

Smart Wheelchair Navigation System with Obstacle Avoidance and Voice Control

Sushma R. Zade¹, Rohan Santosh Patil², Prasad Janardan Kasabe³,
Abhay Navnath Chavanke⁴, Prasanna Amol Kulkarni⁵

Department of Information Technology¹⁻⁵

Sandip Institute of Technology and Research Centre (SITRC), Nashik, India

sushdg@gmail.com¹, rohan003rohan@gmail.com², prasadkasbe912@gmail.com³,

abhaychavanke12@gmail.com⁴, prasannakulkarni54321@gmail.com⁵

Abstract: *Independent mobility is essential for education, work and social life, yet many elderly people and people with paralysis, spinal cord injury or muscular disorders cannot operate manual wheelchairs or joystick-driven electric wheelchairs, and commercial intelligent wheelchairs based on LiDAR, depth cameras and AI processors are unaffordable for many users. This paper presents the design of a low-cost, semi-autonomous smart wheelchair navigation system that combines hands-free voice control with automatic collision prevention. A spoken command is converted to text by an Android smartphone and sent over Bluetooth to an HC-05 module connected to an Arduino Uno. The Arduino simultaneously measures the distance ahead with an HC-SR04 ultrasonic sensor and drives two DC geared motors through an L298N motor driver; whenever an obstacle lies within a safety distance of 20-30 cm, it stops the motors regardless of the command received. The paper reviews existing wheelchair control methods, identifies the need for a minimal, low-cost design with an explicit command-priority rule and a measurable evaluation plan, and describes the architecture, control algorithm, hardware design, safety analysis and evaluation plan. The design will first be validated on a scaled toy-car platform before it is applied to a wheelchair. No experimental results are claimed; measurements will be reported after the prototype has been built and tested.*

Keywords: Smart Wheelchair, Voice Control, Obstacle Avoidance, Ultrasonic Sensor, Arduino Uno, HC-05 Bluetooth, Assistive Technology, Embedded Systems

I. INTRODUCTION

Independent movement is basic to education, employment and social participation. People who have paralysis, spinal cord injuries, muscular disorders or age-related weakness often depend on caregivers, or on wheelchairs that do not match their abilities. A manual wheelchair demands steady upper-body effort, and a joystick-driven electric wheelchair demands hand control that many of these users do not have. Assistive mobility has therefore become a practical research area in embedded systems and healthcare engineering.

Wheelchair navigation is commonly divided into three modes [1]. In manual mode the user steers with a joystick, which suits mild disability but leaves the whole burden of avoiding collisions on the user. In autonomous mode the chair follows a planned route to a destination; this relieves the user but removes control, and it can confuse elderly or cognitively impaired people. The semi-autonomous mode lies between the two: the user decides where to go, while sensors guard against hazards. The system proposed in this paper is semi-autonomous.

A smartphone is an attractive interface for such a system. Most people already own one, it contains a microphone, speech recognition and Bluetooth, and it needs little training to use [1]. Using it as the voice front end avoids building special input hardware for every user.



Many published solutions, however, rely on costly sensors, cameras or complex software, or on input methods that suit only some users. This work proposes an economical design built from an Arduino Uno, an HC-05 Bluetooth module, an HC-SR04 ultrasonic sensor, an L298N motor driver and DC geared motors.

The contribution of this paper is a minimal design that combines voice control with ultrasonic collision prevention under a fixed command-priority rule, in which a safety decision from the distance sensor always takes precedence over a voice or manual command, and that adds a safe stop on loss of the wireless link. The paper presents the design, the control algorithm, the safety analysis and the evaluation plan, and the design is intended to be validated first on a scaled toy-car platform.

II. BACKGROUND AND THEORETICAL FOUNDATION

A. Smart Wheelchairs and Control Modes

A smart wheelchair adds sensors, an embedded processor and a communication link to a powered chair so that it can interpret commands and monitor its surroundings. Input methods range from joysticks and touch screens to voice, head movement and eye movement, and the suitable method depends on the user's disability [1]. The control mode (manual, autonomous or semi-autonomous) decides how much of the driving decision is left to the user and how much to the machine.

B. Embedded Control

A microcontroller can read sensors, receive commands and drive actuators in a continuous loop. The Arduino Uno, based on the ATmega328P, is widely available, inexpensive and programmable in Embedded C with the Arduino IDE, which makes it suitable for an educational prototype. In the proposed design it repeatedly checks for a new command, measures distance, applies the safety rule and sets the motor outputs.

C. Ultrasonic Distance Measurement

The HC-SR04 emits ultrasonic pulses at 40 kHz and raises an echo signal whose duration equals the round-trip time of the reflected wave [2]. The microcontroller measures this duration t and computes the distance d as:

$$d = (v \times t) / 2 \quad (1)$$

where v is the speed of sound, about 343 m/s at 20 °C. In practice, d in centimetres is approximately t in microseconds divided by 58. The specified measuring range of the sensor is 2 cm to 400 cm. Ultrasonic sensing is inexpensive and can detect transparent, reflective and dark solid objects, but it can be disturbed by acoustic noise at similar frequencies, temperature changes and air currents [1].

D. Bluetooth Serial Communication

The HC-05 is a Bluetooth module that exchanges serial (UART) data with the microcontroller and communicates wirelessly with the smartphone over a range of roughly ten metres [3]. It turns commands sent from the phone into serial characters that the Arduino reads, and it can pass data in the other direction. Its data pins use 3.3 V logic, so the Arduino transmit line must be reduced to that level with a voltage divider.

E. Motor Driving

A microcontroller pin cannot supply the current that a motor needs. An H-bridge driver such as the L298N accepts low-current logic signals and switches a higher-current supply to the motors, and it can reverse the polarity to change direction [4]. The L298N contains two H-bridges, so it can drive two motors independently, which lets a differential-drive vehicle move forward, move backward, turn and stop. Geared DC motors give high torque at low speed, which suits a slow and stable assisted-mobility platform [5].

F. Command Priority Rule

Let c be the command received (F, B, L, R or S for forward, backward, left, right or stop), d the measured front distance and d_{th} the safety threshold (20-30 cm in the proposed design). The command applied to the motors, c' , is:

$$c' = S \text{ if } (c = F \text{ and } 0 < d \leq d_{th}); \text{ otherwise } c' = c \quad (2)$$

This rule means that an obstacle-based stop overrides a forward command, while backward, turning and stop commands stay available so that the user can move away from the obstacle.



III. RELATED WORK

A. Brain-Computer Interface Wheelchairs

Brain-computer interface (BCI) chairs let the user steer through brain signals. A hybrid BCI has been used to control the direction and speed of a simulated or real wheelchair [6], and a brain-controlled chair has been demonstrated for familiar environments [7]. As summarised in [1], BCI control needs several commands to start the chair and to set speed and direction, and long use is mentally tiring. Adding automated navigation to a P300-based BCI chair reduces this burden [8], but some such designs depend on cameras fixed in the surroundings and on complex software, which makes them costly and limits them to prepared environments [1].

B. EOG, Sip-and-Puff, Tongue and Facial-Pose Control

Electro-oculography (EOG) interprets eye movements as commands [9]. An EOG chair with seven ultrasonic sensors, tilt detection, health monitoring and GPS-based alerts has been reported [10], but the number of sensors and separate units makes it complex and costly [1]. Sip-and-puff controllers recognise commands from breath pressure but need calibration for each user [11]. Tongue-controlled systems require a magnet on the tongue or in the mouth [12], [13], which limits them to specific users [1]. Facial-pose navigation relies on a model of the destinations that a particular user visits [14].

C. Smartphone-Based Multi-Input Wheelchair

A smartphone-based wheelchair with three inputs (touch, phone-motion and voice) and four infrared sensors for obstacle detection was reported in [1]. It shows that a phone can act as a low-cost, multi-purpose interface. Voice control in that system had a response time of about two seconds, which is a useful benchmark for the present work.

D. Recent IoT, Voice and Vision-Based Systems

Recent work includes an IoT-based voice-activated wheelchair for healthcare and mobility assistance [15], an IoT-based intelligent wheelchair with vision-assisted and adaptive safety control [16], and a simulation-based design of a multimodal, vision-guided smart wheelchair [17]. These works show the current interest in voice, vision and IoT features. Vision-based designs generally need cameras and more processing power than a single distance sensor, which raises cost and complexity.

E. Early Robotised and Multimodal Systems

Early robotised wheelchairs such as VAHM [18] established embedded sensing and control, but without the smartphone interfaces that are available today. A multimodal wheelchair that combines several input methods with obstacle avoidance [19] improves usability at the cost of more components. A literature review of smart wheelchairs [20] gives broader background on the field. A recent design-stage paper proposes an Android application, a Bluetooth module, an Arduino Uno and a motor driver, with an obstacle sensor whose detections are reported through the application [21]. Like the present work it uses low-cost hardware, but it reports no measured results. Table I summarises the comparison.

TABLE I
COMPARISON OF RELATED APPROACHES

Approach	Strength	Limitation
BCI-based [6], [7], [8]	Hands-free, no muscle control needed	Several commands, mental load, costly and complex setup
EOG-based [9], [10]	Uses eye movement	Many sensors and units, costly
Sip-and-puff, tongue [11], [12], [13]	Works in severe paralysis	User-specific calibration or intrusive hardware
Smartphone with IR sensors [1]	Low cost, several inputs	About 2 s voice response time
Vision-assisted and vision-guided [16], [17]	Richer sensing of the surroundings	Cameras and more processing than a single distance sensor
Android app with Arduino Uno [21]	Low-cost hardware; app reports detected obstacles	Design-stage report; no measured results



IV. PROBLEM STATEMENT

Elderly and physically challenged people often cannot use manual or joystick-controlled wheelchairs, most conventional wheelchairs have no obstacle detection, and advanced smart wheelchairs are too expensive for many users. The research problem addressed in this paper is the design of an affordable semi-autonomous wheelchair system that can:

- accept spoken commands through a smartphone without any physical control;
- transmit commands wirelessly to an embedded controller;
- measure the distance to obstacles ahead continuously;
- stop automatically when an obstacle is near, overriding the user command;
- move forward, move backward, turn left, turn right and stop on command;
- stop safely when the wireless link is lost;
- use low-cost, readily available components; and
- remain open to later additions such as GPS, IoT and AI.

V. RESEARCH GAP AND RESEARCH CONTRIBUTION

The reviewed work shows that different systems cover different parts of the problem. BCI, EOG, sip-and-puff and tongue systems serve severe disability but are costly, intrusive or need calibration, and vision-based designs need cameras and more processing. Smartphone-based designs [1], [21] show that a phone can provide a low-cost interface, and [1] combines several inputs with infrared obstacle detection and reports a voice response time of about two seconds.

The present work therefore does not claim to be the first to combine a smartphone interface with obstacle sensing. Its aim is a deliberately minimal and fully specified design: ultrasonic sensing in place of infrared, a single Arduino Uno controller, a command-priority rule in which the safety decision always wins, a safe stop on loss of the wireless link, and a staged build and test plan with defined measurements. The principal contribution of the proposed system is the integration of:

- a smartphone voice interface that reuses the speech recognition already present in the phone;
- a Bluetooth link to a single low-cost microcontroller;
- continuous ultrasonic distance sensing;
- a command-priority rule in which safety takes precedence over commands;
- a safe stop when the wireless link is lost;
- a low-cost bill of materials;
- a staged build and test plan that begins on a toy-car platform; and
- an expandable architecture.

A key design distinction is made between two kinds of input that reach the motors. User commands express intent and may be unsafe. The safety decision protects the user. The design therefore never lets a user command override the safety decision. The system also does not claim autonomous path planning: it does not choose a route, and the user remains responsible for direction.

VI. PROPOSED SYSTEM

The proposed system is a semi-autonomous wheelchair in which the user stays in control of direction and the system watches for obstacles. The major modules are:

- Voice Interface Module (Android smartphone application)
- Communication Module (HC-05 Bluetooth)
- Control Module (Arduino Uno)
- Obstacle Sensing Module (HC-SR04 ultrasonic sensor)
- Safety Decision Module (command-priority rule in the Arduino program)



- Motor Driver Module (L298N)
- Drive Module (DC geared motors)
- Power Module (rechargeable battery pack)

The system has two paths. In the command path, the spoken command travels from the phone through the HC-05 to the Arduino, and from there through the L298N to the motors. In the safety path, the HC-SR04 reports the distance to the Arduino, which decides whether the command may be executed or the motors must be stopped.

VII. SYSTEM FLOW DIAGRAM

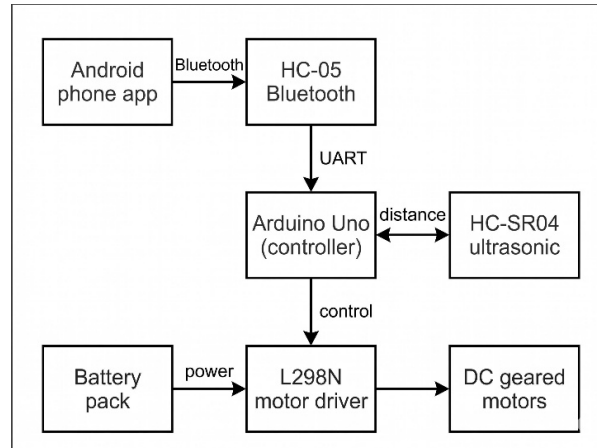


Fig. 1. Block diagram of the proposed system

Fig. 1 shows the architecture of the proposed system. The functions of the blocks are explained below.

A. Smartphone Voice Application

The user speaks one of five commands: Forward, Backward, Left, Right or Stop. The Android application converts the speech to text using the phone's speech recognition, matches the text to a command, and sends one command character over Bluetooth. A set of on-screen buttons gives the same five commands as a backup when voice recognition is unreliable.

B. HC-05 Bluetooth Module

The HC-05 receives the command character from the phone and passes it to the Arduino over a serial link. Its STATE output indicates whether a phone is connected, so the Arduino can detect a lost link.

C. Arduino Uno

The Arduino is the central controller. It reads the command from the HC-05, measures the distance with the HC-SR04, applies the command-priority rule of equation (2), and sets the inputs of the L298N driver.

D. HC-SR04 Ultrasonic Sensor

The sensor is mounted at the front of the vehicle. The Arduino triggers it about ten times per second and converts the echo time to a distance using equation (1).

E. L298N Motor Driver and DC Motors

The L298N converts the Arduino outputs into motor drive. One motor is placed on each side, so movement and turning are obtained by driving the two motors in different directions or stopping one of them, as given in Table II.

F. Power Supply

A rechargeable battery pack supplies the L298N and the motors. The Arduino regulator supplies 5 V to the HC-05 and the HC-SR04. All grounds are connected together.



VIII. METHODOLOGY

The working of the proposed system follows the sequence below.

- The user pairs the smartphone with the HC-05 module and opens the voice-control application.
- The user speaks a command. The phone converts it to text and the application sends the matching command character over Bluetooth.
- The HC-05 passes the character to the Arduino over the serial link.
- About every 100 ms the Arduino measures the distance to the nearest obstacle ahead with the HC-SR04.
- The Arduino compares the distance with the safety threshold. If the distance is within the threshold and the command is forward, the motors are stopped and the command is ignored.
- Otherwise the Arduino sets the L298N inputs so that the motors perform the commanded movement.
- If the Bluetooth link is lost, the Arduino stops the motors.
- The process repeats until the system is switched off.

The development methodology is staged so that each function is tested before the next one is added. In the first stage the vehicle drives forward at low speed and the obstacle-based stop is tested alone. In the second stage the Bluetooth link is added and the vehicle is driven with button commands from the phone. In the third stage voice commands are added. In the fourth stage the complete system is tested and measured. The stages are carried out first on a scaled two-motor toy-car chassis, which is cheaper and safer than a wheelchair and is sufficient to demonstrate the control logic.

IX. CONTROL ALGORITHM

The program running on the Arduino follows the algorithm below and the flowchart in Fig. 2.

- Initialise the serial link to the HC-05 and the trigger and echo pins of the HC-SR04.
- Read any command character received from the HC-05.
- Measure the distance d to the nearest obstacle ahead.
- If the command is forward and d is within the threshold d_{th} , stop the motors and ignore the command.
- Otherwise execute the command according to Table II.
- If the Bluetooth link is lost, stop the motors.
- Repeat from step 2.



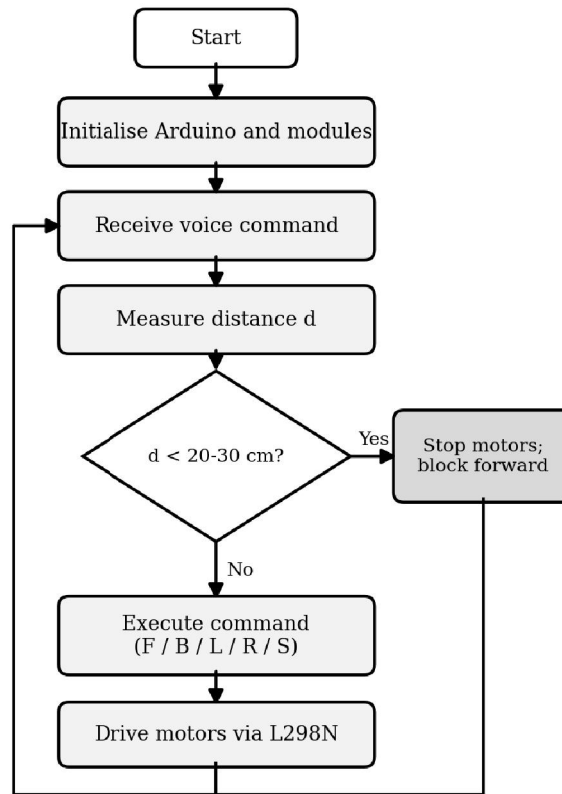


Fig. 2. Flowchart of the control algorithm

TABLE II
MOTOR STATES FOR EACH COMMAND

Command	Left motor	Right motor	Movement	Obstacle within threshold
F	Forward	Forward	Move forward	Blocked: motors stop
B	Reverse	Reverse	Move backward	Allowed
L	Stop	Forward	Turn left	Allowed
R	Forward	Stop	Turn right	Allowed
S	Stop	Stop	Stop	Allowed

X. PROPOSED IMPLEMENTATION FRAMEWORK

The implementation is organised into controller, interface, communication, sensing and actuation components, listed in Table III.

TABLE III
PROPOSED IMPLEMENTATION TECHNOLOGIES

Component	Technology / Purpose
Controller	Arduino Uno (ATmega328P)
Voice interface	Android smartphone application with speech-to-text
Wireless link	HC-05 Bluetooth module (UART)
Obstacle sensor	HC-SR04 ultrasonic sensor



Component	Technology / Purpose
Motor driver	L298N dual H-bridge driver
Drive	Two DC geared motors with wheels
Power	Rechargeable battery pack suited to the motors
Programming	Embedded C using the Arduino IDE
Prototype platform	Scaled two-motor toy-car chassis first; wheelchair later
Test instruments	Digital multimeter, measuring tape, stopwatch

A proposed pin assignment for the Arduino is given in Table IV. It may change during assembly. The Arduino transmit line to the HC-05 receive pin should pass through a divider, for example 1 k Ω and 2 k Ω , to give 3.3 V logic.

TABLE IV
PROPOSED ARDUINO PIN ASSIGNMENT

Arduino pin	Connected to
D10 / D11	HC-05 TXD / HC-05 RXD (through divider), software serial
A0	HC-05 STATE pin (link-loss detection)
D9 / D8	HC-SR04 Trig / Echo
D2, D4, D7, D12	L298N IN1, IN2, IN3, IN4
D3 / D6 (PWM)	L298N ENA / ENB (speed control)

The design keeps all safety decisions inside the Arduino so that they do not depend on the phone or on the wireless link. The phone only sends the user's intent.

XI. SAFETY AND RELIABILITY ANALYSIS

A. Collision Prevention

The ultrasonic sensor allows the controller to stop the vehicle before contact. The stopping distance depends on speed, motor inertia and the delay between measurements, so the threshold of 20-30 cm must be checked experimentally for the chosen speed.

B. Command Priority

Because the safety rule is applied in the Arduino, a mistaken voice command, a delayed command or a faulty phone application cannot drive the vehicle forward into a detected obstacle.

C. Loss of the Wireless Link

If the Bluetooth link is lost while the vehicle is moving, no new Stop command can arrive. The Arduino therefore reads the HC-05 STATE signal and stops the motors when the link is down.

D. Sensor Limitations

An ultrasonic sensor has a limited beam angle, a minimum range of about 2 cm, and can miss soft, angled or very thin objects. It can also be disturbed by acoustic noise and by temperature and air movement [1]. A single front sensor does not cover the sides and rear, and turning commands are allowed while an obstacle is ahead, so a turn may swing the vehicle towards an object that the sensor cannot see. Turning should therefore be tested at low speed.

E. Voice Recognition Reliability and Privacy

Speech recognition can fail in noisy places, and a phone speech service may need an internet connection and may process the audio outside the device. The design keeps the vocabulary to five short words and provides on-screen buttons and a Stop command as alternatives.

F. Power Supply

Motors draw a large current when they start, which can lower the supply voltage enough to reset a microcontroller. The battery pack must be adequate, grounds must be common, and the supply should be checked with a multimeter during motor start.



G. Scaling to a Wheelchair

The toy-car prototype demonstrates the control logic only. A chair carrying a person would need stronger motors and drive electronics, more sensors including side, rear and drop-off detection, speed limiting, a physical emergency stop and extensive safety testing before any use with people.

XII. DISCUSSION

The reviewed research shows that advanced inputs such as brain signals, eye movement and tongue control serve users with severe disability, while camera-based designs need more processing and cost. The smartphone-based system in [1] shows that a phone can provide several inputs at low cost.

The proposed design adopts the smartphone interface and adds a definite rule for what happens when a command and an obstacle conflict. It deliberately avoids path planning, cameras and AI, which keeps the hardware simple and cheap. This also limits what the system can do: it assists the driver, but it does not drive the chair.

XIII. EVALUATION PLAN

No experimental performance values are claimed in this paper. The prototype will be evaluated using the parameters in Table V after it has been built and tested.

TABLE V
PROPOSED EVALUATION PARAMETERS

Parameter	Purpose
Voice recognition success rate	Correct recognition of the five commands in quiet and noisy rooms
Command response time	Time from speaking to motor action, compared with about 2 s reported in [1]
Obstacle detection range and accuracy	Measured distance compared with actual distance
Stopping distance	Distance travelled after an obstacle is detected, at different speeds
Detection reliability	Detection of obstacles of different surfaces, sizes and angles
Bluetooth range	Distance over which commands are received reliably
Link-loss response	Whether and how quickly the motors stop when the link drops
Battery run time	Operating time on a full charge
Component cost	Total cost of the prototype
Usability	Ease of use reported by test users

The prototype will be tested separately for the obstacle-based stop, Bluetooth command control, voice command control, link-loss behaviour and battery operation. Results from other research papers will not be presented as results of the proposed system.

XIV. ADVANTAGES

- Hands-free operation through voice commands.
- Automatic collision prevention that overrides unsafe commands.
- Low cost, using readily available components.
- Simple hardware and program, easy to build and maintain.
- Wireless Bluetooth control from a familiar smartphone.
- Safe stop when the wireless link is lost.
- Suitable as an educational platform for embedded systems.
- Expandable towards GPS, IoT, cameras and AI.



XV. LIMITATIONS

The proposed system has several limitations. A single front ultrasonic sensor does not protect the sides, rear or floor edges. Ultrasonic sensing can be affected by noise, temperature and object surface. Voice recognition may be unreliable in noise and may depend on an internet connection. Bluetooth range is limited to roughly ten metres. The system assists the driver but does not plan a route. The design has not yet been built or measured, and the toy-car prototype does not prove safety for a real wheelchair carrying a person.

XVI. FUTURE WORK

- Additional sensors for side, rear and drop-off detection.
- Offline voice recognition.
- Emergency alert message to a predefined contact.
- GPS tracking and outdoor navigation.
- Camera-based navigation and AI-based assistance.
- IoT-based remote monitoring.
- Speed limiting and a physical emergency stop.
- Testing on a full-size wheelchair with safety review.

XVII. CONCLUSION

This paper presented the design of a low-cost semi-autonomous smart wheelchair navigation system that combines Bluetooth voice control with ultrasonic collision prevention on an Arduino Uno platform. A review of existing control methods showed that advanced inputs and vision-based designs are costly or intrusive, while smartphone-based designs offer a low-cost alternative.

The proposed design builds on this low-cost approach with a few easily available components, a fixed command-priority rule in which an obstacle-based stop always overrides a command, and a safe stop on loss of the link. The design will first be validated on a scaled toy-car platform.

Actual performance, including recognition rate, response time, stopping distance and battery life, must be established by building and testing the prototype. These measurements will be reported in future work and will form the basis for judging the effectiveness of the approach.

REFERENCES

- [1]. A. K. Dalsaniya and D. H. Gawali, "Smart phone based wheelchair navigation and home automation for disabled," 2016.
- [2]. "All about HC-05 Bluetooth module: connection with Android," GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/digital-logic/all-about-hc-05-bluetooth-module-connection-with-android/>
- [3]. Arduino Project Hub, "Ultrasonic sensor with Arduino tutorials." [Online]. Available: <https://projecthub.arduino.cc/>
- [4]. FlyRobo, "DC gear motor." [Online]. Available: <https://www.flyrobo.in/motor-and-driver/dc-gearmotor>
- [5]. J. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, "A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 5, pp. 720-729, 2012.
- [6]. B. Rebsamen, C. Guan, H. Zhang, C. Teo, M. H. Ang, and E. Burdet, "A brain controlled wheelchair to navigate in familiar environments," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 6, pp. 590-598, 2010.
- [7]. I. Iturrate, J. M. Antelis, A. Kübler, and J. Minguez, "A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 614-627, 2009.
- [8]. C.-H. Kuo, Y.-C. Chan, H.-C. Chou, and J.-W. Siao, "Eye-glasses based electro-oculography human-wheelchair interface," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, San Antonio, TX, USA, 2009.



- [9]. T. Pingali et al., "Eye-gesture controlled intelligent wheelchair using electro-oculography," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Melbourne, Australia, 2014, pp. 2065-2068.
- [10]. I. Mougharbel, R. El-Hajji, H. Ghamlouch, and E. Monacelli, "Comparative study on different adaptation approaches concerning a sip and puff controller for a powered wheelchair," in *Proc. 2013 Science and Information Conf.*, 2013, pp. 597-603.
- [11]. J. Kim et al., "The tongue enables computer and wheelchair control for people with spinal cord injury," *Sci. Transl. Med.*, vol. 5, no. 213, 2013.
- [12]. M. E. Lund, H. V. Christensen, H. A. Caltenco, E. R. Lontis, and N. S. Andreasen Struijk, "Inductive tongue control of powered wheelchairs," in *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBC)*, Buenos Aires, Argentina, 2010, pp. 3361-3364.
- [13]. A. Escobedo, A. Spalanzani, and C. Laugier, "Multimodal control of a robotic wheelchair: using contextual information for usability improvement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013.
- [14]. S. Ayyappan "An IoT based voice-activated wheelchair for healthcare and mobility assistance," IEEE Document ID 11554897. [Online]. Available: <https://ieeexplore.ieee.org/document/11554897>
- [15]. P. Vishnu Kumar , "An IoT based intelligent smart wheelchair with vision assisted and adaptive safety control," IEEE Document ID 11546871. [Online]. Available: <https://ieeexplore.ieee.org/document/11546871>
- [16]. K. N. Pillai , "Simulation-based design of a multimodal vision-guided smart wheelchair for secure and energy-efficient mobility," IEEE Document ID 11506780. [Online]. Available: <https://ieeexplore.ieee.org/document/11506780>
- [17]. A. G. Bourhis and Y. Agostini, "The VAHM robotized wheelchair," in *Proc. IEEE Int. Conf. Rehabilitation Robotics*, 1996, IEEE Document ID 655076. [Online]. Available: <https://ieeexplore.ieee.org/document/655076>
- [18]. J. M. Ramos-Arreguin, R. A. Osornio-Rios, E. Cabal-Yeppez, A. Garcia-Perez, et al., "Smart wheelchair with multimodal navigation interface and obstacle avoidance system," in *IEEE Int. Conf.*, 2014, IEEE Document ID 6978290. [Online]. Available: <https://ieeexplore.ieee.org/document/6978290>.
- [19]. R. C. Simpson, "Smart wheelchairs: A literature review," *J. Rehabil. Res. Dev.*, vol. 42, no. 4, pp. 423-436, 2005. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/16320139/>
- [20]. V. Kengale, K. Bansod, C. Sure, M. Dalal, S. Bawane, and M. Pal, "Designing of a smart wheelchair for people with disabilities," in *Proc. 3rd Int. Conf. Intelligent Commun. Technol. and Virtual Mobile Networks (ICICV)*, 2021, pp. 1518-1522, doi: 10.1109/ICICV50876.2021.9388417.

