Navigation and Obstacle Avoidance for Indoor Sweeping Robots Oriented by WorkFlow

Lingsihan Meng

{z5667660@ad.unsw.edu.au}

School of Electrical Engineering, The University of New South Wales, Sydney, 2017, Australia

Abstract. Sweeping robots have been extensively deployed in indoor environments; however, achieving robust navigation and obstacle avoidance in complex and dynamic spaces remains a challenge. Most existing research only concentrates on single specific improvement of algorithms, while the continuity and coordination of the navigation process receive less attention. In this paper, indoor sweeping robot navigation is considered from a workflow perspective, with emphasis on the integration of perception, localization, path planning, obstacle avoidance, and motion execution. Obstacle avoidance is emphasized as the key part in local decision-making, which bridges perception and motion execution and directly determines the navigation safety and continuity. By integrating obstacle-avoidance within a single navigation loop, sweeping robots would be able to handle localization uncertainty, dynamic obstacles and constrained indoor environments. The analysis and the proposed workflow-oriented perspective provide insights for designing more robust and practical autonomous indoor cleaning systems.

Keywords: Indoor sweeping robot; Closed-loop navigation; Obstacle avoidance; Process coordination

1 Introduction

With the development of service robots, household vacuum cleaners have been widely used at home or in the office. dominating a large part of the service robotics market. Research has shown that as demand for autonomy grows and costs of sensing, processing, and production technology, the indoor cleaning robots market is growing steadily. However, different from structured industrial environments, real-world indoor scenes are usually characterized by narrow passages, complex furniture layouts, and frequent pedestrian walking. Such scenarios pose challenges such as unpredictable navigation, unsuccessful obstacle avoidance, and reduced efficiency for robot in longer autonomous vacuuming tasks.

The reasons for this are often attributed to the “complexity of the environment”, but in reality, it turns out that when deploying a system, difficulties stem mainly from the interaction of external factors (moving boundaries, traffic, etc.) with internal limitations (sensor noise, processing lags, and lack of staff). This interplay often aggravates poor coordination between

ICMEEA 2026, April 24-26, Singapore, Singapore

DOI 10.4108/eai.24-4-2026.2364941

perception, localization, planning, and control modules, leading to discontinuous information, inconsistent behaviors, and eventually failures of navigation.

Recently, the literature has focused on proposing improved localization strategies and advanced routing algorithms. However, these studies are typically driven by an interest in the computational performance of the algorithms themselves and rarely consider their effectiveness within a holistic decision-making process. Previous results suggest that failures in navigation are often not due to shortcomings in sophisticated planning algorithms, but are more likely to stem from ambiguous interfaces between modules, broken communications channels, and a lack of coordination across the entire system in decision-making. Thus, it is not enough simply to gather together algorithms that work well in isolation to make a useful robot cleaner. It is time for a paradigm shift in the workflow design perspective applied to self-driving cars.

The aim of this paper is not to review every method for navigation and obstacle avoidance that has been proposed previously, but rather to focus on the design of a workflow model and the analysis of an indoor robot cleaner's navigation function. This study systematically reviews, analyses and compares different technologies and architectures to facilitate communication between modules, communication protocols, and distributed planning algorithms. Particular focus will be placed on the evolution of collision-avoidance strategies, emphasizing their evolution from standalone "reactive modules" to integrated "components within the autonomous robot's control loop."

The subsequent structure of this article is arranged as follows: Chapter Two analyzes the core technical modules of navigation and obstacle avoidance for sweeping robots and their collaborative relationships from a process perspective; Chapter Three focuses on discussing the functional positioning, implementation mechanism and optimization directions of obstacle avoidance technology in the navigation workflow; Chapter Four summarizes the full text and discusses future research directions.

2 The ideal model and key challenges of the navigation workflow

In more recent times, there is a trend to consider autonomy not only as a collection of independent algorithms that may be solved independently from each other but rather as an end-to-end system [1]. In light of these observations, this work develops a model describing ideal vacuum cleaning tours, followed by a discussion of common system-level operational errors observed during real-world deployment.

2.1 Ideal navigation workflow closed-loop model

From a practical perspective, the autonomous navigation of housekeeping robots is considered a continuous process that consists of the following steps: sensing the environment → localization and mapping (which includes localization, environmental modeling, and object recognition) → motion planning (both global and local) → action execution → environment change → sensing. Instead of being linear and sequential, it is an adaptive feedback loop that works in real-time [1].

It is a very kind of loop, looped system of communication is also a simple information transmission, but it is always accompanied by random uncertainties and temporal characteristics. This becomes clear for example if there is an accumulation of measurement errors as well as forecast uncertainties through many iterations and affect the path planning and control signals [2]. It requires good orchestration of activities and migrations between its components during the runtime model and also a restart through instruction protocols - which is necessary for vacuum robot that can have concurrent actions [3].

In practice, the closed-loop of the navigation workflow also means that the decision will be constantly revised during the process and not made once. Every loop around it is going to do an update on the estimate of where the robot is, as well as it is going to re-evaluate the environment and task constraints. For example, temporary obstacles such as people moving, furniture being moved, etc., and the robot needs to change its local plan without reducing the quality of long-term cleaning. This points out that need hold through loop, keep history info can't be abrupt and smooth bear effect.

In addition, from the perspective of workflow-oriented navigation loop can be naturally deal with the uncertainty. And so at each step, a forced binary decision is not made; instead, the uncertainty estimates are propagated forward and reflected in the generated planning and control policies. In this way, it can be more careful in dealing with high-risk situations such as narrow passages or crowds, and remain efficient in open spaces. From the perspective of the system Design, The adaptive behavior is difficult to achieve if the navigation modules are designed and optimized independently.

2.2 Common malfunction patterns in operational disruption

Although it is clearly specified in the theory, many interruptions may occur in home environments due to various external factors, which will result in a complete failure of the entire navigation system.

Progressive error transmission and compounding effects. When GPS does not work in a house, sweeping robot generally uses relative positioning with encoders and inertia sensors. Under such conditions, positioning drift is inevitable. Some researchers have proposed that if the positioning drift situation cannot be corrected, it will affect subsequent path planning, and then cause overlapping or missed sweeping of the sweeping area [4]. Beyond positional drift itself, progressive error transmission also affects the semantic interpretation of the environment. When localization errors accumulate, obstacles may be incorrectly projected onto the map, leading the planner to either overestimate free space or falsely classify navigable regions as blocked. Over time, such misinterpretations significantly reduce coverage efficiency and may cause repeated traversal of already cleaned areas.

Frequent switching between global path planning and local obstacle avoidance may lead to hesitation, trajectory oscillation, and even deadlock if coordination between the two planning layers is not properly maintained [5].

Time synchronization problems. In practical applications, the mutual interference caused by the inconsistency between the global planning and the local obstacle avoidance period is also one of the reasons for the failure of robot navigation. Failure to correctly respond to the global planning information when obstacle avoidance is frequently switched may cause problems such as hesitation, trajectory oscillation and deadlock in robot navigation [5]. From a workflow perspective, time synchronization issues are not merely timing errors but structural inconsistencies between decision layers. When local avoidance operates at a much higher frequency than global planning, the robot may continuously override strategic objectives with short-term reactions. This results in unstable behavior patterns that are difficult to diagnose using traditional algorithm-level debugging methods.

Information semantic discontinuity. Perception module generally give information on obstacles in terms of map or polygonal barrier, but planning module require to know if these obstacle is traversable and how it can be crossed. In the absence of a good semantics layer, the data is continuous but the meaning is not [6]. And semantic discontinuity also make the system be difficult to scale. When other sensors or planning schemes are added, the poor unified representation increases the integration cost and error rate. So as to demonstrated alignment that are more important than consider as second thought after implementation.

Lack of contingency management procedures. Due to the presence of narrow passages and cluttered environment, dead angle is commonly encountered during the operation of the robot. No proper recovery mechanisms or restart policies, it has been observed that local collision avoidance degrades to periodic failure responses and results in collision or stall [7].

CPU contention, and scheduling problems. Autonomous navigation + obstacle avoidance + those tasks that are required to complete also a robot vacuum has to manage another type of concurrent task like charging the battery or disposing off the dustbin. In the absence of proper task manager there can be contention for computation and resource which will deteriorate the performance of the navigation process [3].

3 Process-oriented component coordination approach

In regards to the lower level failures mentioned earlier, this work takes a look and evaluates on how the different key components all work together to keep up the navigation activity and does so at a higher level, instead of just one algorithm on its own.

3.1 Maintaining uninterrupted and coherent state assessment

In the navigation process, a major role played by the state estimator block is to provide a stable estimate of position information so that further processing, such as trajectory generation and obstacle avoidance, can be carried out efficiently. Multi-sensor fusion methods, including Extended Kalman Filters, are often used for reducing positional error in odometry and increasing the accuracy of long-term position tracking [6].

In a non - static environment, state estimation is also to perceive and track moving objects. This can be done via Kalman filter based algorithms, where multiple moving-target tracking methods are used to estimate the locations of all other moving objects in the surroundings at a particular time instance, providing important information for anticipatory route planning [8]. And also these kind of methods have an added advantage of stabilizing the navigation system.

In addition, it is particularly important to maintain state continuity for a long time in the operation of sweeping robots, which are required to run independently for a long time. Where short navigation task maybe is a short stop where there is one stopping point and its all the same like lighting, objects on the floor and everything, but cleaning mission will pass through so many different areas. A good state assessment mechanism should have mechanisms for confidence assessment and recovery such as re-localization triggers above a certain uncertainty.

From its workflow view, it gives perception via state estimation; however, this same state estimation is utilized by all modules for reference. In order to make it more stable, the program can be put in a more advantageous state for switching among the various navigation modes, and it will be able to better prevent a sudden change in behavior.

3.2 Bridging global planning and local execution

Path planning is commonly organised in a hierarchical architecture, in which a global planner determines the long-term route while a local planner performs real-time obstacle avoidance and trajectory adjustment. This hierarchical design improves navigation robustness by balancing long-term efficiency with short-term safety [5].

3.3 Collision prevention as an intrinsic system element

Not as a single safe product but as a constant judgment of the state within the process of movement. Obstacle avoidance is intrinsically a perception, prediction, planning, and control all linked together problem, it is a nice example of system-level coordination. If the collision prevention mechanism works well, then this may act as a mediator that can make the short-term safeguard be compatible with the longer-term objective of coverage and saving power.

This view can also explain the reason why pure avoidance strategy is often invalid for complicated indoor environment. Without the ability to access the history and purpose, the reaction would tend to be oscillated or stuck in local minimum. Embedding obstacle avoidance inside of the navigation loop so the robot is now not just reasoning about motion in response to an immediate threat but also what future state it is trying to avoid.

Evolving from responsive to anticipatory approaches. The traditional reactive method of avoiding obstacles cannot meet the requirements of dynamic Environment. It is possible to combine the forecast of moving obstacles trajectories with short-term motion planning for robot to proactively adjust its motion plan and make it’ s avoidance motion more smooth [8]. It can be seen from recent findings that combining proactive sensing with risk-aware routing will boost the performance of mobility for interacting with uncertain environment [9].

Safety layer in the process. A reactive controller with the highest priority besides the planning-based controller is often required for robots to handle emergency situations. The reactive controller may not be able to choose the optimal action at each individual decision point. But, it is operational safely throughout the entire navigation procedure [1].

Interaction with the localization and mapping process. Pathfinding can destroy the original maps. Using incremental map building and on-the-fly update, the path planning can maintain the position of the system, and avoid obstacles at the same time as adding new obstacles to the global map, reducing system downtime due to information lag [10].

Collision avoidance for autonomous vacuum cleaners should be modelled as a continuous stream of motion decisions embedded within the robot’s locomotion process, not only by itself as an insurance against. Exploration and obstacle avoidance are necessarily a mixture of sensing and mapping, navigation, control and movement management, whose efficiency is strongly conditioned on the coordination between them at the system design layer.

ROS and ROS2 are widely used robotics middleware platforms that provide standardised communication mechanisms between perception, localisation, planning, and control modules. Sensor nodes publish environmental information to localisation and mapping modules, which then provide updated information to global planners and local controllers, enabling coordinated navigation and reliable obstacle avoidance during task execution [11].

ROS2 supports publish–subscribe communication through topics together with services and actions, enabling efficient coordination between navigation modules in dynamic environments. This communication framework enables dynamic route adjustment when new obstacles are detected while maintaining overall navigation objectives [11].

ROS and ROS2 are robotics middleware, which define the general workflow of operations with standard methods for decoupling modules and enabling communication between them. The sensors for environmental perception publish their current state using a network of nodes, which feed the localization module and map builder; these in turn feed the high-level planner and low-level controller modules, respectively. In this scenario, the robot will automatically learn to avoid obstacles by sensing its environment’s state and restricting motion, allowing safety mechanisms to influence motion decisions without disrupting overall task execution [11].

ROS2 also supports interconnection between systems at the workflow layer and provides separate interfaces for this purpose. To transmit information in real time, it uses topics for publish subscribe communication, while services and actions provide direct invocation and long-running commands, respectively. This structure allows for dynamic re-routing around obstacles that may modify existing routes and hence possibly diverge from an original plan to traverse some region, but nonetheless achieve its purpose of covering a given area.

In addition, by using the built-in parameter management and lifecycle management in the middleware, it is possible to change protection layers on the fly: speed limits and navigational strategies according to the terrain or energy constraints. Thus, obstacle avoidance provides a trade-off between short-term survival and long-term performance in terms of coverage efficiency, mission lifetime, etc. Such a duality is particularly important in the case of autonomous vacuum robots operating in dynamic and busy environments [11].

4 Discussion and future prospects

4.1 Fundamental trade-offs in process design

A process-based architecture for navigation is developed, but aspects and their interrelations of the architecture have to be considered. Take for example this trade off between modularity and coupling. An architecture which has more modular parts are easier to maintain, have more flexibility and transparencies which is helpful for doing more robotic works. However, over-segmentation can cause delays in communication between modular units and increase the coordination demand of the system, which may degrade the latency characteristic of the system or, in some cases, harm the integrity property of the system. On the contrary, tightly coupled systems have low latency and efficient coordination between components, but the system will be unclear and debugging, testing and validation will be difficult.

The third difficulty is the balance between safety and effectiveness: safe control methods must be carried out in the context of a real-time environment with a high sense of urgency, responding to unpredictable dangers and disturbances in time. On the other hand, global planning methods aim at optimizing long term goal like task efficiency, route efficiency, which are mostly done using more costly and intensive computation. To meet these two conflicting demands requires balancing so that it is possible to have safety measures predict or override decisions, but routing still works[1].

Apart from considering design and functions, it is necessary to decide which computing power will be allocated to the real-time computing section, and which will be allocated to the long-term computing section. For instance, real-time environmental perception, collision avoidance and motion control require a quick decision in a short time, so the CPU computing power needs to be reserved for them all the time. But the map construction, the global path planning and local coverage all have a longer cycle for processing and have a certain degree of fault tolerance. As mentioned in Section 2.2, if the competition for shared computing resources cannot be effectively regulated, it may result in delayed safety response or a reduction in planning performance. This problem of resource allocation highlights the need to build up priorities strategies and scheduling mechanisms to reach a compromise on how to run both safety-critical computing and the decision making planning so as to make stable navigation happen with a limited on board computing.

4.2 Future directions of development

For future research, several directions can be considered to further improve the development and application of a process-based navigation system:

Artificial intelligence for self-regulating and adaptive process management. AI approaches need not apply only to object recognition or path generation but could be applied to the working process to intelligently control interactions between modules. For example, adaptive algorithms may change data transfer speeds, scoring metrics, or the priority of tasks for detection, routing, and the provision of protective elements on a case-by-case basis or as needed for computationally limited applications, to allow for adaptive and optimal deployment.

Systematic measurement and transmission of probabilistic variables. High-level autonomy will benefit greatly if it can communicate its confidence in decisions based on sensed information, uncertainties in measurements, forecasts, and dynamics, as well as how they propagate through the whole filter/dynamics chain. Thus, when uncertainty is an explicit variable available to all agents in the system, planners and controllers will be able to coordinate actions, thereby improving system robustness and preventing premature decisions in unknown and partially observable environments.

Fault-tolerant, autonomous systems with long life cycles. Long-term autonomy requires navigation systems that can detect degradation, loss of connectivity, or unexpected system activity and then automatically initiate a recovery procedure. Self-healing can be redundancy, adaptive control changes, or deliberate degradation techniques that allow the system to continue operating safely with minimal or no human interaction.

System testing in large-scale simulations. Full-simulation verification is still necessary for the evaluation of the interaction of the system during a full navigation cycle, especially for rare and dangerous failures. End-to -end simulated test: A complete test on a number of aspects such as task assignment, information sharing methods, reaching agreements, and decision priorities before doing anything. It is also minimizing unwanted things that happen in real situations which are used as a test.

5 Conclusion

In this paper, the original motivation mentioned in the introduction is re-examined, that is, how to realize reliable navigation and obstacle avoidance of sweeping robots in complex and ever-changing indoor environments. This study is not only the enhancement of the isolated algorithm but is more on a workflow point of view and thus on the use of the perception, localization, planning, obstacle avoidance and movement.

In this paper, by systematic analysis on the navigation pipeline, the root of safe navigation and smooth operation is to have an obstacle avoidance function. It can be found that it’s necessary for it not just depend on good sensing or intelligent logic, but also on the whole navigation procedure. According to the role, realization mode and closed-loop performance of obstacle avoidance, this work highlights the importance of system-level coordination to enhance robustness and cleaning efficiency. The points mentioned above are very much related to the topic of this article, as they describe why so many existing sweeping robots continue to fail in tight spaces, crowded layouts, or dynamic situations despite improvements in individual technologies. The process-based review shows that the reasons for navigation errors are mainly related to information fragmentation and a lack of inter-module interaction, rather than just a lack of sophisticated algorithms.

The results obtained by this work are of interest for researchers and engineers developing indoor service robots, especially in designing their navigation systems, perception fusion, planning, and control systems. The manufacturers and system designers of sweeping robots might find this perspective useful when building more robust products for realistic home environments as well. Finally, considering obstacle avoidance not as a separate module but as part of the whole navigation pipeline gives us a better direction towards robust, practical autonomous sweeping robots. The future of navigation systems, with an emphasis on process integration and coordination in decision-making, will be able to realize safer, more efficient, and genuinely autonomous navigation through realistic indoor environments.

References

[1] Carvalho, M., Simoni, R., Yoshioka, L.: Autonomous navigation and collision avoidance for mobile robots: Classification and review. arXiv preprint. pp. 1-24 (2024)

[2] Wang, Y., Ahmad, A.: Workflow-oriented navigation frameworks for autonomous indoor mobile robots. Robotics and Autonomous Systems. pp. 104720 (2025)

[3] Kang, H., Lu, J., Li, K., Zhang, Q., Wang, Y.: A decision-making algorithm of multiple reactive tasks for autonomous driving sweepers based on behavior trees. IEEE/ASME Transactions on Mechatronics. pp. 1-12 (2025)
[4] Yang, L.: Autonomous positioning and full traversal path planning technology for indoor intelligent sweeping robot. International Journal of Mechanical Applications. pp. 131-138 (2019)
[5] Li, F., Zlatanova, S., Koopman, M., Bai, X., Diakité, A.: Universal path planning for an indoor drone. Automation in Construction. 95, pp. 275–283 (2018)
[6] Liu, Y., Wang, S., Xie, Y., Xiong, T., Wu, M.: A review of sensing technologies for indoor autonomous mobile robots. Sensors. pp. 1222 (2024)
[7] Zheng, H., Zhang, J., Jiang, M., Liu, P., Liu, D., Qin, T., Yang, M.: Embodied escaping: End-to-end reinforcement learning for robot navigation in narrow environment. arXiv preprint. pp. 1-15 (2025)
[8] Nordström, S., Lindquist, B., Nikolakopoulos, G.: DTAA: A detect, track and avoid architecture for navigation in spaces with multiple velocity objects. arXiv preprint. pp. 1-18 (2025)
[9] Ahmad, A., Khonji, M., Al-Sumaiti, A., Dias, J., Elbassioni, K.: Graph-based local planning with spatiotemporal risk assessment for risk-bounded and prediction-aware autonomous driving. Proceedings of the International Conference on Control, Automation, Robotics and Vision, Dubai, United Arab Emirates, December 2024, pp. 485-492
[10] Deeb, A., Seto, M., Pan, Y.: Piecewise-deterministic quasi-static pose graph SLAM in unstructured dynamic environments. Journal of Intelligent & Robotic Systems. pp. 34 (2022)
[11] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot Operating System 2: Design, architecture, and uses in the wild. Science Robotics. 7(66), eabm6074 (2022)