Enhancing Autonomy of Mobile Robots with Behavioral Tree using ROS2

Lukas Evers*,1, Umut Uzunoglu1, and Ahmed Hussein1 *Senior Member, IEEE*

Abstract— **This paper proposes a state of the art review and a behavioral planning approach to improve the autonomy of mobile robots using the Robot Operating System (ROS2). The study aims to address the issue of human intervention that is often required during the operation of autonomous mobile robots and improve the efficiency and performance of the robots. The system includes a monitoring system, sensor data storage, and behavior tree design and implementation. The effectiveness of the approach was evaluated using the Gazebo simulator, with the behavior tree handling various scenarios such as sensor failures, collisions, unreachable goals, and low battery state. The simulation results showed a significant improvement in the robot's autonomy and resilience to failures, providing evidence for the feasibility of the proposed approach. To further validate the results, additional experiments are being performed in real-world settings.**

I. Introduction

Mobile robots are increasingly being used in a variety of fields, including warehouse logistics, last-mile delivery, and agriculture. Despite being marketed as autonomous, they often require human supervision to operate correctly and are limited in their autonomy to specific environments. The Robot Operating System (ROS) is a popular platform for building and programming robots, but the standard robot still requires human intervention to ensure proper functioning and safety in uncertain environments.

This paper focuses on how behavior planning can increase the robustness and autonomy of robots running ROS2. Behavior planning allows the robot to react to failures and problems and decide on alternative courses of action to mitigate risks and achieve its goals. The paper will explore the possibility of improving current systems by adding a dedicated behavior planning component and provide an implementation of exemplary behaviors and a system architecture for incorporating behavior planning in other robots.

The study aims to demonstrate that the implementation of behavior planning can increase the robot's robustness and decrease the need for human intervention in various scenarios. The paper will present the results obtained from selected scenarios, where failures and problems will be artificially induced to test the robot's ability to behave autonomously. The paper will summarize the results and discuss the acceptance criteria for the requirements. The conclusion will provide an outlook for future work and recommendations.

II. State of the Art

A. Autonomous Driving Navigation Architectures

Autonomous driving software architectures follow the "*Sense - Think - Act*" paradigm [1], as shown in Figure 1. Most architectures implement a hierarchical structure in which sensory inputs are processed to create an environmental representation, leading to a path from the current position to the destination computed by global planning. This is followed by behavioral planning, which ensures that the vehicle obeys traffic rules, and local planning, which translates motion commands while taking into account system constraints. An additional system monitoring layer monitors the execution of other components and triggers actions in case of failures [2]. Vehicles can achieve (Society of Automotive Engineers) SAE levels of autonomy up to levels four and five [3]. Expanding the behavior planning module can lead to higher levels of autonomy [4].

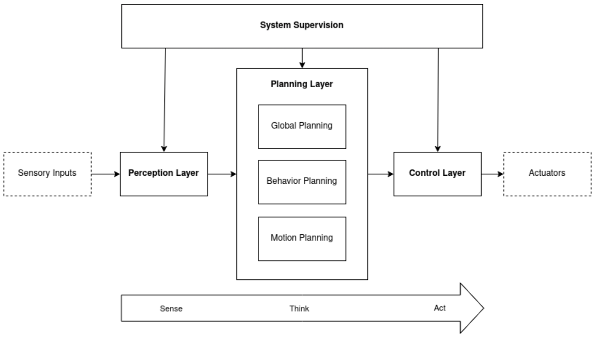

Fig. 1: Common Autonomous Driving Architecture [5], [6]

B. Behavior Types

The behavior planning module is responsible for choosing the best behavior to execute in a specific environment to ensure the best performance for a given task. A behavior planning approach provides the robot with acceptable behaviors for every possible scenario the robot is operating in.

1) Reactive Approach: The first type of behaviors implemented in behavior-based robots were reactive behaviors. This maps sensory input directly to motor commands, similar to reflexes in humans. Reactive behaviors are fast due to their low computational load, and are suitable in scenarios where real-time safety is a concern. However, a system with a solely reactive behavior planning approach will not meet requirements for higher levels of autonomy. Reactive behaviors can still be valuable in improving vehicle safety and reliability [7].

* Corresponding author
1 IAV GmbH, Berlin, Germany
lukas.evers@iav.de, umut.uzunoglu@iav.de, ahmed.hussein@ieee.org

2) Deliberative Approach: Deliberative behaviors are defined as behaviors that involve planning and decision-making. Unlike reactive behaviors, deliberative behaviors follow the ”Sense, Think, Act” paradigm and can decide the best course of action before acting [1]. A behavior planner with a deliberative approach can make decisions based on current and previously processed data, and can predict environmental changes. Deliberative behaviors improve autonomy and mimic human skills, reducing the need for a human operator.

3) Hybrid Approach: A hybrid model combining reactive and deliberative behaviors can create reliable and intelligent systems. The hybrid approach combines the quick reaction times of reactive behaviors with the decision-making ability of deliberative behaviors, leading to more robust robots and higher levels of autonomy. The higher-level deliberative planners can override reactive planners, allowing the system to be more flexible in uncertain environments while still maintaining fast reaction times [8].

C. Behavior Planning Approaches

Robots that use a hierarchical behavior planning approach need a system to determine which behaviors to execute. Different approaches for behavior planning use states to determine which strategies and behaviors are utilized. This article presents and compares Finite State Machines (FSMs), Behavior Trees (BTs), and Partially Observable Markov Decision Processes (POMDPs).

1) Finite State Machines: FSMs are a popular choice for behavior planning in robotics. The fundamental idea behind FSMs is to describe behavior in terms of states that trigger actions. FSMs are defined by a set of states, a starting state, state transitions, final states, and an input alphabet. At any given time, a specific state machine is selected and its actions are executed. The benefits of FSMs include their speed and ease of implementation, especially when using libraries, which can make the development process much simpler and faster. However, as the complexity of the problem increases, so does the effort required to implement the FSM. Despite these limitations, FSMs are still a widely used approach in robotics due to their simplicity and effectiveness in controlling motors [9].

2) Behavior Trees: BTs are a popular approach for modeling AI for non-player characters in computer games. Unlike FSMs, BTs have a hierarchical structure, with nodes that represent actions or sequences of actions. The tree consists of a root node, child nodes, parent nodes, and leaf nodes (execution or action nodes). The behavior tree is executed by ticking the root node, which will then be recursively ticked down the tree until the execution of an action node is triggered. Control nodes, such as ”Sequence,” ”Fallback,” and ”Condition,” determine the order and conditions of execution for the action nodes. BTs provide a flexible and intuitive way to represent complex AI behaviors, and are widely used in the game development industry [10].

3) Partially Observable Markov Decision Processes: POMDPs are a more sophisticated approach for decision-making in uncertain environments. The basic idea behind POMDPs is to model decision-making as a process of choosing actions based on the probability distribution of the system's state. A POMDP consists of a set of states, actions, observations, rewards, and a transition model. POMDPs use belief states, which represent the probability distribution of the system's state, to make decisions. The solution to a POMDP is a policy, which is a mapping of belief states to actions. POMDPs are more complex than FSMs and BTs, but provide a more robust and flexible way to handle uncertainty in robotic decision-making [11].

In conclusion, the behavior planning module of an autonomous driving system plays a crucial role in determining the best behavior to execute in a given environment. The behavior planning approach can range from reactive to deliberative to a hybrid of both. FSMs and BTs are two popular approaches for behavior planning in robotics, while POMDPs are a more sophisticated approach for decision-making in uncertain environments. FSMs are fast and easy to implement, while BTs provide a flexible and intuitive representation of complex AI behaviors. POMDPs are more complex but provide a robust and flexible way to handle uncertainty. The proposed approach for this study will use a BT approach for the behavior planning module, offering a hierarchical structure for modeling AI and a flexible representation of complex behaviors.

III. Proposed Approach

IV. Experimental Work

V. Results and Discussion

VI. Conclusion and Future Recommendations

References

[1] R. Murphy, R. Murphy, and R. Arkin, *Introduction to AI Robotics*, ser. A Bradford book. MIT Press, 2000. [Online]. Available: https://books.google.de/books?id=RVlnL_X6FrwC

[2] M. Zimmermann and F. Wotawa, "An adaptive system for autonomous driving," *Software Quality Journal*, vol. 28, no. 3, pp. 1189–1212, 2020.

[3] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson *et al.*, "Odin: Team victortango's entry in the darpa urban challenge," *Journal of field Robotics*, vol. 25, no. 8, pp. 467–492, 2008.

[4] M. Reke, D. Peter, J. Schulte-Tigges, S. Schiffer, A. Ferrein, T. Walter, and D. Matheis, "A self-driving car architecture in ROS2," in *2020 International SAUPEC/RobMech/PRASA Conference*. IEEE, 2020, pp. 1–6.

[5] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal on Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[6] G. Velasco-Hernandez, D. J. Yeong, J. Barry, and J. Walsh, "Autonomous driving architectures, perception and data fusion: A review," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2020, pp. 315–321.

[7] L. De Silva and H. Ekanayake, "Behavior-based robotics and the reactive paradigm a survey," in *2008 11th International Conference on Computer and Information Technology*, 2008, pp. 36–43.

[8] R. Arkin, R. Arkin, and R. Arkin, *Behavior-based Robotics*, ser. Bradford book. MIT Press, 1998. [Online]. Available: https://books.google.de/books?id=mRWT6alZt9oC

[9] F. Wagner, R. Schmuki, T. Wagner, and P. Wolstenholme, *Modeling Software with Finite State Machines: A Practical Approach*, 05 2006.

[10] M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, "A survey of Behavior Trees in Robotics and AI," *Robotics and Autonomous Systems*, vol. 154, p. 104096, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022000513

[11] S. Feyzabadi and S. Carpin, “Risk-aware path planning using hierarchical constrained markov decision processes,” vol. 2014, 08 2014.