A Human-in-the-Loop Cyber Physical System:
Modular Designs for Semi-Autonomous Wheelchair Navigation

A Major Qualifying Project Report Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science by

Ross S. Desmond
Matthew F. Dickerman
James A. Fleming
Date: April 26, 2013
Report Submitted to:
Taskin Padir, Advisor
Jerry Schaufeld, Co-Advisor
Sonia Chernova, Co-Advisor
Dmitry Sinyukov, Graduate Advisor

ABSTRACT

This project involves the design and development of a prototyping platform and open design framework for a semi-autonomous wheelchair to realize a human-in-the-loop cyber physical system (HiLCPS) as an assistive technology. The system is designed to assist physically locked-in individuals in navigating indoor environments through the use of modular sensor, communication, and control designs. This enables the user to share control with the wheelchair and allows the system to operate semi-autonomously with a human in the loop. The Wheelchair Add-on Modules (WAMs) developed for use in this project are platform-independent. These modules facilitate development and application of semi-autonomous functionalities. By using the WAMs, a team of three can convert similar powered wheelchairs into a semi-autonomous mobility platform in less than ninety minutes.

ACKNOWLEDGEMENTS

We would like to thank our project advisors, Professor Taskin Padir, Professor Schaufeld, Professor Chernova, and our graduate student advisor, Dmitry Sinyukov, for the opportunity to work on this project and for all their support. We would also like to thank the following:

- National Science Foundation

for their generous grant to support the work of Professor Padir. Without the generous support of NSF, this project would not have been possible.

- Cornell University's Systems Engineering department and Intel Corporation

for their financial support and donation of components, as well as the opportunity to demonstrate our work at the Cornell Cup USA technology expo on May 3-4, 2013.

- IEEE, and the Robotics & Automation Society

for the opportunity to present our work at the 5th annual Technologies for Practical Robot Applications (TePRA) conference on April 22-23, 2013.

- Cambridge Science Festival

for the opportunity to demonstrate our work at Robot Zoo on April 13, 2013

- Worcester Polytechnic Institute

for the use of their laboratories and manufacturing facilities and all members of the Robotics Engineering department, including Joe St. Germain, Tracey Coetzee, the students working in the RIVeR lab, and all other students and faculty who supported our project through their efforts and advice.

Contents

List of Figures

List of Tables

List of Equations

1. Introduction

Locked-in syndrome is a rare but devastating condition in which an individual has full cognitive abilities but all voluntary muscles of the body are paralyzed. As a result, the individual is incapable of interacting with the physical world through movement and speech, making independent activities of daily living difficult [1]. As of now, there is no commercially available product that could provide these individuals with control over their mobility and manipulation of surrounding objects.

This project provides a prototyping platform and design framework for a semi-autonomous wheelchair that is capable of integrating human-in-the-loop control. We developed Wheelchair Add-on Modules (WAMs) that can be easily mounted, configured, and customized to fit the design of a wide variety of powered wheelchairs. Through the use of these modules, the team aspires to encourage the rapid exploration of Human-in-the-Loop Cyber Physical Systems (HiLCPS) and shared control applications.

A HiLCPS is defined as a system that augments human interactions with the physical world, most often through the use of body and brain sensors. These advanced, multidisciplinary systems are capable of inferring user intent through the use of human-computer interfaces that go beyond ordinary machine interactions and are able to detect subtle physical and emotional responses. The system reacts accordingly to optimize performance. After interpreting user intent, the HiLCPS often actuates a robotic system to accomplish the task specified by the user. The feedback loop is closed by the user who evaluates the machine's performance. This user-immersive feedback (shown in Figure 1) enables the development of systems that can be used to revolutionize a wide variety of industries and have the potential to impact the daily lives of millions of people.

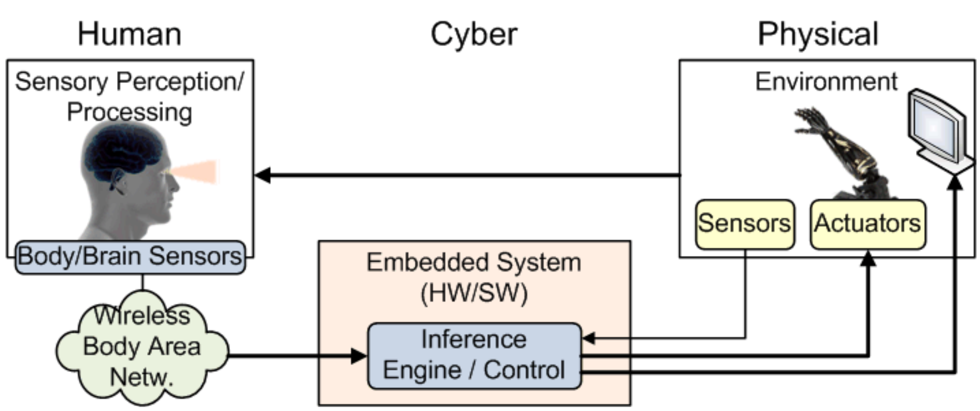


Figure 1: HiLCPS feedback loop [2]

Although this project places an emphasis on improving the lives of individuals with locked-in syndrome, semi-autonomous wheelchairs interfaced with HiLCPS can also be used to improve the lives of others who suffer from medical conditions that severely reduce mobility and prevent the use of traditional powered wheelchairs. Use of such a system will allow patients to live more independently, improve their overall quality of life, and more easily reassume previously held roles within their households and community.

During the design process several challenges had to be overcome. Range-finding, mapping, and odometry sensor modules were developed. For the system's main computer to access the data produced by these sensor modules, a sensor network was developed and implemented with serial communication and standard USB interface. Basic assistive control functionalities were developed to prevent the user from coming to harm and several indoor navigation behaviors were implemented. The WAMs developed in this project allow for the conversion of a commercially available wheelchair into a semi-autonomous wheelchair that can be easily integrated into a HiLCPS.

2. Background

This chapter presents the research that was conducted to better understand the various topics relevant to the successful completion of this project. With the knowledge obtained from this research, we were able to make more educated design decisions. Before attempting to develop an assistive control wheelchair, the assistive technology industry was investigated, paying particular attention to powered wheelchair technologies. Previously completed semi-autonomous wheelchair projects were evaluated, paying particular attention to designs that worked well and those that failed to meet intended project requirements. To implement semi-autonomous functionalities to a wide variety of commercially available wheelchairs, the design of modular systems, electronic sensors, and mapping and navigation algorithms were researched. Guided by the knowledge obtained from pre-emptive research, we were able to make sound design decisions supported by facts and learn from the mistakes and gains of projects with similar objectives.

2.1 Assistive Technology

An assistive technology device is any item, commercial product or equipment, modified to be used to increase, maintain or improve the functional capabilities of individuals with disabilities [3]. This class of technology has been developed for decades to enhance ease of transportation, movement, and communication for individuals with disabilities. Specialized wheelchairs have been customized to provide people with particular disabilities with more freedom and independence. Although these wheelchairs have increased the quality of life for these people, the cost to procure one has been, and still is, very high and is often times not covered by average health care providers. Due to this, most patients are forced to deal with basic, and often times inadequate rehabilitative technologies that place limits on their overall independence.

2.2 Powered Wheelchairs

Electric wheelchairs are a common form of assistive technology. They provide mobility for physically challenged individuals who are able to utilize a joystick. There are many large scale electric wheelchair providers whom have very different designs and features as shown in Figure 2.

Figure 2: Wheelchair examples, marketing company from left: spinlife [4], shermanoaks [5], easymed [6], hoveround [7]

These popular commercially available wheelchairs utilize very different manufacturers for each design and require modification for individuals who are unable to use the joystick. For individuals who are unable to control a conventional powered wheelchair, other assistive devices are able to fit the user's needs with some adaptation.

2.3 Semi-Autonomous Wheelchairs

The access to a means of independent mobility with power wheelchairs promotes a feeling of self-reliance, helping to aid physical and mental struggles caused by the disability. Manual or power wheelchairs can aid thousands of individuals; however a portion of the physically and mentally challenged population is unable to use general powered wheelchairs. Smart wheelchairs are designed to accommodate individuals with a variety of conditions and have been around been around since the mid 1980's [8]. Some wheelchairs used machine vision to identify landmarks, or sonar and IR to avoid drop offs and obstacles.

"A Literature Review of Smart Wheelchairs" stresses the benefits of having access to a means of independent mobility [8]. Individuals who exhibit problems with vision, tremors, low cognitive ability, and reduced or impaired movement find they are unable to use power wheelchairs. The authors show over 85% of responding clinicians seeing a number of patients yearly who lack the precise motor skills to use a conventional joystick.

The Hephaestus Smart Wheelchair system is a series of components to attach to a powered wheelchair to convert it to a smart wheelchair [9]. This system is intended for use with multiple brands of wheelchairs. It uses 16 sonar sensors, with the ability to communicate with 24, which are able to detect obstacles between eight centimeters and one meter. The sonar sensors are mounted to the front tray and rear of the robot as shown in Figure 3.

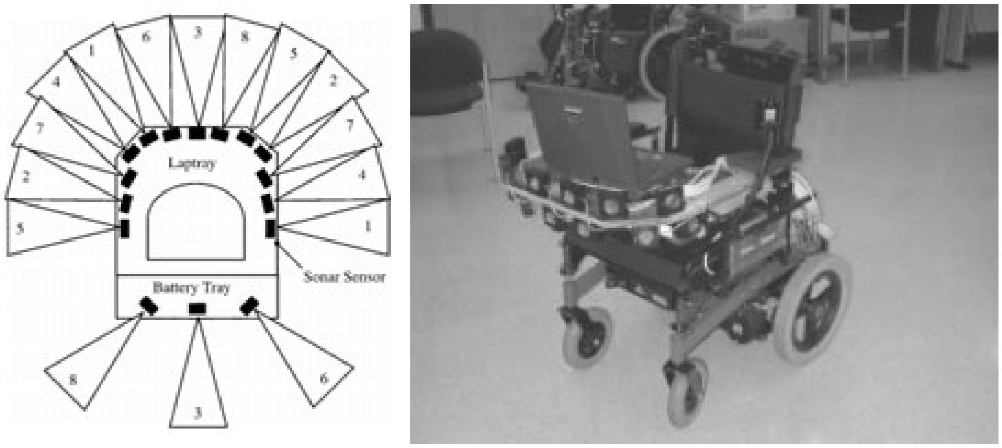


Figure 3: The prototype Hephaestus Smart Wheelchair System

There are also multiple bump sensors on the footrest as a last resort effort to halt the wheelchair in the case of a collision. Their solution utilizes a computer to communicate with the sensors, joystick, and motor controller. The navigation algorithm used for Hephaestus is used to augment the wheelchair operators input and provide control feedback to avoid obstacles.

The Hephaestus system has several blind spots due to the cost of adding the ability to communicate with more sensors also shown in Figure 3. It is also unable to detect drop-offs such as stairs or curbs, cannot communicate with different types of joysticks or sensors, and requires specific mounting surfaces for sensors.

A study of a performance test of a similar collision-avoidance system utilizing behavior based control was conducted to assess reliability and precision [10]. The team studies the “Drive-Safe System” (DDS) which detects obstacles slowing and stopping the wheelchair when needed. It also corrects joystick control through modes such as: obstacle-avoidance, door-crossing, wall-following, corridor-crossing, and override. The mode is determined by the pattern caused by obstacles in certain proximity around the robot. The only modes tested during this study were slow down or stop when obstacles within a set proximity were found.

CanWheel, a research team of 14 clinical researchers and basic scientists, has been evaluating the feasibility and practical applications of smart wheelchairs since 2009 [11]. The research done by CanWheel has identified the needs and experiences of users of powered wheelchairs over the age of 60. Their work aspires to evaluate the effectiveness, impact, and relevance of wheeled mobility devices from the perspective of consumers, caregivers, health care providers, policy makers, and funding agencies. CanWheel also intends to develop a smart wheelchair driven by the needs of the diverse needs of the project's various stakeholders.

Recognizing that the aging population has diverse and dynamic needs, CanChair will also attempt to develop and release common hardware and software design platforms which will others to easily customize commercially available wheelchairs to individual users and adapt the wheelchair as the user ages. CanChair predicts these platforms will "make it easier for researchers and commercial developers to add new capabilities, sensors and interfaces, as well as to migrate to new computer and wheelchair models as they become available". It is clear that they believe a highly customizable, modular design will rapidly accelerate the advancement of smart wheelchair technologies. CanWheel also intends to create multiple prototypes of smart wheelchairs in order to obtain feedback from a broad range of stakeholders throughout the project. The feedback collected will serve as a starting point for future commercialization efforts and additional studies. An example of one of their prototype smart wheelchair is shown in Figure 4.

Figure 4: CanWheel's 2011 smart wheel chair prototype

2.4 Sensing Technologies

In robotic systems, it is important to be able to sense the environment. Data collected can be used to drive control loops, provide users with information, and make the robot more dependable. Oftentimes there will be multiple sensing technologies on a robot that allow for a wider variety of data to be collected. It is useful to understand these technologies before choosing a sensor for a given robot.

2.4.1 LiDAR

LiDAR (Light Detection and Ranging) systems are optical remote sensors that use laser light to detect distances from the sensor. The light they use can be ultraviolet, visible, or near infrared, and the implementation is much more advanced than regular infrared rangefinders. Operation is done either by a laser or a series of lasers which sweep over an area [12].

This technology can be very expensive, especially more advanced models such as the one being used in Google’s driverless cars [13]. The array of points the laser measures can be very accurate, and is currently used for three-dimensional mapping of terrain and height from ground.

2.4.2 Infrared Ranging

Infrared rangefinders are sensors based on infrared electromagnetic radiation, consisting of a transmitter LED and a receiver, which send out a pulse of infrared light, then calculate distance to the target based on the angle of the returned pulse. An example of such a sensor is the Sharp GP2Y0A21 IR sensor, which outputs an analog voltage corresponding to the distance from the sensor. It has a very simple interface to low-level microcontrollers, requiring only an analog-to-digital converter to process the signal. The IR beam tends to be very narrow, as shown in Figure 5.

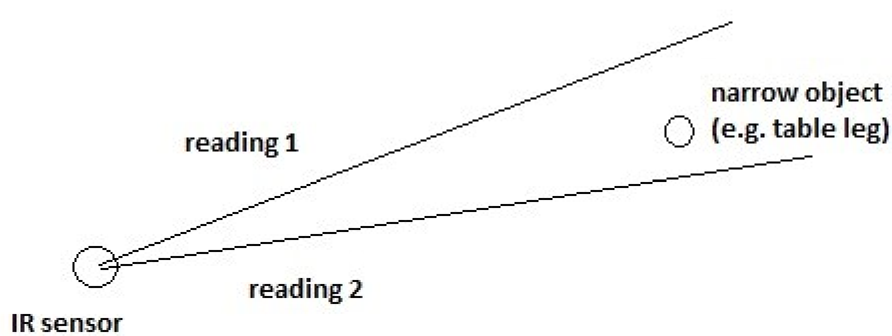


Figure 5: Illustration of IR sensor with its narrow beam missing a thin object such as a table leg.

The narrowness of the IR beam is useful for precisely measuring an obstacle; however this presents problems when attempting to measure narrow objects such as a table leg.

2.4.3 Ultrasonic

Ultrasonic rangefinders operate based on sonic pulses, sending a pulse out and timing how long it takes to receive a response. Different models can operate in air or water (sonar systems, for example). An example would be the Maxbotix LV-EZ0, which outputs an analog voltage corresponding to the distance from the sensor. This sensor has an analog interface, making it simple to work with through the use of a microcontroller.

The maximum distance that these sensors can measure is usually far, with some being over thirty feet [14]. They work best when looking at an obstacle dead-on. When the beam hits an object at a high incident angle, the readings may show different distances than what the actual object distance is, especially if the beam reflects off of multiple surfaces before finally reaching the sensor on the return trip. This effect can be seen in Figure 6. They also rely on the speed of sound being constant at all times, so use in environments with large temperature fluctuations can throw off the readings (Harrison).

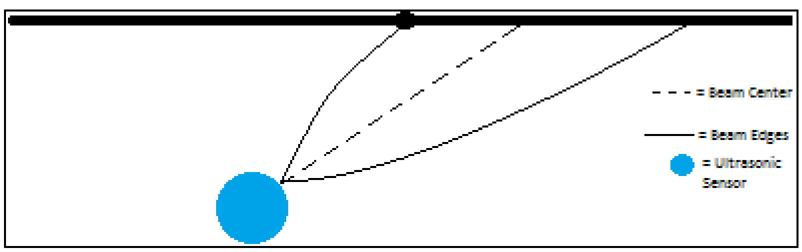


Figure 6: Illustration of the problem ultrasonic sensors face at angles.

2.4.4 Odometry

Odometry is the use of data from sensors designed to detect motion and, using gathered data, estimate position over time. This method of sensing, depending on the implementation, is one of the simplest for estimating position. However it is sensitive to errors due to integration of data over time. As a result, accurate data collection and processing are important to provide the most effective system possible.

Some of the more prominent sensors used for this purpose are rotary encoders, the simplest of which is direct mechanical linkage to a wheel axle. This allows one to measure the angular position of the wheel in time, which translates to velocity data if the wheel is rotating and data is collected at a constant sample rate. Other uses of such an encoder would be indirect mechanical linkage to the wheel, for example if the system included gears. In this case, the encoder might be mounted to the motor shaft, while the wheel is on a separate axle connected to the motor shaft via a gear reduction.

Other sensor options include using analog continuous potentiometers, which have the ability to rotate a full 360 degrees, with a slightly smaller electrical rotation, as shown in Figure 7. This provides the same functionality as the digital rotary encoders in an analog form. Another option might be to use a camera strapped to a robot which watches a point of relative motion (such as the ground), or watch the rotation of the wheel. Through digital processing of the image data, the motion of the robot or wheel can be determined, thus providing odometry data. This method of processing is currently present in surface-independent optical computer mice [15].

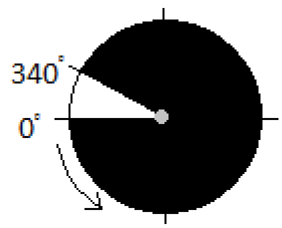


Figure 7: Illustration of the electrical rotation of a continuous potentiometer. The shaft at the center has the ability to rotate continuously, however there is slightly less electrical rotation (340 degrees in this case).

2.4.5 Inertial Measurement

Inertial measurement units (IMU) are sensors that directly measure acceleration, and can be used to obtain information such as gravity, velocity, position and orientation, using a variety of gyroscopes, accelerometers and magnetometers. The IMU uses dead reckoning to estimate velocity and position data, as acceleration can be integrated over time to provide the required data. These sensors are used in part by aircraft to provide raw data about movement (Breed, 2011).

A common usage of these types of sensors is in the Segway Personal Transporter, developed by Dean Kamen and sold through Segway [16]. The Segway is a two-wheeled self-balancing electric vehicle which is speed-controlled by the user shifting their weight forward or backward on the vehicle. They use multiple IMUs as part of their balancing algorithm to prevent the user from falling over.

Embedded IMUs are also available, such as the Sparkfun *ADXL193* which is a single-axis accelerometer that outputs an analog voltage and measures up to 250 gees of acceleration. For rotation, there is the Sparkfun *L3G4200D*, a three-axis gyroscope which can measure up to 2000 degrees per second, or about 5.5 revolutions per second, and communicates over either I2C or SPI for a digital interface.

2.4.6 Visual Sensing

Visual sensors take in image data which is processed either by a separate controller or by the sensor package itself. These sensors are usually cameras, sometimes having additional sensors present, such as the Microsoft Kinect which has IR sensors built in. Depending on the quality of the sensor, they are useful in a variety of environments, being unaffected by wind or light outside the visible spectrum. Processing can be done with an array of cameras (such as stereo cameras, which simulate two eyes), allowing for determining distance to objects in view of the cameras by computing parallax.

Arguably the most common cameras on the market are computer "webcams", designed for communication and low to mid quality video recording. With a USB interface, they can be plugged in to almost any computer system, provided the correct drivers are in place. Often these cameras have lower quality, but can go as high as 15 megapixels [17], and will run at an acceptable real-time pace (around 30 frames per second).

The Microsoft Kinect is a motion sensor which includes both an RGB camera and an IR depth-finding camera, both of which operate at 640x480 pixels at 30 frames per second. It also has a USB interface, allowing it to be plugged in to many computer systems. The Kinect has a history in third-party development, especially in robotics, where the low cost and high capabilities of the sensor allow cost-effective 3D-vision and depth sensing [18].

2.5 Robot Operating System

Robot Operating System (ROS) is a framework originally developed by the Stanford Artificial Intelligence Laboratory in 2007, joined later by Willow Garage and some twenty other institutions [19]. ROS was developed with the goal of being a free and open-source framework that allows a multitude of contributors writing code in multiple languages. In addition, one of the most useful features is being “thin”; in this context, “thin” means having the ability to take code and reuse it outside of its original context [20].

The architecture of ROS contains several fundamental concepts: nodes, messages, and topics. Nodes are individual programs that perform computations. Many nodes can be run at once, each one performing specialized tasks. These nodes can talk to one another though the passing of messages, which are essentially data structures. Messages support all primitive data types (boolean, integer, etc.) and arrays of such. In addition, messages can be composed of other messages. A node sends messages by publishing them to a topic. Other nodes interested in such data subscribe to that topic. In this way, ROS nodes broadcast messages to topics like radio waves, and an arbitrary number of subscribers can listen to any given topic. Multiple nodes may broadcast to a given topic, allowing even more variety for communication.

2.6 Mapping & Navigation

Simultaneous localization and mapping (SLAM) is a technique used in robotics applications to generate a map of a robot's environment while concurrently localizing itself within that map. Using SLAM algorithms, robotic platforms are free to navigate and operate within unknown environments with increased perception. SLAM algorithms address the following two questions: "What does the environment look like?" and "Where is the robot located within the environment?" The SLAM problem has been solved using several different algorithms, but it generally comes in two versions: online and global. The online SLAM problem attempts to estimate the momentary robot pose while the global SLAM problem attempts to determine all poses. The EKF SLAM algorithm was one of the first SLAM algorithms to be implemented. It applies an extended Kalman filter to the online SLAM problem. The EKF SLAM algorithm has been noted to work well when distinct landmarks within the robot's environment are present. The need for these landmarks has been the main drawback of EKF SLAM along with the computational complexity associated with updating the Kalman filter [21].

2.6.2 Bayesian Approach to Motion Planning

One of the medium levels of the traditional artificial intelligence design is motion and path planning. An approach is to create a user adaptive system that allows user input into how the wheelchair reaches its goal [22]. The Bayesian network is built up over time by reading signals such as uncertainty in their decision and recognizing complex user plans. The goal is represented by a pose matrix $P_{goal} = [X_{goal}\ Y_{goal}\ \Theta_{goal}]^T$ and a twist matrix $t_{goal} = [V_{goal}\ \ W_{goal}]^T$ where V and W are linear and rotational velocities respectively. Together these matrices define the goal state and a user plan can be defined as the set of states leading to that goal. The network is built by what the user has in mind to achieve the goal state from its current state. This approach allows the user to feel in control of how the robot will reach its destination. The mapping and path search algorithm the team used was a Voronoi diagram where each node was part of the network.

The trajectory planning algorithm for complex maneuvers used a dynamic model of the wheelchair. A line map was generated of the obstacles around the wheelchair and a fine motion planner created a vector diagram that fills the 3D space with states that point towards the next node in the trajectory to account for error. The algorithm then uses a cost function and assigns a heuristic to each of the reachable nodes from the current node. It then used a probability function to assess the users plan, process plan, and the previous actions to determine the trajectory to take.

This plan was interesting in that it utilized a Markov Decision model for human-computer interaction. The robot computed the high level optimizations of path planning but still allowed for human influence based upon prior decisions and current sensors. This dynamic approach allows for customization by the user while still allowing for successful completion of objectives.

2.6.3 External Sensor Navigation

"A Concept for Control of Indoor-Operated Autonomous Wheelchair" suggests the use of ceiling mounted TV cameras in order to pre-process a map and track motion of the wheelchair [23]. The wheelchair still uses encoders and proximity sensors in order to navigate, however the cameras provide better global navigation and identification of obstacles before the wheelchair gets to the goal. For example, the cameras would be able to provide information of a blocked path that is not in the direct line of sight of the wheelchair. The navigation algorithm used for this particular wheelchair would explore the surroundings using the cameras and proximity sensors. It would then plan a route and attempt to follow the path using wall following. It was able to modify its path if obstacles were identified, and be able to operate in semi-autonomous mode with controlled assistance from the user. This project also used traditional control logic with simplified kinematic equations and a 2D occupation grid for mapping. This project was only completed in simulation and was not tested on a real wheelchair.

2.6.4 Advanced 3D Map-Based Control

The semi-autonomous wheelchair was equipped with a Kinect and improved the mobility of a smart wheelchair through more advanced perception of the environment [24]. This project used the Kinect to generate a point cloud, filter and segment it, then run an object detection algorithm on it. The cloud segmentation was done through RANSAC ("RANdom SAmple Consensus") and then model match the Euclidean cluster to completely identify the object and place it in a 3D semantic map. In order to match the object, common objects are stored in a library that is described by constraints such as area and height. The 3D map would be updated based upon the object's location in the room and local coordinates. The wheelchair would use the 3D map for motion control, target selection, and intention estimated based upon user input. The agent also used a traditional approach to AI with high level computations and low level reactive control from less powerful sensors.

2.6.5 Behavior Based Control

Robot behavior is a control law that is used to constrain a system to achieve and maintain a particular goal state. Behavior based intelligent agents use multiple laws fused together in order to determine the optimal course of action [25]. After a trajectory or goal state has been established, a number of high level behaviors are possible from each state. An example of high and low level behaviors is shown in Figure 8.

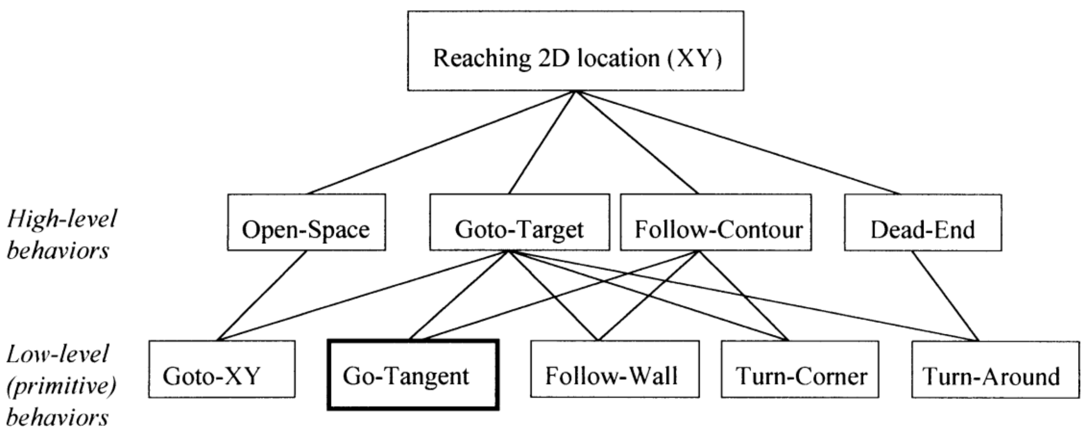


Figure 8: Hierarchy of System Behaviors

The high level behaviors laws are activated by the current state of the system. If the sensors establish that the wheelchair has reached a dead end, that state will be activated and it will follow the low level behaviors [26]. A behavior based model does not use object detection and is a rather primitive/reactive navigator. However BBC (Behavior Based Control) might be useful as a controller input to avoid moving obstacles.

2.7 Modular Systems

Modular systems are composed of many modules that can be altered or replaced without affecting the remainder of the system (Foster, 1995).The overall design philosophy operates partially on the idea that strong coupling between pieces of a system is inflexible. If a section of such a system needed to be changed, then other components would have to be either redesigned or have an adapter created for them. By designing components with a standard interface, the internals of a section of the system can be modified without affecting surrounding sections, and the system will not require redesign. This also allows new components to be designed and added to the system without having to modify the rest of the system. As long as a given component implements the standard interface, the rest of the system can understand it. Systems that implement modular design allow for reusability of components and are more extendable. An example of this design is a computer, where customizability is extremely high; a wide variety of components can be connected together in many configurations to yield a working system.

2.8 Context and Motivation for the Project

We created a system that adheres to design principles such as modularity, configurability, and ease of use. An aim is to provide a product that will be used to accelerate HiLCPS research and development. HiLCPS applications capable of augmenting human interaction with the physical world have the potential to improve the daily lives of millions of people - particularly those enduring physical or mental challenges that adversely affect motor function and mobility, such as locked-in syndrome. This section serves to provide readers with the global context and motivation that inspired this project.

2.9 Locked-in Syndrome

Locked-in syndrome is a rare but devastating condition in which an individual is fully aware and awake but all voluntary muscles of the body are paralyzed. As a result, the individual is incapable of interacting with the physical world through movement and speech, making independent activities of daily living impossible. Despite the traumatic loss of motor function, individuals with locked-in syndrome retain cognitive capabilities and are able to think and reason normally.

Locked-in syndrome is generally caused by damage to the ventral pons (shown in Figure 9), the sector of the brain that is responsible for conducting signals from the cerebrum to the cerebellum, medulla, and thalamus. This damage results in quadriplegia and anarthria (the inability to speak). Neuropsychological assessments have revealed that patients maintain normal brain activity but sometimes patients report that attention span, execute function, intellectual ability, perception, and visual and verbal memory were affected by the condition.

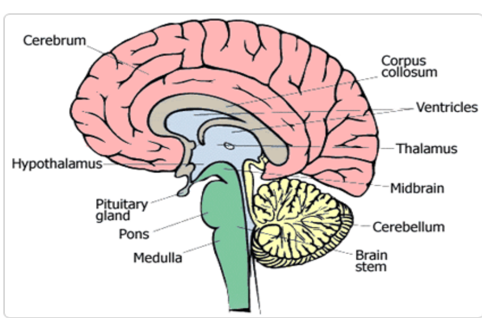


Figure 9 : The pons is used to conduct a number of important signals throughout the brain.

Three categories of classification have been established for locked-in syndrome: classic, incomplete, and total. Classic is defined as quadriplegia and anarthria with preserved consciousness and vertical eye movement. Classic locked-in syndrome is the most common variant of the three. Incomplete is the same as classic with remnants of voluntary movement other than vertical eye movement. Total is described as complete immobility and inability to communicate, with full consciousness.

According to a clinical review on locked-in syndrome conducted by the National Rehabilitation Hospital of Dun Laoghaire, Ireland, the average age of onset ranges between 33.6 to 45.3 years, depending on the patient's country of origin [1]. The study also revealed that the condition had a tendency to be more common in males than females, although a larger sample size could reveal otherwise. In a report titled "Life-sustaining treatment and locked-in syndrome" by Anderson, Dillon, and Burns, the authors found that patients with locked-in syndrome tended to have a worse quality of life on the Spitzer quality of life index than cancer patients but better than terminally ill patients. The report also identified that, of the longest surviving group of patients with locked-in syndrome, 54% had never considered euthanasia, 46% had previously considered it, and none had a "not for resuscitation" order, revealing that locked-in survivors rarely want to end their lives, despite their profound medical condition.

Life expectancy for patients diagnosed with locked-in syndrome has greatly improved in recent years largely due to improved multidisciplinary rehabilitation technology. Ten year survival rates as high as 80% have reported in many developed countries and over 60% of patients are eventually rehabilitated to the point that they are able to live at home with family.

Unable to speak, move, or make facial expressions, patients diagnosed with locked-in syndrome often seek alternative methods of communication with others. To accommodate these individuals, patient-computer interfaces such as infrared eye movement sensors and computer voice prosthetics are being developed by rehabilitation engineers and speech therapists. The use of such communication technology, such as the Eye-gaze Response Interface Computer Aid (ERICA) designed by the UMD Communication Sciences and Disorders program (shown in Figure 10), has proven to have a positive effect on the lives of people with locked-in syndrome, allowing for patients to initiate in dialog, prepare messages, and even access the internet.

Figure 10 : Eye-gaze Response Interface Computer Aid (ERICA)

When individuals with locked-in syndrome return home they are exposed to greater social interaction with friends and family. Equipped with communication devices, their overall quality of life improves and increases the desire to live. Living at home with locked-in syndrome, however, places a significant long term physical and psychological burden on the family of the patient. Individuals with locked-in syndrome require constant monitoring and need assistance with all activities of daily living (ADL) such as eating, dressing, and bathing. In addition, acquiring the needed rehabilitation treatment and technology is often costly and, due to limited funding, most caretakers are inadequately supported to treat such a severe disability.

With the ability to communicate restored through patient-computer interfaces, the next logical step that should be taken to improve the quality of life for individuals with locked-in syndrome is to increase the mobility of the patient. Independent mobility (even when actuated through the use of rehabilitative technology, such as powered wheelchairs) promotes dynamic interactions within a physical environment and provides social and emotional benefits that are necessary for developing a healthy self-esteem. Because no cure currently exists for locked-in syndrome, it is likely that rehabilitative technologies, such as semi-autonomous wheelchairs interfaced with HiLCPS, will become an excellent candidate for providing patients with independent means of mobility.

Although this project places an emphasis on improving the lives of men and women with locked-in syndrome, semi-autonomous wheelchairs interfaced with HiLCPS can also be used to improve the lives of others who suffer from medical conditions that severely reduce mobility and prevent the use of traditional powered wheelchairs. By using such a system, patients will be able to live more independently, improve their overall quality of life, and more easily reassume previously held roles within their households and community.

2.10 Human in the Loop Cyber Physical Systems

A human in the loop cyber physical system (HiLCPS) can be described as a system that augments human interactions with the physical world, most often through the use of body and brain sensors. These advanced, multidisciplinary systems are capable of inferring user intent through the use of a human-computer interface. These human-computer interfaces go beyond ordinary machine interactions and are able to detect subtle physical-emotional responses of users, such as pleasure or fear, and react accordingly to optimize performance. After interpreting user intent, the HiLCPS often actuates a robotic system to accomplish the task specified by the user. The feedback loop is closed by the user who evaluates the machine's performance. This advanced, user-immersive feedback enables the development of systems that can be used to revolutionize a wide variety of industries and has the potential to impact the daily lives of millions of people.

Shared control is a major theme associated with HiLCPS platforms. Shared control essentially moderates control over the system between the human operator and the system itself based upon the situation. The purpose of implementing shared control is often to make a safer or more easily useable system. Sytems that implement shared control can readily transfer control from the human operator to the system's computer during potentially dangerous and complex situations to prevent accidental damage to the system and user. Once the system accomplishes its task, control is restored back to the human operator. Shared control and HiLCPS are ideal for improving the quality of life for a wide variety of individuals who suffer from medical conditions that inhibit mobility. For these individuals, shared control HiLCPS could be the essential technological link for living productive and fulfilling lives.

2.11 The Growing Need for Rehabilitative Technology

In recent years, these devices have become exponentially more advanced and ever more capable of improving the lives of the millions of individuals who suffer from medical conditions that hinder activities of daily living. Rehabilitative technology has shown tremendous potential to reduce the stress currently placed on healthcare providers and caregivers by reducing the amount of human resources needed per patient. While the rehabilitative technology assists the patient, the healthcare professionals are able to shift their focus on diagnosing the rehabilitative process and making more personalized recommendations for swifter recoveries. Essentially, through the integration of advanced rehabilitative technologies, healthcare professionals are able to better utilize their time, training, and skillsets to better meet the needs of the patients they serve. The National Science Foundation (NSF) has recognized the benefits associated with rehabilitative technologies and, in recent years, provided a large number of grants to accelerate the development and advancement of such technologies.

Due to the increasing number of elderly populations within countries throughout the world, the rehabilitative technology industry is expected to grow [27]. Intelligent assistive mobility devices are expected to become increasingly popular due to the exceptional improvement to quality of life they are able to provide these individuals. It is likely that the development of intelligent wheelchairs, capable of providing users with advanced navigation functionalities, will be a logical first step in the process of creating advanced rehabilitative technologies.

Future rehabilitative technologies should also be highly customizable such that they can best accommodate patients on an individual basis. This can generally be accomplished through the use of modular devices and add-on features. This kind of customizable design reduces the costs to procure the assistive device and ensures greater levels of customer satisfaction. Often a rehabilitative technology that works with one individual may not work for another due to the nature of the individuals' medical conditions or complications. Future competitive rehabilitative technology will have to be designed such that it is able to meet the individual needs of patients of various lifestyles and medical conditions.

3 Decision Process

We used modern software engineering techniques such as decision flow charts to identify feasible solutions for the project. After defining the project goal we described key terms such as: modularity, cost-effectiveness, research and development time cost, and ease of implementation. While researching methods and technology for possible solutions to our objectives we rated them from a scale of one to ten in the previously mentioned categories. Then we added each component to a decision flowchart, listing their advantages and disadvantages. After careful deliberation, we would add reasoning as to which components are necessary and eliminate those that either do not fit the criteria or are not worth doing in the scope of the project.

3.1 Modularity

A modular component needs a great degree of interoperability with the rest of the system. The component should be able to be moved to another part of the system and, with minimal effort, be able to have the system working. In addition, the system can have a higher degree of scalability, such as the ability to add many more sensors and provide greater performance with minimal effort. Most importantly, components can be replaced with a new one that has the same interface, and the system will not break. As well as the components being individual, writing the software in a modular fashion is just as important. By utilizing good design practices, future project teams can easily keep a healthy code base, as well as being able to implement new features and update old ones.

Due to the need for a robust interface between hardware and software across multiple platforms, the development cycle of the interface will be much longer, as it must encompass more.

3.2 Cost-Effectiveness

This term is defined as the price of the component versus the amount to which the component fulfills project use cases. The more a product is able to perform towards our goal it is considered more valuable and is weighted as such. If a product’s cost outweighs its predicted performance ratio this term is rated lower.

3.3 R&D Time Cost

Research and development time cost is the amount of time it would take for the team to research the technology, acquire the materials, and develop the component. This category is important because the scope of our project spans seven months. The more efficient this term is (the lower time cost) the higher the rating of the component.

3.4 Ease of Implementation

The ease of implementation value is determined by the time it would take to implement a solution after research and development has occurred. This is most relevant when we decide to use a fully commercially developed solution the cost to integrate that product into our system. This term includes designing mechanical mounts, connecting to our power supply, and implementing the features in software. The larger this term, the easier the product is to implement.

3.5 Systems Engineering Approach

Systems engineering allows us to identify the exact requirements to be met for successful completion of the project. The first step is to identify the stakeholders of the project. Stakeholders are anyone or anything that imposes requirements. In Appendix B: Systems Engineering the Stakeholder Table displays the first table identifying each stakeholder. They are associated with a primary key (ID), a description and role, a method of needs elicitation, and their relevance to our goal. The method of needs elicitation is the way of procuring the stakeholders needs to build requirements.

In Table 1 we identify the four core stakeholders influencing the direction and needs of the project. These four are: the wheelchair operator, the National Science Foundation, Cornell Cup, and other roboticists. "Other roboticists" is a key stakeholder as we developed subsystems for use in other robotics projects. This is reflective of our final goal to create a commercially viable product to convert a powered wheelchair to a semi-autonomous one. Subsystems are a product in itself, and must therefore be designed to suit both our wheelchair system and the needs of other roboticists. The stakeholders introduce needs to our project, Table 9. The needs vary from specifics such as payload (one human) to concepts like modularity. The main concepts created by our stakeholders are modularity, manipulation of environment, safety, navigation, and ease of configuration. Therefore each component designed must follow and contribute to these main concepts.

Table 1: Section of Stakeholder Table in scope of MQP

ID	Stakeholder	Description	Role	Method of needs elicitation
1	Physically/mentally challenged individuals	User, operator of the wheelchair	User, operator of the wheelchair	Survey or utilize available surveys
2	National Science Foundation	Government agency providing funds for research	Provides funds for four year research	1) Proposal
3	Cornell Cup	Competition using Intel atom boards for undergraduate projects	Provides Intel atom boards	1) Feedback on progress
4	Other Roboticists	Robotic hobbyists and professionals, including other smart wheelchair companies	Use separate components of the system as part of their projects	1) Blogs and popular robotics activity. 2) personal experience

The system requirements derived from the needs table in which the team must address to warrant a successful end result are listed below:

- System must be able to detect static and dynamic obstacles.
- System must detect obstacles outside of 6 inches and within 20 feet.
- System must be able to detect and avoid cliffs, such as a stairwell.
- System must be fully functional within an indoor environment (similar to that of the first floor of a common household).
- Sensors must becompatible with commonly used powered wheelchairs.
- System must be able to support a minimum of 30 sensors.
- System must be able to retrieve odometry data from the wheelchair.
- System must have the ability to conduct 3D mapping of indoor environments.

Systems engineering allowed us to create concrete requirements to meet in order to have a working and complete product that fulfills out main goal.

3.6 Decision chart

The decisions the team made in respect to fulfilling system requirements was based on a pro/con flowchart. The flowchart began by highlighting key high level components organized into three sections: mechanical, electrical, and software. These sections were broken up, depicted in the top left of Figure 11, into the major subsystems of the project.

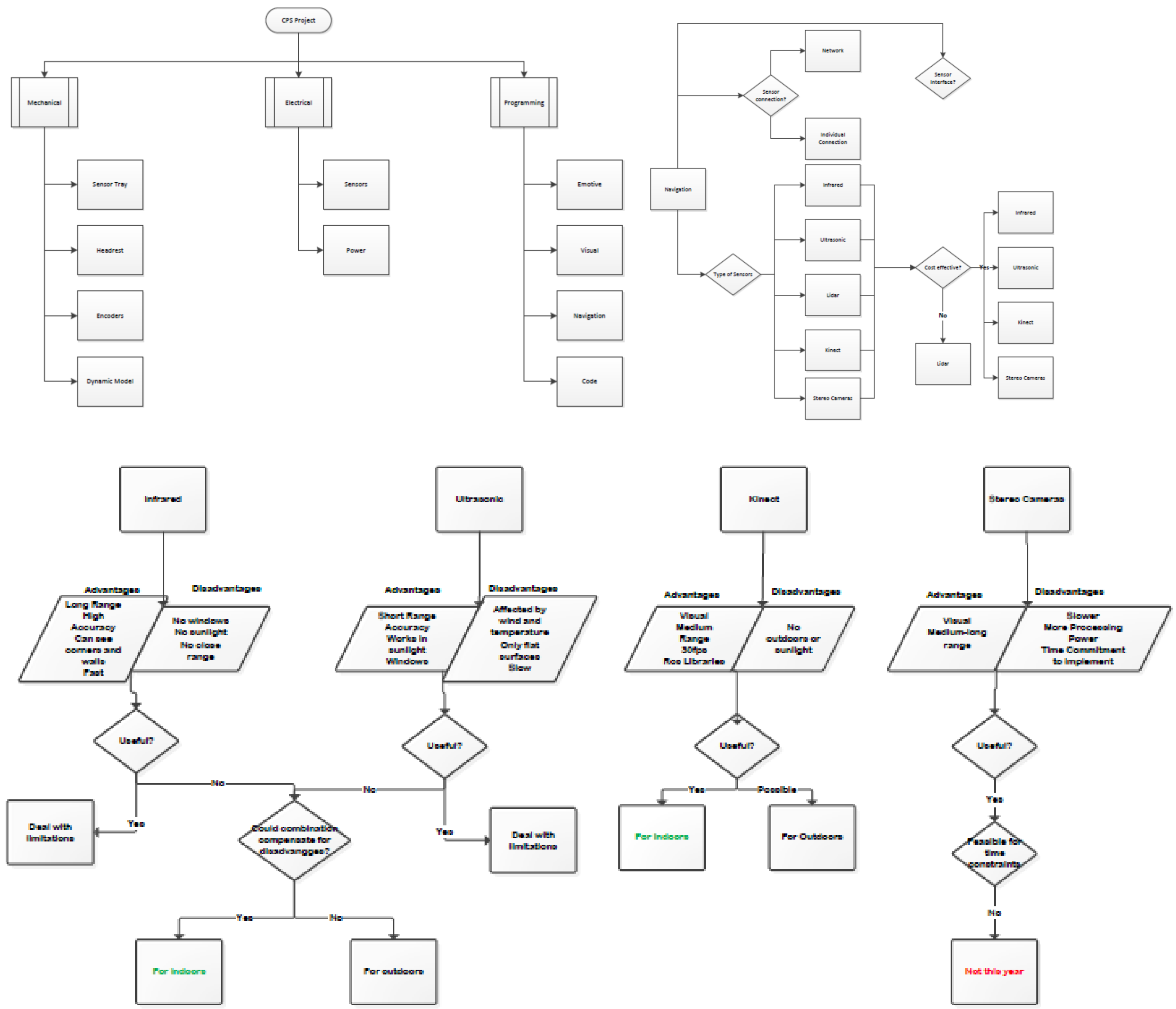


Figure 11: Section of decision chart general (left), more specific (right), and in depth (bottom)

The top right of Figure 11 shows a decision process for navigation, which is a top level requirement for the system. Each of the sensors and processes shown in that flowchart is expanded into sub charts, listing the advantages and disadvantages of each sensor. Alter careful deliberation with the project team we debated whether or not the component was useful towards the project's goals. These were determined by the template in Table 2: Component Weight Chart. Each category is given a value from zero to ten, zero being the least beneficial. This value is the percentage that is multiplied to the weights shown under each attribute.

Table 2: Component Weight Chart Template

Attributes (Relative Weighting)							
Option	Modularity (0.25)	Cost Effectiveness (0.2)	R&D Time Cost (0.2)	Ease of Implementation (0.15)	Life Cycle & Replacement (0.1)	Aesthetics (0.1)	**Total Score**

These weights were chosen for the strength each attribute had toward our project. The resultant score was between zero and one, with the higher score being the best option.

3.7 Specific Challenges & Requirements

The needs of the project imposed several important requirements for the project. These requirements include specifications of the environment in which the system operates, sensor options, odometry data collection, safety features, and other high priority decisions. The environment must be fully specified as all other components must comply with the requirements driven by outside interference. Next safety specifications driven by stakeholders are fully defined so that components are designed to fulfill the stakeholder's needs.

3.7.1 Environmental Requirements

To begin development, the team had to first define the environment in which the chair would operate. Operating in a wide variety of conditions increases the complexity of the system, as a larger number of potential scenarios would have to be taken into account. To begin to address the problem, the team decided to limit the environment to an indoor one. We outlined these requirements:

- There shall be no holes in the floor of any kind. A hole is an area which has a depth greater than 30mm.
- The minimum distance between any part of the ceiling or doorway(s) shall not be less than the maximum height of the wheelchair.
- The minimum width of a doorway shall not be less than the maximum width of the wheelchair in any orientation.

- There shall be no vertical deformations of any shape upon the driving surface of the chair whose height exceeds 30mm.
- The maximum elevation change from any point on the floor to another point on the floor may not be greater than 30mm.

3.7.2 Identifying Sensor Requirements

A critical design decision for this project was identifying what sensors to use. Knowing that the system would only be operated within simple indoor environments, we could use a wide variety of commercially available sensors; even sensors that would normally suffer in performance if used outside due to the sun's radiation or wind such as IR or ultrasonic sensors. While selecting sensors to use with this project, we carefully considered the following criteria for each sensor:

- Sensor's ability to detect obstacles
- Cost of sensor
- Implementation time of sensor

Ideally, the range-finding sensors used on the wheelchair would be able to detect all common household obstacles, be relatively low cost, and take little time to integrate into the project. Figure 12 illustrates the team's thoughts on IR and ultrasonic sensors, the Microsoft Kinect, LiDAR, and stereo vision according to the sensor criteria listed above.

Sensor	Outdoor Performance	Obstacle Detection	Cost	Implementation Time
Infrared	**Poor** Interference from sun's radiation greatly degrades data quality	**Good** Difficult to detect reflective and transparent objects such as mirrors and glass windows	**Excellent** Low cost sensor (~$10)	**Excellent** Easily integrated into project
Ultrasonic	**Good** Temperature and wind mildly affect data quality	**Good** Difficult to detect obstacles with curved surfaces	**Excellent** Low cost sensor (~$25)	**Excellent** Easily integrated into project
Kinect	**Poor** Interference from sun's radiation greatly degrades data quality	**Good** Color camera coupled with IR projection are able to detect most obstacles	**Good** Moderately priced sensor (~$100)	**Excellent** Easily integrated into project
LiDAR	**Excellent** Minimal adverse affects associated with outdoor operation	**Good** Difficult to detect reflective and transparent objects such as mirrors and glass windows	**Poor** Expensive sensor (~$2000)	**Excellent** Easily integrated into project
Stereo Vision	**Excellent** Minimal adverse affects associated with outdoor operation	**Good** Slow frame-rate and response time, unsuitable for dynamic obstacle detection	**Good** Moderately priced sensor (~100)	**Poor** Requires time consuming calibration, and programming

Figure 12: Evaluating the use of range-finding sensors for this project

To account for user safety, the sensors chosen should be able to detect obstacles as close as 5cm and as far as 10m. The lower bound of 5cm was selected so that any obstacle that gets too close to the system would be detected and initialize an immediate response to avoid either collision or harm to the user. The upper bound of 10m was selected so that the indoor environment could be mapped and high level obstacle avoidance algorithms could be used to respond to detected obstacles sooner, making the system safer and more reliable to use. Although each individual sensor does not have to meet both the upper and lower bound requirements, the combination of all sensors integrated into the system should be capable of meeting this requirement. Additionally, the placement of the range-finding sensors used in this project should be best suited for obstacle detection. System “blind spots”, or areas where no range-finding sensor is facing, should be reduced appropriately while also considering the costs and challenges associated with adding additional sensors.

3.7.3 Move to decision making for low level sensors

One of the major challenges with using IR based sensors, is being able to detect transparent obstacles such as sliding glass doors or windows. Therefore the team decided to use ultrasonic sensors in tandem to IR based sensors. The ultrasonic sensors are capable of detecting transparent obstacles. The drawback of using ultrasonic sensors is that using too many of them in close vicinity can result in interfering signals and curved or sound absorbent obstacles tend to be sensed as farther away than they truly are. Luckily IR based sensors are not affected by these types of obstacles and will provide accurate results. By using IR based sensors in conjunction with ultrasonic sensors, a diverse amount of obstacles can be detected.

3.7.4 Platform-Independent Odometry Collection

Odometry data is critical for closing the motor controller feedback loop and implementing basic dead-reckoning algorithms used to identify the current position of a robot with respect to some global frame of reference. The team quickly recognized that a method of collecting odometry data across a diverse number of commercially available powered wheelchairs would need to be developed for the successful implementation of this project. To accomplish this, the team had to abstract the features common to a large number of commercially available wheelchairs and build the design around these abstractions. Identifying the common features, however, proved to be challenging due to the lack of standardization within the powered wheelchair community. Additionally, the developed odometry collection module would have to be easy to mount on to a commercially available wheelchair, requiring only minimal additional hardware or tools.

3.7.5 Interchangeable Sensor Design

One benefit associated with prototyping platforms is the ability to rapidly interchange one component for another. The ability to rapidly swap in or out sensor packages was identified as a desirable quality for quickly testing HiLCPS platforms and soon became a requirement for the successful completion of this project. By enabling users to easily reconfigure the system's sensor suite, a more customizable system can be developed and more performance evaluation tests can be conducted. The following design requirements were developed to accommodate for interchangeable sensor design:

- System can accommodate a wide variety of sensors weighing less than 5 lbs.
- Mounting of sensors is simple and requires minimal hardware and tools
- Built in wire management features included to maintain system aesthetics
- Sensors can be swapped in and out within a twenty minute time window

3.7.6 Safety Features

Due to the nature of rehabilitative technologies operating so closely with humans, several built in safety requirements were established. These safety requirements were designed to not only protect the user from coming to harm, but also others within the system's region of operation. The key safety features agreed upon are listed below:

- System will not collide with static or dynamic obstacles while navigating
- System will be able to detect unsafe changes in elevation
- If an unsafe change in elevation is detected, the system will come to a halt and disable the ability for the user to travel in that direction

3.7.7 Assistive Control Behaviors

A trademark functionality of HiLCPS platforms is shared control between the user and the system. In this project shared control arises through the use of assistive control behaviors. Assistive control behaviors are designed to make navigation within an indoor environment easier on the user. By using assistive control behaviors, the user can provide the system with a general direction of where the user wants to go, but the system makes all the decisions on how to best get there. By adding this element of shared control, the mental strain of navigating through an indoor environment is reduced, allowing the user to instead, shift focus to more desirable topics of daily living. The requirements for assistive control behaviors are listed below:

- Speed control based upon proximity to obstacles
- System assists user with navigating parallel to walls
- System can override user input if in violation of safety rule sets
- System assists user with turning corners
- System assists user navigate through doorways
- All assistive behaviors are weighted to increase or decrease effect on system

3.7.8 Sensor Network

The electrical architecture of the robot had to be able to support a variety of sensors, as well as an unspecified number of sensors that might grow at any time. Many sensors, such as the Sharp IR and Maxbotix ultrasonic, have analog interfaces. This presented a problem, as no commercially available computers have a user-accessible analog interface for arbitrary inputs. The first step is digitizing the input data. This is not a particularly difficult problem to solve, as there are dedicated integrated circuits (IC) which can do the conversion to a digital signal and talk to a microcontroller. In addition, there are microcontrollers which have built-in analog to digital converters (ADC), which allows for operation without an external IC. Simplicity was another desired outcome of the solution to the sensor architecture, so the decision was made to use a microcontroller with an integrated ADC.

The next decision was on how to talk to a computer with a microcontroller. There are ICs that do this, such as those made by Future Technology Devices Incorporated (FTDI). These ICs are also sold as part of completed, working modules which can natively talk to microcontrollers, such as those manufactured by Gravitech. Both utilize USB as the connection to the computer, and talk over a Universal Asynchronous Receive/Transmit (UART) serial connection. There were pros and cons to using both as can be seen in **Table 3**.

Table 3: Pros and Cons of modular IC component

	Integrated Circuit	Gravitech Module
Pros	• Low production cost, only pay for IC and supporting components. • Depends only on IC manufacturer	• Module is already constructed and works with microcontrollers natively. • If module malfunctions, can be replaced easily. • Addition to the platform requires less change (single 1x6 header). • Removal of feature is easy.
Cons	• There would be an associated R&D time cost in development of the circuit that would work with an FTDI IC. • If the IC in question failed, the entire platform that depended on it would be unable to talk to a computer. Replacement would require unsoldering the IC and insertion of a new one, which would take time and require soldering skills. • If one wanted to use the sensor platform for purposes that did not include talking to a computer, then they would have to pay the additional cost of the IC when it was built.	• Higher cost than integrating FTDI IC directly into platform. • Dependent on outside source to provide modules which there may be less reliable than being dependent on IC manufacturer (FTDI ICs are used in other products)

After reviewing the pros and cons of each option, the decision was made to go with the Gravitech modules because it was a more modular component than an IC.

The next step was to choose the microcontroller to be used for communication. There were already two requirements for the microcontroller:

- Must have an analog-to-digital converter to be able to use analog signals
- Must be able to talk UART with external devices

In addition to these requirements, there were several considerations that the team had come up with which would make usage of the microcontroller easier. These considerations were partially based on the theme of modularity:

- Microcontroller should be re-programmable without much work.
- Microcontroller should be swappable if it were to break, or upgradeable to an extent if more program memory or random access memory is required.
- Reasonable human error in usage of the microcontroller should minimize the damage done and allow for normal operations. Unreasonable errors include intentional over-volting to an extreme degree, physically damaging the microcontroller, etc.

- Throughout the life of the product, the user should not have to worry about the microcontroller's operation being faulty as a result of varying temperatures, excessive reprogramming, reasonable voltage dips, or plugging the microcontroller in incorrectly.

While many microcontrollers can meet the requirements, there are fewer microcontrollers that can address even a majority of the considerations. The team was already familiar with microcontrollers from Atmel, which was the starting point for microcontroller research. Samples of various microcontrollers were obtained from Atmel to determine their feasibility for use on the sensor platform. As a result, the ATMega168 was chosen for having an ADC and a UART module, as well as its ability to operate at voltages as low as 2.7V, its re-programmability (up to 100,000 programming cycles) and its high temperature range (-40^{0}C to 85^{0}C).

With the components chosen for low-level communication and routing of data to the computer, the next step was to outline the requirements for the sensor platform. Given the theme of modularity, the team would need to be able to address a wide variety of sensors.

The primary sensors to be used on the robot were distance sensors. However, there are various methods of determining distance through use of different sensor technologies. Each has different advantages and disadvantages that needed to be considered. Given the advantages and disadvantages, the decision to use both ultrasonic and infrared rangefinders for low-level sensors was made. The specific models chosen for use are the Sharp GP2Y0A21YK0F and the Maxbotix LV-EZo model 08502. The Sharp IR was chosen because it was suited for measuring short distances with higher accuracy than longer distances, and the Maxbotix ultrasonic was chosen because it could measure larger distances. These sensors would not be used for mapping. The Microsoft Kinect was chosen for use in mapping, as its infrared beams cover a wide enough area to be suitable. In addition, a Hokuyo LiDAR was chosen for high-accuracy mapping. The primary reason for choosing both ultrasonic and IR sensors was that a combination of both allowed for a wider variety of objects to be detected (such as glass doors).

With the sensors chosen, the platform on which the chosen components needed to be developed. Firstly, for the platform to be as flexible as possible, it needed to be able to support even more sensors, including those that run at 3.3V, as well as digital sensors. Since the microcontroller operates at 5V, a regulator would be needed to convert power down to 3.3V. In addition, there would need to be a digital I/O interface which would allow for operation of potential digital sensors.

3.7.9 Odometry

Multiple options of commercial products are available to extract odometry data. Some of these encoders include absolute, incremental, magnetic, and camera based. After research and sparse development we were able to weight each of these options in the categories described previously. The decision weight chart is shown in Table 4. Using this table were able to identify the best solution for our project objectives. The Wheel-on-Wheel encoder had the highest total score and was pursued further.

Table 4: Weight chart used to select method of extracting odometry data

Attributes (Relative Weighting)							
Option	Modularity (0.25)	Cost Effectiveness (0.2)	R&D Time Cost (0.2)	Ease of Implementation (0.15)	Life Cycle & Replacement (0.1)	Aesthetics (0.1)	Total Score
Shaft Encoder	0: Motors may require significant changes for use	10: Only purchase two encoders	10: Commercial product	2: Requires drilling out motor and inserting centered shaft	10: High life cycle, easy replacement	10: Invisible or close to	0.63
Wheel-on-Wheel	9: Use on most wheelchairs without modifications to motors	8: Requires mount, wheel, and encoders	5: Requires research in materials and modular design, manufacture time	10: Can be easily mounted and integrated	9: Plastic parts easy to replace, lower life cycle	8: Depends on mechanical mount	0.805
Mouse Camera	8: Would work on any wheelchair, depends on surface	6: Depends on Camera, only requires one	1: Requires research into type of camera and drivers	5: Requires underside mount and significant software	5: Would require replacement Camera	10: Invisible or close to	0.565
Wireless Encoder	9: Would work on most wheelchair wheels	4: Would require pendulum or IMU	1: Requires significant research and development	5: Requires mounting system and significant software	8: Easy to replace with new module	8: Depends on mechanical mount	0.56

3.7.10 Visual Sensor Mounts

The two mapping sensors chosen for compatibility with the project were the Kinect and LiDAR. This presented the issue of how and where to put these visual sensors. The LiDAR for the project has a

270 degree planar view and the Kinect has a 43^o vertical by 57^o horizontal field of view [18]. The Microsoft Kinect also has approximately a two foot dead space where a point cloud cannot be generated accurately. In order to utilize the maximum potential view of each of these sensors they must be mounted strategically.

Figure 13: Original Power Wheelchair

Figure 13 displays the bare bones of a commercially available electrical wheelchair. There is very little surface area away from the user allowing for either sensor to be used efficiently. Due to the need for modular mounting capabilities, our components needed to fit as many wheelchairs as possible. After assessing powered wheelchairs such as those in Figure 2 the team found a number of wheelchairs utilize a headrest and nearly all of them a joystick. The joystick, being far enough away from the user and out in front of the wheelchair was in an optimal location for efficiently using the 270^o of visibility. This however brings up the issue of the distance from the ground and at what objects we would be seeing. Therefore the location of the joystick is optimal, although the mount will need be able to change its orientation.

Using the Kinect's 2' dead space and 43^0 vertical sight to our advantage, above the head of the user would allow for the team to see all obstacles in front of the system. Other models of powered wheelchairs utilize similar headrests and would therefore be a viable option for a configurable mounting component.

4 Methodology

Our goal was to create a product capable of providing handicapped individuals with the ability to live more independently and assist in mobility. The system must be as adaptable as possible to various other powered wheelchairs. The resulting prototype has the ability to be interfaced with a brain/computer interface as well as simple joysticks. It is able to assist operators through behavior based control and simultaneous localization and mapping [28].

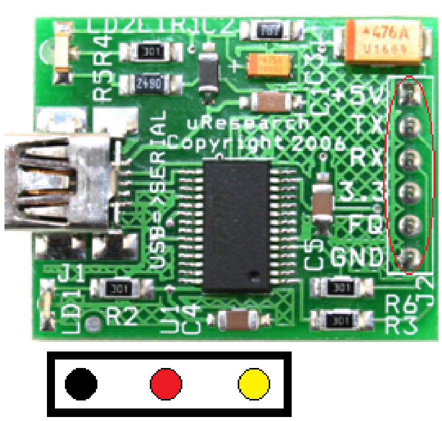
4.1 Objectives

The systems engineering approach identified stakeholders and, in turn, needs. These needs were combined and converted into a step-by-step general process which we call our objectives. The completion of the following objectives signifies the finished needs in the scope of our project.

- Define variability in electric wheelchairs
- Identify sensors for semi-autonomous navigation
- Design modular solutions for mounting sensors, Wheelchair Add-on Modules (WAM)
- Create a sensor network to communicate with the master computer, WAMNet
- Write software to utilize an unspecified number of sensors
- Demonstrate assisted control through use of sensor and software packages

4.2 WAMNet

Development of the network that would utilize the IR and ultrasonic sensors began with placing previously determined components. Each component (microcontroller, Gravitech USB-UART and the chosen sensors, needed an interface to the PCB that would house all of them, as shown in Figure 14. In addition to these components, there needed to be additional components to enable operation of the PCB.



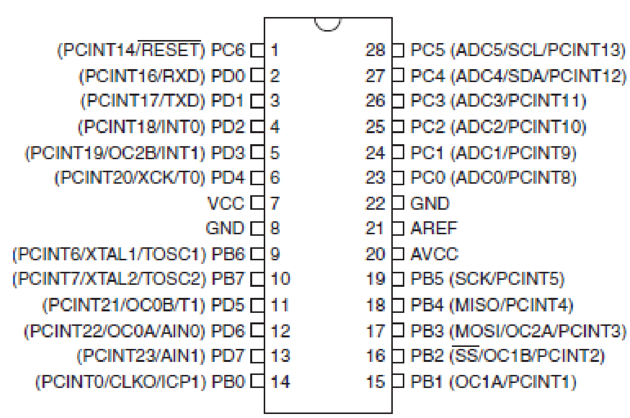


Figure 14: Diagram of the pinouts of sensors (bottom left), the Gravitech module (top left) and the ATMega168 (right). Sensor pinout is, from left to right, ground, power, and signal.

Operation of the microcontroller requires several additional components, first and foremost a clock source to ensure consistent operation. The choice of clock speed had several considerations associated with it:

- There are only certain clock speeds that will generate accurate speeds for UART transmission
- Higher clock speeds mean computationally intensive algorithms can be run faster
- Higher clock speeds increase power consumption of the chip
- Higher clock speeds increase the minimum voltage the chip can run at

The microcontroller would not be performing any computation-intensive algorithms by design; it was meant to route large amounts of data without any internal processing. Therefore a high clock speed was not essential. However, not every clock speed can generate every baud rate for serial communication. Since the default for computers is 115200 baud (symbols/second), we needed to choose a clock speed that could drive the UART module at this baud rate, as shown in Equation 1. In addition, a lower clock speed could be chosen to save on power consumption and decrease the required voltage for operation, shown in Figure 15.

$$BAUD = \frac{fosc}{16(\boldsymbol{UBRR}n + 1)}$$

Equation 1: Calculation of baud rate from clock speed and internal UBRR register.

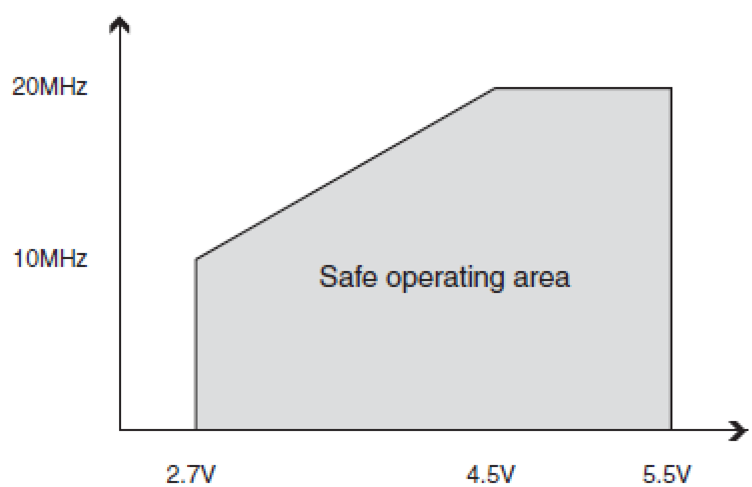


Figure 15: Maximum ATMega168 frequency vs. supply voltage

Since UBRRn register can only bet set to integer values, an integer value and clock speed needed to be found that could generate a baud rate of 115200. A plot of frequency versus the UBRRn register was created and used to decide upon possible clock speeds, as shown in Figure 16.

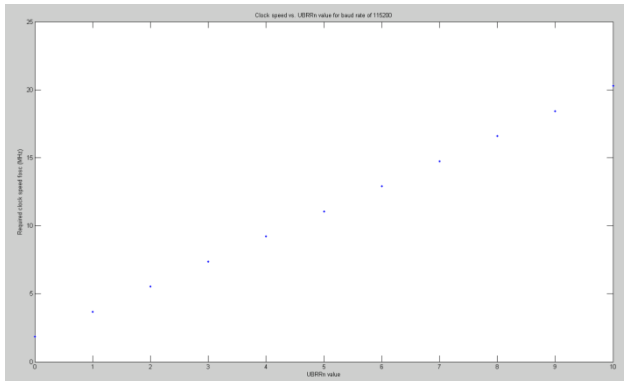


Figure 16: Plot of frequency required vs. UBRRn register setting.

From Figure 15 and Figure 16 the chosen clock speed was 3.6864MHz, for UBRRn equal to 1. This enabled lower power consumption, as well as operation even if the voltage were to dip significantly, and would still enable serial communication at 115200 baud. We could have dropped the clock speed down to 1.8432 MHz, however power consumption would already be very low with the microcontroller at 3.6864 MHz, with current draw being approximately 2mA at 5V, calculated from the graph in Figure 17. This also gives us extra computing headroom in case future projects need to use more computationally intensive algorithms.

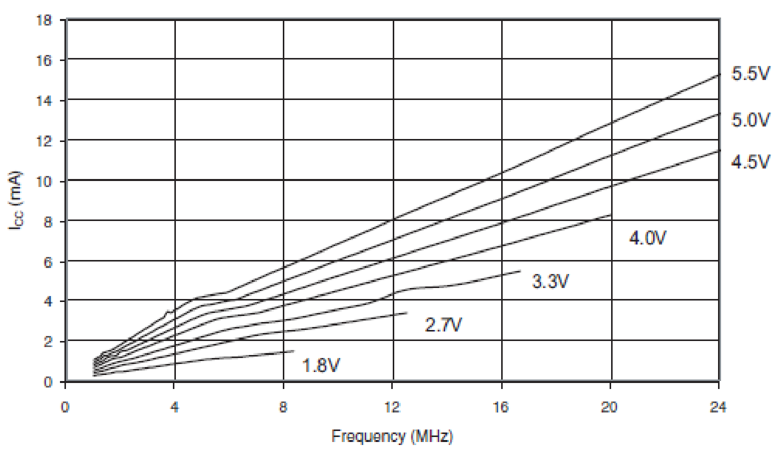


Figure 17: Graph of active power consumption of ATMega168 vs. clock speed

The rest of the parts were chosen without much discretion, such as a 3.3V regulator, headers for the board and capacitors for the crystal oscillator. To preserve modularity as much as possible, all components that could be bought in a through-hole mounting version were bought; it takes less skill to solder and de-solder a through-hole device than a surface-mount device. In addition, a DIP socket for the ATMega168 was chosen so swapping out broken microcontrollers would be easier. A decoupling capacitor was placed at the output of the 3.3V regulator and at the input of the 5V power supply for the board to help keep any potential transient noise in the circuit low.

With all components defined, the PCB layout needed to be designed. For the layout we used National Instruments' Ultiboard software. A key focus of the layout is that no two traces run directly on top of one another between each layer of the board to reduce interference, and an effort was made to bring the board size small. Using the circuit schematic the layout was produced by hand to optimize the routing, as can be seen in Figure 18.

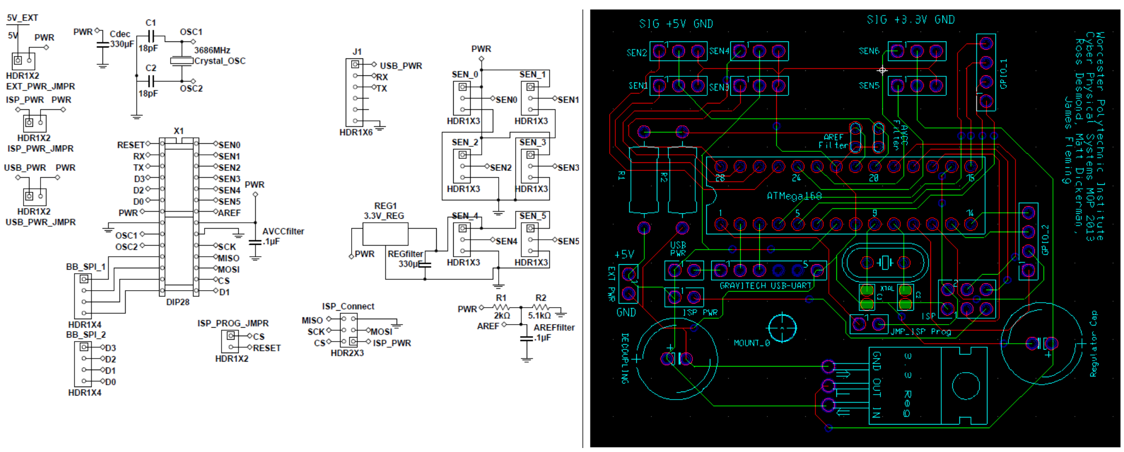


Figure 18: Production sensor hub schematic (left) and circuit layout (right)

This board is the version that was deployed on our project for use as the sensor network backbone. It features mostly through-hole components for ease of replacement, and was designed to be as simple as possible while being reconfigurable to a degree. There is almost no component on the board that isn't being used on the project with the exception of the digital I/O, which is there to provide potential use for the future without having to remake the hardware.

The next step was to design the architecture of the sensor network. The architecture of the routing modules was completed, but now there needed to be a way to connect multiple USB-enabled devices to a single computer. Fortunately, this solution already exists in the form of powered USB hubs. This allowed for a simple bus architecture which enables an easily-expandable network of sensors, as shown in Figure 19. This architecture allowed us to connect up to 127 sensor hubs together through powered USB hubs, and the interface would work without issues.

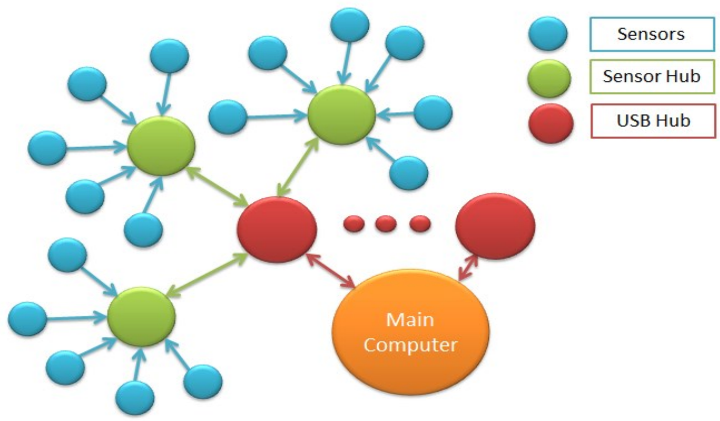


Figure 19: Sensor network architecture

For the purpose of our project the name became the Wheelchair Add-on Module Network (WAMNet). It is functional and will work with the sensors that we are using, while still being potentially useful in other robotics applications. On the system, we outlined a series of locations where it would be desirable to place sensors, described in Figure 20. The location of these sensors reflects the desired measurements to sense:

- Cliff data is absolutely essential to the project. Without it, the chair would be able to drive over stairwells, which would likely injure the driver. There are two IR sensors pointing straight down to detect such cliffs.
- We wish to avoid crashing into walls, as this poses a safety risk to the user as well as the chair. For this use we place sensors on the front plate facing forward.
- Sensors on the back of the chair will help prevent the chair from backing up into walls and people. This is a safety concern for the individuals around the chair, as the user may not be able to see them.
- Sensors on the side allow us to detect obstacles on either side of the chair, further making it aware of obstacles that it may encounter in the future.

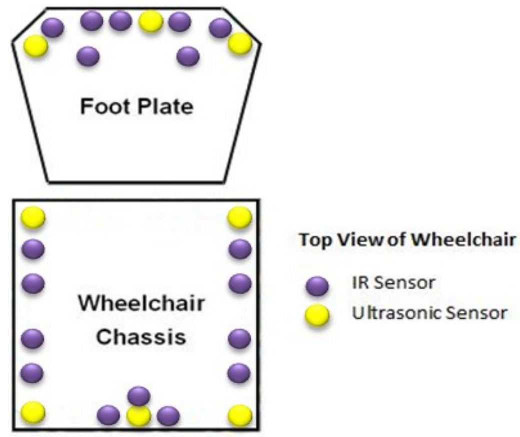


Figure 20: Sensor Location Diagram

With the given number of sensors, the team was confident that it could implement a degree of semi-autonomous behavior with the sensor packages that would enable a higher level of safety in use of a powered wheelchair for physically challenged individuals. With the layout of the sensors, it was decided that seven WAMNet hubs would go on the chair. Two located on the front, two on each side, and one on the back.

The next step was to interpret the data being sent back from the sensors. Due to the nonlinear nature of some types of sensors, there is a need to parameterize the data.

The first sensor that was parameterized was the Sharp IR. To do this, the team measured multiple data points, each of which consisted of a sensor reading and a distance from the sensor. Multiple curve fits were attempted to find the most accurate. The curve fits were:

Polynomial: $f(x) = \sum_n^t p_{t-n} * x^n : n = 0 \dots t$

Exponential: $f(x) = a * e^{b*x} + c * e^{d*x}$

Rational: $f(x) = \frac{\sum_n^t p_{t-n} * x^n}{\sum_w^j q_{j-w} * x^w} : n = 0 \dots t, w = 0 \dots j$

The most accurate curve fit was found through manual testing, and was a fourth order polynomial fit, as shown in Figure 21.

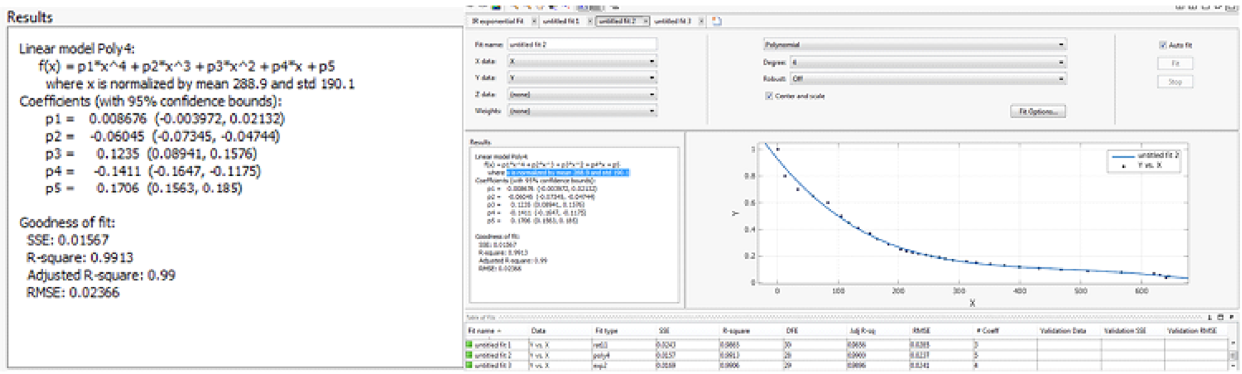


Figure 21: Results of IR parameterization. X-axis is sensor reading, Y-axis is distance (meters)

The same process was repeated for the ultrasonic sensors, as shown Figure 22.

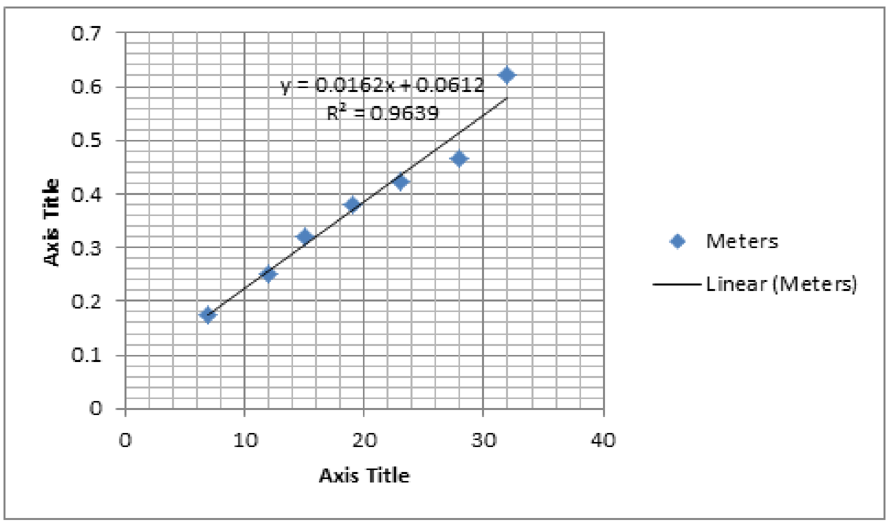


Figure 22: Results of Ultrasonic parameterization. X-axis is sensor reading, Y-axis is distance (meters)

With a function computed for both the ultrasonic and IR sensors, all measurements in the sensor driver could be in meters. This is a much more intuitive data format and is particularly useful for generating data structures such as point clouds and laser scans.

4.3 Encoder

The encoder module chosen through the systems engineering approach was the Wheel-On-Wheel encoder (WOW). The WOW encoder design addresses the variability between similar commercially available power wheelchairs. The wheel on the WOW was chosen to have the maximum friction coefficient on common wheelchair wheels. The wheel mount must be a configurable distance away from the wheelchair's motors and adaptable to motor length and diameter. Figure 23 compares two different powered wheelchairs with different wheel and motor size.

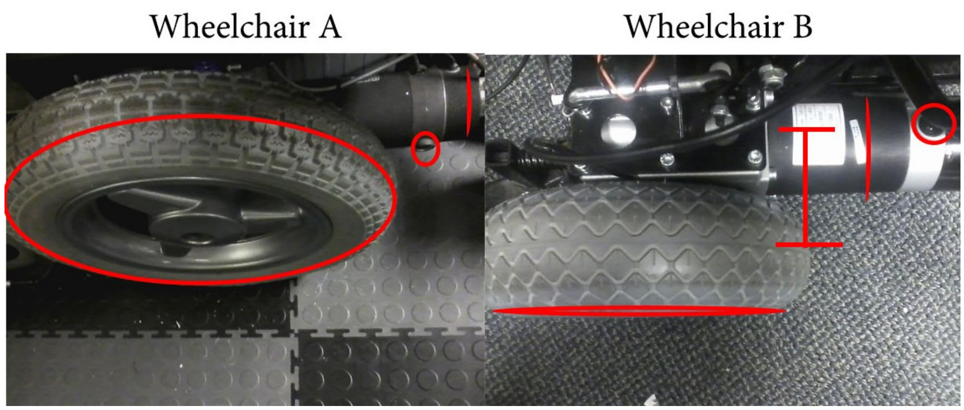


Figure 23: Wheel and motor of two different wheelchairs

The WOW encoder addresses the following variability:

- Wheel Diameter, WOW allows for a known circumference of module.
- Motor Diameter, WOW utilizes pipe clamps and matching C-Brackets to be configured to different wheelchairs.
- Distance from center of motor to center, addressed through a configurable lever arm
- Distance from perpendicular tangent line of the wheelchair wheel and the floor, a configurable slider is enclosed in the module using a screw to secure the slider in place at a desired position.

Figure 24: WOW encoder module

The lever arm holding the wheel and encoder utilizes a spring which provides more than 5 lb's of force in the linear direction. The lever arm is kept perpendicular to the module to allow the wheel on the WOW to be parallel with the wheel on the wheelchair. The linear slider and screw fixture shown in Figure 24 allows the entire lever arm with spring to move for adjusting to the diameter of the wheel and where the module could be mounted. The final module is shown on the left in Figure 24 and the technical drawings are located in Appendix C: Technical Documentation.

In order to test the reliability and accuracy of the WOW encoder it was necessary to added shaft encoders to the motor. This requires drilling a hole in the rear of the motor and centering a smaller shaft. In order to do this several centering techniques were employed such as a centering mount for the shaft. This was ineffective and required CNC machining to center the shaft in the motor and mount shaft encoders.

4.4 Footplate

The wheelchair footplate was originally molded plastic and rounded. In order to test and protect our system during the prototype phase a new footplate needed to be created that overhung the sensors in front of the user.

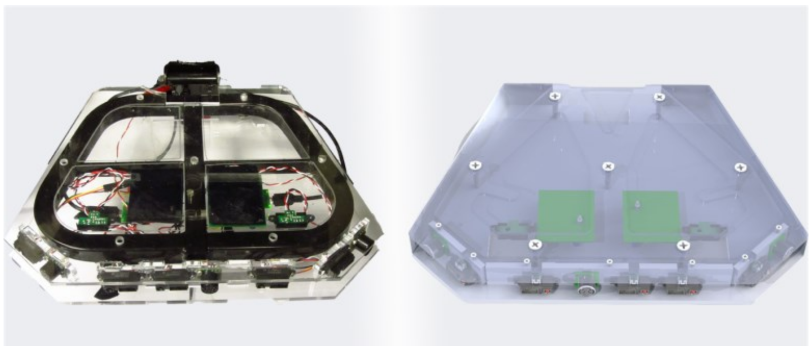


Figure 25: Acrylic footplate (left) and aluminum model (right)

The prototyped footplate was made out of acrylic, Figure 25, and overhung each sensor by a quarter inch. There are four sensors one the front (three IR and one ultrasonic) and two on each side (one IR and one ultrasonic). This arrangement provides maximum coverage of the front and sides while utilizing the IR and ultrasonic advantages to sense different obstacles. Finally there are two IR sensors pulled inside the footplate looking down, these two sensors are critical of edge detection and are the sole sensors responsible for identifying cliffs.

The new footplate allows for all the electronics to be pulled inside, keeping them safe from tampering or harm. This footplate is not as modular as the rest of the WAM's, as it was developed for the prototype only and other sensor cases created are responsible for protection elsewhere on the robot.

4.5 Sensor Cases

The section 2.2 Powered Wheelchairs shows various different types of popular powered wheelchairs. The body of each wheelchair is significantly different, creating a nearly impossible problem of designing a singular mounting solution for our entire sensor suite. After breaking down our WAMNet into separate components they are easier to mount individually and on a case by case basis. Depending on the needs of the operator, it is beneficial to allow a person to configure the wheelchair with individual sensors. Therefore our solution is to build individual sensor cases that do not utilize screws.

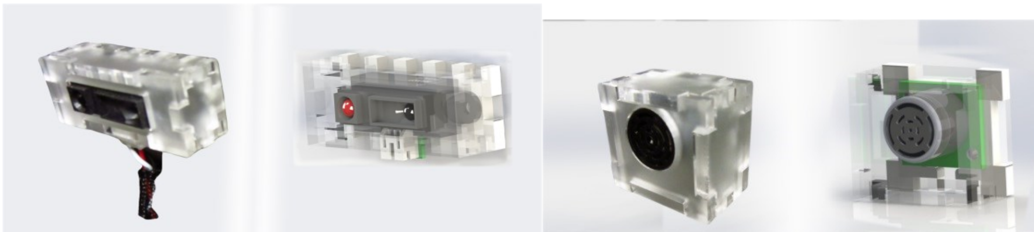


Figure 26: Infrared Sensor and case (left). Ultrasonic sensor and case (right)

The infrared and ultrasonic cases keep the main sensors protected from minimal collisions while keeping it stable. The back of the each mount uses VHB, a two-sided foam locking solution to securely mount to plastic or metal surfaces. Each of sensors is strategically placed along the wheelchair to offer maximum coverage.

4.6 LiDAR/IMU WAM

Systems engineering alluded to utilizing the mount of the joystick for the LiDAR. The issue with this location is the height. At this height the team would be unable to identify obstacles below three feet with the LiDAR. While we are using the WAMNet, we need to be able to rearrange the LiDAR to an optimal position in the future based upon results of SLAM. The configurability presents another issue, the LiDAR is planar and therefore the angles roll, pitch, and yaw must be known to transform the LiDAR's data to the global coordinate frame. The inertial measurement unit is able to measure the accelerations and rotations on the component. The IMU is also used for SLAM and is necessary for localization. Therefore mounting the LiDAR and IMU together is beneficial, Figure 27.

Figure 27: LiDAR/IMU Module

The module is a 3 Degree of Freedom configurable mount featuring a spherical joint. It is able to be locked into any roll, pitch, and yaw position. The IMU in mounted on the inside of the mount to protect it from collision or tampering, and is can be used to find the transformation from LiDAR to wheelchair coordinate frames.

4.7 Assisted Navigation

Technology discussed in Chapter 2.2 Powered Wheelchairs clearly shows a need for smart wheelchairs. Our objective was to demonstrate a safe and dependable navigation technique. Behavior based control theory allows for the system to react to dynamic and static obstacles alike while the user takes over the high level navigation. The reactions and assisted control over the system will help those either mentally or physically challenged drive an electric wheelchair.

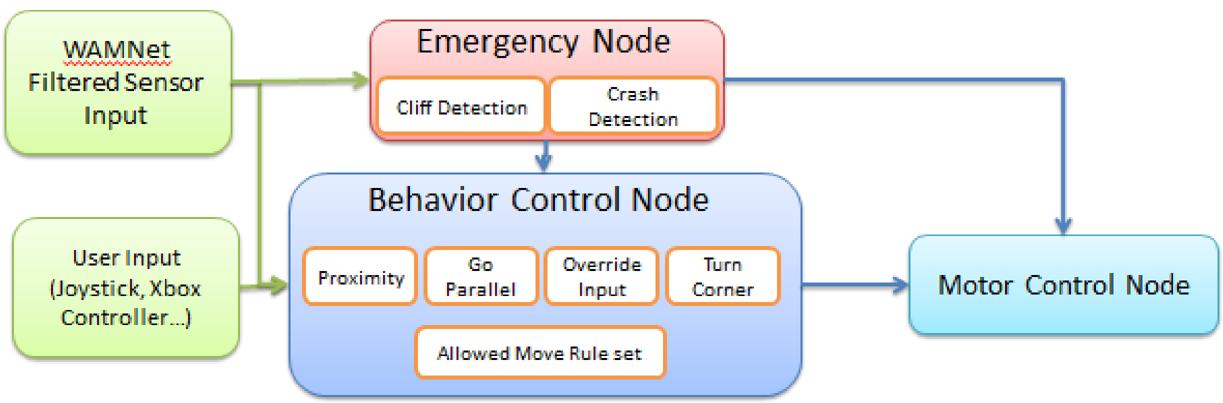


Figure 28: Behavior Based Control of the system

The behavior control node alters the input velocities based on interpreting filtered data from the WAMNet. Figure 28 displays a visual representation as to how the behavior control node operates. The velocities are altered through multiple stages of piecewise functions that limit the amount of change in the original control loop.

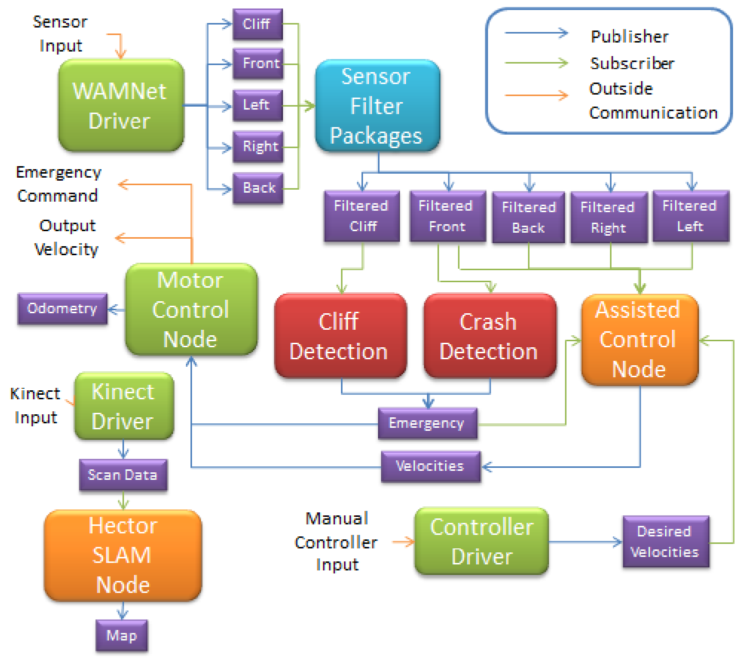


Figure 29: Overall ROS Architecture

In **Figure 29** the entire ROS architecture for this project is shown. This picture describes the logic and process from raw sensor and user input to final velocity control. The purple nodes are topics, green nodes are drivers which interact directly with raw sensor data, orange nodes are high level logic, and red nodes are low level reactive behaviors. Sensor Filter Packages is blue; it is a collection of nodes that perform filtering.

4.7.1 Proximity Behavior

The proximity behavior is function based. As implemented, the function is linear which is activated when either the front or rear sensors see obstacles and when the wheelchair is converging on the object. The linear function used was created to take a maximum and minimum distance from the robot, and then output a number between zero and one. The output represents a percentage of the input velocity which the system should be allowed to go. The linear velocity function is shown in Equation 2.

$$v_f \coloneqq linear\ velocity\ function$$

$$v_i \coloneqq user\ input\ linear\ velocity$$

$$X \coloneqq Array\ of\ sensor\ distances\ from\ front\ or\ rear$$

$$S_{min} \coloneqq Minimum\ sensor\ distance\ for\ zero\ velocity$$

$$S_{max} := Maximum\ sensor\ distance\ for\ zero\ velocity$$

$$k_w := proportional\ constant\ \in 0{:}1$$

Equation 2: Linear Velocity Control

$$v_f(X, v_i) = \left(\frac{\sum_{k=0}^{j} \overline{X_k}}{n} * \frac{1}{S_{min} - S_{max}} + \frac{S_{max}}{S_{max} - S_{min}} \right) * v_i * k_w + vi$$

The first term, $\frac{\sum_{k=0}^{j} \overline{X_k}}{n}$, is "x" in the linear fit equation: $y = mx + b$. This term represents the moving average over n sensors for a history of j previous sensor readings. M represents the inverse of the minimum and maximum possible sensor readings to create a linear coefficient that will allow the minimum sensed distance to be zero when reached. Finally the b term, $\frac{S_{max}}{S_{max} - S_{min}}$ is a divider to offset the function to reach a zero velocity state when the robot reaches a certain distance from an obstacle.

4.7.2 Go Parallel Behavior

This behavior affects the angular velocity of the system. Since the system can only rotate around the z axis, there are only two degrees of freedom. Therefore there is only one angular velocity: roll that is denoted by ω. On our system there are six sensors, 4 IR and 2 ultrasonic, on each side of the robot. This behavior is activated and weighted higher when the sensors are able to find a planar surface on either side of the robot. While driving, both sides are actively checking for walls or surfaces on either side of the robot, altering the control loop using a PID controller. This behavior uses the linear regression technique of least squares to identify the best fit linear line next to the wheelchair. The slope of the planar surface is used in the PID loop to allow the robot to converge to a zero slope. The final angular behavior control equation is shown in Equation 3.

$$\omega_i := input\ angular\ velocities\ from\ user$$

$$\omega_f := output\ angular\ velocity\ function$$

$$X := Array\ of\ \ x\ direction\ distances\ from\ sensors$$

$$Y := Array\ of\ y\ direction\ distances\ from\ sensors$$

$$kpb := weight\ constant\ on\ this\ particular\ behavior$$

$$kp, ki, kd := PID\ loop\ tuned\ constants\ for\ a\ slightly\ underdamped\ system$$

$$leastSquares := linear\ regression\ function\ for\ best\ fit\ line$$

Equation 3: Angular Velocity Control Equation

$$\omega_f(X,Y,\omega_i) = \omega_i + kpb\left(\sum_{n=0}^{SensorArrays} (kp * leastSquares(X,Y)_n + ki * \sum_{k=0}^{h} leasetSquares(X_k, Y_k)) + kd * (leastSquares(X,Y) - leastSquares(X,Y)_{previous})\right)$$

4.7.3 Override Input

The override input utilizes a predetermined rule set which includes absolute safety features. One feature is to not allow the user to continue driving into an object when it is within a certain distance from the wheelchair. Another is to keep the user from driving over a cliff as defined in the system requirements. These features are activated by the emergency node after the emergency stop has been activated.

4.8 Headrest

To provide additional space on the wheelchair to mount sensors too, a headrest sensor mount was created. A key quality of the headrest sensor mount is the ability to rapidly mount or interchange sensors. Individual sensors, such as the Microsoft Kinect, could be mounted onto modularly designed sensor plates. These sensor plates are able to be swapped in and out with other sensor plates as seen necessary by the user. The ability to rapidly change the positioning or even type of sensor used for an application makes the headrest sensor mount an excellent add-on for a HiLCPS prototyping platform.

Originally, the headrest sensor mount was built into a headrest. Our thought process was that the headrest (including the built in sensor mount) could be purchased as a single unit and used to replace a previously existing headrest. A prototype for this type of a headrest sensor mount was constructed. Stand offs attaching to generic sensor attachment plates protruded from the headrest. Sensors could then be mounted to the generic sensor plates. This design was ultimately rejected for several reasons:

1. It failed to meet aesthetic necessities
2. The design could not be reliably reproduced
3. Wire management quickly became an issue
4. Mounting sensors to the generic sensor plates proved to be challenging and frustrating

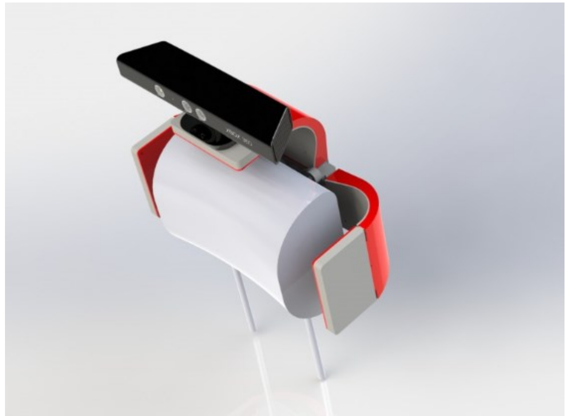


Figure 30: Headrest mount concept

To address the problems of the first headrest mount design, a new headrest design was developed Figure 30. One of the most distinct changes to the design was that the headrest mount was now completely separate from the headrest. The headrest mount was designed to attach to the headrest via the two headrest prongs and small clamp-on shaft collars. This design allowed users to use their original headrest and augment it with a stylish headrest sensor mount. The new headrest sensor mount was also designed to have built in wire management structures to provide the user with an aesthetically pleasing and simplified method of wiring the individual sensors to the main controller. The headrest sensor mount can accept three sensor mounting plates that can be interchanged with ease.

Figure 31: Headrest mount SolidWorks final design

These sensor mounting plates can be attached to the top, left and right of the headrest sensor mount and include built in wire management structures to provide users with a more aesthetically pleasing final product Figure 31.

4.9 WAMNet Software

With the sensor network hardware created, the code for the microcontroller had to be written. The code was written in embedded C using Eclipse. The code flow is outlined in Figure 32.

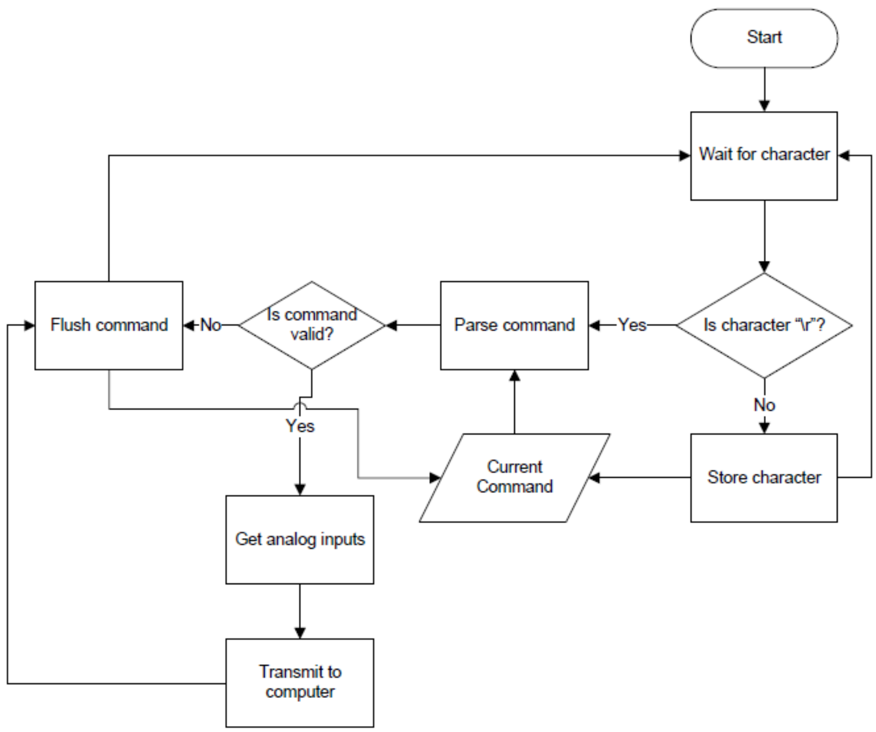


Figure 32: WAMNet Hub Code Flow Diagram

The command format that the sensor driver accepts is the form "sX\r", where X is the analog port (0-5) that the user wishes to read from. This simple command reduces overhead in data transmission and allows for theoretically faster data transmission than a larger command.

With the code for the sensor hardware written, a ROS node needed to be written to allow the rest of the software components to interact with it. Since all communication is done over a serial connection through a USB bus, the WAMNet hubs show up as FTDI devices on the computer. Code was written in Python due to its flexibility and the ability to divide up software components into individual modules.

The chosen method of reading the ports was Python's built-in serial library, which provides all the functionality needed to connect to a device over a UART connection. The method of reading needed to be fast, as the data rate was only 115200 baud, with 10 bits/baud giving us 11520 bytes per second. Reading from many boards in series would give a long wait time for boards not being read from. As a result, the refresh rate of the sensor data coming in to the computer would be very low, giving slower reaction times to events that might be threatening to the wheelchair. To solve this problem, it was decided that the boards would read in multiple threads running in parallel, allowing for all of them to potentially be read at once, giving much faster refresh rates. Fresh data would be stored in the WAMNet driver. At the same time the reading threads were running, a thread would be running which published sensor data to multiple topics. Other nodes would listen to these topics to obtain relevant data to their function. There are several requirements for the sensor network that were elicited:

- To convey the theme of modularity, extension of the network must be easy on the software side.
- Configuring the various topics which data can be published to should be easy.
- Multiple topics should be able to listen to the same data from a given sensor.

To meet the first requirement, we used the ROS parameter database, which can store many parameters under a given name and be accessed from any ROS node. The configuration of the sensor network was stored in a parameter file according to the board's location on the robot, the sensor name, and which port on the particular board it was. The format can be seen in Figure 33, where the board ID is 13WD80JZ, the ports are 0 and 1, the sensors are Maxbotic ultrasonic, and the sensor names are abbreviated "Front Ultrasonic" and "Front Left Ultrasonic".

```
sensorboard_driver:

  13WD80JZ:
    p0:
      type: MAXBUS
      name: FU

    p1:
      type: MAXBUS
      name: FLU
```

Figure 33: Format for sensor network parameter file

This format is easily extendable, requiring only a text editor. The ROS parameter database handles the parsing of the file, and parameters can easily be accessed through Python or C++ code.

Configuration of the topics falls into the same configuration file with a slightly different structure. In this case, there is a given topic name which contains a list of sensors underneath it. Upon launch of the WAMNet driver, the listed topics will publish the sensor data they are assigned. An example of such a configuration can be found in Figure 34.

```
data_topics:
  cliff_data:
    - CRI
    - CLI
  front_plate:
    - FFRI
    - FRU
    - FRI
    - FI
    - FU
    - FLI
    - FFLI
    - FLU
```

Figure 34: Example configuration of a topic to publish data

The format once again allows for easy extension and configurability of topics. In addition, a topic can have an arbitrary number of sensor readings that it publishes, and there can be an arbitrary number of topics. There were several topics decided upon which would run, which are outlined in Table 5.

.

Table 5: Topics and descriptions

Topic	Sensors
cliff_data	Cliff Right IR, Cliff Left IR
front_plate	Far Front Right IR, Front Right Ultrasonic, Front Right IR, Front IR, Front Ultrasonic, Front Left IR, Far Front Left IR, Front Left Ultrasonic
left_side	Left Side Front Ultrasonic, Left Side Front IR, Left Side Center Front IR, Left Side Center Back IR, Left Side Back IR, Left Side Back Ultrasonic
right_side	Right Side Front Ultrasonic, Right Side Front IR, Right Side Center Front IR, Right Side Center Back IR, Right Side Back IR, Right Side Back Ultrasonic
robot_back	Back Left IR, Back Right IR, Back Ultrasonic, Back Bottom IR

Once topics are configured, the next step was data interpretation. To keep code as modular as possible, there was a folder placed in our code which was designed to hold very simple but specific modules. These modules would be data interpreter modules, which had only an “interpret” function inside of them. This method would be accessed by the WAMNet driver to interpret a sensor of a specific type. In this way, new sensor interpreters could be added simply with the addition of a small module.

5 Results

With the system fully assembled, Figure 35, the team a number of testing procedures have been developed to validate the completed system and this section presents results or these tests. Development and testing occurred simultaneously for most components to ensure viable operation. Final testing consisted of overall analyses and summaries of testing for each measurable component.

Figure 35: Fully Assembled System

5.1 WAMNet

The first test of the WAMNet was ensuring that proper communication between a computer and the WAMNet hub was occurring. One of the desired pieces of information was the error rate of the communication link. To test this, known data was streamed from the board back to the computer and saved to a text file. This text file would then be scanned to confirm the data sent back was as intended. Any erroneous data would constitute an error. Initial conditions were as follows:

- Wheelchair powered on. For each series of tests motor is run at various speeds to generate electromagnetic interference similar to what would be encountered during operation
- Sensor board plugged in to 5V power. 5V rail measured 5.04V
- Battery voltage was 24.0V, as expected

- PuTTY set up to connect to the board over a serial connection at 115200 baud

The results of the test were relatively conclusive. For 100 kB of data gathered, there were no errors of any kind. While this says nothing about the actual error rate, it provides us a lower bound on the error rate. Supposing the very next byte received was an error, the error rate would be approximately one byte per 100,000.

With communication reliability verified, the next desired piece of data is the refresh rate of the sensor network. The higher the rate the better, as higher rates can allow for better control loops. To test this, a counter was set up inside the WAMNet driver. This counter would print a number which increments with each completed sensor reading. The average rate would then be calculated by timing the network for a specified period (three minutes), then checking the number of readings measured. The average rate would be the number of readings measured divided by the time period. Initial conditions were as follows:

- WAMNet driver run as normal with all desired sensor boards and sensors plugged in.
- Configuration file is written such that all sensors to be used on the project are read during operation.

The results of the test yielded 15661 sensor readings over the course of three minutes. This equals out to around 5220 per minute, or 87 per second. This gives us an 87 Hz refresh rate of the sensor network. This means that new data will be available for publishing every 87 Hz. The team believes this is appropriate for important ROS nodes running cliff and crash detection.

5.2 Wheelchair Add-on Modules

The following sections are results to verify the functionality and requirements set forth for each WAM. Using these WAMs a team of three can reasonably provide a powered wheelchair with assisted control functionalities less than ninety minutes. Some WAMs were tested through use in the prototyping phase and were evaluated based on practical application.

5.2.1 Footplate

The acrylic footplate was able to withstand any accidents through the prototyping phase. The wire management of the first prototype was negligible.

5.2.2 Wheel-on-Wheel Encoder

The WOW encoder was compared to a traditional shaft encoder output. This was done by placing the wheelchair on aluminum blocks to keep the wheels off the ground. The shaft encoder and respective WOW encoder were connected to the motor controller. The two encoder data were plotted via Matlab. The shaft encoder was mounted to the power wheelchair and directly compared, then scaled to the WOW encoder such as in Figure 36.

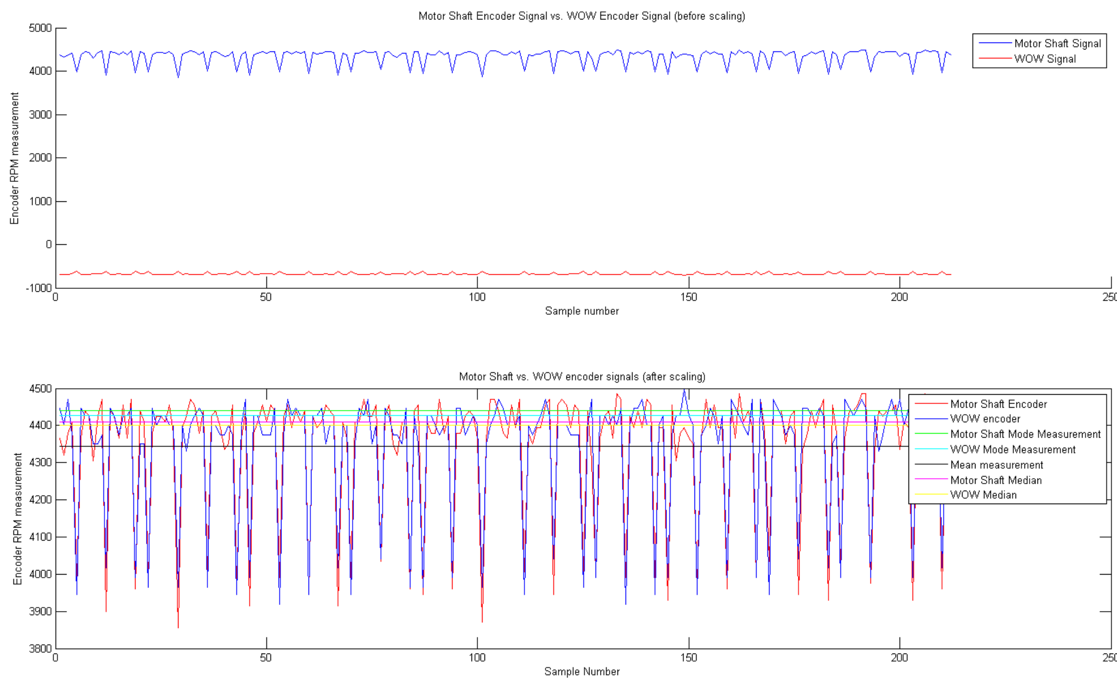


Figure 36: WOW encoder versus shaft encoder at high speeds

Table 6 shows a table of the results found in Figure 36. This compares the performance of the WOW to the encoder on the shaft of the motor at high speeds.

Table 6: Encoder data at high speeds (scaled)

Encoder Type	Mean	Median	Mode	Max	Min	Standard Deviation
Shaft	4343.844	4410	4440	4485	3855	171.0865

WOW	4343.844	4400.9472	4426.609	4497.178	3919.794	164.1623

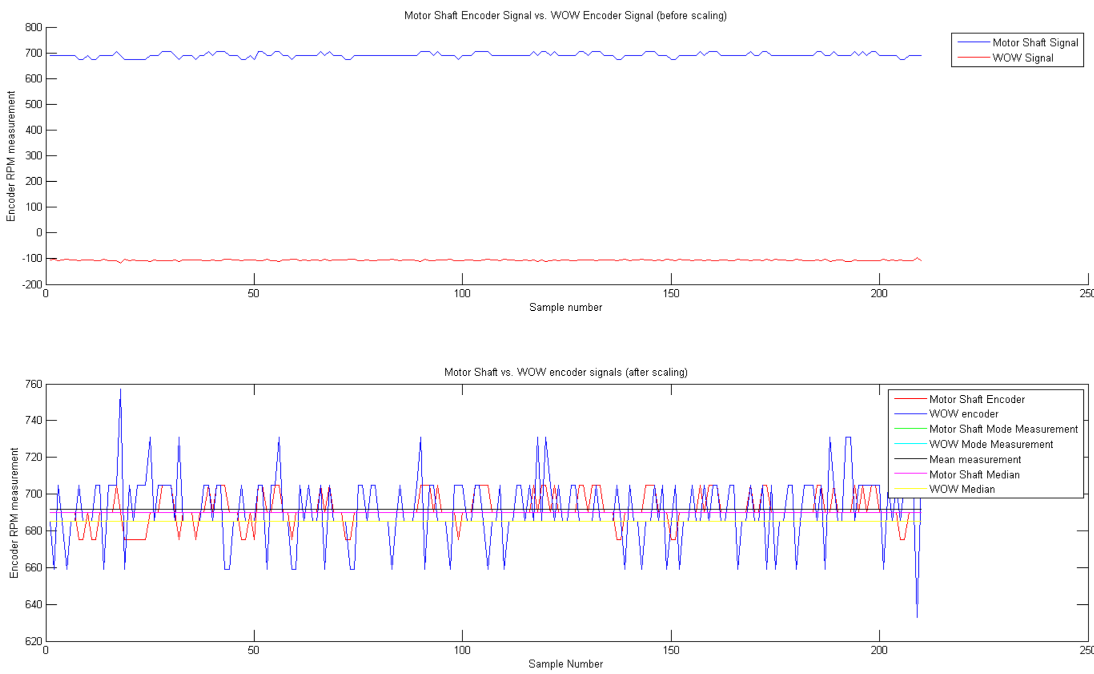


Figure 37: WOW encoder versus shaft encoder at low speeds

Table 7 shows a table of the results found in Figure 37. This compares the performance of the WOW to the encoder on the shaft of the motor at low speeds.

Table 7: Encoder data at low speeds (scaled)

Encoder Type	Mean	Median	Mode	Max	Min	Standard Deviation
Shaft	691.6429	690	690	705	675	8.7107
WOW	691.6429	685.2103	685.2103	756.9942	633.0038	18.0614

5.2.3 Headrest

The headrest was able to withstand 15 lbs. of pressure from sensors. This was tested by placing 5 lb. weights on each of the three arms extending from the headrest mount. The mount itself connecting to the two legs of the headrest used four clamp-on shaft collars. The collars above and below the plastic mount needed to be retightened periodically. The Kinect was stable while driving on flat or slightly abrasive surfaces. Using the Kinect on the headrest we were able to use Hector Slam to map one floor of a building accurately.

5.3 Assisted Navigation

The behavior based control node first utilized a Proportional-Integral (PI) controller. The PI controller caused the wheelchair to oscillate when the user attempted to drive away from the wall, causing a significantly under-damped solution Figure 38.

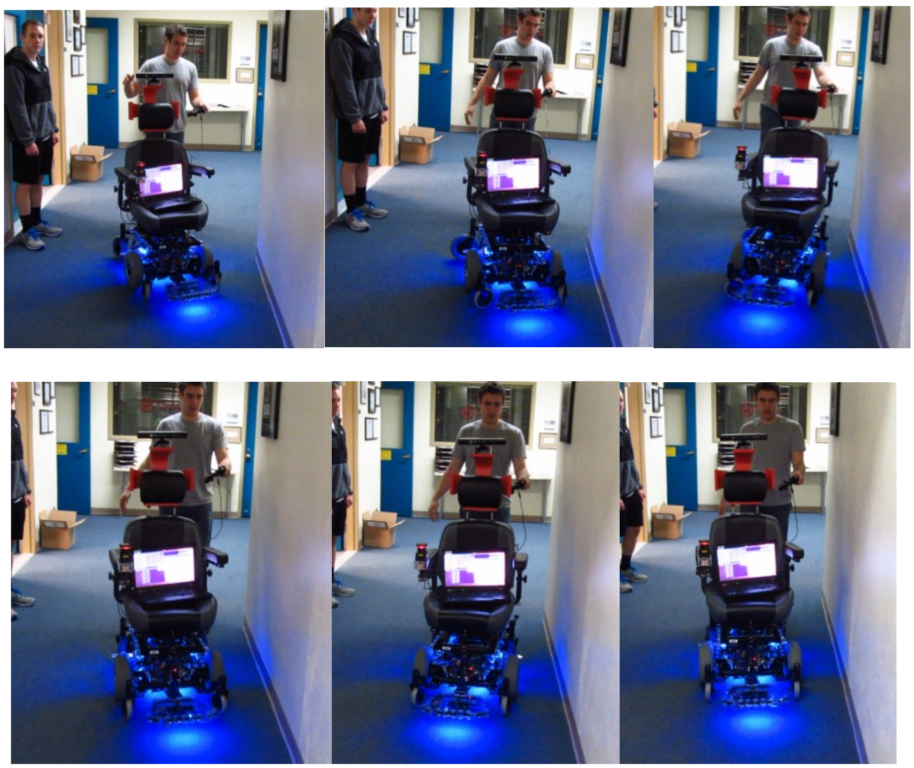


Figure 38: Time lapse of PI Oscillation

The PID controller was tuned slightly over damped; the system did not oscillate and would converge to a zero slope over time, and therefore be optimal for assisted control. This is optimal as it does not overly force the operator to stay parallel to the wall.

The proportional proximity controller was a linear function and would never reach a zero speed. The emergency node would trigger before the function limited the user to a negative velocity. The proximity controller was able to slow the wheelchair to a safe speed before collision without the emergency node. The full video demonstration can be found as an attachment on the MQP project website under the WPI Gordon Library.

6 Analysis

This section analyzes the results for the system. It includes the WAMNet, WAM's, and control of the wheelchair.

6.1 WAMNet

The WAMNet sensor network performed well in our implementation of the system. Forming the backbone of emergency behaviors and safety features, it effectively routes data from the sensors to the ROS nodes that require the data. Configuration for the network is located in a single file, and setup is as simple as plugging in a new device and adding the ID into the configuration file. Allowing software to use the newly added device requires only creating a topic that uses sensors attached to it.

Regarding usage for other robotic applications, it effectively turns analog sensor readings into digital ones inside of a modern PC. However, it is still limited to applications that use ROS; to overcome this, new software would need to be written which could utilize the sensor network on different computer architecture (such as ARM).

6.2 Wheel-On-Wheel Encoder

The WOW encoder was able to compare extremely well to the shaft encoder. At high speeds the two types of encoders were nearly the same, whereas at low speeds the shaft encoder was less sensitive than the WOW. However the mean each type of encoder was the same.

6.3 Assisted Navigation

Behavior control helped the user by avoiding or stopping before obstacles. The proximity and go-parallel behaviors combined allowed the wheelchair to approach a wall at a 60 degree or less angle and correct itself to be parallel to the wall. The proximity node slowed the wheelchair as it approached to the wall. When the side sensors were in range of the wall the parallel behavior was activated and the wheelchair would avoid the wall. The combination of these behaviors allowed the wheelchair to navigate through doorways if the user points the chair in the general direction of the doorway, Figure 39.

Figure 39: Assisted Doorway Navigation

The cliff sensors were able to stop the wheelchair when moving forward and perpendicular to the stairs, Figure 40: Cliff Detection Lapse.

Figure 40: Cliff Detection Lapse

6.4 Headrest

The headrest worked, holding the Microsoft Kinect just over the head of people less than 5' 11". In order to accommodate individuals greater than this height, a new base must be swapped out from the assembly. The entire headrest and assembly would oscillate when given a pulse input signal, such as the wheelchair hitting a bump. This only happens without a user in the seat, otherwise the headrest is stable.

7 Discussion

This section serves to interpret the overall performance of the system. It describes how our solution was able to meet initial requirements and propose future improvements for each WAM and controller.

7.1 Was goal achieved?

The team's project goal was to design modular components that, when combined as system, would provide a wheelchair with a degree of semi-autonomy that would assist users of powered wheelchairs. We believe that goal has been accomplished. The WAMNet provides effective routing of large amounts of sensor data to software running on the wheelchair. The WAM's were able to be mounted onto another type of wheelchair and the entire system was assembled under 90 minutes. Emergency features such as cliff detection and crash detection were successfully implemented, providing a degree of safety. Proximity detection features allow us to control the speed of the wheelchair based on surrounding obstacles, providing another degree of safety to users in the chair, as well as surrounding users. The wall following behavior weights could be tuned to be stronger based on the individual operating the robot. It steered the system away from walls and keep them parallel in a corridor. Finally, assisted navigation through doorways relieves the user of some of the challenges of navigating through doorways.

7.2 Recommendations

The WAMs could use some future work is the following areas:

- The wheelchair headrest mount should utilize a damper, such as a ribbing of rubber on the inside of the mount to reduce the oscillation from a step input to the system.
- The WOW encoder should use a swing arm made out of Delrin and a guard should be made around the arm to protect from accidents.
- The current footplate is made out of acrylic; however more suitable footplate would be made out of aluminum utilizing the same CAD drawings.
- Behavior control should implement other reaction based behaviors such as:
    - Follow person: use the front sensors to attempt to match the velocity and movement of an object in front
    - Keep distance from objects: attempt to push off walls or obstacles based on proximity to avoid obstacles more aggressively

- Detect corner: will help the wheelchair to assist the user in getting out of a corner
- The WAMNet should be less CPU intensive, currently it uses 6 of 8 cores in order to run the drivers and get the data from the sensor hubs

While the system performs well now, these adjustments will make the system perform better overall and protect it from damage.

7.3 Social Considerations

Up until this point we have tested and verified individual components and system integration/capabilities. The next step is to consider human testing as this product is intended for use in clinical scenarios. This would require IRB approval for human trials to collect feedback and continue development. This would allow the behavior based controller to be tuned to an operator's preference.

8 Conclusion

This project encompassed elements from disciplines such as electrical and computer, mechanical, robotics, and systems engineering. Upon submission of this report, all system requirements have been achieved, warranting the project a complete success! Through the use of the WAMs and WAMNet, we were able to provide a prototyping platform for a wide variety of commercially available wheelchair designs. All components were tested to verify functionality. Assistive control behaviors successfully implemented, enabling users to navigate within indoor environments with increased ease of use and safety, allowing them to focus shift focus from the challenges of operating a powered wheelchair to more desirable aspects of daily living.

References

[1] E. Smith and M. Delargy, "Locked-in Syndrome," *BMJ,* vol. 330, pp. 406, 2005.

[2] G. Schirner, D. Erdogmus, K. Chowdhury and T. Padir, "The Future of Human-in-the-Loop Cyber-Physical Systems," *Computer,* vol. 46, pp. 36, 2013.

[3] (). *Definition of Assistive Technology*.

[4] (). *Spinife Experts in Motion*.

[5] (). *Sherman Oaks*.

[6] (). *EasyMed*.

[7] (). *Hoveround*.

[8] R. C. Simpson, "Smart Wheelchairs: A Literature Review," *Journal of Rehabilitation Research & Development,* vol. 42, pp. 423, 2005.

[9] R. C. Simpson, D. Poirot and F. Baxter, "The Hephaestus Smart Wheelchair system," *IEEE Transactions On,* vol. 10, pp. 118, 2002.

[10] P. Edmund F. LoPresti, P. Vinod Sharma, P. Richard C. Simpson and L. C. Mostowy, "Performance testing of collision-avoidance system for power wheelchairs," *Journal of Rehabilitation Research and Development,* vol. 48, pp. 529, 2011.

[11] (2013). *CanWheel Project* [CanWheel]. Available: http://www.canwheel.ca/our-research/current-projects/.

[12] (). *Robotics Fundamentals Series: LiDAR*.

[13] A. Priddle and C. Woodyard, "Googe discloses costs of its drivesless car tests," *USA Today,* June 14, 2012.

[14] (). *Pro's and Con's of Range Sensors*.

[15] (). *Microsoft's Sleek New IntelliMouse Explorer and IntelliEye Technology Toss Out the Mouse Ball Today at COMDEX*.

[16] (). *BAE SYSTEMS and Segway Inc. Announce Partnership to Market Segway™ Personal Transporter in the UK*.

[17] (). *Logitech HD Pro Webcam C920*.

[18] (). *Kinect for Windows Sensor Components and Specifications*.

[19] (). *Repositories - ROS Wiki*.

[20] (). *ROS: an open-source Robot Operating System*.

[21] S. Thrun, W. Burgard and D. Fox, "Simultaneous localization and mapping," in *Probabilistic Robotics*Anonymous Cambridge, MA: MIT Press, 2006, pp. 309.

[22] E. Demeester, A. Hüntemann, D. Vanhooydonck, G. Vanacker, H. Van Brussel and M. Nuttin. User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving. *Autonomous Robots 24(2),* pp. 193-211. 2008. Available: http://dx.doi.org/10.1007/s10514-007-9064-5. DOI: 10.1007/s10514-007-9064-5.

[23] D. Stefanov, A. Avtanski and Z. Bien. "17 A concept for control of indoor-operated autonomous wheelchair," in *Advances in Rehabilitation Robotics*, Z. Bien and D. Stefanov, Eds. 2004, Available: http://dx.doi.org/10.1007/10946978_17. DOI: 10.1007/10946978_17.

[24] C. Su, S. Rakheja and H. Liu. 3D semantic map-based shared control for smart wheelchair. pp. - 41-51. 2012. . DOI: 10.1007/978-3-642-33515-0_5.

[25] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation,* vol. 2, pp. 14, 1986.

[26] P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell and H. J. Spoelder, "Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT,* vol. 52, pp. 1335, 2003.

[27] Machiel Van der Loos, H.F. and D. J. Reinkensmeyer, "Rehabilitation and Health Care Robotics," *Springer Handbook of Robotics,* pp. 1223, 2008.

[28] R. Desmond, M. Dickerman and J. Fleming, "Development of Modular Sensors for Semi- Autonomous Wheelchairs," *IEEE International Conference on Technologies for Practical Robot Applications (TePRA),* April 22-23, 2013.

Appendices

Appendix A: Authorship

Content	Author
Introduction	Matthew Dickerman
Background (Intro)	Ross Desmond
Assistive Technology	Ross Desmond
Powered Wheelchairs	Ross Desmond
Semi-Autonomous Wheelchairs	Ross Desmond
Sensing Technologies	James Fleming
Robot Operating Systems	James Fleming
Mapping & Navigation	Ross Desmond
Simultaneous Localization & Mapping	Matthew Dickerman
Bayesian Approach to Motion Planning	Ross Desmond
External Sensor Navigation	Ross Desmond
Advanced 3D Map-Based Control	Ross Desmond
Behavior Based Control	Ross Desmond
Modular Systems	James Fleming
Context and Motivation	Matthew Dickerman
Locked-in Syndrome	Matthew Dickerman
Human in the Loop Cyber Physical Systems	Matthew Dickerman
The Growing Need for Rehabilitative Technology	Matthew Dickerman

Decision Process	Ross Desmond
Modularity	Ross Desmond
Cost-Effectiveness	Ross Desmond
R&D Time Cost	Ross Desmond
Ease of Implementation	Ross Desmond
Systems Engineering Approach	Ross Desmond
Decision Chart	Ross Desmond
Specific Challenges and Requirements	Matthew Dickerman
Environmental Requirements	Matthew Dickerman
Identifying Sensor Requirements	James Fleming
Platform-Independent Odometry Collection	Matthew Dickerman
Interchangeable Sensor Design	James Fleming
Safety Features	James Fleming
Assistive Control Behaviors	James Fleming
Sensor Network	James Fleming
Odometry	Matthew Dickerman
Visual Sensor Mounts	Ross Desmond
Methodology	James Fleming
Objectives	Ross Desmond
WAMNet	James Fleming
Footplate	Ross Desmond

Sensor Cases	Ross Desmond
LiDAR/IMU WAM	Ross Desmond
Assisted Naviation	Ross Desmond
Headrest	Matthew Dickerman
WAMNet Software	James Fleming
Results	James Fleming
WAMNet	James Fleming
Wheelchair Add-on Modules	Ross Desmond
Assisted Naviation	Ross Desmond
Analysis	James Fleming
WAMNet	James Fleming
Wheel-On-Wheel Encoder	Ross Desmond
Assisted Navigation	Ross Desmond
Headrest	Ross Desmond
Discussion	James Fleming
Was goal achieved?	James Fleming
Recommendations	Ross Desmond
Conclusion	Matthew Dickerman

Appendix B: Systems Engineering

Table 8: Stakeholder Table

ID	Stakeholder	Description	Role	Method of needs elicitation	Relevance to CC and MQP
1	Physically/mentally challenged individual	User, operator of the wheelchair	User, operator of the wheelchair	Survey or utilize available surveys	TRUE
2	National Science Foundation	Government agency providing funds for research http://www.nsf.gov/	Provides funds for four year research	1) Proposal 2) Recommendations	TRUE
3	Cornell Cup	Competition using intel atom boards for undergraduate projects http://www.systemseng.cornell.edu/intel/	Provides intel atom boards	1) Feedback on progress	TRUE
4	Other Roboticists	Robotic hobbyists and professionals, including other smart wheelchair companies	Use separate components of the system as part of their projects	1) Blogs and popular robotics activity. 2) personal experience	TRUE
5	Domestic residents	People who live in the same household	Assist the locked-in individual when operating system	Survey or utilize available surveys	FALSE
6	Professional Caretakers	Hired to take care of user, close personal relationship	Be aware of how to use the system and be prepared to take user in and out of the wheelchair	1) Research services of professional caretaker companies. 2) Survey or utilize available surveys	FALSE
7	Doctor (medical Professional)	Hired to evaluate and prescribe treatment to the user, recommends rehabilitation methods	Evaluate and recommend the system	1) Personal interaction with doctors	FALSE
8	Trainer	Professionals to train operator and care takers of the system	Needs to configure and understand system	1) Project the needs	FALSE
9	Maintenance	Professionals caretaker of the system itself	Debugging and replacement of system components	1) Project the needs	FALSE
10	Assembler	Professionals responsible for adapting the system to an existing wheelchair	assemble the system	1) Project the needs	FALSE

11	Government: Healthcare regulations		Provides restrictions, regulations, and specifications for assistive technology	1) Identify standards and regulations through public resources	FALSE
12	Health Insurance Companies	insurance companies that may subsidize system	Subsidize cost of system	1)explore policies related to existing assistive devices	FALSE
13	Property Insurance companies	Insure damage and accident costs	Insure damage and accident from system	1)explore policies related to existing assistive devices	FALSE
14	General population	Anyone who interacts with the system apart from those previously covered	Interact directly with the system	1) Survey or utilize available surveys	FALSE

Table 9: Project Needs

ID	Need	Description	Cost	Source/Stakeholders	Priority
1	Modularity	It shall be possible to use individual components of the system for other robotic applications	high	S1,S2,S4	2
2	Mobility	The system shall be able to transport the LI in a household environment	low	S1,S2	1
3	Manipulation of environment	The system shall allow the use to manipulate common household objects	high	S1,S2,S4	10
4	BCI command driven	The system shall be able to accept commands from BCI	medium	S1,S2	5

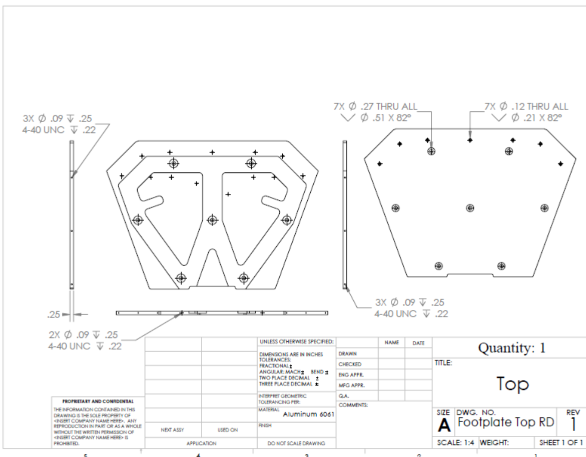
5	Safety	If an individual component breaks the system shall react accordingly. If the system breaks any individual component shall switch to the safest mode. System shall be able to report that the system is not working.	high	S1,S2	8
6	Security	System shall provide at least moderate protection against theft of personal data	high	S1,S2	9
7	Semi-autonomous navigation	System shall be able to navigate autonomously based on LI input	medium	S1,S2,S4	1
8	Payload	The system should be able to carry one human	low	S1,S2	1
10	External Operation	The system shall have a means to control the system outside of the LI.	low	S1,S2	3
11	Recharge	The system shall have a means to be recharged by the LI	medium	S1	4
12	Ease of LI extraction	It shall be easy to remove the LI from the system	low	S1,S5,S6	5
13	Flexibility of system inputs	It shall be possible to control the system via different input devices including BCI, joystick, keyboard	medium	S1,S5,S6	2
14	Self-feeding	The system shall be able to feed the LI by command with food from the common household	high	S1,S2,S5,S6	3

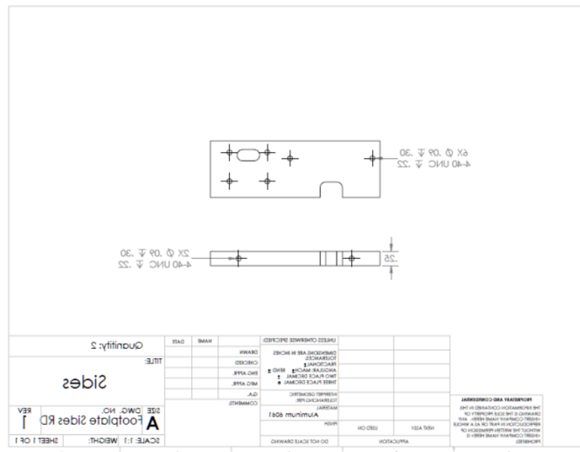
15	Means of training	The system shall have a way to be easily adapted to various LI conditions	high	S1,S5,S6	4
16	Maintainability	Replacement of components should be cost and time effective.	medium	S1,S8	6
17	*Atom board*	*The system must use an atom board*	*low*	*S3*	*1*
18	Ease of configuration	The integration of components to the system should minimize the possibility of error	Medium	S4,S9,S10	1
Definitions:	LI: locked-in individual				

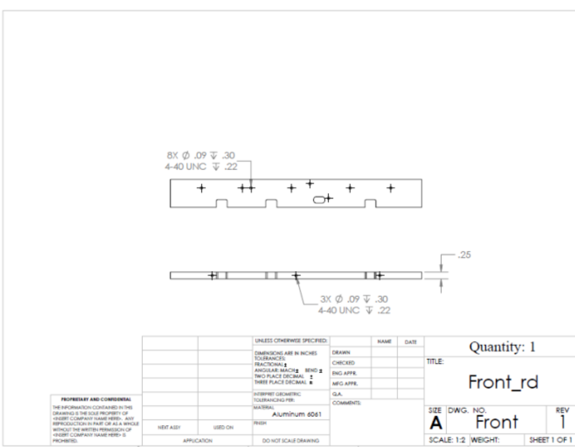
Appendix C: Technical Documentation

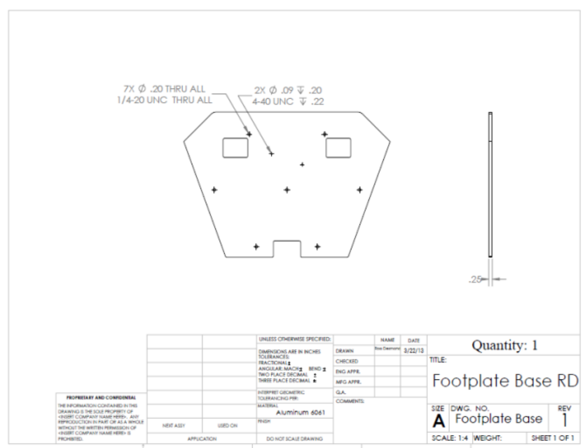
This section provides technical documentation for the footplate, WOW encoder, headrest sensor mount, and WAMNet. Full CAD models of each WAM have been provided for reference.

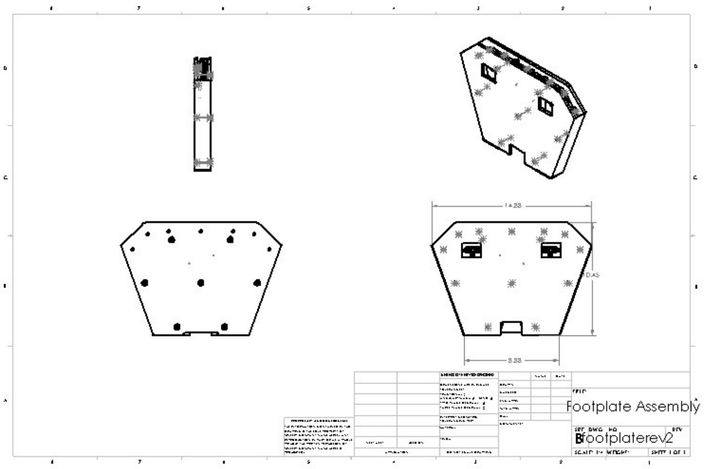
Footplate



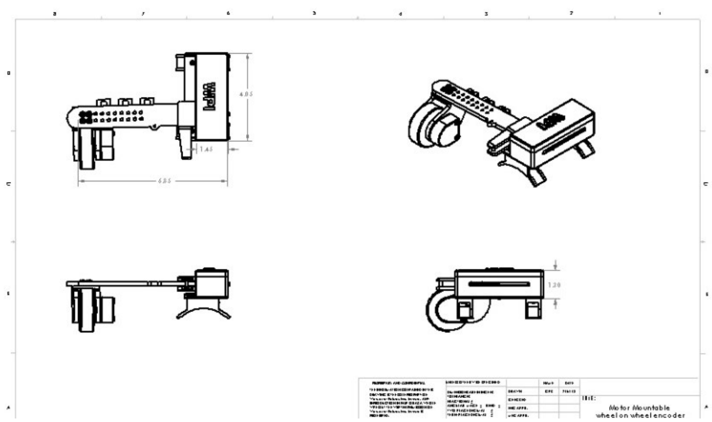




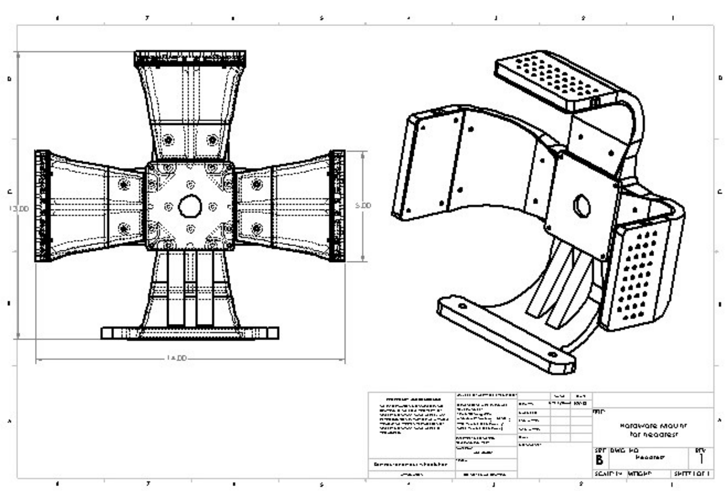




Wheel-on-Wheel Encoder



Headrest Sensor Mount



WAMNet Specifications

Interface	USB	
Sensors per Hub	Max: Two 3.3V, Four 5V	
Transmission Speed	CPS: 82 Hz per sensor *(for 7 hubs)*	
Number of Hubs	CPS: 7 Maximum: 127	
Communication with Hubs	Individual basis	
Communication	Serial at 115200 Baud	
Power Options	External 5 Volt or USB Power	
Number of Sensors on Project	8 Ultrasonic, 18 IR	Total: 26
Maximum Number of Sensors	Max 3.3V: 254, Max 5V: 508	Total: 762