An Intention Detecting Semi-Autonomous Intelligent Wheelchair System

(Summer 2012 Progress)

Nicholas Brent Burns
University of Texas at Arlington
Computer Science and Engineering Department
Research Experience for Undergraduates – Summer 2012
Arlington, TX USA
nburns@mavs.uta.edu

***Abstract*—Assistive devices are essential in improving the quality of life for the elderly and disabled. Powered wheelchairs are one of these tools and they provide the user freedom and independence. Self-reliant individuals become less of a burden on others and can prolong life through self-sustainability. When it comes to computer aided devices, individuals prefer to be assisted rather than controlled by these systems.**

***Index Terms*—Circuits, Control systems, Intelligent systems, Semi-autonomous systems.**

I. Introduction

The Computer Science and Engineering (CSE) Department at the University of Texas at Arlington (UTA) has an abundant amount of diverse research projects in progress at any given time.

One of the larger collaborations is known as the SmartCare project. It is a joint effort combining the knowledge and resources of the College of Nursing and the College of Engineering at UTA. The Research Experience for Undergraduates (REU) program, headed up by Dr. Huber, has been collaborating with the College of Nursing's Kathryn Daniel, Ph.D., RN. "SmartCare, a technology development center will develop, build, and deploy the next generation of intelligent care technologies enabling a significant nationwide improvement in in-home health care" [6].

One of the many projects currently under way is known as the Intelligent Wheelchair Project. It is a preexisting endeavor which I selected as my REU project for the summer of 2012. I was the sole student working on this project under the supervision and assistance of Drs. Huber and Záruba.

This paper details the overview of this project, past work on the wheelchair, background research, goals for the project, a detailed description of the hardware and software, current state of the project, obstacles faced, and future work needed to complete the Intelligent Wheelchair system.

II. Past Work

The idea of designing an intelligent wheelchair system for the CSE Department at UTA was first conceived by Dr. Manfred Huber, Dr. Gergely V. Záruba, and Ph.D. Candidate John H. C. Staton. These individuals started this project in 2008 from the ground up.

An abundant amount of work was previously completed when I took over the project at the beginning of the 2012 summer. Dr. Záruba originally created the first version of the circuit and the WheelchairClient software. Detailed descriptions of the previous work completed on the actual wheelchair, hardware, and software is found in the following sections *V. Hardware* and *VI. WheelchairClient Software*.

III. Background Research

To prepare for this project before my REU tenure began, I researched various topics and read many published papers to increase my knowledge of the technical areas that would be vital to successfully implementing concepts of the Intelligent Wheelchair.

The first paper I read was John Staton's Thesis *An Assistive Navigation Paradigm for Semi-Autonomous Wheelchairs Using Force Feedback and Goal Prediction*. His paper focuses on a more theoretical and simulated approach for autonomous path planning systems with an intuitive human controller through the use of a force-feedback interface [5].

Even though Staton did physically work on the wheelchair, his background and skills are in the area of algorithmic and formulaic approaches to the intelligent aspect of the wheelchair project. His work will be vital in the development of the semi-autonomous computer navigation controller and other aspects of the WheelchairClient software.

All of the above mentioned works are cited in the *References* section.

IV. ULTIMATE GOALS

While the final version of the Intelligent Wheelchair is not yet complete, below is a list of all the functionality we wish the wheelchair to one day possess.

- Have the semi-autonomous computer navigation controller efficiently and accurately avoid obstacles, navigate around corners, and reverse the wheelchair successfully into an elevator
- Integrate external sensors (laser range finders) into the system to increase the navigation controller's capabilities
- Integrate encoders to measure wheel rotation
- Design and implement a PID Controller to minimize wheelchair movement error that utilizes data from the encoders, sensors, and joystick
- Have the navigation controller compensate for "shaky" user input through the wheelchair's joystick to produce the desired effect rather than actual input
- Construct containers to hold all external hardware
- Have fluid and user friendly wheelchair movement
- Provide thorough documentation to impart to the next researcher(s) of this project

V. HARDWARE

As stated in the section *II. Past Work*, Dr. Záruba had previously designed and constructed the circuit that interfaces the wheelchair and laptop software. At his suggestion and through my desire I built an entirely new circuit that would serve the same functionality as the former. However this would replace certain components, add an encoder chip, and upgrade the microcontroller from a PIC16F876A to a PIC18F4620.

This custom circuit is enclosed in a grey enclosure case and I will refer to this circuit as the "Grey Box Circuit" or GBC for short. Dr. Záruba altered the joystick box (Action Box) on the wheelchair to allow access to certain features of the wheelchair's capabilities.

He contacted the company that constructed the wheelchair to determine the circuitry and design of their hardware in order to utilize certain connections. He spliced into the wheelchair's joystick box and made use of such features as a +5V and +15V supply, common ground, the joystick's X and Y coordinates, Speed (V), and many other features.

A. Description of the Powered Wheelchair

The wheelchair used for this project is a powered wheelchair controlled by a simple joystick and speed control knob interface. The two wheels are powered by two separate car batteries. The output of the joystick box is fed into a "Black Box" below the seat of the wheelchair. The output from the Black Box feeds into two motors that drive the two wheels. Little is known about the capabilities of the circuitry and software inside. Dr. Záruba informed me that he made no alterations to the contents of the Black Box since it contains a Neural Network chip that is most likely used to stop the wheelchair from moving when positioned on an incline and the joystick is at rest.

Fig. 1. Picture of the Powered Wheelchair

Dr. Záruba and Dr. Huber placed two plastic gears on the inside of the wheelchair's wheels for the encoder's to utilize. Any and all changes we made to the circuitry of the wheelchair takes place before outputs from the joystick box reach the Black Box. This limitation must be taken into consideration for all experiments and modifications we perform on this wheelchair system.

The wheelchair will have two encoders placed near the gears of the wheels and two laser range finder sensors placed on the front and rear of the wheelchair. None of these have been physically implemented on the wheelchair as of today, but the locations and approaches for mounting them have already been conceived. The encoders will be contained in a spring controlled contraption to allow the encoders to be lifted in case any debris is lodged between the encoder's gears and wheel's gears. This must be done because the encoder gear interface is not sealed in an enclosure. The laser range finders will be simply mounted on the front and back with an offset of about 1 foot from the center of the wheelchair to the right.

B. Major Components Used

- PIC18F4620 – The microcontroller which controls the functionality of the circuit
- HCTL-2032 – The Encoder Chip that interfaces the two encoders with the PIC microcontroller
- FT232R USB UART – USB-to-Serial chip that connects the laptop to the GBC
- AD5204 – Digital Potentiometer to send precise values to the wheelchair
- FQP13N10L MOSFET – 100V LOGIC N-Channel MOSFET to aid in switching between hardware and software control of the wheelchair
- MES20 Encoders – Two incremental encoders with a resolution of 1024 pulses per revolution used to measure wheel rotation as seen in "Fig. 4"
- Laser Range Finder Sensors – Used for obstacle and door detection (specs unknown)
- 400SD4 Gateway Notebook – Laptop used to run WheelchairClient software and communicate with wheelchair via the GBC

C. Description of the Grey Box Circuit (GBC)

The overall purpose of the GBC is to facilitate communication between the wheelchair and the WheelchairClient software running on the laptop. It allows the wheelchair operator to switch between joystick control and software control of the motors driving the two wheels. The ultimate desire is to use the GBC's capabilities to seamlessly switch between user (joystick) control and software control in order to fulfill the semi-autonomous functionality of the overall system.

Again, the prime objective of this system is not to control and override the wheelchair user but rather to assist the individual in any appropriate situation where the aid of an intelligent system would be beneficial.

I first designed the GBC using *ExpressSCH*, a schematic design program. I based most of my circuit design off of the previous design built by Dr. Záruba.

1) Physical Connections and Descriptions

To power the components on the GBC I used connections from the wheelchair's circuitry via a DB15 Serial interface that Dr. Záruba built.

The only programmable component on the GBC is the PIC18F4620 Microcontroller. Therefore this is the chip that will manage the circuit and coordinate certain operations for desired effects. The PIC has output connections to the Digital Potentiometer, the Encoder Chip, the USB-to-Serial Chip, and the MOSFET.

The MOSFET is connected to a series of two relays that control which signal (X, Y, and Speed (V)) is fed into the wheelchair's Black Box. This is the switching mechanism that drives the wheelchair through actual joystick input or software generated signals from the Digital Potentiometer. This switching of control can be generated by either of the two physical push buttons, one on the joystick box and the other on the Grey Box.

The PIC and Encoder Chip have basic Digital I/O connections to send appropriate commands in order to receive the count from each encoder.

The PIC and USB-to-Serial Chip have bidirectional communication in order for the PIC to communicate with the laptop. The USB-to-Serial Chip is connected to the laptop via a USB cable with USB Mini-B and USB-A connectors.

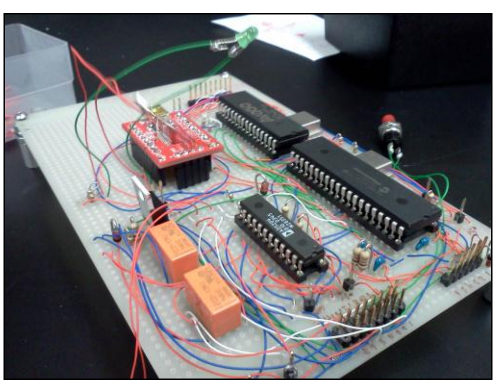

Fig. 2. Picture of the Completed GBC outside of the Enclosure Case

2) The Microcontroller's (PIC) Code Description

The majority of the microcontroller's code was written originally by Dr. Záruba. His first version of the circuit used the PIC16F876A microcontroller. The new GBC uses the PIC18F4620, therefore some of his code and algorithms had to be altered to handle the different specifications of the new PIC.

The overall structure he designed remains the same with some additions I made in order to handle such things as a new external clock with a different frequency, new timer calculations, encoder handling, different design for switching control, new USB-to-Serial Chip, and various other features.

After the PIC has been configured correctly, control is given to the "main routine" which constantly checks if a command has been received from the WheelchairClient software via the USB-to-Serial Chip. If a command was sent from the laptop's software, the Hamming-Encoded command is decoded and the appropriate command is executed. This could be such things as "Get X", "Set Speed", "Get Encoder Reading", etc.

While the main routine is executing, a pre-configured timer is generating an interrupt 15 times a second (15Hz). When this interrupt is generated, the PIC converts the analog signal from the wheelchair's X, Y, and Speed (V) connections into digital values that the PIC can utilize after the conversion is complete.

VI. Wheelchair Client Software

The software and computer navigation controller, that in essence makes the wheelchair "intelligent", is known as the WheelchairClient software. All of this software is written in C++ using Microsoft's Visual Studio 2010 and runs entirely on the UTA Gateway laptop connected to the wheelchair.

A. Overview

The WheelchairClient software will contain all intelligent functionality such as the PID Controller, the semi-autonomous computer navigation controller, sensor and encoder data interpretation, and random testing of software controlled wheelchair functionality.

As stated previously, Dr. Záruba was the first to create the WheelchairClient software. His version that was given to me at the beginning of the summer provided the following functionality at the time.

- Capable of communicating and interpreting data from the wheelchair and Grey Box Circuit
- Capable of retrieving the X and Y coordinates of the joystick and the speed from the control knob
- Capable of sending X, Y, and Speed (V) commands to the wheelchair to "override" the actual joystick's input
- Capable of making the wheelchair follow the software's commands or the hardware's (joystick) commands continuously

B. PID Controller

Even though a PID Controller has yet to be implemented into this system, I will define what a PID Controller is and outline the functionality it will theoretically provide for this system in the future.

1) Definition

PID controller stands for proportional–integral–derivative controller. Basically it is a feedback loop that calculates the amount of error between desired output and actual output. The system attempts to minimize error by adjusting specific control inputs. Essentially it gets your output where you want it.

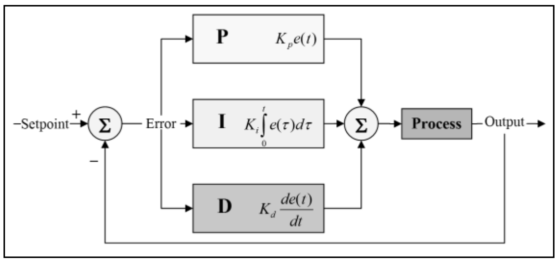

Fig. 3. Flowchart Representation of a PID Controller System

2) Application in this System (Encoders)

One of the uses the PID Controller will serve for the Intelligent Wheelchair Project is to compensate for "shaky" user input through the joystick. If the person driving the wheelchair has tremors or some other condition which causes their joystick hand to shake, the WheelchairClient software will need to recognize this condition and then enact the PID Controller to adjust and alter certain conditions so the input fed into the motors reflects what the user intended to do instead of what they actually gave to the joystick.

This form of intention detecting and error correcting could also be applied to the following situation; the user is attempting to drive perfectly straight (forward) but is slightly veering off to the right or left. The software could make a probabilistic assumption that the user is intending on moving directly north instead of the actual path travelled due to the joystick input. The PID Controller then could make small adjustments by driving or slowing down the appropriate wheel.

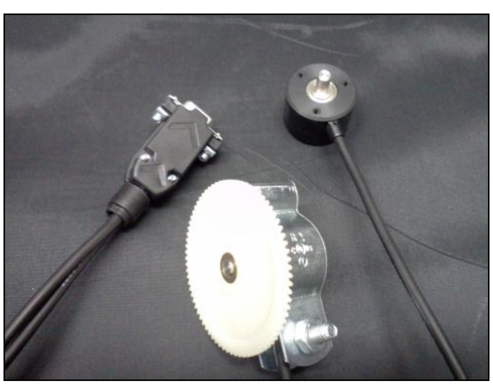

Fig. 4. Picture of the Encoders (unattached)

3) Application in this System (Encoders and Sensors)

As stated above, the laser range finder sensors are crucial for obstacle avoidance and door detection. Each sensor has nearly a 360 degree line of sight. Since there is one sensor on the front and rear of the wheelchair, the sensors have nearly a 100% view of its surroundings.

Another use of the PID Controller would be in obstacle avoidance and door detection. The external sensors (laser range finders) would be responsible for identifying an obstacle or door frame.

For a positive identification of an obstacle (another person, chair, etc.) the WheelchairClient software would assist the user in maneuvering safely around or coming to a gradual halt by adjusting inputs to the motors. If the user seems to be making the same adjustments the software is, the software would stop overriding inputs and give complete control back to the user as long as they continue this trend.

The PID Controller would be used in conjunction with the laser range finder sensors to adjust motor inputs so the wheelchair stays close to the projected safe path. The difference between the projected safe path and the actual path of the wheelchair would be the error the PID Controller would be attempting to minimize.

This same approach could be used when a door frame is detected and the WheelchairClient software assumes the user is actually attempting to navigate through that door.

C. Future Functionality of the Software

As previously stated the PID Controller, sensor handling, and intention detecting algorithms have not yet been implemented into this system. Although the graphical user interface (GUI) for the software already has many useful features, below is a list of possible additions that the GUI could also allow.

- Graphical representation of the current position of the joystick
- Visual manipulation of joystick and speed control
- Visual illustration of the wheelchair's path travelled
- Visual illustration of the wheelchair's predicted path
- Display for the sensor data and possibly a visual representation of obstacles or door frames
- Debug functionality to assist in constructing the system

VII. EXPERIMENTS PERFORMED

Over the summer I performed experiments to measure the range of values for the joystick and speed control knob. These values are shown in "Table 1". Within the joystick box the X, Y, and Speed (V) values are determined by using potentiometers connected between GND and 5V. Using the analog-to-digital converter on the PIC microcontroller, these raw values being sent to the wheelchair's Black Box are also being sent to the WheelchairClient software. Enabling the software's *Soft Follow* feature I am able to continuously receive the X, Y, and Speed (V) values every 300ms while positioning the joystick and speed control knob in various locations.

TABLE I. JOYSTICK AND SPEED VALUES

	Min.	At Rest in Center	Max
X	176	508 – 510	511
Y	160	504 – 508	511
Speed (V)	0	N/A*	511

*Position of Joystick has no effect on Speed

These results are not as ideal and consistent as I hoped they would be. However a lot can be done using these values. I have written preliminary WheelchairClient code that deduces the current quadrant of the joystick by comparing the values of X and Y.

More experiments must be performed once the encoders are mounted onto the wheelchair to measure the amount of wheel rotation. These experiments will consist of recording a large amount of readings and comparing X, Y, and Speed (V) values with encoder data.

Such experiments will include:

- Driving the wheelchair North, South, and turning “on a dime” to maneuver East and West
- 90/180/270/360 degree rotations
- Driving around a corner
- Coming to a gradual halt (obstacle simulation)
- Averting around an object (obstacle simulation)
- Distance travelled measurements
- Effect of battery life on wheelchair operations
- Effect of weight of wheelchair user on wheelchair operations
- Comparing different speeds
- Response time of joystick/software input against actuators driving the wheels

VIII. OBSTACLES – SUMMER 2012

I encountered a number of problems over the course of my REU work on the Intelligent Wheelchair Project. Most of the complications were numerous hardware and circuit problems. Constructing the Grey Box Circuit took roughly 5 weeks to complete since I had no extensive prior experience building a circuit of that magnitude. Since the REU program was only 10 weeks long, this consumed too much of my time.

A. Grey Box Circuit Problems

During the construction of the GBC I faced many problems with the circuitry not behaving correctly or circuit components failing or burning out. The first MOSFET I placed in the circuit drew too much current and burned out. Some loose connections and faulty wiring caused some nuisances and many days of debugging. Even though I learned a lot by discovering and fixing these problems, I would have preferred to spend more time altering the WheelchairClient software to accommodate for the PID Controller and other intelligent functionality.

B. Encoder Chip Problem

The two encoders seemed to be working correctly and they are sending signals to the encoder chip. However, the encoder chip is not sending the correct count to the PIC. I followed the code structure outlined in the encoder chip’s datasheet to set up the encoder chip from the PIC but it only sent a value of zero as the count. I spent the last two weeks of the REU program attempting to debug this dilemma but no success has been made.

C. Miscellaneous Hardware Problems

One problem that took a while to pinpoint was the joystick not resetting to a zero input for the X and Y coordinates. I eventually discovered that two resisters in the joystick box had burned out. The resister shunt’s purpose was to “lie” to the Black Box by using a voltage divider to bring down 5V to 2.5V so the Black Box would think the joystick was at rest in the center. Once I replaced all the resistors the problem was fixed.

IX. FUTURE WORK

Although my REU tenure for the Summer of 2012 has come to an end, I do plan to continue working on the Intelligent Wheelchair Project throughout the Fall 2012 semester. I hope to make significant progress and have a working Beta version of the wheelchair by the winter break. I will continue to seek advice and aid from professors, graduate students, and fellow undergraduate students.

A. Immediate Work to be Done

As stated in *VIII. OBSTACLES – SUMMER 2012*, the current problem halting progress on the project is the Encoder Chip problem. I am currently trying to fix this problem so the project can move forward and intelligent software can be designed. I am getting assistance from Drs. Huber and Záruba to quickly correct the false count value being sent to the PIC.

I have already utilized an oscilloscope and multimeter in attempts to debug the problem but no peculiar behavior is shown. I will attempt to change the order and introduce delays into the PIC’s Encoder setup algorithm and hopefully this will shed some light on the matter.

After the Encoder Chip problem is solved, the encoders can then be mounted onto the wheelchair. I am currently designing the final contraption for the encoders while attempting to debug the encoder chip problem. I will also begin performing the wheelchair experiments I described in *VII. EXPERIMENTS PERFORMED*.

B. Impending Work to be Done

Once the previously stated problems are no longer causing havoc, I may then start testing the accuracy and range of the laser range finder sensors. I will first teach myself how to interface the sensors with the laptop directly and then I will run experiments and mount them to the front and rear of the wheelchair.

After the encoders and sensors are working correctly and their data is being interpreted accurately, I may then begin utilizing this data for the design of the PID Controller and other intelligent functionality.

ACKNOWLEDGMENT

I would like to thank Dr. M. Huber, Dr. G. Záruba, J. Staton, all of the graduate students in the LEARN Lab, and all of my fellow REU Students for their assistance. This summer's REU program was the best CSE experience I have had in my college career. I gained an abundant amount of knowledge on what to do and what not to do.

Again, I deeply thank Dr. Huber and Dr. Záruba for assisting me throughout the summer and teaching me unconventional ways to approach and design complex systems.

REFERENCES

[1] eCircuit Center Author. *PID Controller* [Online]. Available: http://www.ecircuitcenter.com/circuits/pid1/pid1.htm

[2] H. A. Yanco and J. Gips, "Preliminary investigation of a semi-autonomous robotic wheelchair directed through electrodes," Artificial Intelligence Lab., Mass. Inst. Of Tech., Cambridge, MA, Comp. Science Dept., Boston College, Chestnut Hill, MA, June 1997.

[3] H. Uchiyama, W. D. Potter, M. A. Covington, J. Tarver and R. Eunice, "Perceptual navigation for semi-autonomous wheelchair operations," Univ. of Georgia, Athens, GA.

[4] H. Uchiyama *et al.*, "A semi-autonomous wheelchair with helpstar," Univ. of Georgia, Athens, GA.

[5] J. H. C. Staton, "An assistive navigation paradigm for semi-autonomous wheelchairs using force feedback and goal prediction," M.S. thesis, Dept. Computer Science, Univ. of Texas at Arlington, Arlington, TX, 2008.

[6] K. Daniel, *SmartCare* [Online]. Available: http://learnlab.uta.edu/smartcare/

[7] R. Chen, L. Chen, and L. Chen, "System design consideration for digital wheelchair controller," IEEE Trans. on Ind. Elec., Aug. 2000.

[8] T. Auyeung, "Robot programming in "C"," Feb. 2006.

[9] T. Bourke, "Development of a robotic wheelchair," Nov. 2001.

[10] V. Sharma, R. C. Simpson, E. F. LoPresti, and M. Schmeler, "Clinical evaluation of semiautonomous smart wheelchair architecture (drive-safe system) with visually impaired individuals," 2001.