I - Wheel - A Robotic Wheelchair with Mobile Controlled Navigation System

Samarjeet PrithvirajChavan[1], Vivek P Deodeshmukh[2],
Electronics and Telecommunication Department, D. J. Sanghvi College Of Engineering , Mumbai University, Biomedical Department, D. J. Sanghvi College Of Engineering , Mumbai University

Abstract— *Many people those using wheelchairs are not able to control an electronic wheelchair with joystick interface. The iWheel" robotic wheelchair system is a general purpose navigational assistant in environments that are accessible for the disabled (e.g., ramps and doorways of sufficient width to allow a wheelchair to pass). A reactive system does not use maps for navigation. One of the advantages of this strategy is that users can navigate in multiple locations and environments. This report describes indoor navigation in the "iWheel" system; outdoor navigation is currently under development.*

Index Terms—*Indoor Navigation, Navigation Control, Robotic Wheelchair, Wheelchair Automation.*

I. INTRODUCTION

The aim of the "iWheel" project is the development of an electronic powered wheelchair system that provides assistance in navigation in indoor and outdoor environments, which allows its user to drive more easily. An electronic powered wheelchair is usually a semi-autonomous system. Full solution to Artificial Intelligence problems do not need to be found before a useful system can be built. A robotic wheelchair can take advantage of the intelligence of the user by asking for help when the system has difficulty in navigation.

There are two basic requirements for any robotic wheelchair system. First and most important, a robotic wheelchair must be able to navigate safely. Any errors must at least be able to prevent harm from coming to the occupant. Second, the system must interact effectively with the user. Other features may include outdoor as well as indoor navigation, automatic mode selection based upon the current environment, and capable of easily adapting to various user interfaces.

II. SYSTEM ARCHITECHTURE

As shown in Fig. 1, the Bluetooth module is connected to an android device via a Bluetooth link. The transmitter of Bluetooth module is connected to receiver of arduino and receiver of arduino is connected to transmitter of Bluetooth module. The advantage of this connection is that, both arduino and Bluetooth module become inter-controllable. Any signal henceforth received at Bluetooth module is transmitted to arduino through module's transmitter and any signal from arduino is received by module at its receiver.

Arduino is connected with Electronic Speed Controller's (ESC's) via three pin's viz, PWM (3; 6), DIRECTION (7; 4) and BRAKE (8; 5). Arduino is connected to both ESC's, thus making itself easy to control ESC's and thus the motor's. *PWM* pins decide state of the motor's (.i.e. ON/OFF). *DIRECTION* decides whether the motor have to rotate in front or reverse direction. *BRAKE* pins when ON stop the rotation and do not allow movement in any direction.

Both motors are connected to their independent ESC's, thus reducing the complexity of the system and making it easy for arduino to control them, which itself is controlled by Bluetooth module.

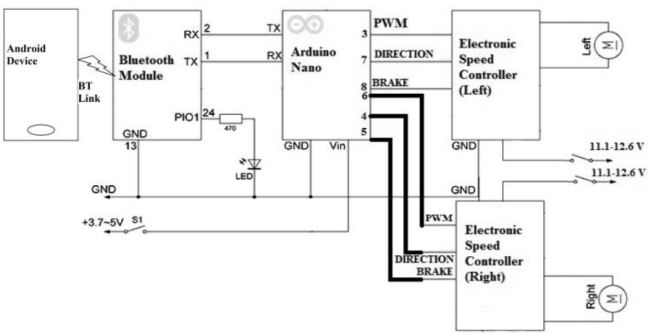

Fig. 1: Hardware block Diagram of the System.

Above shown is the block diagram of the system. Here, as shown above, an Android device, a Bluetooth module, Arduino Nano, two Electronic speed controllers and corresponding motor's form the hardware of the system.

III. MAJORE SYSTEM COMPONENTS

A. *ARDUINO NANO*: The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (*Arduino Nano 3.x*). It works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech. The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TXLEDs will flicker if digital pins 0 or 1 are high. The ATmega328has 32 KB, (also with 2 KB used for the boot loader). The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM. Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), anddigitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions. The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

B. *LR7843:* It finds application in, High Frequency Synchronous Buck Converters for Computer Processor Power and High Frequency Isolated DC-DC Converters with Synchronous Rectification for Telecom and Industrial Use. It has Very Low RDS (*on*) at 4.5V VGS, with Ultra-Low Gate Impedance and Fully Characterized Avalanche Voltage and Current.

C. *BLUETOOTH MODULE*: Bluetooth Wireless module provides a simple interface for connecting to Arduino Nano. The module provides a method to connect wirelessly with a PC or Bluetooth enabled device to transmit/receive embedded data such as GPS data, ADC voltage reading and other parameters. It has following specifications and feature's:

- JY-MCU Bluetooth Wireless Serial Port Module with free extension cable designed for easy use with Arduino® boards and Firewing® boards.
- Supply voltage can be between 3.6 to 6V DC. IOs are 5V tolerant.
- Bluetooth V2.0+EDR (*Enhanced Data Rate*) 3Mbps.
- Bluetooth SPP (Serial Port Protocol).
- Easy to connect this module with any standard Bluetooth device, just search and key "1234" pass code.
- Baud rate: 38400 bps.
- Module requires no setup.
- Dimensions: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm).

Table. 1:Pinout of a Bluetooth Module

Pin No.	*Signal Description*
1	Key (No pin)
2	VCC +3.6v to +6v DC
3	GND- Ground Connection
4	TXD -Tx from module
5	RXD- Rx for the module

- LED indicate show connection status, when it is flashing means non-connection, normally on mean it connect successfully.
- Backplane set anti-reverse diode with 3.3V LDO, input voltage: 3.6~6V,unpaired current: about 30 MA, Paired: about 10 MA, input voltage can't exceed 7V.
- Interface level 3.3V, can be directly connected the various SCM (*51, AVR, PIC, ARM, MSP430, etc.*), the microcontroller can also be directly connected without MAX232.
- Effective distance: 10 meter's.
- After repaired, can used as full-duplex serial, supports 8 data bits, 1 stop bit, no parity communication format.
- Easy to connect this module with any standard Bluetooth device, just search and key "1234" pass code.
- Running in slave role: Pair with BT dongle and master module, simply establish connection between device with Bluetooth function GPS, PC, PDA, PSP phone, Industrial control, etc.

D. *MOTOR'S:* Serpent industrial encoded motors with carbon brushes are used to drive the chair. Encoders are used to determine the position, velocity and direction of a motor shaft or other mechanical motion. They provide information required for the precise control of a variety of applications, such as positioning a rotary table, pick and place, machine assembly, packaging, robotics and more. Regardless of type, all encoders provide a method of orientation detection that's used as a reference point for position control. An encoder is a special sensor that captures position information and relays that data to other devices. The position information can be determined using one of three technologies: optical, magnetic or capacitive. Optical encoders are the most accurate of the standard styles of encoders. When choosing an optical encoder, it's important that the encoder has extra protections built in to prevent contamination from dust, vibration and other conditions common to industrial environments.

Table. 2: Technical Specifications of Motor's

Shaft Diameter	8mm
Mounting Panel	40 mm X 4 Holes
Motor Diameter	65mm
Motor Length	170mm
Motor Weight	1200 grams
Rated Torque	128 Kgcm
Stall Torque	348 Kgcm
No Load Speed	92 rpm
No-Load Current	600 ma

We have used motors with optical pickup incremental encoders. An incremental encoder only reads pulses to provide information about the relative motion of the shaft. It has no information about location when powered up; it can only show how far the shaft has moved since the encoder was powered up. It reports back these position changes with *electrical pulses.* These pulse streams can either be single channel *(one output wire from the encoder)* or dual channel *(two wires)*. Think of an incremental encoder as a tape measure with no numbers on it, only tick marks: you can tell how far you've moved, but you don't know exactly where you are unless you measure from a known spot.

E. *BATTERY:*

Table. 3: Technical Specification of Battery

Capacity	4200mah		
Voltage	11.1v		
Continuous Discharge Rate	20C		
Burst	30C		
Charger Time	Recommend (by input 1A)		
Size	L: 13.5 cms	B: 4.2 cms	H: 2.1 cms
Weight	334 gms		

F. *BATTERY CHARGER:* Corresponding lithium polymer battery charger is used. This charger has following features,

- AC 100~240v or 12V DC input
- Microprocessor controlled
- Delta-peak sensitivity
- Individual cell balancing
- Li-ion, LiPo and LiFe capable
- Ni-Cd and NiMH capable
- Large range of charge currents
- Store function, allows safe storage current
- Input voltage monitoring.
- Battery break in and cycling

Table. 4: Technical Specifications of Battery Charger

Input Voltage	11~18v
Circuit power	Max Charge: 90W / Max Discharge: 10W
Charge Current Range	0.1~10.0A
Discharge current range	0 .1~2.0A
Discharge current range (LiPo, LiFe/Li-ion)	2.0- 4.2v/cell
Ni-MH/NiCd cells	1~16
Li-ion/Poly cells	1~6
Pb battery voltage	2~20v
Weight	520g
Dimensions	146x148x58mm

IV. NAVIGATION SYSTEM FOR INDOOR ENVIRONMENT

The direction of mobile robotics research is towards the development of autonomous navigational systems. An electronic powered wheelchair must be capable to interact with its occupant, making the robotic system partially-autonomous rather than fully-autonomous. A fully-autonomous mobile robot is often only given its destination point and a directional map. An electronically powered wheelchair should not subscribe to this method. The occupant may decide to change directions during journey. The wheelchair must be able to accept input from its occupant throughout the journey. The wheelchair should have the ability to take on a autonomous role if the user desires, but the robot and the occupant will have to work together.

For the occupant to successfully and efficiently navigate in various environments, the system uses reactive navigation .i.e. the system navigates according to the occupant's will. Because there is an intelligent occupant giving high-level navigation commands to the electronic wheelchair, the common limitation of a reactive navigation system (lack of planning) is eased down. The system tries hard to carry out the occupant's commands while keeping him safe, leaving the planning that a reactive system typically omits to the occupant. If the occupant wants the system to be more automatic, maps or direction log of commonly traveled paths such as the home and the office could be saved in the android program. Path planning for indoor robotics has been studied [2] and could be implemented on iWheel.

Wheelchair controls are divided into two levels of control. One that involves avoiding obstacles and keeping the chair centered in a hallway is called Low-level control. Giving the directions to the wheelchair to take it to a desired location is comprised in High-level control. For a electronic powered wheelchair occupant who has good control of the joystick, these two types of control can be easily managed at the same time. The occupant can avoid obstacles on the path by moving the joystick to make the proper adjustment. This is analogous to driving a car; people make many small adjustments to their route to keep the car in the proper lane and to avoid obstacles like potholes.

When a power wheelchair occupant does not have perfect control of a joystick or has no control of a joystick at all, low-level control does not easily blend into high-level control. It is not possible to make small adjustments easily. For a occupant driving using an alternative access method, low-level control adjustments require as much effort as high-level directional commands. A robotic wheelchair can assist a occupant in this group by taking over low-level control, requiring the occupant to use the access method only to give high-level directional commands like "right" or "left"[1].

V. GRAPHICAL USER INTERFACE

A robotic wheelchair system is expected to be more than just a navigation system. It is expected from the system to keep its occupant safe from harm and help or guide in navigation. The "iWheel" system solves the adaptation problem through the addition of a general user interface that can be customized easily for each occupant[1].

The graphical user interface is built on Android and Arduino and can be easily customized for various access methods. To date, the interface has been made for single access method, which is traditional navigation system. Here, four buttons are used to direct the wheelchair. The navigation command portion of the interface used in interface control consists of four directional buttons. The user controls the standard speed of the robot by clicking on the check box on the upper part of the screen. The robot may move at a slower pace according to the occupant's desire when the current task requires a slower speed to be carried out safely. The actual speed of the robot is displayed by the robot when check box is unchecked.

In the "iWheel" system, the occupant gives the high-level commands ("forward," "left," "right," "back") through the graphical user interface. The system carries out the occupant's command using common sense constraints such as obstacle avoidance. The robots low-level control acts to keep the wheelchair and its occupant safe. For example, if the occupant instructs the chair to go forward, the robot will carry out the command by taking over control until another command is issued. While executing the high-level "forward" command, the chair will prevent the occupant from running into walls or other obstacles. If the chair is completely blocked in front, it will stop and wait for another command from the occupant. If it is drifting to the right, it will correct itself and move to the left. This navigation method allows people who have trouble with fine motor control but who have the ability to issue high-level commands to control a powered wheelchair.

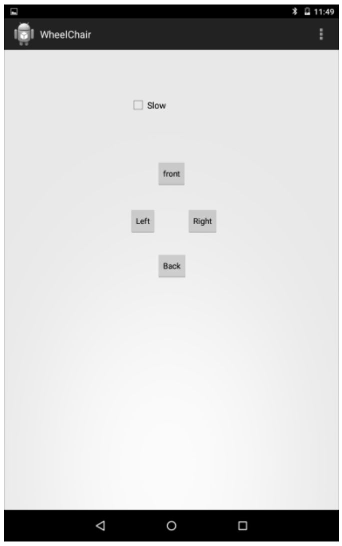

Fig. 2: GUI of Android Program.

A. ACCESS METHODS: Access methods are devices used by occupants to drive wheelchairs or control computers. Many different access methods for powered wheelchairs are used. Joystick is the most commonly known and used access method. If a occupant has sufficient control with a joystick, no additional assistance is necessary. Then, these occupants would not need robotic wheelchair's since they are able to drive without the system. If a person does has a fine control, joystick movement can be limited through the addition of a plate which restricts the joystick to limited/basic directions. Occupants in this group might be helped by a electronic system. If they push the joystick forward, the fine control could be taken over by the robotic system. If a occupant is unable to use a joystick, there are other access devices which can be employed. A button or group of switches can be used to control the wheelchair. If a occupant has the ability to use multiple switches, different buttons can be linked to each navigation command. The multiple buttons can be on the wheelchair tray, mounted around the occupant's head or placed anywhere that the occupant will be able to reliably hit them. Another access method for wheelchairs is a sip and puff system. With this method, the user controls the wheelchair with blowing or sucking on a tube. If the occupant can control the air well enough, soft and hard sips or puffs can be linked to control commands. This is analogous to the multiple button system above.

If the user has only one switch site, the wheelchair must be controlled using single switch scanning. In this mode, a panel of lights scans through four directional commands (forward, left, right and back). The occupant clicks the switch when the desired command is lit. If the occupant is traveling forward and drifts left, he must stop, turn the chair to the right and then select forward again. This mode of driving is very slow and difficult; it is the method of last resort. Obviously, a electronic wheelchair system could help this group of occupants. Most research on robotic wheelchairs has not focused on the issue of access methods. Most of the current systems are driven using a joystick (e.g., [3], [4], and [5]). A few researchers have used voice control for driving a robotic wheelchair (e.g., [6]). Voice control can be problematic because a failure to recognize a voice command could cause the occupant to be unable to travel safely. Additionally, some members of our target community are non-verbal.

VI. WORKING OF THE SYSTEM

As stated above, Android software is connected to Bluetooth Module via Bluetooth link. Bluetooth module which is connected to arduino receives character data from android application and transmits those characters to arduino. Arduino which in turn is directly connected to ESC's controls flow of PWM, its DIRECTION and BRAKING.

Working of the system can be explained as follows,

i. Occupant connects his android device with Bluetooth module.

ii. Occupant then selects speed mode.

iii. Occupant presses a navigational direction key (front, Back, Left, Right) on screen. Each key on screen is related to a character value (F/f, B/b, L/l, and R/r). When the key is pressed by the user, corresponding character value is generated by the program and is transmitted to Bluetooth module. A set of characters (F, B, L, R) is assigned for higher speed and another set of characters (f, b, l, r) for lower speed operation. Thus, a single character value is generated for any single key press.

iv. Bluetooth module receives this single character value and passes it to arduino nano.

v. Arduino nano receives this single character value and decides which motor to run in forward direction or in reverse direction.

vi. Suppose arduino receives 'F/f', both ESC's thus both motor's receive PWM's, high DIRECTION and low BRAKE's.

vii. If arduino receives 'B/b', both ESC's thus both motor's receive PWM's, low DIRECTION and low BRAKE's.

viii. If arduino receives 'L/l', both ESC's thus both motor's receive PWM's, left low and right high DIRECTION and low BRAKE's.

ix. If arduino receives 'R/r', both ESC's thus both motor's receive PWM's, right low and left high DIRECTION and low BRAKE's.

Following is the state table followed by the system. Here, states of the system vary according to the button pressed (front, Back, Left, and Right) on the android application."F/f, B/b, L/l, R/r" are the sets of characters sent by Bluetooth module or received by arduino nano. Arduino nano can generate and transmit PWM to Left or Right (L/R) ESC at frequency between 0-255(256). Direction of the motor can be in forward or reverse direction indicated by 1 or 0 in the state diagram. Braking is indicated as '1' in following state diagram. '0' indicates free wheel.

Table. 5: State diagram of the system.

BUTTON PRESSED		FRONT	BACK	LEFT	RIGHT
CHARACTER		F/f	B/b	L/l	R/r
PWM	L	200	200	200	200
PWM	R	200	200	200	200
DIRECTION	L	1	0	0	1
DIRECTION	R	1	0	1	0
BRAKE	L	0	0	0	0
BRAKE	R	0	0	0	0

SUMMARY

This research project is aimed towards a developed usable, low-cost assistive robotic wheelchair system for disabled people. In the initial work towards this goal, an indoor navigation system and a graphical user interface have been developed. The robotic wheelchair must work with the occupant to accomplish the occupant's goals, accepting input as the task progresses, while preventing damage to the occupant and the robot.

REFERENCES

[1] H.A. Yanco, A. Hazel, A. Peacock, S. Smith, S., and H. Wintermute, "*Initial Report onWheelesley: A Robotic Wheelchair System*", In Proceedings of the Workshop on DevelopingAI Applications for the Disabled, held at the International Joint Conference on Artificial Intelligence, Montreal, Canada, August 1995.D. Kortenkamp, R.P. Bonasso, and R. Murphy, "*Artificial intelligence and mobile robots: case studies of successful robot systems*", MIT/AAAI Press, 1998

[2] M.A. Perkowski and K. Stanton, *"Robotics for the handicapped*",Northcon Conference Record, pages 278 (284, 199)

[3] H. Wakaumi, K. Nakamura and T. Matsumura,"*Development of an automated wheelchair guided by a magnetic ferrite marker lane*" Journal of Rehabilitation Research and Development, vol. 27(1) : 27(34), Winter 1992

[4] R.C. Simpson, S.P. Levine, D.A. Bell, L.A. Jaros, Y. Koren and J. Borenstein,"*Navchair: an assistive wheelchair navigation system with automatic adaptation*"Assistive Technology and AI, LNAI 1458, pp. 235–255, 1998

[5] J.D. Crisman and M.E. Cleary,"*Progress on the deictically controlled wheelchair*"Assistive Technology and Artificial IntelligenceLecture Notes in Computer Science Volume 1458, 1998, pp 137-149

[6] D.P. Miller and M.G. Slack."*Design and testing of a low-cost robotic wheelchair prototype*"Autonomous Robots,vol 2:77 - 88, 1995.