Unblocking the Path: *VLM*-Assisted Robot Navigation in Indoor Environments

Anonymous for submission

***Abstract*—Autonomous navigation is a fundamental capability for mobile robots, yet traditional methods largely treat the robot as a passive agent that follows preplanned paths while avoiding obstacles. Such approaches are effective in structured environments but fall short in human-centric indoor spaces where progress often requires active interaction, such as opening doors or using elevators. Large foundation models, and in particular vision-language models (*VLM*s), offer a new opportunity to address these challenges by combining scene understanding with high-level reasoning.**

In this work, we develop a navigation system that integrates classical geometric planning with *VLM*-based reasoning to enable robots to actively resolve situations where passive path following would fail. When blocked, the robot queries a *VLM* with sensory inputs and task context to determine what is preventing progress and how to overcome it. To ensure successful localization of interaction objects, such as door buttons, we couple the *VLM* with an open-vocabulary detector that grounds language-based reasoning into concrete visual cues. We implement this system on a real robot and evaluate it through proof-of-concept experiments, demonstrating the potential of *VLM*-assisted navigation for unblocking tasks in complex indoor environments.

I. Introduction

Autonomous navigation is a fundamental capability for mobile robots, supporting a wide range of real-world applications. Classical navigation pipelines typically build or access an explicit geometric map, plan a path to the goal, and execute the plan while avoiding obstacles. In parallel, end-to-end learning approaches have been explored to directly map sensory inputs to control commands without explicit maps. While both families of methods have achieved considerable success, they generally assume that the robot is a passive agent whose progress depends on map accuracy or the generalization of trained policies. As a result, when critical passages are blocked or when accessible space is limited, these methods may fail to complete the task, as illustrated in Fig. 1.

Such failures highlight a broader limitation: conventional navigation excels in open or structured environments but struggles in human-centric indoor spaces. Multi-story buildings, diverse floorplans, and dynamic infrastructure introduce challenges that cannot be resolved by simply replanning around obstacles. For instance, a closed door blocks passage until it is actively opened, and reaching a goal on another floor may require recognizing and operating an elevator. These scenarios show that effective indoor navigation often requires not only geometric reasoning but also deliberate interaction with the environment—capabilities that remain underexplored in existing systems.

Large foundational models have recently demonstrated impressive multimodal reasoning abilities, combining vision, language, and other modalities to generate semantically meaningful guidance. Among them, vision–language models (*VLM*s) are particularly relevant for robotics, as they can jointly interpret visual inputs and contextual prompts to produce actionable suggestions. Such models enable a robot not only to identify objects in complex scenes but also to infer how those objects can be used to overcome blockages. Rather than relying on hardcoded heuristics or narrowly trained policies, the robot can flexibly query a *VLM*, making it better suited for navigation in diverse and previously unseen environments.

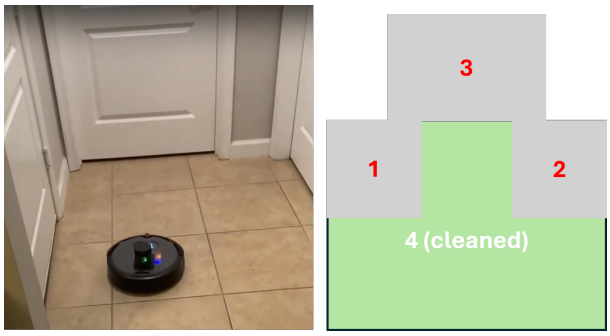


Fig. 1: A vacuum cleaner can still leaves three room uncleaned (marked in red) due to blockage even if it has tried all reachable space.

In this work, we develop a navigation system that enables robots to actively resolve situations where passive path following would fail, such as being stopped by a closed door or needing to reach a goal on another floor. When the robot becomes blocked, it queries a *VLM* with sensory observations and task context to determine what is blocking progress and how to overcome it. Current general-purpose *VLM*s often succeed in reasoning about suitable unblocking strategies but fail to directly localize the relevant interaction objects, such as door buttons. To bridge this gap, our system integrates the *VLM* with an open-vocabulary detector that accepts detection requirements from the language model, thereby grounding high-level reasoning into concrete visual cues. The resulting system is implemented on a real robot and evaluated through proof-of-concept experiments. The contributions of this paper are summarized as follows:

- We enable mobile robots to navigate in complex indoor environments while actively unblocking themselves by leveraging *VLM*-based reasoning for interaction decisions.
- We ensure successful localization of interaction objects

by integrating a general-purpose *VLM* with an open-vocabulary detector that accepts detection requirements from the language model.
- We implement and validate the proposed navigation system on a real robot, demonstrating proof-of-concept success in tasks that require active interaction with the environment.

II. Related Work

Planning safe and efficient paths and controlling mobile robots to follow them has long been a central problem in autonomous robotics. The safety of navigation is affected by the robot's current state and dynamics, as well as constraints imposed by the environment. Simultaneous Localization and Mapping (SLAM) methods [1], [2] are widely used to build environment maps so that the robot can localize itself and identify obstacles. Given such maps, path planning algorithms including Dijkstra, A^*, and Rapidly-Exploring Random Trees (RRT) [3], [4] can generate collision-free paths toward goal locations. Accurate robot dynamics models are then needed to compute control inputs for following the planned path. Recent work has explored data-driven dynamics learning with Gaussian processes [5]–[7] and neural networks [8], [9]. Safety has also been incorporated through model predictive control with constraints [10], [11], often solved via time discretization and system linearization [12], [13], or through control barrier functions [14], [15]. Motion constraints such as minimum turning radius must also be considered during trajectory generation [16]–[18]. While many of these approaches focus on global path planning, they can also be adapted for local replanning when unexpected obstacles appear. Once a path is determined, a variety of tracking algorithms can be used to execute it robustly [19].

Large foundational models have recently been applied to robot navigation to provide semantic reasoning beyond geometric planning. Vision-language models have been used to construct action-aware costmaps for traversable obstacles, enabling navigation through elements such as curtains without additional training [20]. Other work has investigated privacy-aware path planning, where *VLM*s guide robots to consider social and privacy constraints in office environments [21]. Large models have also been integrated into semantic mapping and reasoning pipelines to achieve zero-shot object navigation toward targets specified in natural language [22]. Beyond navigation targets, *VLM*s have been applied to improve scene understanding and object manipulation through zero-shot visual grounding [23], and to guide navigation in human-centric environments by reasoning about social cues and implicit language input [24]. More recently, multimodal approaches such as OpenNav [25] extend large models to open-world navigation, translating rich scene understanding into executable plans, while Vi-LAD [26] demonstrates that distilling attention from vision-language models improves socially aware motion planning in dynamic environments. These works illustrate the potential of large models to support higher-level reasoning in navigation. In this paper, we emphasizes *active unblocking*, where the robot couples *VLM*-based reasoning with open-vocabulary detection and physical interaction to restore progress when passive planning fails.

III. Methodology

In this section, we present our method for enabling robots to navigate in complex indoor environments while actively unblocking themselves. The system combines classical path planning, *VLM*-based reasoning, and open-vocabulary object detection to decide when and how to interact with the environment. An overview of the method is shown in Fig. 2. We begin by describing the navigation stack used to plan initial paths, then introduce the *VLM* reasoning process for blockage handling, followed by the integration with open-vocabulary detection for localizing interaction objects, and finally outline how the robot executes interaction actions and continues navigation.

A. Global Mapping and Path Planning

The robot is provided with a floorplan or a map of the environment, which enables it to plan an initial path to a given goal location. In our implementation, we adopt the ROS2 Navigation Stack [27], [28] to handle localization, map representation, and path planning. This provides the robot with a collision-free path under the assumption that obstacles are either absent or traversable. The map is constructed when the space is connected (e.g., doors are open) to facilitate the robot's self-exploration, which may not hold in subsequent navigation tasks. As a result, the planned path may become infeasible when the robot later encounters closed doors or other barriers, motivating the need for reasoning and interaction capabilities described in the following subsections. Fig. 3 shows such a case.

B. VLM-*Based Reasoning for Blockage Handling*

The robot is equipped with onboard LiDAR and other sensors that allow it to detect obstacles along the planned path. When such obstacles are encountered, for instance a closed door that was marked as open during mapping, the traditional navigation stack can only attempt to replan around them. However, a feasible detour may not always exist, especially in single-access rooms, corridor entrances, or exits like Fig. 3. In these situations, the robot requires higher-level reasoning to determine how to actively interact with the environment to continue its mission.

To address this challenge, we incorporate a vision-language model (*VLM*) as a semantic reasoning module. In this work, we tested with *GPT-4o* and *GPT-5* models. When the robot detects that its progress toward the goal is blocked, it captures an image of the scene and provides it to the *VLM* together with a contextual prompt for blockage analysis. The *VLM* processes this input to generate an unblocking suggestion, allowing the robot to move beyond passive replanning and actively restore navigability. While the approach can, in principle, generalize to various types of blockages, in this work we focus on closed doors that require pressing a button to continue navigation.

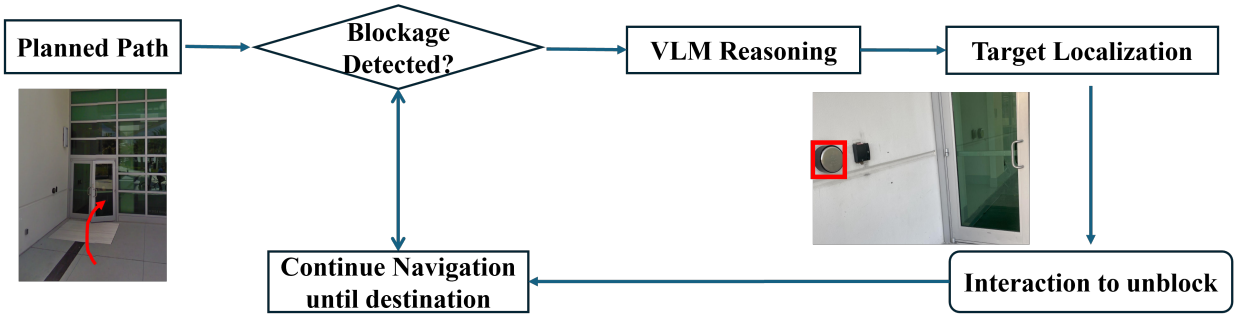
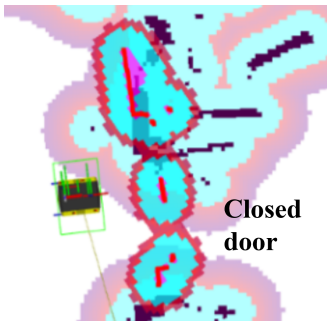


Fig. 2: Overview of the proposed method. The robot follows a planned path, invokes *VLM*-based reasoning when blocked, uses CLIP-based detection to localize interaction objects, performs the required action, and continues navigation.

(a) Image view of the navigation scene, showing the robot's goal and the closed door along the path.



(b) Visualization in RViz with the map, sensor data, and the planned path passing through the closed door.

Fig. 3: Example of path blockage during navigation. The path planned guides the robot to a closed door ($left$) and trys to go through it. The $right$ image presents a map view. No alternative path exists on the same floor, requiring the robot to actively handle the blockage.

Fig. 4: Examples of images provided to the *VLM* for blockage analysis. The two images are captured at different distances from the door, approximately 1.5 m and 0.5 m, demonstrating how the input viewpoint can be adjusted.

In addition, we heuristically set the distance from which the image is taken when querying the *VLM* for analysis. Figure 4 shows two examples: one image captured (*left*) from approximately 1.5 m away and another *right* from about 0.5 m, illustrating how input scale can be adjusted by our method.

To structure the interaction with the *VLM*, we adopt a model-context-protocol (*MCP*) design. This approach improves reliability by explicitly encoding information about the navigation task, the robot's physical capabilities, and the desired output format. As summarized in Table I, the protocol specifies general navigation instructions, enumerates feasible robot actions, defines the expected analysis of scene images, and constrains the output to a machine-readable format.

TABLE I: Model Context Protocol (MCP) prompt structure used for *VLM*-based blockage handling.

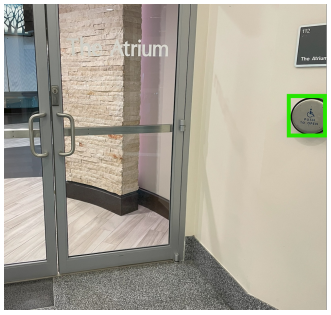
Context Element	Example Content
General Instructions	You are assisting a ground robot performing an autonomous navigation mission. The robot's path may be blocked because its planning map assumes certain passes or checkpoints are always open.
Robot Capabilities	Equipped with a manipulator; can push or press/click a button; not strong enough to rotate handles or push open heavy doors.
Task Instruction	When an image is provided, analyze it and return: • Obstacle Identification — what is blocking the path • Interaction Opportunities — objects that can clear the blockage • Action Recommendation — push / press / none
Output Format	Return fields for Obstacle, Interaction Object, and Recommended Action. Generate an annotated image with bounding boxes as a downloadable file link.

During deployment, the robot simply issues the runtime query 'unblock me' together with the scene image, while the MCP context ensures that the *VLM* consistently interprets the request and produces actionable guidance.

C. Integration with Open-Vocabulary Detection

When performing blockage handling with recent *VLM*s, including GPT-4o and GPT-5, we observed that these models are capable of perceiving the environment and reasoning about blockages. However, we also found that their direct localization of interaction objects is unreliable: bounding boxes drawn by the model are often misplaced and require multiple iterations of self-correction to converge. Figure 5 illustrates this issue, where the initial prediction fails to align with

(a) Initial predicted bounding box is misplaced from the actual door button.



(b) Correct bounding box after three iterations of self-correction.

Fig. 5: Direct object localization from a *VLM* (*GPT-5 in this example*) is unreliable, often requiring multiple iterations to refine bounding boxes.

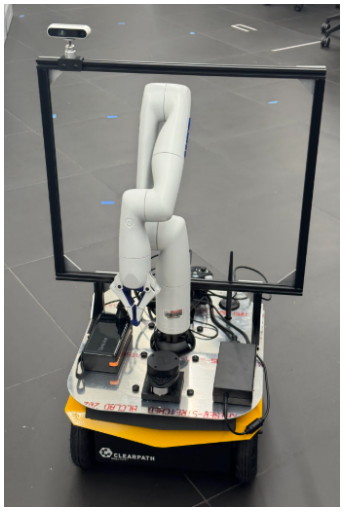


Fig. 6: The experimental platform built on a Clearpath Jackal mobile base, equipped with a 2D LiDAR, an Intel RealSense camera, and a manipulator for interacting with door buttons.

the actual button, and a correct localization is only achieved after three rounds of refinement. Such iterative behavior is impractical for real-time navigation.

To address this limitation, we integrate the *VLM* with an open-vocabulary object detector based on CLIP. In this paper, the *VLM* first generates a high-level suggestion describing the required interaction object (e.g., 'door button'). This textual cue is then passed to the CLIP-based detector [29], which searches the scene image for corresponding visual instances and outputs bounding boxes around candidate objects. This design allows the *VLM* to focus on semantic reasoning, while CLIP provides accurate and efficient localization. Together, they enable the robot to ground language-based reasoning into concrete visual cues that can be directly used for planning and manipulation.

D. Unblocking and Navigation Continuation

Once the interaction object is successfully localized, the robot performs the corresponding action to resolve the blockage. In this work, the robot navigates to a location adjacent to the detected button and presses it to open a closed door. After the environment is unblocked, the robot resumes following its planned path toward the goal. The ROS2 navigation stack continues to provide localization and motion control, while our system monitors progress to detect any further blockages. If additional obstacles are encountered along the way, the same reasoning–detection–interaction cycle is triggered, enabling the robot to repeatedly unblock itself until the destination is reached. This iterative process allows the robot to maintain robust navigation performance in complex indoor environments where multiple interventions may be required during a single mission.

IV. Experiments

We conduct experiments in real-world indoor environments to evaluate the effectiveness of the proposed navigation method. We first describe the experimental setup, including the robot platform and test environments. We then present representative results highlighting key steps of the navigation process, focusing on blockage reasoning, interaction-object localization, and manipulation. The experiments show that the proposed method enables the robot to actively unblock itself and complete navigation tasks that cannot be achieved by a standard passive navigation strategy.

A. Experimental Setup

Our robot platform is built on a Clearpath Jackal mobile base equipped with a 2D LiDAR, an Intel RealSense depth camera, and a manipulator mounted on top (Fig. 6). The 2D LiDAR is primarily used for localization and for detecting blockages along the planned path. The camera provides visual input to the *VLM*, as well as image streams used by the open-vocabulary detector to localize interaction targets. In this work, the interaction target is the button used to open doors.

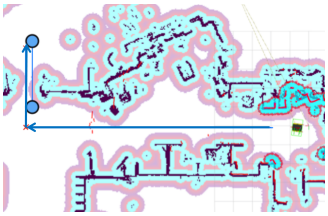
Experiments were conducted in an office building environment with corridors, offices, and multiple doorways. In these settings, the robot was tasked with navigating from one location to another while encountering closed doors that blocked the initially planned path.

B. Task Completion

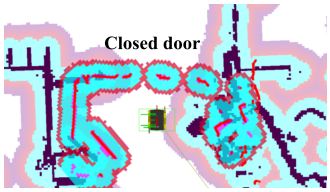
We first evaluate whether the proposed method enables the robot to complete navigation tasks that cannot be achieved by passive planning (*Nav2* only).

When relying only on passive replanning, as can be seen from 7a, the robot first assumes there is a collision-free path to the destimation. However, as it gets closer, it found the opening is actually blocked, while no alternative free path exists. Since no feasible detour exists, the robot fails to reach the destination.

When relying only on passive replanning, the robot initially assumes that a collision-free path to the destination exists, as shown in Fig. 7a. However, as the robot approaches the goal,



(a) Initially planned global path.



(b) Blockage discovered near the destination; navigation fails.

Fig. 7: Failure of passive navigation. The robot is unable to complete the task when the planned path is blocked by a closed door.

(a) With the blockage reasoning, the robot detects and click the door button to unblock itself.

(b) The map shows an opened space after clicking the door button.

Fig. 8: Success of our method: the robot actively unblocks the environment and completes the navigation task.

it discovers that the apparent opening is actually blocked, and no alternative free path is available. Since no feasible detour exists, the robot fails to reach the destination (Fig. 7b).

In contrast, our method actively handles the blockage by reasoning about the situation, localizing the interaction object, and pressing the button to open the door. As a result, the robot successfully reaches the goal in an end-to-end manner (Fig. 8). See Subsection IV-C for detailed evaluations of the blockage reasoning and detection.

To quantify these observations, we measured task success rates over multiple runs in the office environment. Throughout our tests, as long as *CLIP* successfully detects and localizes the door button, the robot always succeed in navigating to the goal locations.

C. Results on VLM-Based Blockage Reasoning

To illustrate the reasoning–detection–interaction details, we show a representative example of the robot encountering a closed door. The *MCP* prompt structure has been introduced in Sec. III. Table II shows the output from *GPT-5* when queried with a scene image shown in Fig.9. The *VLM* correctly identified that the closed door blocked the path and suggested pressing the nearby button as the unblocking action. Based on this suggestion, the CLIP-based detector localized the button in the image.

TABLE II: Representative *VLM* response (GPT-5) to the blockage prompt.

Field	Output
Obstacle	Closed door blocking the path
Interaction Object	Door button on the right side
Recommended Action	Press the button

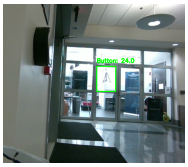


Fig. 9: Examples of button detection and localization using CLIP after the *VLM* suggests pressing the button. The first two subfigures show successful cases, while the third illustrates a representative failure caused by side-view detection under strong background lighting. Across all trials, CLIP achieved an 85% success rate in button localization.

To further evaluate robustness, we collected 20 additional images from different door locations and processed them with the reasoning–detection pipeline. In all cases, the *VLM* correctly identified the blockage as a closed door, and the CLIP-based detector localized the button with an overall success rate of 85%. Fig. 9 presents three example results. A common failure mode, illustrated in Fig. 9, occurs when the robot views the button from a steep side angle under bright background conditions, which reduces the detector's reliability.

D. End-to-End Navigation Sequence

To visually summarize the entire navigation process, we present four representative snapshots of one trial (Fig. 10). The sequence illustrates the key stages: (a) the robot follows the initially planned path, (b) encounters a closed door that blocks progress, (c) reasons about the blockage and presses the button, and (d) continues along the newly opened path to reach the destination.

V. Conclusion and Future Work

In this paper, we presented a method that enables a robot to actively resolve situations where passive replanning fails. By combining classical path planning with *VLM*-based reasoning and CLIP-based open-vocabulary detection, the robot is able to identify blockages, localize relevant interaction objects, and perform actions such as pressing a button to open a door. Experiments in an office environment demonstrated that the proposed approach successfully unblocks navigation and allows the robot to complete tasks that are otherwise unachievable by standard passive methods.

Our experiments further revealed that reliable button localization is critical for task success. A simple heuristic approach of stopping at a fixed distance from a blockage is insufficient, as it may not guarantee that the interactive object is within

Fig. 10: Representative sequence of one navigation task. From left to right: (1) Following a planned path. (2) Blockage encountered. (3), (4) Reasoning and executing unblocking solution. (5) Path cleared. (6) continuing and completing the navigation task.

view. Instead, actively searching for and localizing interactive elements, such as door buttons, emerges as a promising direction to ensure robust execution of unblocking behaviors.

While these results are promising, the current implementation addresses only one type of blockage: closed doors requiring button interaction. In practice, robots may encounter a wider variety of obstacles and dependencies, including elevators, temporarily blocked hallways, or complex multi-step interactions. The success of unblocking in these broader scenarios will depend critically on how the robot's model context is specified and how effectively the *VLM* can interpret that context.

In future work, we aim to extend the proposed method to handle a wider spectrum of indoor blockages, improve robustness under challenging visual conditions, and systematically investigate what forms of context, map representation, and state information are necessary for robots to flexibly unblock their way in the manner humans naturally do. Ultimately, this line of research moves toward robots capable of interactive and adaptive navigation in complex human environments.

REFERENCES

[1] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 11(1):24, 2022.

[2] Julio A Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A Castellanos. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 2023.

[3] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, and Robin De Keyser. Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28, 2016.

[4] Iram Noreen, Amna Khan, and Zulfiqar Habib. Optimal path planning using rrt* based approaches: a survey and future directions. *International Journal of Advanced Computer Science and Applications*, 7(11), 2016.

[5] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2013.

[6] Juraj Kabzan, Lukas Hewing, Alexander Liniger, and Melanie N Zeilinger. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370, 2019.

[7] Lukas Hewing, Juraj Kabzan, and Melanie N Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 28(6):2736–2743, 2019.

[8] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

[9] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[10] Anil Aswani, Humberto Gonzalez, S Shankar Sastry, and Claire Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

[11] Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.

[12] José Manuel Bravo, Teodoro Alamo, and Eduardo F Camacho. Robust mpc of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745–1751, 2006.

[13] Jun Zeng, Bike Zhang, and Koushil Sreenath. Safety-critical model predictive control with discrete-time control barrier function. In *2021 American Control Conference (ACC)*, pages 3882–3889. IEEE, 2021.

[14] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[15] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

[16] Leopoldo Armesto, Vicent Girbés, Antonio Sala, Miroslav Zima, and Václav Šmídl. Duality-based nonlinear quadratic control: Application to mobile robot trajectory-following. *IEEE Transactions on Control Systems Technology*, 23(4):1494–1504, 2015.

[17] Ernő Horváth, Csaba Hajdu, and Péter Kőrös. Novel pure-pursuit trajectory following approaches and their practical applications. In *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 000597–000602. IEEE, 2019.

[18] Jiaming Wu and Dongjun Chen. Trajectory following of a tethered underwater robot with multiple control techniques. *Journal of Offshore Mechanics and Arctic Engineering*, 141(5):051104, 2019.

[19] Nguyen Hung, Francisco Rego, Joao Quintas, Joao Cruz, Marcelo Jacinto, David Souto, Andre Potes, Luis Sebastiao, and Antonio Pascoal. A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments. *Journal of Field Robotics*, 40(3):747–779, 2023.

[20] Zhen Zhang, Anran Lin, Chun Wai Wong, Xiangyu Chu, Qi Dou, and K. W. Samuel Au. Interactive navigation in environments with traversable obstacles using large language and vision-language models. *arXiv preprint arXiv:2310.08873*, 2023. Available at https://arxiv.org/abs/2310.08873.

[21] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Panav: Toward privacy-aware robot navigation via vision-language models. *arXiv preprint arXiv:2410.04302*, 2024. Available at https://arxiv.org/abs/2410.04302.

[22] Congcong Wen, Yisiyuan Huang, Hao Huang, Yanjia Huang, Shuaihang Yuan, Yu Hao, Hui Lin, Yu-Shen Liu, and Yi Fang. Zero-shot object navigation with vision-language models reasoning. In *International Conference on Pattern Recognition*, pages 389–404. Springer, 2025.

[23] Sichao Liu, Jianjing Zhang, Robert X Gao, Xi Vincent Wang, and Lihui Wang. Vision-language model-driven scene understanding and robotic

object manipulation. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 21–26. IEEE, 2024.

[24] Amirreza Payandeh, Anuj Pokhrel, Daeun Song, Marcos Zampieri, and Xuesu Xiao. Narrate2nav: Real-time visual navigation with implicit language reasoning in human-centric environments. *arXiv preprint arXiv:2506.14233*, 2025.

[25] Mingfeng Yuan, Letian Wang, and Steven L Waslander. Opennav: Open-world navigation with multimodal large language models. *arXiv preprint arXiv:2507.18033*, 2025.

[26] Mohamed Elnoor, Kasun Weerakoon, Gershom Seneviratne, Jing Liang, Vignesh Rajagopal, and Dinesh Manocha. Vi-lad: Vision-language attention distillation for socially-aware robot navigation in dynamic environments. *arXiv preprint arXiv:2503.09820*, 2025.

[27] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[28] Steve Macenski, Shrijit Singh, Francisco Martin, and Jonatan Gines. Regulated pure pursuit for robot path tracking. *Autonomous Robots*, 2023.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.