Design and Implementation of A Voice Controlled Indoor Autonomous Robot Kit

Ren Zhong[1], Zhaofeng Tian[2], Qiren Wang[2], Mingyu Guo[2] and Weisong Shi[2]

[1]*Computer Science, Wayne State University* [2]*University of Delaware*

***Abstract*— The integration of voice interaction and autonomous driving into robotic systems holds immense potential for enhancing mobility aids in environments such as homes and healthcare facilities. Despite advancements in voice technology and navigation systems, existing indoor robots and mobility aids often lack intuitive, secure, and adaptive functionalities. This paper introduces VOCAR (Voice-Controlled Autonomous Robot Kit), a modular system designed to enable natural voice-based interaction and efficient autonomous navigation for mobile robots. VOCAR features personalized voiceprint authentication, a large language model (LLM)-based intent recognition pipeline, and advanced planning modules for obstacle avoidance, narrow passage traversal, and energy-efficient trajectory generation. It employs a unified CAN bus protocol to ensure compatibility with diverse robotic platforms. The feasibility of VOCAR is demonstrated through its implementation on a powered wheelchair, showcasing its ability to enhance user autonomy and interaction in dynamic indoor environments. This work highlights VOCAR's potential to transform mobility aids by providing a secure, adaptive, and energy-conscious solution for voice-controlled autonomous navigation.**

***Index Terms*—large language model application, energy-efficient indoor autonomous driving, human-machine interaction**

I. Introduction

Endowing objects with the ability to act and communicate has long been a compelling vision, inspiring both imagination and innovation. Recent advancements in voice technology and autonomous driving have brought this vision closer to reality, enabling robots to perform tasks such as detecting falls in elderly individuals [1] or conducting house-cleaning activities. However, while specialized robots with autonomous navigation [2]–[4] or voice interaction [5] demonstrate exceptional performance in specific domains, most indoor robots and mobility aids still lack the integration of these functionalities. This gap raises a critical question: how can these advanced capabilities be made accessible for frequent and localized interactions in everyday indoor settings such as homes or healthcare facilities? For example, elderly or disabled individuals in care centers often rely on assistance to navigate shared spaces. Enhancing wheelchairs with autonomous driving and voice interaction capabilities could significantly reduce their dependence on caregivers, fostering greater independence and improving their overall quality of life.

Analyzing the requirements for developing an adaptive system reveals several critical functionalities that must be addressed. First, the system must support reliable and context-aware voice interaction. In real-world scenarios, especially in noisy or crowded environments, the system must be able to filter and respond exclusively to commands from authorized users. This calls for the implementation of reliable user authentication mechanisms that can distinguish qualified users based on unique speech characteristics such as tone, accent, or speaking style. Beyond authentication, the system must also excel at intent recognition, ensuring that user commands are not only understood but also translated into precise operational tasks for the robot. Meeting these requirements is essential to maintain reliability and usability, particularly in dynamic and complex settings where miscommunication could lead to operational errors.

The second requirement focuses on achieving adaptive and safe autonomous driving to accommodate the diversity of robot configurations and operational environments. Mobile robots vary widely in size, shape, and kinematic capabilities, such as four-wheeled versus two-wheeled systems. This diversity necessitates a flexible control architecture capable of adapting to different mechanical designs and constraints. Additionally, the system must ensure collision-free navigation in both open outdoor spaces and constrained indoor environments. Addressing this requirement demands an advanced perception and planning module capable of interpreting environmental data in real time and making decisions that prioritize safety and efficiency.

Privacy is also critical considerations for indoor environments, particularly in sensitive settings such as homes or healthcare facilities. To protect user privacy, the system must avoid relying on camera-based sensors for navigation and interaction. Instead, it should prioritize alternative sensing modalities, such as LiDAR or ultrasonic sensors, which provide the necessary data while preserving privacy. Moreover, incorporating advanced functionalities increases computational and operational demands, making energy optimization essential for prolonged usability and system efficiency.

Finally, the system must address the need for a versatile communication framework to ensure compatibility with a wide range of robot platforms. Robots can vary significantly in their locomotion systems, with some requiring differential steering for four-wheeled designs while others depend on balance-aware algorithms for two-wheeled configurations. To meet this requirement, the system must incorporate a modular command translation layer capable of converting high-level operational instructions into specific control signals tailored to the mechanics of each robot. This functionality not only

ensures seamless operation across diverse platforms but also supports scalability, enabling the integration of future robotic systems without major redesigns or modifications.

To address the outlined challenges, we propose the design of VOCAR, Voice-Controlled Autonomous Robot Kit, which aims to enhance mobile robots by combining voice-based human machine interaction and autonomous driving into an unified system. The system begins with an initialization phase, where the user engages in a conversation with an Large Language Model (LLM) based chatbot to register personalized voiceprints, ensuring only authorized individuals can control the robot. Simultaneously, the user manually guides the robot through the operational environment, constructing a detailed multi-layered map containing semantic, positional, and navigation goal data. During this process, VOCAR also collects kinematic parameters to calibrate the robot for smooth and accurate navigation. Once initialized, the system enables intuitive real-time interaction, continuously monitoring environmental audio to identify commands from authorized users using voiceprint-based verification. Recognized commands are processed through a speech-to-text module and parsed by the LLM-based chatbot, which matches them to predefined actions. For navigation, VOCAR generates safe and energy-efficient trajectories, dynamically switching between narrow passage planner and open area planner depending on the environment, while optimizing both path smoothness and energy consumption. Communication between VOCAR and the robot's hardware is facilitated by a unified CAN bus protocol, ensuring compatibility with diverse robotic platforms. By integrating these elements, VOCAR delivers a secure, adaptable, and efficient system for enhancing mobile robots.

The contributions of this work are summarized as follows:

1. ***Comprehensive Analysis and System Design.*** We provide an in-depth analysis of the challenges in developing voice-controlled autonomous driving systems for mobile robots in dynamic indoor environments.

2. ***An advanced voice command processing pipeline.*** We propose a real-time voice recognition module that combines streaming speech-to-text conversion with LLM-based parsing, ensuring natural and reliable command execution while filtering out malicious or unintended inputs.

3. ***A safe and energy-efficient navigation module.*** VOCAR's navigation system dynamically combines narrow passage planning, obstacle avoidance, path smoothness optimization, and energy efficiency to generate safe, user-friendly, and power-conscious trajectories.

4. ***A unified communication protocol for robotic platforms.*** We design a unified CAN bus protocol that ensures seamless integration with a wide variety of robotic platforms, enabling scalable and reliable operation across diverse hardware configurations.

The remainder of this paper is organized as follows: Section II reviews the related work. Section III outlines the design goals and provides an overview of VOCAR. The detailed design of the system is described in Section IV, followed by an initial implementation on a power wheelchair in Section V. Finally, Section VI concludes the paper and discusses the future work of VOCAR.

II. Related Work

A. Voice Recognition

Several previous works have focused on voice recognition. In 2016, DeepSpeech2 was introduced by Amodei*et al.* [6]. This pioneering work introduced an end-to-end deep learning approach capable of recognizing both English and Mandarin speech with near-human accuracy. It demonstrated the potential of deep learning to handle diverse speech recognition tasks across various languages and acoustic environments. The system replaced traditional hand-engineered components with neural networks, marking a significant step toward a unified speech recognition system, adapting to various contexts and languages with minimal modifications. Furthermore, Han*et al.* [7] demonstrated remarkable progress in conversational speech recognition through densely connected LSTMs and an innovative parameter averaging adaptation method. Their system achieved record-breaking performance on both Switchboard and CallHome evaluation sets, notably surpassing human-level transcription accuracy on the Switchboard dataset, establishing a new benchmark in the field of speech recognition technology. Additionally, in recent years, frameworks like Gesper [8] and MC-SpEx [9] were developed for voice recognition in complex situations and noise environments. Gesper introduces a novel two-stage speech reconstruction approach that reverses the traditional order - performing restoration before enhancement. The system employs complex spectral mapping-based generative adversarial networks (CSM-GAN) for speech restoration, followed by full band-wideband parallel processing for enhancement, effectively addressing various speech quality issues like noise, coloration, discontinuity, and reverberation. MC-SpEx introduces a novel speaker extraction system that improves upon previous approaches by implementing multi-scale interfusion through weight-shared ScaleFusers and a multi-scale interactive mask generator to better leverage multi-scale information. The system also introduces conditional speaker modulation to enhance speaker embedding utilization, resulting in state-of-the-art performance in extracting target speakers' speech from mixed audio.

With the development of voice recognition, voice-controlled robots have played a big role in assisting people for a few years. This technique has been embedded into fields such as robotic arms, robotic vehicles, and wearable devices. Mishra *et al.* [5] introduced their voice-controlled personal assistant robot. This paper presents the development of a voice-controlled robotic assistant that can perform various tasks through smartphone commands processed via cloud servers and communicated over Bluetooth. The system demonstrates promising results in remote voice control applications, with potential use cases in domestic, healthcare, and industrial settings. The emergence of lightweight speech recognition toolkits has greatly facilitated the integration of voice control into embedded systems. Specifically, open-source packages

like Vosk and Kaldi have enabled real-time voice recognition capabilities on resource-constrained hardware platforms such as NVIDIA Jetson series, smartphones, and raspberry-pis. These accessible tools have accelerated the development of voice-controlled applications across various domains, from home automation to industrial control systems, making voice interaction more prevalent in our daily lives.

B. Energy Efficient Navigation and Smooth Control

An indoor navigation system requires path planning, trajectory generation, and control to navigate a robot from a start point to a goal point, addressing concerns such as collision avoidance (safety), smoothness (comfort), and energy efficiency (economy). Planning methods can be categorized into three classes: search-based, sampling-based, and optimization-based methods. Dijkstra [10], A* [11], D* [12] are classical Grpah search-based methods to generate a global path from the start to the goal. Subsequently, Hybrid A* is proposed for autonomous driving applications, which considers vehicle kinodynamics in curve sampling and leverages pruning tricks to improve the search efficiency [13], [14]. More recently, the jump points method leveraging pruning tricks has improved the computing efficiency compared to the A* method [15], [16]. In terms of sampling-based method, PRM [17], RRT [18], [19] and its derivatives [20]–[22] are commonly used global path planning in robotics field. Polynomial-based trajectories can be generated by sampling start-end state pairs in the Frenet road system for autonomous driving [23]. While easy to be adapted to various scenarios, search-based and sampling-based methods have limits in explicitly incorporating the agent's kinodynamic constraints, and trajectory's smoothness and curvature. A bi-level or front-rear end methods, that leverage search-sampling-based methods to generate global path fast, and optimization-based methods to analytically or numerically derive an objective-dependent optimal trajectory that satisfies application constraints. Under the optimization-based branch, gradient-based planning methods leverage the distance gradient from the obstacle [24], or potential gradient in costmap [25], Euclidean Signed Distance Field (ESDF) [26], [27] to tackle the collision avoidance to the obstacles. While some other works use convex decomposition to build safe corridors for robots to navigate in multiple polytopes [16]. The smoothness of the trajectory is important for maneuverability and comfort. To enable agile UAVs to precisely track the planned trajectory, snap (second derivative of acceleration) can be minimized in piece-wise polynomials [28] to maximize the smoothness. To reduce the discomfort for a passenger in an intelligent vehicle, jerk (first derivative of acceleration) needs to be minimized in a trajectory [23]. Energy efficiency plays an important role in the economic performance of the trucking industry and intelligent vehicles [29], [30]. The energy concerns also apply to mobile robots whose energy performance will highly impact task completeness with a limited single-charge range [31].

C. Robot Intra-system Communication Interface and Protocol

Serial communication remains a cornerstone of many robotic systems, particularly due to its reliability and simplicity. Serial communication protocols such as Universal Asynchronous Receiver-Transmitter (UART), Serial Peripheral Interface (SPI), and Inter-Integrated Circuit (I2C) are extensively employed for intra-system data exchange.

UART is one of the simplest serial communication protocols, operating on a node-to-node basis without requiring a clock signal. It is often used for communication between microcontrollers and peripheral devices or for transmitting data to a central processing unit. For instance, [32] describes a customized UART-based communication protocol that was developed to transmit sensor data in an educational robotics system. However, its limitation lies in its node-to-node nature, which makes it less suitable for systems requiring multiple devices to communicate on a shared bus. SPI addresses some of UART's limitations by allowing multiple devices to communicate efficiently using a master-slave architecture. With its high-speed data transfer capabilities and full-duplex operation, SPI is well-suited for scenarios where rapid communication is essential, such as interfacing with sensors, memory modules, or actuators. Despite its advantages, SPI requires more pins and complicated hardware designs when interfacing with multiple devices. I2C uses a two-wire (SDA and SCL) architecture that supports multiple devices on the same bus, making it particularly advantageous for short-distance communication between sensors, microcontrollers, and other components in a system. In [33], an I2C-based interface was implemented to facilitate the transfer of data packets between a motherboard and various actuators in a robotic system. The design showcased I2C's ability to handle multiple devices efficiently. However, I2C's transmission speed is limited to 3.4 Mbps, which might become a bottleneck in high-data-rate applications or systems requiring extensive real-time communication.

Controller Area Network (CAN) is also widely used in robotics for its reliability in noisy environments and support for distributed control. CAN enables prioritized messages with deterministic timing, making it faster and more stable than UART and I²C in real-time applications. Its built-in error detection and correction mechanisms further enhance its robustness. [34] demonstrated a CAN-based network coordinating multiple STM32 micro-controller nodes in an autonomous robotic platform. This setup showcased CAN's ability to provide fast and stable motion control in distributed systems. Its message-based protocol allows efficient integration of multiple nodes, with each message's identifier enabling prioritization and scalability. While CAN's standard frame supports only an 8-byte payload, modern versions like CAN FD increase this limit to 64 bytes, addressing data throughput needs for advanced robotics [35]. Overall, CAN remains a key protocol for real-time, reliable communication in robotic systems, especially for modular and distributed architectures.

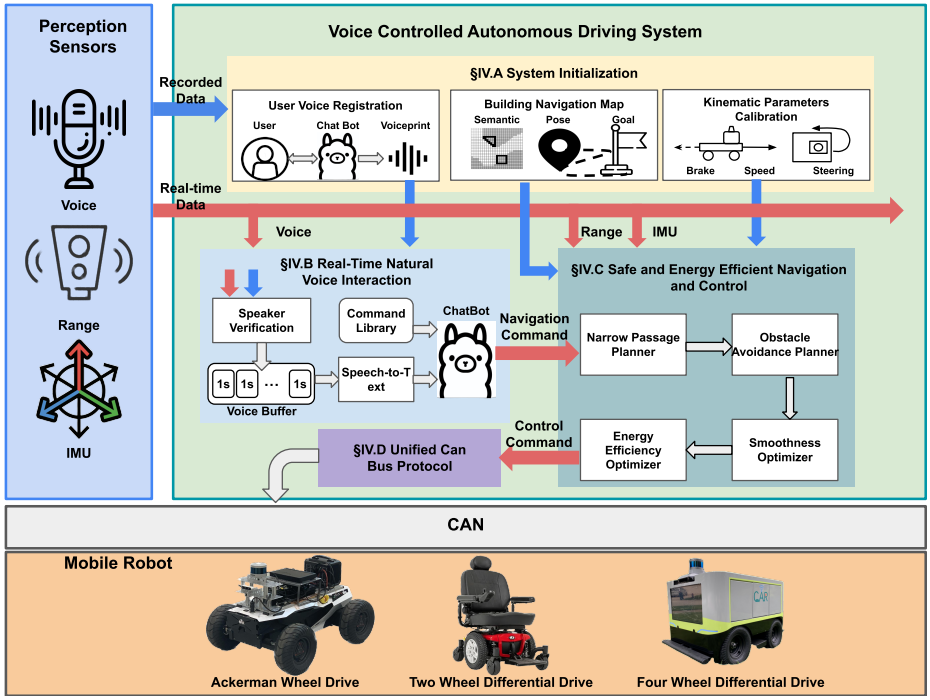


Fig. 1: VOCAR Overview. VOCAR is a kit designed to enhance mobile robots with autonomous driving capabilities and voice interaction. It is compatible with robots featuring diverse kinematic models. The system leverages a CAN network for seamless integration. The system combines a robust perception module with a voice-controlled autonomous driving framework, incorporating features such as voice verification, recognition, and safe, energy-efficient navigation in dynamic indoor environments.

III. System Overview

A. Design Goals

As a system designed to enhance robots with voice-based interaction and autonomous navigation, VOCAR adheres to the following key objectives:

Protect User Privacy and Safety. The system must safeguard user privacy by avoiding sensors or data collection mechanisms that may capture sensitive information. Additionally, it must ensure safe navigation by avoiding collisions with obstacles and maintaining smooth, predictable movements, particularly when users are near or interacting with the robot.

Enable Natural Interaction. The system must support intuitive communication through natural language commands, minimizing the learning curve. It should also ensure quick response times and include safeguards against unauthorized or unintended commands.

Optimize Energy Consumption. The system must minimize energy consumption to extend the robot's operational time while maintaining all essential functionalities.

Enhance Network Resilience. The system must remain functional even in environments with limited or unreliable network connectivity. Core functionalities must operate reliably during network shortages.

B. Design Overview

To achieve the aforementioned design goals, we propose VOCAR, a voice-controlled autonomous system for mobile robots. By integrating advanced components, VOCAR provides modular and adaptable functionality across diverse robotic platforms. The system comprises four primary design modules, as illustrated in Fig. 1.

System Initialization. During initialization (Section IV-A), VOCAR engages the authorized user in a brief conversation with a Large Language Model (LLM)-based chatbot. Audio clips recorded during this interaction are used to capture the user's voiceprints, ensuring secure access. Next, the user guides the robot through the operational environment to construct a multi-layered map containing semantic, pose, and navigation goal information. Simultaneously, the robot's kinematic data is collected to calibrate its navigation parameters. These initialization processes establish critical baseline data, ensuring that only authorized users can control the robot and providing a robust foundation for dynamic path planning in indoor environments.

Real-time Voice Command Recognition. VOCAR enables natural and intuitive interaction through real-time voice command recognition (Section IV-B). During operation, the system actively monitors environmental audio and uses voiceprint-based speaker verification to filter out unauthorized or invalid inputs. Verified audio clips are aggregated into a buffer and processed by a speech-to-text module. The resulting text commands are parsed by the LLM-based chatbot and matched against a predefined command library. This module employs

streaming voice recognition for quick responses to short commands. It integrates mechanisms to resist interference from malicious or unintended commands, ensuring secure, reliable, and user-friendly control.

Safe and Energy-Efficient Autonomous Driving. The autonomous driving module (Section IV-C) focuses on generating smooth and predictable trajectories to enhance user comfort and safety, especially when users are near or riding the robot. Upon recognizing a command, the system determines whether a narrow passage planner is required based on the semantic map. It then uses an obstacle avoidance planner to generate a safe path, which is further refined by a smoothness optimizer and an energy efficiency optimizer. This layered approach ensures robust navigation, balancing safety, efficiency, and user experience.

Unified CAN Bus Protocols. To support seamless communication across a wide range of robotic platforms, VOCAR introduces a universal CAN bus protocol (Section IV-D). This protocol standardizes interactions between the system's components and underlying hardware, ensuring compatibility and streamlining integration.

IV. Software Design

After VOCAR is deployed on a robot, it runs a SLAM algorithm to determine the current state of the robot and begins the initialization of the system (Section IV-A), which includes automatic calibration of kinematic parameters, construction of environmental maps, and user voiceprint registration. Once initialization is complete, VOCAR continuously collects voice signals from the environment and performs low-latency voice command recognition (Section IV-B). Based on the recognized commands, the autonomous driving system (Section IV-C) generates safe and smooth driving paths adapted to the robot's motion capabilities, allowing the robot to autonomously perform the tasks assigned by the user. Leveraging a unified CAN bus protocol (Section IV-D), the communication system ensures robust and adaptable interactions between VOCAR and the connected robots, enabling seamless motion control across diverse robotic platforms.

A. System Initialization

Before VOCAR can operate in its regular mode, both the autonomous driving and voice recognition modules require essential data to meet system objectives. Specifically, the voice recognition module needs to construct a voiceprint model for the user, while the autonomous driving module requires an environmental map and the robot's kinematic parameters.

Qualified User Voice Registration. To register a high-quality user voice profile that accurately captures the characteristics of the user's speech under varying tones and speaking speeds, VOCAR engages the user in interactive dialogues. These dialogues are dynamically generated by a large language model (LLM) and converted into audio using text-to-speech (TTS). To protect user privacy, the prompts for LLM are carefully designed to exclude any content related to personal or sensitive information. After collecting approximately ten minutes of audio data, VOCAR processes the recordings to extract high-quality feature vectors, known as x-vectors [36], which form the basis for speaker verification. This method ensures the creation of a robust and reliable speaker model while adhering to strict privacy and data security standards.

Construction of the Base Map. As a foundational component of autonomous driving systems, the localization map constructed during the initialization phase consists of three hierarchically aligned layers: localization, semantics, and navigation goals.

1. *Localization Layer.* In conventional robotic frameworks, the localization and semantic layers are typically combined into a unified occupancy grid that contains essential data. However, occupancy grid-based localization methods often require additional wheel odometry sensors to enhance estimation accuracy. Since not all robots are equipped with such sensors, VOCAR adopts a depth-sensor-based localization approach.

To initialize the localization layer, users manually operate the robot to traverse all regions that need to be accessed. During this process, VOCAR collects data to build an initial point cloud map. To ensure precision, VOCAR performs automatic loop closure detection and global optimization on the collected point cloud data after traversal is complete. Additionally, VOCAR allows users to manually add loop closure information to refine the map further, ensuring no misalignments that could cause localization errors remain in the final point cloud map.

2. *Semantic Layer.* The semantic layer defines navigable regions and enables the execution of path planning within the navigation system. After constructing the localization layer, VOCAR uses the same dataset to build a semantic map based on an occupancy grid. Since semantic mapping algorithms rely only on 2D depth data, VOCAR extracts 2D depth information from the 3D depth data collected during traversal.

To define navigable regions accurately, VOCAR processes the extracted 2D depth data based on the robot's physical dimensions. To avoid discrepancies between the semantic map and the localization map, VOCAR bypasses the positional estimates generated by the semantic mapping algorithm and instead retrieves position information directly from the localization layer. This approach simplifies the calibration process between the localization and semantic layers.

Beyond defining navigable regions, the semantic layer also includes narrow passage information derived from the localization map. This ensures that even constrained spaces are accurately represented, facilitating more robust path planning.

3. *Navigation Goal Layer.* The navigation goal layer stores locations that the robot may be commanded to navigate to. VOCAR provides two methods for adding these destination points: **1) Real-time Method:** Users manually operate the robot to reach a desired destination and add it via a voice command. Since these destinations are defined using the preliminary point cloud map, VOCAR recalculates coordinates after map optimization to ensure positional accuracy. **2) Offline Method:** Users can add destination points through

VOCAR's remote management system, using the semantic layer map as a reference.

Kinematic Parameter Calibration. Autonomous driving systems, after fine-tuning, are capable of planning smooth and efficient driving trajectories. However, to ensure that the robot can accurately follow these trajectories, VOCAR determines the kinematic constraints of the robot during the initialization phase. These constraints, such as maximum linear velocity, angular velocity, and deceleration limits, are derived from motion data collected during the mapping process, which reflects the robot's actual operational capabilities.

During initialization, VOCAR records the linear velocity (v) and angular velocity (ω) computed by the SLAM algorithm, as well as the acceleration (a) measured by the IMU. These data points enable VOCAR to establish key kinematic constraints:

- *Maximum Linear Velocity (v_{max}):* The highest linear velocity recorded during traversal, representing the robot's safe speed for straight-line motion.
- *Maximum Angular Velocity (ω_{max}):* The highest angular velocity observed, defining the robot's rotational speed limits.
- *Maximum Deceleration (a_{min}):* The steepest deceleration slope measured by the IMU, which determines the minimum stopping distance:

$$d_{min} = \frac{v^2}{2|a_{min}|}.$$

- *Minimum Turning Radius (R_{min}):* Calculated as:

$$R_{min} = \frac{v_{max}}{\omega_{max}},$$

reflecting the robot's ability to navigate tight curves.

By establishing these kinematic constraints during initialization, VOCAR ensures that the planned trajectories remain within the robot's physical limits. This guarantees safe and predictable motion while maximizing trajectory adherence in autonomous navigation.

B. Real-Time and Natural Voice Interaction

Traditional robots are typically operated using physical controller. A controller can authenticate operators through methods such as password protection and allow user to precisely operate the robot with no latency. However, controllers have notable drawbacks: they require users to learn specific control schemes and are generally not portable. To facilitate more convenient interaction between users and robots, VOCAR adopts voice input as the primary control method. Voice-based interaction provides a natural way for users to communicate with the robot.

Speaker Verification. This module is designed to identify authorized users before interpreting commands. Due to the requirement of recognizing short commands, such as stop and back, VOCAR validates the input voice every 1 second clip. Upon detecting voice activity using a low-latency Voice Activity Detection (VAD) algorithm, the system extracts speaker-specific embeddings, known as x-vectors, from the active speech. These embeddings, computed by a speaker encoder, represent the unique characteristics of speakers. For authentication, the extracted x-vector is compared with stored embeddings of authorized users using cosine similarity:

$$\text{Similarity Score} = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|} \tag{1}$$

where $\mathbf{x}_1$ represents the x-vector of the detected speech, and $\mathbf{x}_2$ is the pre-registered x-vector of an authorized user. A predefined threshold is applied to the similarity score to determine whether the speaker is authorized. If the threshold is exceeded, the input clip is buffered for command interpretation.

Natural Language Command Interpretation. This module is responsible for interpreting natural language commands from users to control the robot. It combines speech-to-text processing and a large language model (LLM) to understand user intentions and translate them into actionable commands. The system processes user inputs iteratively, accumulating text data until a complete command is identified or deemed irrelevant.

Algorithm 1 Natural Language Command Interpretation Workflow

Require: Verified 1-second audio segment a_i, predefined command list C, accumulated text buffer $T \leftarrow \emptyset$
Ensure: Executed command or reset buffer

```
while Verified audio stream a_i is active do
    Convert a_i to text t_i using SpeechToText(a_i)
    Append t_i to the buffer: T ← T ∪ t_i
    Query the Large Language Model (LLM) with accumulated text T: matched_command ← LLM(T, C)
    if matched_command ≠ ∅ then
        if Required parameters for matched_command are complete then
            Execute matched_command
            Clear text buffer T ← ∅
        else
            Continue accumulating text
        end if
    else
        if No valid user command is detected in T then
            Clear text buffer T ← ∅
        end if
    end if
end while
```

As described in Algorithm 1, the workflow begins with speech-to-text conversion, where the system transforms verified audio segments into textual data. This text is concatenated with previously accumulated text to construct a more complete input for the LLM. The LLM processes this input and matches it against a predefined command list to determine if the current text corresponds to a valid command. If a match is found and the required parameters are complete, the command is issued, and the accumulated text buffer is cleared. If no match exists and the text indicates that the user is not addressing the

robot, the system also clears the buffer to reset for the next interaction. The predefined command list is divided into two categories as shown in Table I

TABLE I: Examples of Real-time and Complex Commands in VOCAR System

Real-time Commands	Complex Commands
"Stop"	"Navigate to the kitchen"
"Slow down"	"Find the nearest charging station"
"Turn left"	"Go to the living room and wait for me"
"Turn right"	"Locate Bob in the meeting room"
"Move forward"	"Take me to the second floor"
"Accelerate"	"Navigate to room 105"

This design ensures robust and secure voice interaction by combining speaker verification and natural language understanding. The use of short utterance speaker verification guarantees that only commands from authorized users are processed, enhancing system security. Simultaneously, the natural language command interpretation dynamically accumulates text to enable accurate recognition of both short and long user commands. By leveraging real-time audio processing and contextual understanding, the module maintains high responsiveness while accurately translating user intentions into actionable commands.

C. Safe and Energy Efficient Navigation and Control

Autonomous navigation systems face challenges ranging from smooth, energy-efficient path planning in normal scenarios to safe navigation in narrow spaces. While effective safety path planning solutions exist, VOCAR builds upon these foundations to enhance trajectory smoothness and minimize energy consumption by integrating dynamic planning mechanisms. Furthermore, it incorporates a specialized module for navigating constrained environments, leveraging environment mapping, precise contour adaptation, and reinforcement learning-based control.

Improve Smoothness and Energy Efficiency through Trajectory Planning. Trajectory planning for mobile robots is crucial for achieving safe and efficient navigation in dynamic and constrained environments. The process involves optimizing multiple criteria, including smoothness and energy efficiency. These objectives are mathematically formulated as a trajectory optimization problem, defined as:

$$\begin{aligned} \min_{(\mathbf{z}(t),\mathbf{u}(t))} \quad & J(\mathbf{z}(t),\mathbf{u}(t),[t_0,T]) \\ \text{s.t.} \quad & \dot{\mathbf{z}}(t) = f_z(\mathbf{z}(t),\mathbf{u}(t)), \\ & f_p(\mathbf{z}(t)) \not\subset \mathcal{W}_{\text{int}}, \forall t \in [t_0,T]; \\ & g_{\text{init}}(\mathbf{z}(t_0)) = 0, \\ & \underline{g}_t \le g_{\text{itm}}(\mathbf{z},\mathbf{u}) \le \overline{g}_t, \\ & \underline{g}_T \le g_{\text{end}}(\mathbf{z}(T)) \le \overline{g}_T. \end{aligned} \tag{2}$$

Here, $\mathbf{z}(t)$ represents the state variables, $\mathbf{u}(t)$ the control inputs, and f_z the state transition function. The objective function J integrates criteria for smoothness and energy efficiency over the planning horizon $[t_0, T]$:

$$J = \int_{t_0}^{T} \Big(\underbrace{\|d_o(t), v(t) - v_d, a(t), j(t)\|^2_{2,\mathbf{w}_g}}_{\text{Smoothness Objectives}} + \underbrace{w_e \cdot f_e(t)}_{\text{Energy Efficiency}} \Big) dt, \tag{3}$$

where $\mathbf{w}_g = [w_o, w_v, w_a, w_j]$ assigns weights to obstacle proximity d_o, velocity tracking $v(t) - v_d$, acceleration $a(t)$, and jerk $j(t)$. The energy efficiency term is weighted by w_e. The combined weight vector is $\mathbf{w} = [\mathbf{w}_g, w_e]$.

1. *Smoothness Optimization.* Smoothness ensures comfortable and safe trajectories by minimizing abrupt changes in motion. This is particularly critical for passenger comfort and system reliability. Smoothness is achieved by:

- Including acceleration $a(t)$ and jerk $j(t)$ in the objective function.
- Assigning weights w_a and w_j within $\mathbf{w}_g$ to penalize large values of $a(t)$ and $j(t)$.
- Directly embedding these terms into the cost function J, ensuring optimization over the entire planning horizon.

Minimizing acceleration and jerk reduces sudden changes in speed and motion, resulting in smoother trajectories that improve passenger comfort and reduce mechanical stress on the robot's hardware. This improves ride comfort and prevents mechanical stress on the robot's hardware.

2. *Energy Efficiency Optimization.* Energy efficiency reduces power consumption, which is essential for extending the operational time of autonomous driving [37], [38] and mobile robots [31], especially those with limited energy resources. Energy efficiency is addressed by:

- Introducing a dedicated energy term $f_e(t)$ in the objective function J.
- Weighting this term by w_e, which allows prioritization of energy optimization.
- Coupling the control inputs $\mathbf{u}(t)$ with the state transition model f_z, ensuring that optimized trajectories inherently consume less energy.

The inclusion of $f_e(t)$ in the objective function ensures that the trajectory planning explicitly considers energy consumption at every time step. By penalizing high energy usage through the weight w_e, the optimization process prioritizes trajectories that minimize unnecessary energy expenditure, such as rapid accelerations or inefficient motion patterns. Additionally, coupling $f_e(t)$ with the system dynamics f_z ensures that energy-efficient behavior aligns with the physical constraints of the robot, resulting in a practical and efficient implementation.

By carefully designing the objective function J and tuning the weight vector $\mathbf{w}$, the trajectory planning process of VOCAR achieves a balanced optimization of smoothness and energy efficiency, ensuring that the planned trajectories are safe, comfortable, and resource-efficient, making it suitable for enhancing real-world robotic applications.

Narrow Passage Navigation. Navigating narrow spaces, such as corridors, doorways, or tightly packed environments, is a

fundamental challenge in robot navigation as Fig. 2 shows. The obstacle avoidance planner will refuse to navigate through the doorway because the wall is too close to the robot. VOCAR addresses this problem through three aspects, map, robot contour and control.

Fig. 2: A robot fails to pass indoor narrow passage

1. *Mapping Narrow Spaces.* During the initialization phase, VOCAR identifies and maps narrow regions in the robot's operational environment. By analyzing the environment map, it detects constrained areas and generate a safe corridor for navigation. This mapping process ensures that the robot is aware of spatial limitations, enabling subsequent planning to focus on feasible paths within these regions. Awareness of these constraints lays the foundation for effective navigation in tightly constrained spaces.

2. *Precise Contour Adaptation.* Once narrow regions are identified, VOCAR plans trajectories that closely follow a tight contour around the robot's body to minimize unnecessary clearance. This is achieved by dynamically adapting the planned path to fit the robot's precise shape, using methods such as those described in [39]. By maximizing space utilization while avoiding collisions, this contour-based approach reduces the risk of collision and ensures the robot can traverse narrow passages effectively without compromising safety.

3. *Reinforcement Learning-Based Control.* To handle complex and dynamic narrow spaces, VOCAR employs a deep reinforcement learning (DRL) algorithm [40], [41] to enable adaptive and precise control. The system directly maps sensor inputs to control commands, allowing the robot to learn and execute complex maneuvers, such as sharp 90-degree turns. This capability ensures smooth and safe navigation in highly constrained environments by adapting to real-time conditions and executing precise actions tailored to the surrounding context.

By combining mapping, contour learning, and reinforcement learning-based control, VOCAR's narrow passage navigation module ensures safe, efficient, and adaptive navigation in tightly constrained environments. This hierarchical approach addresses the challenges of narrow spaces comprehensively, balancing planning precision with real-time control adaptability.

D. Unified CAN Bus Protocol (Uni-Bus)

The diversity of communication protocols in mobile robotics presents significant challenges. The voice-controlled box prioritizes the use of the CAN bus for communication due to its robustness, flexibility, and ability to meet the demanding requirements of mobile robotics platforms. Unlike other protocols such as UART, SPI, or I2C, which are constrained by limitations like point-to-point communication, short distances, or lack of robust error handling, the CAN bus supports deterministic timing, fault tolerance, and multi-node communication, making it ideal for stable and scalable interactions between the Jetson Orin Nano and the robot's microcontroller.

- *Unified Design for Compliant Robots:* To standardize communication across diverse mobile robots, the proposed CAN bus design defines consistent destinations for key commands, such as brake, forward, and steering. This ensures that robotic systems adhering to the unified design can communicate seamlessly with the voice-controlled box, enabling uniform motion control.
- *Adapting to Non-Compliant Systems:* For robots that do not follow the unified CAN bus design, a CAN bus hacking approach is employed. Inspired by methods used in automotive applications like cabana [42], this involves reverse-engineering the robot's proprietary CAN messages. By mapping these proprietary commands to the box's standardized format, the system achieves compatibility with non-compliant robots.

TABLE II: Node ID and CAN ID Allocation for Motion Control

Function	**Node ID**	**CAN ID**
Servo Motor	1	0x182
Left Drive Motor	2	0x183
Right Drive Motor	3	0x184
Front Left Motor	4	0x185
Front Right Motor	5	0x186
Rear Left Motor	6	0x187
Rear Right Motor	7	0x188

TABLE III: PDO Mapping for Motion Control Devices

PDO Type	**Index**	**Sub-Index**	**Length**	**Description**
TPDO	0x6064	0x00	32	motor actual position
TPDO	0x606C	0x00	16	motor actual velocity
RPDO	0x607A	0x00	32	motor target positioin
RPDO	0x60FF	0x00	16	motor target velocity

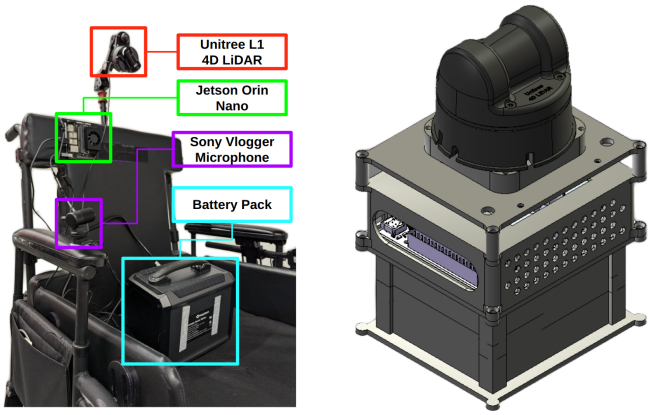
The CAN bus is widely adopted across various industries due to its flexibility, but this flexibility also introduces variability in implementations. For instance, the same data packet can carry different meanings depending on the CAN header and destination configurations, which vary across brands and systems. A unified design ensures consistency and ease of integration for compliant robots, while the reverse-engineering approach extends compatibility to non-compliant systems.

We propose to use CAN bus's variant CANopen as the baseline for our goal. By combining a structured Node ID and CAN ID allocation with an optimized PDO (Process Data Object) mapping, the design supports a wide range of configurations, including 2-wheel robots, 4-wheel robots, and wheelchairs. At Table II each actuator or motor is assigned a

unique Node ID, and its corresponding CAN ID is calculated as 0x181 + NodeID, ensuring a predictable and scalable addressing scheme. As shown in Table III, the PDO mapping enables real-time exchange of critical process data such as motor position, velocity, and control commands, reducing bus traffic and ensuring deterministic communication. The design is future-proof, accommodating reserved Object Dictionary ranges for custom extensions. The Node ID allocation and PDO mapping tables below outline the addressing scheme and real-time data configuration for motion control devices.

V. Implementation

We implemented a prototype VOCAR system, as shown in Fig. 3a. The entire software stack is built on ROS2 (Robot Operating System 2), a middleware framework that supports modularity, scalability, and real-time performance. The primary hardware platform, the NVIDIA Jetson Orin Nano, delivers the computational power for real-time processing. Its specifications include a 6-core ARM Cortex-A78AE CPU, a 1024-core Ampere GPU, and 8 GB of LPDDR5 RAM, with a configurable power envelope of 7–15W. The robot platform itself is a differential-drive power wheelchair, integrating LiDAR and IMU sensors to enhance environmental perception. The prototype is accompanied by a conceptual design of VOCAR Fig. 3b, which integrates essential hardware into a compact box for streamlined installation.



(a) Wheelchair Prototype (b) Concept Design

Fig. 3: Illustration of the VOCAR prototypes. (a) A prototype implementation on a wheelchair platform, showcasing the integration of sensors and computational units for real-world testing. (b) A conceptual design integrating essential hardware into a compact box, enabling streamlined deployment for diverse mobile robot platforms.

To enable voice-based control, VOCAR integrates a locally deployed Meta-Llama-3-8B-Instruct language model [43], chosen for its balance between capability and computational efficiency. Deploying the model on an NVIDIA Jetson Orin Nano removes the need for network connectivity, enabling secure offline operation. The language model is configured with a tailored prompt that encodes predefined control commands, allowing the system to interpret and respond to spoken instructions reliably. We validated a total of 120 natural commands from those listed in Table I. Without optimization, the system achieved an average task execution time of approximately 1.5 seconds, with an overall accuracy of 91.67%.

During initialization, users interact with the chatbot using a Sony Vlogger microphone, which captures high-quality, noise-cancelled audio. The recorded audio is processed to extract x-vectors using the pre-trained Pyannote model [44], creating unique voiceprints for each user. The voiceprints are securely stored for speaker verification, ensuring that only authorized users can interact with the robot.

Simultaneously, the system generates a multi-layered map of the environment to support navigation as shown in Fig. 4. Using hdl-graph-SLAM [45] and interactive SLAM [46], localization maps are constructed from data collected by a Unitree L1 LiDAR and IMU sensors. The semantic layer, built using Cartographer [47], includes manual annotations for navigation goals and narrow passages. This mapping process ensures that VOCAR operates effectively in dynamic indoor environments, balancing autonomy with adaptability.

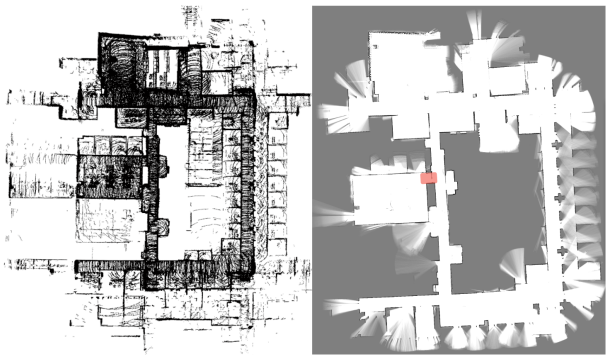


Fig. 4: Map representation of the operational environment. Left: The point cloud map generated during initialization, providing detailed spatial information for localization and navigation. Right: The semantic map, which overlays navigation goals and identifies critical areas, such as the narrow passage highlighted in red for specialized planning and control.

During operation, VOCAR continuously monitors environmental audio, verifying each spoken command through voiceprint analysis using cosine similarity 1 of x-vectors. Once verified, audio is converted to text using the vosk-model-small-en-us-0.15, a lightweight speech-to-text engine optimized for low-latency performance. The text is then parsed by the chatbot, which references an embedded command list to interpret the user's intent. This pipeline ensures secure, reliable, and real-time recognition of voice commands, even in noisy environments. By filtering commands through x-vector authentication before forwarding them to the chatbot, VOCAR further safeguards against unauthorized or malicious inputs.

For navigation, VOCAR employs a layered approach to path planning that prioritizes safety, energy efficiency, and smooth-

ness. The system uses the TEB planner for obstacle avoidance, dynamically adapting trajectories to environmental changes. In narrow spaces, the system switches to a pure pursuit strategy, ensuring precise control over the robot's movements. This modular design ensures robust performance across diverse operational scenarios. While details of energy and smoothness optimizations are covered in the system design, these features are integrated seamlessly into the planning process.

VOCAR implements a unified CAN bus protocol to ensure compatibility across a wide range of robotic platforms. This protocol standardizes communication between the software stack and the underlying hardware, allowing the system to control the wheelchair platform used for testing. Communication with the wheelchair is established via a CAN2RNet interface, ensuring reliable data exchange.

VI. Summary and Future Work

A. Summary

This work proposes VOCAR, a voice-controlled autonomous system designed to ensure secure and adaptable functionality for mobile robots. The system integrates several advanced components: a locally deployed LLM-based chatbot for natural language interaction, a voiceprint-based authentication mechanism for secure access, and a multi-layered mapping framework for precise indoor navigation. We implemented the system on a differential-drive power wheelchair. The implementation utilized the *NVIDIA Jetson Orin Nano* and the *ROS2* framework to enable real-time processing. The system's key modules—voice command recognition, safe and energy-efficient path planning, and universal hardware compatibility via a CAN bus protocol—were successfully integrated.

B. Future Work

Although the current implementation of VOCAR demonstrates the feasibility of the design, there are several areas for further evaluation:

- **Comprehensive Testing:** Systematic evaluations will be conducted to assess the system's performance under real-world conditions, including noisy environments and diverse indoor layouts. Metrics such as command recognition accuracy, latency, and navigation success rates will be measured.
- **Scalability and Generalization:** Future work will explore the integration of additional robotic platforms and environments to evaluate the generalization of the universal CAN bus protocol and mapping capabilities.
- **Dynamic Environment Handling:** Enhancements to the mapping and navigation modules will focus on real-time dynamic obstacle detection and avoidance, increasing the robot's adaptability to changing environments.
- **User Experience Improvements:** User studies will be conducted to refine the voice interaction pipeline, ensuring more natural and responsive communication.
- **Implementation of Concept Design:** Efforts will be directed toward realizing the conceptual design of VOCAR, consolidating all essential hardware into a single compact box. This streamlined implementation will simplify installation and enhance portability across various robotic platforms.

By addressing these areas, VOCAR can evolve into a more robust, scalable, and user-centric solution, paving the way for broader adoption of voice-controlled autonomous systems in indoor robotics.

References

[1] P. Kaur, Q. Wang, and W. Shi, "Fall detection from audios with audio transformers," 08 2022.

[2] S. Bedaf, P. Marti, F. Amirabdollahian, and L. de Witte, "A multi-perspective evaluation of a service robot for seniors: the voice of different stakeholders," *Disability and rehabilitation: assistive technology*, vol. 13, no. 6, pp. 592–599, 2018.

[3] U. Reiser, T. Jacobs, G. Arbeiter, C. Parlitz, and K. Dautenhahn, "Care-o-bot® 3–vision of a robot butler," in *Your Virtual Butler: The Making-of*. Springer, 2013, pp. 97–116.

[4] N. Mitsunaga, T. Miyashita, H. Ishiguro, K. Kogure, and N. Hagita, "Robovie-iv: A communication robot interacting with people daily in an office," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5066–5072.

[5] A. Mishra, P. Makula, A. Kumar, K. Karan, and V. Mittal, "A voice-controlled personal assistant robot," in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. IEEE, 2015, pp. 523–528.

[6] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*. PMLR, 2016, pp. 173–182.

[7] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane, "The capio 2017 conversational speech recognition system," *arXiv preprint arXiv:1801.00059*, 2017.

[8] W. Liu, Y. Shi, J. Chen, W. Rao, S. He, A. Li, Y. Wang, and Z. Wu, "Gesper: A restoration-enhancement framework for general speech reconstruction," *arXiv preprint arXiv:2306.08454*, 2023.

[9] J. Chen, W. Rao, Z. Wang, J. Lin, Y. Ju, S. He, Y. Wang, and Z. Wu, "Mc-spex: Towards effective speaker extraction with multi-scale interfusion and conditional speaker modulation," *arXiv preprint arXiv:2306.16250*, 2023.

[10] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: his life, work, and legacy*, 2022, pp. 287–290.

[11] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[12] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE international conference on robotics and automation*. IEEE, 1994, pp. 3310–3317.

[13] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[14] ——, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[15] D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 25, no. 1, 2011, pp. 1114–1119.

[16] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[17] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[18] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[19] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan," *Algorithmic and computational robotics*, pp. 303–307, 2001.

[20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[21] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2014, pp. 2997–3004.

[22] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[23] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 987–993.

[24] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.

[25] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[26] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[27] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[28] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[29] L. Liu, W. Li, D. Wang, Y. Wu, R. Yang, and W. Shi, "Fuel rate prediction for heavy-duty trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8222–8235, 2023.

[30] Z. Tian, L. Liu, and W. Shi, "A pulse-and-glide-driven adaptive cruise control system for electric vehicle," *International Transactions on Electrical Energy Systems*, vol. 31, no. 11, p. e13054, 2021.

[31] L. Liu, R. Zhong, A. Willcock, N. Fisher, and W. Shi, "An open approach to energy-efficient autonomous mobile robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 569–11 575.

[32] E. Binugroho, E. S. Ningrum, and A. R. A. Besari, "Design of communication protocol based on 9 bit uart on adroit education robot," in *International Electronics Symposium*, 2014.

[33] A. F. Ribeiro, P. Silva, I. Moutinho, V. Silva, and N. Pereira, "Optimization of fast moving robots and implementation of i2c protocol to control electronic devices," 2005.

[34] D. Pan, Q. Gao, P. Zhao, J. Zeng, P. Xu, and H. Xiang, "Design and test of a distributed control system of weeding robot based on multi-stm32 and can bus," in *Journal of Physics: Conference Series*, vol. 2203, no. 1. IOP Publishing, 2022, p. 012019.

[35] Z. Lin, T. Wang, Q. Gao, and Y. Liu, "Design of robot platform based on can bus," in *2011 International Conference on Electrical and Control Engineering*, 2011.

[36] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," 04 2018, pp. 5329–5333.

[37] Z. Tian, L. Xia, and W. Shi, "Emato: Energy-model-aware trajectory optimization for autonomous driving," *arXiv preprint arXiv:2412.08830*, 2024.

[38] ——, "Slope considered online nonlinear trajectory planning with differential energy model for autonomous driving," *arXiv preprint arXiv:2412.09424*, 2024.

[39] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Toward agile maneuvers in highly constrained spaces: Learning from hallucination," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1503–1510, 2021.

[40] Z. Tian and W. Shi, "Design and implement an enhanced simulator for autonomous delivery robot," in *2022 Fifth International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 2022, pp. 21–29.

[41] Z. Tian, Z. Liu, X. Zhou, and W. Shi, "Unguided self-exploration in narrow spaces with safety region enhanced reinforcement learning for ackermann-steering robots," in *2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*. IEEE, 2024, pp. 260–268.

[42] commaai. (2018) cabana:tool developed to view raw can data. [Online]. Available: https://github.com/commaai/cabana

[43] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 02 2023.

[44] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Proc. INTERSPEECH 2023*, 2023.

[45] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, 04 2019.

[46] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3d graph slam for map correction," *IEEE Robotics and Automation Letters*, vol. 6, pp. 40–47, 01 2021.

[47] S. Sen, B. Hecht, A. Swoap, Q. Li, B. Boatman, I. Dippenaar, R. Gold, M. Ngo, S. Pujol, and B. Jackson, "Cartograph: Unlocking spatial visualization through semantic enhancement," 03 2017, pp. 179–190.