving movement data collected by the odometry sensor. Which will in turn be used to correct any defects in the platform's movement.

Assumptions: It is assumed that the odometer makes correct readings each time.

Responsibilities: This subsystem is responsible for correcting any performance deficiencies in the platform and improving performance.

2.2 Actuation Layer

The actuation layer is the bottom layer of the design. It is the layer that will directly interface with the hardware of the vehicle. This layer will provide the step between the instruction and application of the processor running the vehicle, and the actual movement that the vehicle executes. First, there will be a motor controller that receives the instructions from the processor. This controller will take the instructions and transform them into voltages that will determine the speed and direction of the spinning wheels. It will send the voltage to the motors, which will use that to create movement. Another subsystem of the layer is the E-Stop, which will interface with the controller and the brake of the vehicle. The E-Stop will be a hardware switch that the user will manually engage. It will then notify the brake to engage, and the motor controller to cut power to the motor. The last subsystem is the odometry sensor. This will be continuously getting movement data from the spinning wheels and sending it back to the processor for a feedback system. It will allow the processor to know how fast the wheels are spinning, and this will allow for real time corrections to the movement. All of the communication of this layer will be to the Processing layer, because it will interface directly with the micro pc.

2.2.1 Actuation Interface

General Description: The Actuation interface will receive the data sent from the Processing ROS interface. It will then direct that data on to the motor controller to be used for motion. It also will take in data from the odometry sensor and pass that back to the Processing layer to be used by the Mini PC.

Assumptions: The interfaces will have a good connection and there will be minimal noise introduced to those connections.

Responsibilities: The Actuation interface will handle the communications with Processing layer and the interface provided there. It really is just a pass-through, it will send the data it received from other subsystems to the right destination. It will need to have a secure connection, as any interference or loss of data can result in problems for the vehicle, as motion and other actions can pose harm to surroundings.

2.2.2 Motor Controller

General Description: The Motor Controller will take in the data from the Processor, passed through the actuation interface, and convert it into voltages to send to the motor to control the movement.

Assumptions: Data is sent in the proper format and contains real instructions.

Responsibilities: The motor controller will be receiving data from the ROS framework, as that is the method that the Processor is sending the data. That data will be formatted a specific way, so the controller can decode the instructions and then turn that into voltage. The controller will be a dual-channel, so it can interface with two motors at the same time. It will send each motor a unique signal of voltage that will instruct it on how to move.

2.3 Motor

General Description: The motor will be taking the voltage commands from the controller and creating movement. It will turn the wheels in speeds and directions based on the input from the controller.

Assumptions: Voltage signals are in the correct range.

Responsibilities: There will be two independently controlled motors. They will take voltage inputs and turn those into movement. They will not have any feedback or output. They will spin forwards and backwards and the speeds will be controlled by the voltages they receive from the controller.

2.4 E-Stop

General Description: This is a mechanical switch that will send a signal to the controller and brake that will force the vehicle to stop all movement.

Assumptions: User understands the operation of the switch, and the lines connected are correct.

Responsibilities: The E-Stop will be connected to the motor controller and the brake with a closed circuit. This circuit will be engaged, in a HIGH signal. When the E-Stop is engaged, that signal will be cut, and the motor controller and the brake will recognize the change in voltage and execute the proper actions. The E-Stop will have to be manually dis-engaged by the user.

2.5 Brake

General Description: The brake is an electronic switch that can be triggered by voltage, and it then stops all movement in the wheels.

Assumptions: Mechanical component in wheels works properly.

Responsibilities: This switch will be connected to the motor controller, as well as the E-Stop. It will be on the circuit from those devices, and will respond to the change in the voltage from HIGH to LOW. When this change is detected, the brake will engage, stopping all rotation by the wheels.

2.6 Odometry Sensor

General Description: The odometry sensor is constantly gathering rotational data from the wheel, to determine at any time what the speed of the vehicle is. The data acquired from this sensor is sent to the actuation interface and on to the Processor where it will be used as feedback.

Assumptions: The image on the wheel is attached correctly, so that the sensor acquires accurate data, and the Processor can interpret correctly the data sent.

Responsibilities: The odometry sensor needs to be attached on the motor. It will constantly be monitoring the rotation of the wheel. The data that it collects will be sent to the processor by way of the actuation interface. The data will be sent without modification, so it is up to the processor to decode what is sent.

3. Testing Considerations

This section describes the way in which the Wheelchair Platform architecture will be verified and validated. These testing considerations will ensure that the designed system can support the requirements described in the System Requirements Specification. By evaluating layer reliability, the overall reliability and performance of the platform as a whole can also be evaluated and improved upon.

3.1 Overall Considerations

Modularity: All system layers will be independent. The individual system layers will not depend upon internal mechanisms of another layer.

Code Review: Code will be frequently reviewed to verify that it conforms to specification requirements. Frequent code reviews will lead to early error identification and fault prevention.

3.2 UI Layer

Internal-System Interaction: The UI Layer will send and receive data to and from the Network Layer. The UI layer will not directly communicate with the Processing and Actuation layers

External-System Interaction: The subsystems of this layer will not communicate with any external resources.

3.3 Network Layer

Internal-System Interaction: The Network Layer will send and receive data to and from the UI and Processing layers. The Network layer will not directly communicate with the Actuation layer.

External-System Interaction: The subsystems of this layer will not communicate with any external resources.

3.4 Processing Layer

Internal-System Interaction: The Processing Layer will send and receive data to and from the Network and Actuation Layers.

External-System Interaction: The subsystems of this layer will not communicate with any external resources.

3.5 Actuation Layer

Internal-System Interaction: The Actuation Layer will send and receive data to and from the Processing. The Actuation layer will not directly communicate with the UI and Network layers.

External-System Interaction: The subsystems of this layer will not communicate with any external resources.

APPENDIX C

DETAILED DESIGN SPECIFICATION

1. Introduction

This document provides the detailed design for the Wheelchair Platform project, an autonomous vehicle. As defined in the Architecture Design Specification (ADS) the system consists of four layers - User Interface, Network, Actuation and Processing layers. The subsystems for each of these layers are also detailed in the ADS and will be summarized at the beginning of this document.

The component design sections shall further define the subsystems by identifying the key modules and components that make up each subsystem. Each module is described in these sections by providing the purpose, dataflow, physical data structures, dependencies, and process (including pseudo-code whenever applicable). The interfaces between each module will also be described within this document as specifically as possible.

At the end of the document, the detailed design shall discuss about quality assurance to verify that the design meets the requirements defined in the System Requirements Specification (SRS) document. This will be illustrated through the use of a requirements traceability matrix. Finally, the completion of the Wheelchair Platform project shall be laid out by the acceptance plan.

2. Architecture Overview

This section gives a general overview of the architecture for Sherpa Drone and all of the layers outlined in the Architectural Design Specification document.

2.1 Architecture Description

The architecture is broken down into four layers. These layers are UI (user input) layer, Network layer, Processing layer, and Actuation layer.

2.1.1 UI Layer

The UI Layer is responsible for providing a system that allows receiving user input. It controls the appearance and accepts user input through a laptop. It sends any processing tasks to the next layer down, the Network Layer. This layer will not directly communicate with the Processing and Actuation layers.

2.1.2 Network layer

The Network layer controls communication between the User Interface Layer and Processing Layer. The Network Layer consists of packets. The User Interface Layer and the Processing Layer will communicate through the Network Layer by sending information in the form of packets. The Network Layer will accept instructions in the form of packets from the User Interface Layer, and then convey those instructions to the Processing Layer by relaying packets. The guideline used to define this layer was drawn from the necessity to have a medium in order to convey information from the User to the Processor, and vice versa.

2.1.3 Processing Layer

The Processing layer receives packets of instructions from the network layer and translates them into machine-readable code. This layer will calculate the position of the platform and handle movements from one point to another point and angular spin. The processing layer will be executed with the help of the ROS platform. The processing layer will also receive voltage and other movement related information recorded by the odometry sensor. This information will be used to correct any movement defects.

2.1.4 Actuation Layer

It is the layer that will directly interface with the hardware of the vehicle. This layer will provide the step between the instruction and application of the processor running the vehicle, and the actual movement that the vehicle executes. First, there will be a motor controller that receives the instructions from the processor. This controller will take the instructions and transform them into voltages that will determine the speed and direction of the spinning wheels. It will send the voltage to the motors, which will use that to create movement. Another subsystem of the layer is the E-Stop, which will interface with the controller and the brake of the vehicle. The E-Stop will be a hardware switch that the user will manually engage. It will then notify the brake to engage, and the motor controller to cut power to the motor. The last subsystem is the odometry sensor. This will be continuously getting movement data from the spinning wheels and sending it back to the processor for a feedback system. It will allow the processor to know how fast the wheels are spinning, and this will allow for real time corrections to the movement. All of the communication of this layer will be to the Processing layer, because it will interface directly with the micro pc. The guideline used to define this layer was the need to interface the processor and the actual hardware of the vehicle.

2.2 UI Module Descriptions

2.2.1 Internal-System Interaction

The UI Layer will send and receive data to and from the Network Layer. The UI layer will not directly communicate with the Processing and Actuation layers.

2.2.2 External-System Interaction

The subsystems of this layer will not communicate with any external resources.

2.2.3 Screen Subsystem

The Screen Subsystem is a module whose responsibility is to provide users with a platform where they can enter their desired instructions.

2.2.4 Input Subsystem

Input Subsystem is a module that has the sole responsibility of accepting the input received through the screen.

2.3 Network Module Descriptions

2.3.1 Internal-System Interaction

The Network Layer will send and receive data to and from the UI and Processing layers. The Network layer will not directly communicate with the Actuation layer.

2.3.2 External-System Interaction

The subsystems of this layer will not communicate with any external resources.

2.3.3 Packets Subsystem

The Packets Subsystem will be used to create packets and send those packets to the Network Interface. A packet consists of a packet number, protocol, destination address, and originating address.

Assumptions: The interfaces will have a good connection and there will be minimal noise introduced to those connections.

Responsibilities: The Packets Subsystem is responsible to process information received from the User Interface Layer, and sent to the Processing Layer.

2.3.4 0MQ Subsystem

0MQ is a multiplatform intelligent socket library for messaging. Its libraries are to be used to establish communication like that of a client and server.

2.4 Processing Module Descriptions

2.4.1 Locomotion Subsystem

The locomotion module functions by receiving instructions from 0MQ. It will receive instructions to travel from one point to another, or to cover a certain distance.

Assumptions: All user instructions are translated to become machine-readable.

Responsibilities: This subsystem is responsible for executing movement instructions correctly and efficiently.

2.4.2 Movement Types Subsystem

The Movement Types module functions by receiving instructions from 0MQ. It will receive instructions to travel from one point to another, or to cover a certain distance.

Assumptions: All user instructions are translated to become machine-readable.

Responsibilities: This subsystem is responsible for executing movement instructions correctly and efficiently.

2.4.3 Movement Correction Subsystem

The Movement Correction module functions by receiving movement data collected by the odometry sensor. Which will in turn be used to correct any defects in the platform's movement.

Assumptions: It is assumed that the odometer makes correct readings each time.

Responsibilities: This subsystem is responsible for correcting any performance deficiencies in the platform and improving performance.

2.5 Actuation Subsystem

2.5.1 Actuation Interface

The Actuation interface will receive the data sent from the Processing ROS interface. It will then direct that data on to the motor controller to be used for motion. It also will take in data from the odometry sensor and pass that back to the Processing layer to be used by the Mini PC.

Assumptions: The interfaces will have a good connection and there will be minimal noise introduced to those connections.

Responsibilities: This subsystem is responsible for motor control directions and data transfer via odometry sensor and Processing layer to facilitate the Mini PC.

3. Module Design Specifications

3.1 Overall Detailed Design Principles

3.1.1 Scalability

The Wheelchair Platform must be scalable to give users the freedom to add or remove valves or sensors as they choose within the limits of the system. This affects what hardware expansion elements are built into the system.

3.1.2 Ease of Use

The application interface must be intuitive and user-friendly. This means building the UI interface with simple and easy to understand components. Ease of use also means limiting the number of functions a user must go through to control the system.

3.1.3 Maintainability

The Platform should be designed and implemented in such a way that the system requires minimal maintenance. Sensors should be replaceable and reinstallation should be possible at the user level. This will mean that the central control unit will have to be built with a plug and play type hardware interface for all sensors.

3.1.4 Compatibility

Since the Platform is initially designed, as predecessor teams shall install a basic product with limited features assuming additional competitive features, the Wheelchair Platform should be designed to not rely on any proprietary equipment to be installed. The controlling unit shall be compatible with majority of related future technologies.

3.1.5 Portability

The system is designed for the IGVC competition. So it must be easily moved and able to operate independently.

3.1.6 Modularity

The system design should contain layers and components that can be modified or replaced with ease.

3.2 User Interface Layer

The User Interface Layer is the first layer of the system architecture design. It is the layer that will communicate with the Network Layer and the User using 0MQ. The User Interface Layer consists of Screen and the 0MQ interface. The User Interface Layer will communicate with Processing Layer through the Network Layer, which makes it the foremost layer in the system that needs to be implemented in order of precedence. The UI Layer will send user instructions in the form of 0MQ messages from the User Interface Layer, and the Network Layer transmit those instructions to the Processing Layer by relaying the instructions produced packets.

3.2.1 Screen Module

• Module Prologue

The screen module consists the code required to display system information to the user. The module will be implement using Pythons Tk GUI interface, Tkinter. It will consist of necessary GUI elements for viewing relevant system information such as speed, angle, and connection status.

• Interfaces

• Physical Data Structure

• Module Processing

3.2.2 Input Module

- Module Prologue

The screen module consists of the code required to send information into to the system. The module will be implemented using Pythons Tk GUI interface, Tkinter. It will consist of text fields, buttons and necessary GUI elements for sending commands to the vehicle. It will enable the setting of vehicle attributes as well as allow for incremental changes attributes.

- Interfaces
- Physical Data Structure
- Module Processing

3.2.3 0MQ Module

- Module Prologue
- Interfaces
- Physical Data Structure § Module Processing

3.3 Network Layer

The Network Layer is the second layer of our architectural design. It is the layer that will communicate with the User Interface Layer and Processing Layer using the 0MQ. The Network Layer consists of packets and the 0MQ interface. The User Interface Layer and the Processing Layer will communicate through the Network Layer by sending information in the form of packets. The Network Layer will accept instructions in the form of packets from the User Interface Layer, and then convey those instructions to the Processing Layer by relaying packets. The guideline used to define this layer was drawn from the necessity to have a medium in order to convey information from the User to the Processor, and vice versa.

3.3.1 Packets Module

- Module Prologue
- Interfaces
- Physical Data Structure
- Module Processing

3.4 Processing Layer

The purpose of the Processing Layer is to receive packets of instructions from the network layer and translate them into machine-readable code. This layer will calculate the position of the platform and handle movements from one point to another point and angular spin. The processing layer will be executed with the help of 0MQ messages. The processing layer will also receive voltage and other movement related information recorded by the odometry sensor. This information will be used to correct any movement defects.

3.4.1 Movement Types Module

- Module Prologue

The movement types module consists of sub modules for the different movement types. The different movement types are forward/backward movement, movement in an arc to the left and right and turning clockwise and counterclockwise. The appropriate sub module then passes their respective outputs to the locomotion module, which handles the processing to actually engage the motors to get the vehicle to move as desired.

- Interfaces
- Physical Data Structure
- Module Processing

3.4.2 Movement Correction Module

- Module Prologue
- Interfaces
- Physical Data Structure
- Module Processing

3.4.3 Locomotion Module

- Module Prologue

The locomotion module uses programmatic movement commands from the movement types module and error information from the movement correction module to produce output that the motor controller can use to engage the motors to achieve the desired real world movement result.

- Interfaces
- Physical Data Structure § Module Processing

APPENDIX D

SYSTEM TEST PLAN

1. Introduction

The purpose of the System Test Plan document is to provide a methodology by which the correctness of the Wheelchair Platform project shall be tested upon. The methodology holds the details that shall be followed at every level of abstraction for the system. The methodology shall provide metrics and procedures to ensure that the Wheelchair Platform adheres to all requirements provided in the System Requirements Specification. The following sections will reflect on and summarize the critical and high priority requirements from the System Requirements Specification (SRS), provide an overview of the ADS and DDS, and will then define the STP laid out by Team SNAPH. The System Test Plan shall discuss Unit testing, Component/Function testing, Integration testing, and System Verification testing.

2. References

2.1 Project Charter

This document contains the scope statement, main objectives and purpose of the project.

2.2 System Requirements Specification

This document is a description of the system to be developed.

2.3 Architecture Design Specification

This document contains information on how and why the product will be used.

3. Test Items

3.1 0MQ Version 4.1.3

- Communicate with microcontroller Raspberry Pi 2 onboard the Wheelchair Platform

• Limitations: No user interface for simple delivery

3.2 Raspberry Pi Version 2.0

• Communicate with laptop network controller

• Limitations: No separate pin for reverse mode

3.3 Sabertooth Dual 60A 6V-30V Motor Driver

• Relay instructions from microcontroller to movement

• Limitations: No separate pin for reverse

3.4 Wheelchair Platform Version 1.0

• Test platform movements

• Limitations: No object avoidance sensors

3.5 E-Stop AC 600V 10A Red Mushroom 22mm

• Emergency Stop Button to kill motor

• Limitations: Button physically on motor, not wireless

4. Approach

4.1 Overview

This section describes in detail how Team SNAPH will carry out the testing approach and maintain records of the tests.

4.2 Overall Test Strategy

Team SNAPH will utilize both Black Box and White Box testing (as per the ADS and DDS testing plans) to ensure that the system will meet the requirements defined in the SRS. The system must also comply with the ADS and DDS specifications, which were written to reflect the SRS in a stable and fundamental way. The tests will be documented utilizing the following strategy:

• Test ID

• Test Type (unit, component, singular, etc.)

• Test Date

• Results and outputs

• Pass or Fail

• Comments, details, and plans to correct errors, and severity of errors.

4.3 Hardware and Software Configurations

There are a number of hardware and Software components that must be tested to ensure that they are reliable enough to handle their tasks in the Sentinel Project. The plan enacted in this section will ensure that the defects in the hardware are minimal.

• Test Name/Test ID

• Test Type Hardware/Software

• Fixed/Ignored

• Results

• Pass or Fail

• Comments, details, contingency plans and severity of errors.

Any bugs or defects that are critical to the systems success will be evaluated and carried out as a team in order to better resolve and fulfill the needs of all subsystems. Some issues will likely be non-severe and can be handled according to our own development.

4.4 Testing Metrics

The success of the prototype will be measured through our test cases and based on their priority in our SRS. A successful prototype will need to ensure that all high priority requirements are fulfilled and the acceptance criteria are met. This ensures that the prototype will be acceptable to the end user and all stakeholders before the product is considered for further production. Other test cases will be on a pass or fail basis and will be presented accordingly regardless of whether or not they actually fulfilled the project requirement.

4.5 Testing Requirements

If a change must be made in the source code to better implement a failed component in testing, regression testing will be carried out to ensure that any changes did not introduce new errors to subsequent systems that utilized information that may have been changed. The team will log the results of their changes and the regression testing.

5. Features to be Tested

5.1 Customer Requirements

5.1.1 Instruction Based Movement

Description: The system shall allow users to instruct the wheelchair to travel from its current position to a desired position. The movement must be continuous, and the wheelchair must not stray from its track.

Testing Approach: The team will send inputs via an input device in a wireless fashion. The team will receive a message after each action, to denote that the action was completed successfully.

5.1.2 Feedback Control Mechanism

Description: The system performance shall be evaluated regularly, providing the user with improved performance.

Testing Approach: To ensure this function the team will collect information about how the system is functioning with the help of the encoder.

5.1.3 Angular movement (spin)

Description: The system shall accurately rotate a certain angle based on user instruction.

Testing Approach: The team will send inputs via an input device in a wireless fashion. The team will receive a message after each action, to denote that the action was completed successfully.

5.1.4 Payload

Description: The system shall carry a payload of at least 20lb.

Testing Approach: The team will put a load of 20lb or greater on the wheelchair base, and observe that it is able to withstand the pressure of the load. The team will have to observe this while the wheelchair is in motion.

5.2 Packaging Requirements

5.2.1 Platform Size

Description: The platform shall be less than 7 feet long, 4 feet wide and 6 feet tall.

Testing Approach: During the development phase, the team will ensure that the platform size is less than 7 feet long, 4 feet wide and 6 feet tall by using a measurement scale.

5.2.2 Platform Housing

Description: The components of the wheelchair platform shall be encased into a plastic housing for security.

Testing Approach: The team will ensure that the motor controller, Raspberry pi and all other components are protected from outside exposure by testing the reliability of the plastic housing.

5.3 Performance Requirements

5.3.1 Vehicle Speed

Description: The Wheelchair Platform must maintain a minimum speed of one mile per hour. The Wheelchair Platform must not exceed a speed of five miles per hour.

Testing Approach: The team will test the speed of the wheelchair platform based on the number of rotations performed by the tires.

5.3.2 Vehicle Battery

Description: The Wheelchair Platform battery must pass the threshold operation time of one hour.

Testing Approach: The team will ensure that the battery performs efficiently for one hour by keeping it active for at least an hour.

5.3.3 Vehicle Payload

Description: The Wheelchair Platform shall carry a 20-pound payload.

Testing Approach: The team will put a load of 20lb or greater on the wheelchair base, and observe that it is able to withstand the pressure of the load. The team will have to observe this while the wheelchair is in motion.

5.3.4 Distance Traveled

Description: The Wheelchair Platform shall travel a continuous 44-foot distance.

Testing Approach: The team will ensure that the wheelchair platform travels a continuous 44- foot distance by instructing the wheelchair to travel from its current position to a desired position.

5.3.5 Vehicle Emergency Stop

Description: The Wheelchair Platform emergency stop shall force the vehicle to stop within 0.5 seconds of the request.

Testing Approach: The team will ensure the functionality of this feature by instructing the platform to move, and pressing the e-stop button to stop the motion. The e-stop must be a red button, clearly visible to the regular eye.

5.3.6 Directional Capabilities

Description: The Wheelchair Platform shall determine a specified degree at which it faces when interpreting user direction.

Testing Approach: The team will send inputs via an input device in a wireless fashion. The team will receive a message after each action, to denote that the action was completed successfully.

5.4 Safety Requirements

5.4.1 Emergency Stop

Description: A mechanical emergency stop button must be located on the center rear of the vehicle and should break power to the motor when pressed.

Testing Approach: The team will test this function by checking if there is any continuing power to the motor after the emergency stop button is stopped.

5.4.2 Kill Motor on Stall

Description: The motor should not actuate when the vehicle is stalled.

Testing Approach: The team will test this function by preventing the motor from receiving any power.

5.4.3 Maximum Current Draw

Description: The current draw shall not exceed what the power supply can provide.

Testing Approach: The team shall ensure this function by setting a limit on the maximum current that can be drawn.

6. Features NOT to be Tested

6.1 Customer Requirements

6.1.1 Debugging Mode

Description: The debugging mode shall show information of all vehicle hardware input to the system. The debugging mode shall allow the user to manually adjust individual vehicle hardware components.

Reasoning: Does not call for testing.

6.2 Packaging Requirements

6.2.1 Full User Manual

Description: A user manual shall be provided with instructions to operate the wheelchair platform. It shall be a printed document.

Reasoning: Does not call for testing.

6.3 Performance Requirements

6.3.1 Micro-PC

Description: The Wheelchair Platform shall contain a Micro-PC with a minimum speed of 700MHz quad-core processor and 512MB of RAM.

Reasoning: Does not call for testing.

6.4 Safety Requirements

6.4.1 Electrical Packaging

Description: All electrical connections shall be safely packaged to reduce the risk of electric shock to operators and maintainers.

Reasoning: Does not call for testing.

6.4.2 Electrical Grounding

Description: All electrical connections shall be properly grounded to reduce the risk of electric shock to operators and maintainers. Reasoning: Does not call for testing.

6.4.3 Safe Edges

Description: The product shall not contain sharp edges to reduce the risk of cuts to operators and maintainers.

Reasoning: Does not call for testing.

6.4.4 Fuse Protection

Description: The power supply shall be fused.

Reasoning: Does not call for testing.

7. Test Deliverables

This section outlines the different test artifacts that will be made available to the project sponsor upon project completion.

7.1 System Test Plan

The test plan shall include general test approach, test risk, items to be and not to be tested, as well as the item pass/fail criteria.

7.2 Test Cases

Individual test cases shall consist of the following information and will be recorded in spreadsheet.

- Test ID: Unique identifier to identify the test case
- Test Item: The hardware, module, subsystem, layer, or feature that will be tested
- Purpose: Reason test is being performed as it relates to overall final product
- Description: A brief summary of the test
- Input: The inputs that will be needed to run the test
- Expected Output: The result that should occur when test is run
- Priority: Level of importance
- Note: Extra information tester may find important to note such as system configuration

7.3 Test Results

Test results will be kept in a spreadsheet after tests are run and will consist of the following information.

- Test Result ID: Unique identifier to identify test result
- Test ID: Unique identifier to tie result to test case
- Test Date: The date of test
- Tester: The person responsible for carrying out the test
- Actual Output: The result given by the test
- Result: If actual output matched expected output, pass. Otherwise, fail
- Note: Extra information tester may find important to note

7.4 Defect Report

When a test case is run that produces output different from the expected output, the following details will be noted and stored in a defect report spreadsheet.

- Defect ID: Unique identifier to identify defect
- Description: Brief description of defect
- Severity: Impact on system
- Configuration: Configuration and environment related details
- Steps to Reproduce: As detailed of a description to reproduce defect
- Reported by: Name of team member who observed defect
- Date: Date when defect was observed
- Status: Open or closed
- Fixed by: Name of team member who fixed defect
- Date Closed: Date defect was fixed
- Note: Extra information

REFERENCES

Intelligent Ground Vehicle Competition. "Official Competition Rules." The 24th Annual Intelligent Vehicle Competition. Web. 12 Nov. 2015. < http://www.igvc.org/rules.htm>.

Hintjens, Pieter. *ZeroMQ: Messaging for Many Applications*. Sebastopol, CA: O'Reilly & Associates, 2012. Print.

Cook, David. *Robot Building for Beginners.* New York: Springer Science Business Media. Print.

Trivedi, Amit Ranjan, et al. *Design and Implementation of a Smart Wheelchair.* Proceedings of Conference on Advances In Robotics - AIR '13 (2013). Print. Shiomi, Masahiro, et al. *User-friendly Autonomous Wheelchair for Elderly Care Using Ubiquitous Network Robot Platform.* Proceedings of the Second International Conference on Human-agent Interaction - HAI '14 (2014). Print.

Martinez, Aaron, and Enrique Fernández. *Learning ROS for Robotics Programming: A Practical, Instructive, and Comprehensive Guide to Introduce Yourself to ROS, the Top-notch, Leading Robotics Framework*. Birmingham: Packt, 2013. Print.

BIOGRAPHICAL INFORMATION

Pragya Neupane is from Kathmandu, Nepal. She has been attending the University of Texas at Arlington since Fall 2013 to pursue the Honors Bachelor of Science degree in Computer Science and Engineering. She is interested in image processing, database management, and software project management. She plans to work with software and application development after obtaining her undergraduate degree. After about five years of receiving technical experience, she wishes to enroll in a Master's degree program in Business Administration (MBA).