

# Obstacle Avoidance and Voice Control for Autonomous Car

ANNWESH HAZRA<sup>1</sup>, SIDDHARTHA SEN<sup>2</sup>

<sup>1, 2</sup>*Dept of CSE, Institute of Engineering and Management, Kolkata*

***Abstract-*** This paper presents a voice-controlled, obstacle-avoiding car system designed to enhance accessibility and safety in transportation. The system uses an Arduino UNO microcontroller interfaced with a Bluetooth module to receive voice commands from an Android application. These commands are processed to drive DC motors via an L298D motor driver. Obstacle detection is achieved using ultrasonic sensors, allowing the car to avoid collisions autonomously. This innovation can aid differently-abled individuals, support operations in hazardous environments, and promote hands-free vehicle control. Future enhancements include integration with IoT and AI to enable global remote access and intelligent navigation, making it a significant step toward smart and inclusive mobility solutions.

## I. INTRODUCTION

An innovative prototype of a smart car unit equipped with voice control and obstacle avoidance features has the core idea is to design a system that enables hands-free vehicle operation using voice commands, which can be especially useful for differently-abled individuals, in hazardous environments, or in areas where direct human control is not feasible at disaster zones, chemical plants, or even military zones. The primary mode of communication between the user and the car is established via a Bluetooth module, which acts as a bridge between the Android mobile application and the car system. The user speaks into the mobile app, which captures and processes the voice command. These commands are then converted into digital signals and transmitted through Bluetooth. The system supports a communication range of up to 100 meters, offering flexibility and mobility to the user.

Recent advancements in engineering particularly in mechanical design, electronics, and embedded systems have paved the way for the creation of intelligent machines that can perform complex tasks with little to no human involvement. Among the most

impactful of these innovations are autonomous vehicles [1], which integrate real-time sensing with automation to improve safety, efficiency, and usability across a wide range of environments and industries. Autonomous systems are especially valuable in situations where human operation may be difficult, unsafe, or even impossible, such as in disaster response zones, high-risk industrial areas, or assisting individuals with physical impairments. One of the key technologies enabling these vehicles to operate independently is obstacle detection and avoidance [3][4]. To achieve this, ultrasonic sensors [5] are commonly used. These sensors function by sending out high-frequency sound waves and measuring the time it takes for the echoes to bounce back from nearby objects. This allows the system to accurately calculate distances in real time and take quick decisions—such as stopping, slowing down, or changing direction: to prevent collisions and ensure smooth navigation. In parallel, voice recognition technology adds another powerful layer of control. By using simple spoken commands, users can direct the vehicle without the need for physical interaction making the system especially helpful for people with limited mobility. These voice inputs are captured and processed through a Bluetooth-connected Android app, translated into digital commands, and interpreted by a microcontroller, such as an Arduino UNO [8]. The microcontroller then acts as the ‘brain’ of the vehicle, executing the commands through motor drivers to manoeuvre the car accordingly.

By combining autonomous navigation and voice-based control, the proposed system becomes more than just a technical experiment, it becomes a practical solution. It has the potential to support mobility-impaired individuals, reduce human risk in hazardous environments, and serve as a building block for next-generation smart transportation systems. As technology continues to evolve, further integration with Artificial Intelligence (AI) and the Internet of

Things (IoT [12]) could enable remote operation, intelligent decision-making, and real-time monitoring—bringing us closer to a future where safe, accessible, and autonomous travel becomes the norm.

## II. EXISTING MODEL AND PROPOSED SYSTEM

The evolution of automotive technologies in recent decades has led to remarkable advancements in vehicle automation, safety systems, and user interfaces. However, despite these innovations, a critical shortcoming persists: modern vehicles are not adequately designed to serve the mobility needs of elderly individuals and persons with physical disabilities. Conventional vehicle control mechanisms—such as steering wheels, pedals, and gear levers—require fine motor skills and physical coordination, posing substantial barriers for users with age-related limitations or motor impairments. This often results in a loss of independence, reduced mobility, and diminished quality of life.

To address this challenge, there is a growing need to develop alternative vehicular control systems that prioritize accessibility, safety, and ease of use. In this context, voice-controlled vehicle systems present a promising solution by enabling users to operate vehicles through natural language commands, thereby eliminating the need for manual input. When integrated with real-time obstacle detection and avoidance [3][4] capabilities, such systems can significantly enhance safety and autonomy for mobility-impaired individuals.

This paper proposes a voice-operated, obstacle-avoiding vehicle system tailored specifically for users with limited physical abilities. The system is built upon an embedded platform incorporating voice recognition modules and sensor-based navigation. Users can issue simple voice commands to initiate movement, change direction, or halt the vehicle, while ultrasonic or LiDAR sensors continuously monitor the environment to detect obstacles and initiate corrective actions. These features are designed not only to ensure operational safety, but also to restore a degree of autonomy and confidence in users who may otherwise be dependent on caregivers or specialized transport services.

By integrating accessible human-machine interaction with real-time safety mechanisms, the proposed system seeks to create a more inclusive mobility solution. This research contributes toward the broader objective of developing intelligent transportation technologies that are both technically robust and socially responsive, ultimately fostering greater independence for underserved user populations.

## III. HARDWARE CONFIGURATION

### i. Arduino Uno

The Brain of the entire system. It is a small programmable board that receives inputs from sensors and sends commands to control parts like motors (It is like the manager telling everyone else what to do based on the situation).



Fig. 1. Structure and parts of Arduino Uno

### ii. Ultrasonic sensor (HC-SR04)

This is like the eyes of the robot. It sends out sound waves and listens for echoes to detect how far things are. If something comes too close, it alerts the Arduino, which can make the car stop or turn.

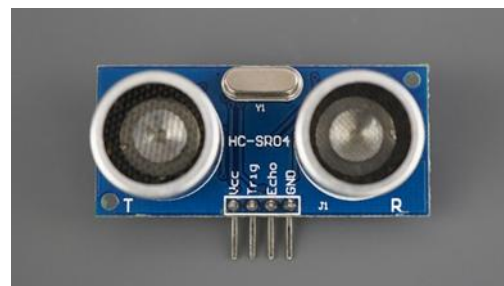


Fig. 2. The Ultrasonic sensor (HC-SR04) used

iii. Motor Driver (L293D)

Acts like a middleman between the Arduino and the motors. The Arduino itself can't directly power motors because they need more current, so the motor driver helps deliver that power safely and lets us control direction and speed.

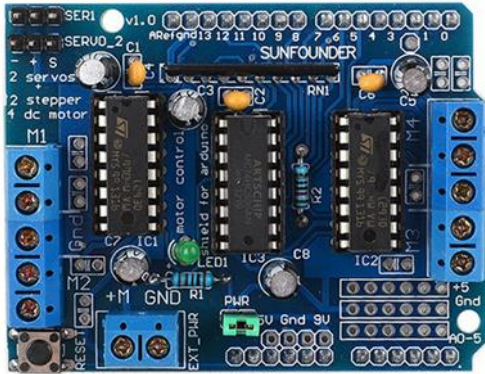


Fig. 3. The Motor Driver (L293D) used.

iv. Servo Motor (Tower Pro SG90)

This motor is used for precise turning or rotation. It might be used to rotate the ultrasonic sensor so the robot can “look” left or right, or it could help in steering.



Fig. 4. The Servo Motor (Tower Pro SG90) Used.

v. Geared Motors (x2)

These are the legs of the robot as they make it move. The gears inside reduce speed but increase strength, so the car moves at a good speed and can carry some weight.



Fig. 5. The Geared motors used respectively.

vi. Robot Chassis

This is the body or structure that holds everything together; wheels, motors, sensors, etc.

A good chassis keeps the whole system balanced and easy to move.



Fig. 6. The Robot Chassis used in this model.

vii. Power Supply

The energy source for the whole robot. Usually a battery pack, it gives power to all the compartments.

viii. Keil  $\mu$ Vision IDE

A software tool used to write and test code for microcontrollers (especially non-Arduino ones). Even though Arduino has its own IDE, Keil might be used if working with more complex embedded systems or doing simulations.

ix. Embedded C Programming Language

This is the language we use to talk to the microcontroller. It helps us write the logic like “if obstacle is close, turn left” or “move forward for 5

seconds.” It is simple, fast, and perfect for hardware-level tasks.

#### IV. SOFTWARE CONFIGURATION

##### A. ARDUINO IDE

The Arduino IDE (Integrated Development Environment) is an essential software platform used for programming Arduino boards. Designed with simplicity in mind, it offers an easy-to-navigate interface that caters to both beginners and experienced developers. This environment allows users to write, compile, and upload code directly to Arduino microcontrollers with ease. It features a built-in text editor with syntax highlighting and auto-complete functionality, helping users code more efficiently. The programming language it supports is based on C and C++, making it relatively straightforward for newcomers to grasp. A standout feature of the Arduino IDE is its collection of built-in examples, or "sketches," which offer ready-made code for common hardware modules and projects—greatly reducing development time. Additionally, the software includes a serial monitor, allowing for real-time communication and debugging between the computer and the Arduino board. The IDE is compatible with a wide variety of Arduino models, including the Uno, Mega, and Nano, and it runs seamlessly across Windows, macOS, and Linux platforms. Overall, the Arduino IDE is a versatile and beginner-friendly tool that plays a vital role in transforming innovative ideas into functional electronic projects.

##### B. EMBEDDED C

Embedded C is the language used to write the core logic that controls how your robot behaves. It gives direct access to hardware-level operations like reading sensor data, driving motors, and responding to input. Using Embedded C ensures that your robot runs efficiently, as the code executes quickly and with minimal overhead. This is especially important when the robot needs to react to obstacles or commands in real-time.

##### C. PYTHON AND VOSK

Python is often used on a computer or Raspberry Pi to interact with the Arduino. When paired with Vosk, which is a speech recognition [6] toolkit, it allows your robot to follow spoken instructions. What makes Vosk

useful is that it does not need internet access to work. This setup is helpful for adding voice control in environments where connectivity is limited or unavailable, making the robot more flexible and interactive.

##### D. MATLAB

MATLAB [2][9] is a high-level computing platform widely used for designing, simulating, and analysing systems in robotics and automation. Its importance lies in its ability to model control systems, process sensor data, and simulate robot behaviour before physical implementation. With specialized toolboxes, it supports real-time signal processing, image analysis, and hardware interfacing. MATLAB is especially valuable in academic and research settings due to its accuracy, visualization capabilities, and compatibility with microcontrollers like Arduino and Raspberry Pi. It allows for rapid development and testing of algorithms, helping ensure system reliability and performance.

#### V. IMPLEMENTATION AND DESIGN

The robot's design begins by defining its purpose—such as obstacle avoidance or voice-controlled movement—followed by assembling essential components like the Arduino Uno, HC-SR04 ultrasonic sensor, L293D motor driver, DC motors, a servo motor, and chassis. Once the structure is built, components are electrically connected: motors to the driver, sensors and servo to Arduino pins, and optionally a Bluetooth module for wireless control. The robot is programmed using Arduino IDE in Embedded C, with logic for movement, obstacle sensing, and control commands. For voice control, a Python script using Vosk processes voice input and communicates with Arduino via serial. Alternatively, the Blynk app enables mobile control over Bluetooth. Before deployment, simulation in tools like Tinkercad or Proteus can help test logic. The code is uploaded, and power is supplied through batteries. Final steps include testing, calibrating sensor responses, adjusting motor speeds, and documenting the project with wiring diagrams, code, and observations.

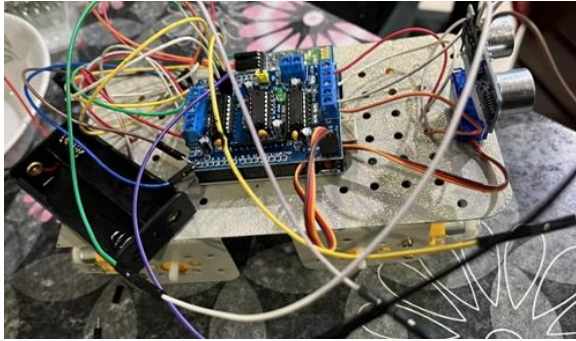


Fig. 7. Design of the model.

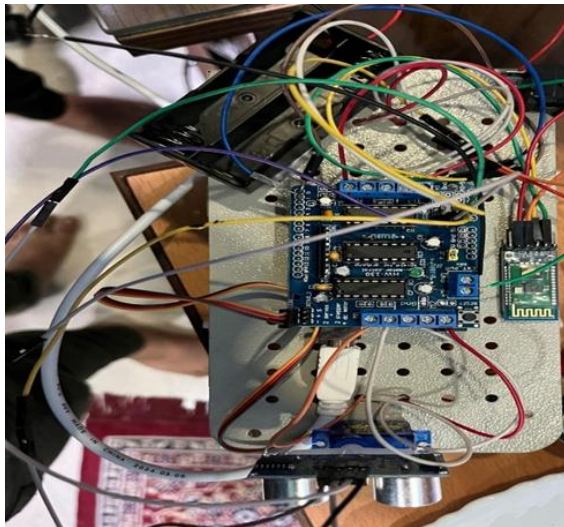


Fig. 8. Model with all the components connected.

## VI. ARDUINO CODE

```
const int TRIG_PIN = 9;
const int ECHO_PIN = 10;
void setup() {
    Serial.begin(9600);

    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    pinMode(3, OUTPUT); // forward
    pinMode(5, OUTPUT); // backward
    pinMode(6, OUTPUT); // forward
    pinMode(11, OUTPUT); // backward
}

void loop() {
    long duration;
    float dcm; // distance in cm

    digitalWrite(TRIG_PIN, LOW);
```

```
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    duration = pulseIn(ECHO_PIN, HIGH);
    dcm = duration / 29.1 / 2;

    Serial.print(dcm);
    Serial.print(" cm, ");

    if (dcm < 30) {
        // Obstacle detected, stop
        digitalWrite(3, LOW); // forward off
        digitalWrite(6, LOW); // forward off
        digitalWrite(5, LOW); // backward off
        digitalWrite(11, LOW); // backward off
        delay(500);
    } else {
        // No obstacle, move forward
        digitalWrite(3, HIGH); // forward on
        digitalWrite(6, HIGH); // forward on
        digitalWrite(5, LOW); // backward off
        digitalWrite(11, LOW); // backward off
    }

    delay(10); // Small delay to stabilize readings
}
```

## VII. ARDUINO FLOWCHART

This flowchart outlines the working of an obstacle-avoiding robot. It starts by initializing sensors and motors, then continuously checks the front distance. If the path is clear (more than 20 cm), it moves forward. If an obstacle is detected, it stops, moves back, and checks both left and right distances using a rotating sensor. Based on which side is clearer, it turns in that direction and resumes the process.

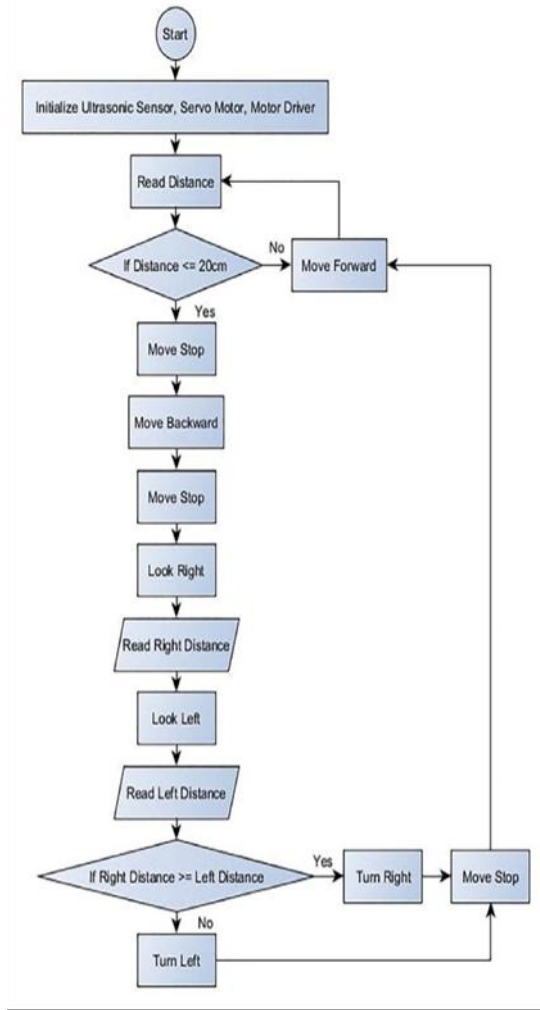


Fig. 9. Detailed Arduino Flowchart

## VIII. EXPERIMENTAL RESULTS

### A. ENVIRONMENTAL TESTING

The experiments were conducted in a controlled environment replicating real-world conditions, including:

- Obstacle types: Solid walls, cylindrical objects, and human-sized objects.
- Voice command clarity: Tested in quiet and moderately noisy settings.
- Range testing for Bluetooth connectivity.

### B. VOICE COMMAND ACCURACY

Assess the car's ability to recognize and respond accurately to voice commands.

Method: Commands such as 'Forward,' 'Backward,' 'Left,' 'Right,' and 'Stop' were given in various tones and speeds.

Results:

Table 1. This table provides the real data on voice command accuracy of the trail.

Command Type	Attempts	Successful Responses	Accuracy (%)
Forward	50	48	96
Backward	50	47	94
Left	50	45	90
Right	50	46	92
Stop	50	49	98

### C. OBSTACLE AVOIDANCE ACCURACY

Evaluation of the effectiveness of the ultrasonic sensor in detecting obstacles and making autonomous navigation decisions.

Method: The car was placed in paths with obstacles at varying distances (10cm, 20cm, 30cm, etc.).

Results:

Table 2. Real data of obstacle avoidance accuracy.

Obstacle Distance	Detection Accuracy (%)	Autonomous Navigation Success Rate (%)
10 cm	100	95
20 cm	100	97
30 cm	100	98

### D. BLUETOOTH RANGE

Determining the operational range of the Bluetooth module.

Method: The car was operated at varying distances from the command source (5m, 10m, 15m, etc.)

Results:

- Effective range: 15 meters in an unobstructed environment.
- Slight delay in command execution beyond 12 meters.

### E. SAFETY PERFORMANCE

Test of the auto-braking system to prevent collisions.  
Method: The car was directed towards an obstacle while measuring braking response time.

Results:

- Average braking response time: 0.4 seconds.
- Zero collisions recorded during 50 trials.

### F. PERFORMANCE IN NOISY ENVIRONMENT

Objective: Test the impact of background noise on voice command accuracy. Results:

- Accuracy in quiet environment: 95%.
- Accuracy in noisy environment: 75% (due to misinterpretation of commands).

## IX. FUTURE SCOPES

The current model of the obstacle-avoiding and voice-controlled robotic [6] car demonstrates fundamental capabilities in autonomous navigation and human-robot interaction. However, the system can be significantly enhanced through the following advancements:

One of the most promising upgrades involves the integration of camera modules and computer vision algorithms. This would allow the robot to identify objects, recognize pathways, and classify obstacles using image processing techniques, expanding its ability to handle complex, real-world environments. A smartphone application could be designed to stream live video from the camera, enabling users to remotely monitor and control the vehicle with greater precision.

In addition to visual feedback, the following future enhancements can be explored:

- **AI-Based Path Planning:** Implementing advanced algorithms such as Dijkstra's algorithm or the A\* search algorithm would enable the robot to compute optimal paths autonomously and adapt to dynamic environments.
- **IoT [12] Integration:** Linking the robotic system to cloud platforms could facilitate real-time remote monitoring, control, and predictive maintenance through over-the-air software updates.

- **GPS-Based Outdoor Navigation:** Adding GPS modules would enable location-based path planning [11] and autonomous outdoor exploration, making the system applicable for delivery robots or search-and-rescue tasks.
- **Gesture Recognition [6] Control:** Using smartphone inertial sensors or external cameras, gesture-based control systems could replace or supplement voice commands, enhancing hands-free operation in noisy environments.
- **Personalized Voice Commands:** Machine learning techniques could be applied to develop speaker-dependent voice models, allowing the system to respond exclusively to authorized users, thus increasing both security and command reliability.

These enhancements would not only broaden the practical applications of the robotic car but also make it more adaptive, intelligent, and suitable for deployment in industries, healthcare assistance, and smart mobility solutions.

## X. CONCLUSION

This device demonstrates the effective combination of real-time obstacle detection and voice-controlled navigation in a single autonomous vehicle framework. The use of ultrasonic and infrared sensors allows the system to detect and respond to nearby obstacles, significantly reducing the risk of collisions [3][4]. Meanwhile, the speech recognition module enhances the accessibility and usability of the vehicle by enabling voice-based control, making it particularly useful for users with limited mobility [5][6]. The integration of these features not only improves operational safety but also demonstrates the potential for the system's application in both personal assistive devices and autonomous driving solutions.

Looking forward, the incorporation of advanced technologies such as artificial intelligence, cloud connectivity, and IoT [12] could allow the system to achieve fully autonomous operation, adaptive path planning [11], and real-time environment analysis, further enhancing its performance in real-world scenarios.

IX. REFERENCES

- [1] A. K. J, R. Rajalakshmi, M. Gowtham, R. Geetheshwar and S. R. Srividhya, "Eyes on the Road: Unleashing the Power of Autonomous Vehicles with Cameras," 2025 International Conference on Intelligent and Cloud Computing (ICoICC), Bhubaneswar, India, 2025, pp. 1-5, doi: 10.1109/ICoICC64033.2025.11052026. keywords: {Bluetooth;Automation;Accuracy;Navigation;Robot sensing systems;Video surveillance;Real-time systems;Acoustics;Gyroscopes;Collision avoidance;Voice-Controlled robot;Arduino UNO [8];Bluetooth communication;Assistive technology;Gesture control;Obstacle avoidance;Mobile application;Smart automation},
- [2] L. Chen, H. Guo, Q. Meng and J. Liu, "Talk-to-Drive [2]: A Voice-Driven Framework for Driving Style Adjustment in Motion Planning\*," 2025 Joint International Conference on Automation-Intelligence-Safety (ICAIS) & International Symposium on Autonomous Systems (ISAS), Xi'an, China, 2025, pp. 1-6, doi: 10.1109/ICAISISAS64483.2025.11051870. keywords: {Adaptation models;Simulation;Speech recognition;Predictive models;Mathematical models;Planning;MATLAB [2][9];Autonomous vehicles;Vehicles;Predictive control;Talk-to-drive;motion planning;personalized driving styles;artificial potential field},
- [3] Shreya Sule, Kritak Gupta, Viraj Desai. "Autonomous Car: The Future of Roadways." *International Journal of Research in Technology & Management*, Vol. 2, October 2014.
- [4] Al-Qadasi, Y. et al. "Fully Autonomous Wheelchair for Indoor Mobility: A Hybrid Approach to Mapping and Navigation with ROS." *Journal of Science and Technology*, 30(2), January 2025. <https://doi.org/10.20428/jst.v30i2.2435>.
- [5] Sourabh Marne, Gayatri Deshpande, Drishti Bomble, Jay Kharat. "Human Voice Controlled Robot Embedded with Real Time Obstacle Detection and Avoidance." *International Journal of Engineering and Advanced Technology*, Vol. 8, Issue 5.
- [6] Yasar Ali Memon, Imaaduddin Motan, Muhammad Ali Akbar, Sarmad Hameed, Moez Ul Hasan. "Speech Recognition System for Voice Controlled Robot with Real Time Obstacle Detection and Avoidance." *International Journal of Electrical, Electronics and Data Communication*, Vol. 4, Issue 9, September 2016.
- [7] P. Narendra Ilaya Pallavan, S. Harish, C. Dhachinamoorthi. "Voice Controlled Robot with Real Time Barrier Detection and Advertising." *International Research Journal of Engineering and Technology (IRJET)*, Vol. 6, Issue 1, January 2019.
- [8] Md Nazmul Howlader, Rimjhim Akter Bristi. "Autonomous Two-Wheeler Vehicle Built on the Arduino." *Authorea*, February 10, 2025. <https://doi.org/10.22541/au.173920163.31413357/v1>.
- [9] Junfeng Mou. "Research on Autonomous Navigation and Obstacle Avoidance Algorithms for Robotic Vacuum Cleaners by Fusing LiDAR and Visual Information." *SPIE Proceedings*. DOI: <https://doi.org/10.1117/12.3060490>.
- [10] Anonymous. "Advanced Data Classification Framework for Enhancing Cyber Security in Autonomous Vehicles." *Automation*, 6(1), 2025.
- [11] Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3310-3317, 1994.
- [12] H. Sundmaecker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the Internet of Things," *European Commission*, 2010.
- [13] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed., MIT Press, 2011.