Development of Smart Semi-Autonomous Wheelchair

Chiraag A[1], Tejas BU[2], Vrishab Kushalappa[3], Tavishi Bansal[4] and Prof. Asha S Patil[5]

[1,2,3,4] *Final year students, Department of Electrical & Electronics Engineering, PES University, Bangalore*

[5] *Professor & project guide, Department of Electrical & Electronics Engineering, PES University, Bangalore*

Abstract.

This research paper focuses on the development of a semi-autonomous smart wheelchair based on the line follower ideology [4,7], to assist the physically challenged, mainly in interior spaces. In this paper the different modes of operation, overall hardware and software architecture along with communication protocols have been discussed. For people having disabilities or the elderly, requirement of mobility support could restrict their independence, on account of third-party reliance. Hence, a cost effective, reliable, autonomous system has been developed to tackle this issue. The system has three main modes, the autonomous line follower mode, remote web control mode and the manual mode. In the manual mode, the user can either make use of the touchscreen GUI application or voice commands to manoeuvre the wheelchair.

1. INTRODUCTION

The US Access Board, Washington, DC, in 2015, reported that the percentage of operators using powered wheelchairs was 48% and manual chairs was 50% [1]. This abnormality could be explained by the hefty price tags the powered wheelchairs have. Hence, this project not only aims at devising a semi-autonomous wheelchair but also to keep the costs as low as possible.

For handicap individuals, a manual wheelchair using brute strength could work well to an extent. But those with crippled legs/hands will not find it feasible to operate a manual wheelchair and instead would have to rely on another person [2]. This is rather an inconvenience for both the parties and hence, a power operated wheelchair is the best solution. Adding onto this issue, disorders like gait are estimated to increase from 10 % for the population aged 60–69 years and more than 60 % for people over 80 years [3]. In most cases, the conventional joystick operation could also prove to be a difficult task for the disabled and falls out of favour. The simple solution as discussed in this paper is to develop an autonomous powered wheelchair which would drastically ease the operation and improve independence of the disabled.

This paper is organized into the following sections. Section II describes, Implementation and System configuration, Section III, IV & V describe the three modes of operation while Section VI & VII, deals with conclusion and future prospects respectively.

2. IMPLEMENTATION AND SYSTEM CONFIGURATION

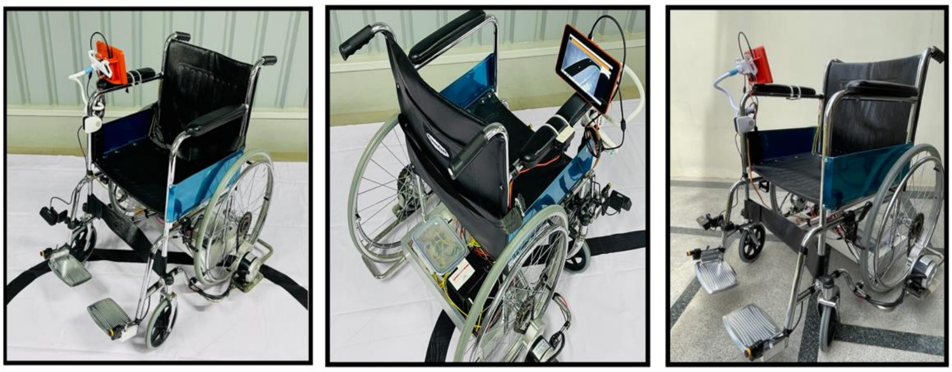

Figure 2.1. Final product: Semi-Autonomous Smart Wheelchair (left to right: wheelchair on track, top-view with active touchscreen interface, front-view)

A differential drive setup is used in converting a manual wheelchair to an electric wheelchair. Two PMDC motors are attached to the rear wheels of the wheelchair, using a chain drive setup. Custom fabrication was done on the manual wheelchair to host this system. Mild steel is used to extend the platforms on the outer sides of either wheels. A plate is provided on both the sides for the motor to be placed onto. Arc welding is used to attach the framework to the chassis of the manual wheelchair, as evident from Fig. 2.1. Motors are placed just below the sprocket attachment. This design philosophy is followed to help facilitate the forward/backward motion of the wheelchair and avoid chain slips and loss of torque or power transmission.

Fig. 2.3, shows the block diagram of the system. It mainly consists of Raspberry Pi as the heart, for computer vision [5]. Interfaced with it via I2C protocols, is the 7-inch Raspberry Pi touchscreen, which serves as the control panel and display unit for our wheelchair. A webcam with inbuilt microphone is connected to the Raspberry Pi using USB, for detecting the predefined black line, and for remote video streaming [6]. This whole setup is powered by a 10,000mah power bank. The Arduino UNO serves as the slave microcontroller and performs motor manipulation operations based on the commands received from the master, Raspberry Pi [5]. It is interfaced to the Pi via 3.3V to 5V logic level converter and makes use of UART communication protocols. The UNO is connected to a dual channel 12A Sabertooth motor controller, which is setup to be used in simplified serial mode via UART protocols.

The motor controller is connected to two 24V 250 W PMDC motors which are hooked onto the rear wheels of our wheelchair to work as a differential drive setup. To the Arduino, two optical sensors are interfaced and are mounted on either side at the front of our wheelchair to detect obstacles and come to a halt until the obstacle is removed. A buzzer is interfaced to the Arduino to go ON, when an obstacle is in the path of traversal [5]. This whole setup is powered by two 12V 12AH lead acid batteries connected in series to provide a combined energy capacity of 24AH and voltage of 24V.

MODEL	SPECS	V	NO LOAD		RATED LOAD				
			SPEED RPM	CURRENT A	TORQUE N.M.	SPEED RPM	CURRENT A	P-OUT W	EFFICIENT η
1016Z2	250W24V	24	434± 5%	≤1.8	6.65± 5%	357±5%	≤13.7	250	≥76%

Figure 2.2. 250W 24V PMDC Motor Specification

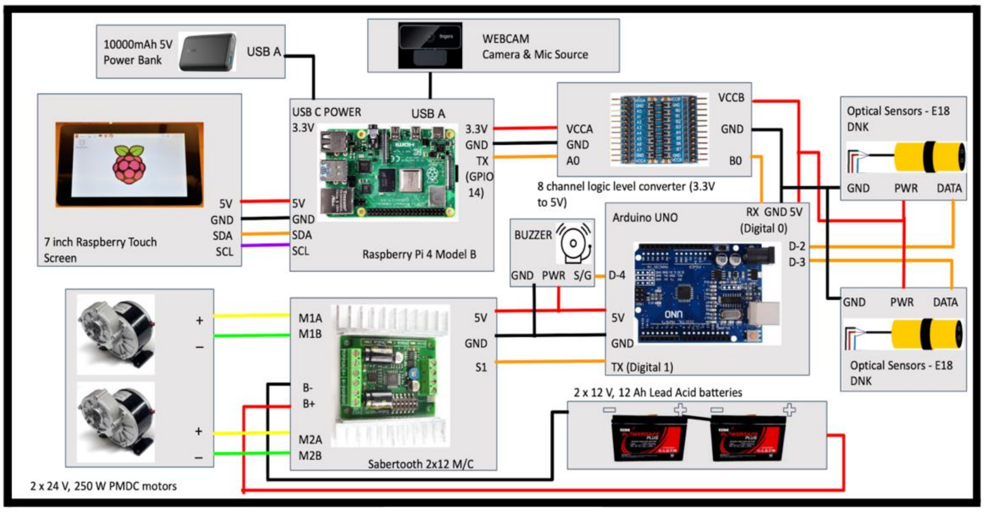

Figure 2.3. System Block Diagram

2.1. *UART Communication Protocols*

The choice of communication protocols that was selected to establish between Raspberry pi and the Arduino UNO was UART. The reason being, this kind of communication requires less no. of wires, it’s a 2-way communication and there’s no assigned master-slave components as such, as this is a multi-master way of communication. The communication protocols in serial mode of operation of the Sabertooth motor controller and the Arduino UNO were fixed to use UART protocols. The UART protocol is established in this way, Raspberry Pi to Arduino UNO to Sabertooth Motor controller in this direction only.

In this system, the assigned GPIO or digital RX and TX pins on Raspberry Pi and Arduino UNO are made use of. To configure these pins Software serial is used on Arduino. The connections between these components can be understood more clearly from the above block diagram.

2.2. System Flowchart

Fig. 2.4 shows the flowchart of the full operation and modes of the wheelchair. The system basically has three modes, autonomous mode, web-controlled mode and the manual mode.

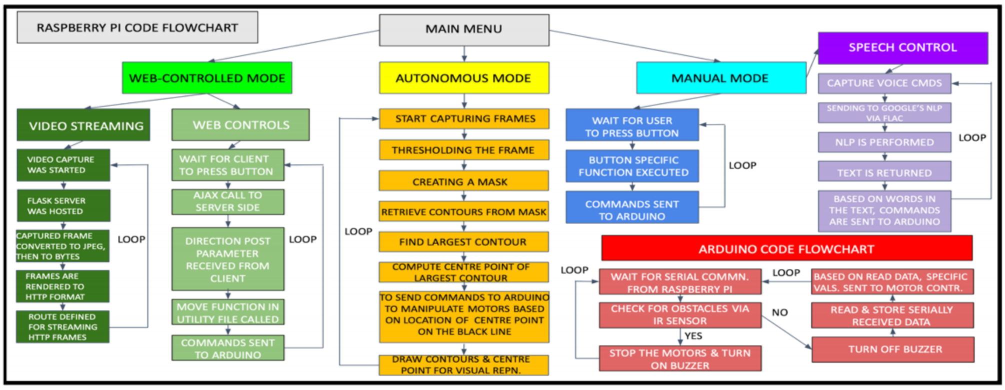

Figure 2.4. System Flowchart

3. AUTONOMOUS LINE FOLLOWER MODE

To achieve line follower capabilities, computer vision using OpenCV library was utilized. On the frames received from the webcam, thresholding and masking are applied to only detect the dark/black line or a track over a light background. The frames are converted to grayscale since we are only looking for the black path. Noise is eliminated using Gaussian blur. In the thresholding process, a pixel below 65 turns white and anything above turns black. After which contours are retrieved from the frame using the masked image. Now, to avoid errors of drawing contours around another black object, only the contour with the largest area will be selected, which is obviously going to be the track and next, the center point of this track is found and based on the position of this centre point on the original frame, wheelchair will be turned right/left or will move forward. This is the ideology behind our line follower algorithm. This system can be put to use in interior spaces, where a predefined track can be laid down precisely and a controlled environment can be created to control the errors that could occur.

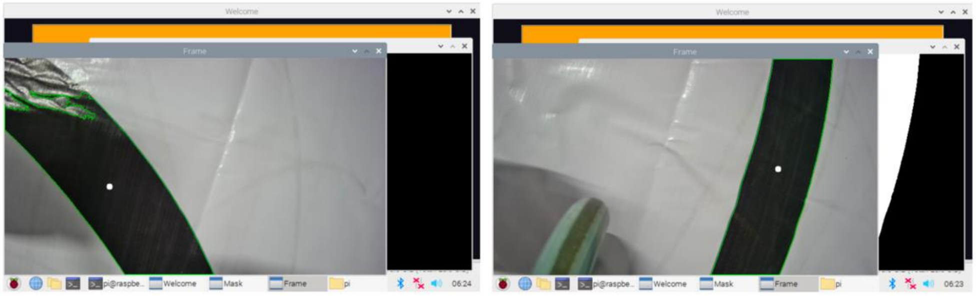

Figure 3.2. Black line contour detection (left to right: *case a and case b*)

As it can be seen from *case a* in Fig. 3.1, the center point is located on the left part of the frame, which is less than the 300-pixel range and thus, the wheelchair will be turned in the left direction, and the path to be traced is also a left curve.

As it can be seen from *case b* in Fig. 3.1, the center point is located on the right part of the frame, which exceeds the 320-pixel range and thus, the wheelchair will be aligned in the right direction, since the path to be traced is curved right.

Hence, the main goal of this algorithm is to align the system along the center point of the largest detected contour. The algorithm is constantly checking for the center point of the contour and is highly sensitive. This sensitivity can be relaxed to make the system work smoother, depending on the precision of the line follower algorithm. Altering the range of pixel values can also increase the lenience of the system. In our case, preference is given to sensitivity, precision and accuracy of the algorithm and the system constantly tries to adjust the wheelchair orientation and position.

4. WEB-CONTROLLED MODE & REMOTE VIDEO STREAMING

In this mode, the wheelchair can be controlled remotely from any internet enabled device using the IP address of the Raspberry Pi. To assist the user, real-time video streaming of 180^0 forward vision of the wheelchair is made available. This mode requires the system to be connected to internet. Currently, the system works on Wi-Fi, however, this restricts the wheelchair to be operated in interior spaces with Wi-Fi provision. Provision of cellular connection to the system would expand the range to outdoors as well.

4.1. Web Controls

The webpage for the GUI controls is developed using HTML, PHP & JavaScript. This page can be accessed using the URL, via any internet enabled device, from anywhere in the world. When a user presses on a direction button, a specific function is called and a letter specific to each button is passed as a parameter to the function. Then, this function makes an AJAX call in the background, and sends the direction value to the backend.

The interface for the web controls using apache web server was developed successfully as shown int Fig. 4.1. The wheelchair can be controlled using the GUI web page. The corresponding blackened communications were set up using AJAX calls. Hence, now making use of the Raspberry Pi's IP address, the wheelchair can be manoeuvred easily and remotely. It can be seen in the above figure, the webpage is being accessed by the Raspberry Pi's IP address i.e., 192.168.1.8. This way using these four buttons, the wheelchair, can be controlled remotely via the internet.

Figure 4.3. GUI controls on webpage

4.2. Remote Video Streaming

The programming was done on python & HTML. The motion JPEG frames are broadcasted over HTTP protocols, onto the Flask mini framework. This means, this broadcast can be visible from any web browser. This is called through context path, in which the frames are uploaded to the HTML page, from where it can be accessed anywhere.

4.3. Voice Control System

Speech command system was made possible using the speech_recognition module for python along with Google's web speech API. This method, takes the voice input from the mic, and then uploads the same to Google's cloud NLP, and then retrieves the processed words, which represent output of the input voice command as text, which we can use to our own benefit. The voice control system will make use of the simplest method of Speech-to-text API conversion i.e., synchronous recognition request. This method can process commands with maximum data of one minute. A synchronous request is sent to the Google web speech API, which basically blocks the user from giving any more voice commands, which is a safety feature to avoid unambiguous control of the wheelchair. On an average of 2 seconds, the speech to text processing is performed.

5. GUI Application And Manual Mode

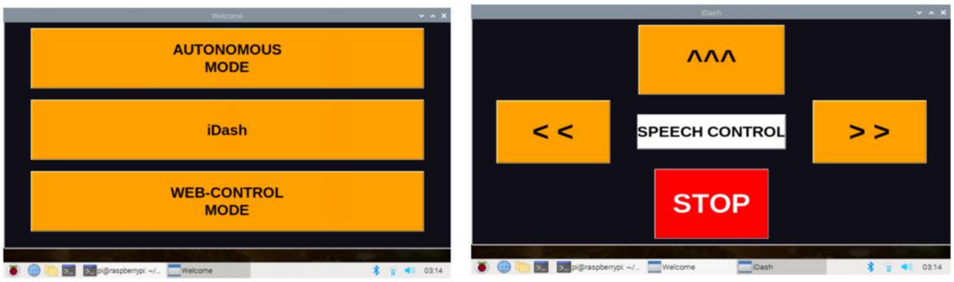

Figure 5.4. Graphic User Interface (left to right: Home Screen and Manual Mode Screen)

5.1. *GUI Application*

The graphic user application was developed in python2 making use of the Tkinter library, which basically, brings in GUI features to python programming language. A menu-based approach was adopted to develop this application. Using the Tkinter library, buttons, window and different colours were applied to the application to give it a colourful and fresh look. The user interaction with the wheelchair was made to feel easy and convenient.

5.2. *Manual Mode*

In the manual mode, the patient or the user can control the wheelchair by touch-based commands through the touch control panel. For this too, a GUI python program is created, which is executed, when the manual mode button is pressed. Similar to the controls, seen on the web control mode, even in this mode, there are four buttons to control the motion of the wheelchair, namely, forward, left, right and stop buttons.

The voice control system can also be activated, from the button here. Basically, when the control buttons are pressed, a specific byte of value corresponding to each button is sent to the Arduino UNO using UART protocols. Say, for example, when the forward button is pressed, character ‘1’, is sent as serial communications, and when right, left or stop buttons are pressed, then the Raspberry Pi, starts sending, ‘2’, ‘3’ and ‘4’ byte size values at 38400 baud rates, to the Arduino. The baud rate is chosen to be 38400, since the Arduino has to communicate with the motor controller using UART protocols itself, and we make use of standard RX & TX pins for this setup. Based on the values received, the Arduino performs the desired action, either to move, turn left or right or stop the wheelchair, by again sending motor values to the motor controller, via UART protocols, which manipulate the motors.

6. CONCLUSION

According to the methodology plan, a smart semi-autonomous wheelchair is developed, with three modes, Autonomous mode, Web-controlled mode and Manual mode. The wheelchair can autonomously commute on its own in a controller environment, making use of a predefined path. It can also be controlled manually by the user or by the internet, and as a use case, proves to be helpful for the hospital staff to control a patient remotely, in case of any mishaps. Through remote video streaming, the wheelchair can be monitored in terms of its position and orientation in an environment. The patient can also give voice commands, which is a convenient feature, to control the wheelchair. A safe speed of 4kph can be achieved and the system can support a total weight of 100kg, inclusive of the wheelchair weight.

The core functionality of serving as an automated wheelchair relies on Computer Vision. High accuracy with less processing overhead was achieved by choosing an optimal frame rate, 18 fps through trial and error. The developed codebase assures safety from bugs and software crashes. The attached infrared sensors avoid obstacles and safeguards the patient from potential collisions. Concluding, an autonomous wheelchair in a controlled environment with smart features like voice controls, remote control over the web with video streaming and touchscreen UI has been developed to cater to the needs of many.

7. FUTURE PROSPECTS

Further research on this project would focus on making the system fully autonomous. ROS using Simultaneous Localisation and Mapping Algorithms could be used to achieve this. In this, the idea is to map an environment, in different ambient conditions before hand, and with this database in hand, to help the wheelchair commute from one point to another inside this environment, avoiding obstacles. The LiDAR sensor could be used to achieve such a system, to map and help the wheelchair to navigate on its own. As of now, only obstacle detection is developed. As an extension of this section, obstacle avoidance could also be developed making use of computer vision.

8. REFERENCES

[1] Shubham Pandey, Shubham Chandewar, Krishnamoorthy A, “Smart Assisted Vehicle for Disabled/Elderly Using Raspberry Pi”, Vol. 6, No. 2, July 2017, pp. 82~87, ISSN: 2089-4864, DOI: 10.11591/ijres.v6.i2.pp82-87, 2017

[2] G. Bourhis, K. Moumen, P. Pino, S. Rohmer and A. Pruski, “Assisted navigation for a powered wheelchair. Systems Engineering in the Service of Humans”,

Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, France, pp. 553–558, 1993.

[3] W. Pirker and R. Katzenschlager, "Gait disorders in adults and the elderly, A clinical guide," Wien Klin Wochenschr., vol. 129, no. doi: 10.1007/s00508-016-1096-4, p. 81–95, 2016.

[4] A. Parsad, "Line Following Robot,"Dept. Elex. & Comm. Eng., Visvesvaraya Technological University, Banglore, India, 2005.

[5] Ayush Wattal, Ashutosh Ojha, Manoj Kumar. “Obstacle Detection for Visually Impaired Using Raspberry Pi and Ultrasonic Sensors”. National Conference on Product Design (NCPD 2016), July 2016.

[6] Matsumoto, O., Komoriya, K., Hatase, T., Yuki, T., & Goto, S. (2008). Intelligent Wheelchair Robot "TAO Aicle". Service Robot Applications. doi:10.5772/6064

[7] Bajestani, S.E.M., Vosoughinia, A., “Technical Report of Building a Line Follower Robot” International Conference on Electronics and Information Engineering (ICEIE 2010), vol 1, pp v1-1 v1-5, 2010.

[8] Development and Applications of Line Following Robot Based Health Care Management System Deepak Punetha, Neeraj Kumar, Vartika Mehta, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, Issue 8, August 2013 p2446-2450

Biographies

Chiraag A received their bachelor's degree in Electrical & Electronics Engineering and Minor Degree in Computer Science from PES University, Bangalore in 2022. He has interned at Deloitte USI as a Full Stack Web Development intern. He is currently working as a Full Stack Software Engineer at Byju’s Think & Learn Pvt. Ltd. His interests are web development, system software development including embedded systems.

Tejas BU received their bachelor's degree in Electrical & Electronics Engineering from PES University, Bangalore in 2022. He has interned at Continental Automotive as a Software Testing intern. He is currently working as a Software Development Engineer at IBM. His interests are data science, machine learning & artificial intelligence.

Vrishab Kushalappa received their bachelor's degree in Electrical & Electronics Engineering from PES University, Bangalore in 2022. He has interned at ABB and Toyota Kirloskar Motor Pvt. Ltd. He is currently working as an Associate Software Engineer at Harman Connected Services, Product Development division. His interests are embedded systems, system software development etc…

Tavishi Bansal received their bachelor's degree in Electrical & Electronics Engineering from PES University, Bangalore in 2022. She has interned at Delta X Automotive as an Embedded Engineer intern. She is currently working as an Analyst at Deloitte USI. Her interests are embedded systems, VLSI etc…

Mrs Asha. S. Patil has obtained MTech from VTU and pursuing Ph.D. at PES University. She is currently working as an Assistant Professor in PES University, Bangalore. She has published 4 papers in conference proceedings. Her areas of interest are VLSI, Microcontrollers, Data Communication, etc.