AGVs vs AMRs: A Comparative Study of Fleet Performance and Flexibility

Rita Tomé Silva
Department of Electrical and Computer Engineering
Faculty of Engineering of the University of Porto
Porto, Portugal
0009-0007-9895-9090

Marina Brilhante
Centre of Robotics in Industry and Intelligent Systems
INESC TEC
Porto, Portugal
0009-0008-9508-0379

Héber Sobreira
Centre of Robotics in Industry and Intelligent Systems
INESC TEC
Porto, Portugal
0000-0002-8055-1093

Diogo Matos
Centre of Robotics in Industry and Intelligent Systems
INESC TEC
Porto, Portugal
0000-0002-5422-2122

Pedro Costa
Department of Electrical and Computer Engineering
Faculty of Engineering of the University of Porto
Porto, Portugal
0000-0002-0435-8419

***Abstract*—Autonomous Mobile Robots (AMRs) and Automated Guided Vehicles (AGVs) have emerged as key innovations in the industry world, with AMRs offering flexibility and adaptability for dynamic environments, while AGVs provide high accuracy for repetitive tasks; thus, this research proposes a study of fleets of both AGVs and AMRs to enhance productivity and efficiency in industrial settings. Several tests were performed where the duration of a mission, the success and collision rate, and the average number of disputes per mission were analyzed in order to obtain results. In conclusion, while AGVs tend to be more reliable and consistent in task completion, AMRs offer greater flexibility and speed.**

***Index Terms*—AGV, AMR, Robot Fleet, Path Planning, TEA*, TEB Local Planner**

I. Introduction

With the industry's ever-changing demands, manufacturers are being pushed to innovate and find new ways to add value to their products. In this context, Autonomous Mobile Robots (AMRs) and Automated Guided Vehicles (AGVs) have emerged as key technological advancements that are increasingly being deployed in warehouses, distribution centers, manufacturing facilities, and even healthcare environments [1].

The core guidance systems of AGVs have undergone significant evolution since their introduction in the 1950s, advancing through mechanical, optical, inductive, inertial, and laser-based methods over the years. Traditional AGVs are limited to following fixed, predefined paths along a guide system. In contrast, AMRs offer greater flexibility, capable of navigating to any accessible and collision-free location within a designated area [2].

AGVs and AMRs offer a solution to streamline the material handling process, reduce error rates, and enhance worker productivity [3]. However, despite their similar function, they are equipped differently and respond in distinct ways, exhibiting their respective advantages and disadvantages depending on their area of use.

The most significant distinction between these robots is in their navigation capabilities. AMRs possess the capacity to dynamically plan the optimal route to a designated destination, detect potential obstacles, and modify their trajectory to avoid such obstructions. In contrast, AGVs follow a predefined fixed route, which is frequently guided by wires, magnetic strips, or sensors [4].

In terms of predictability, AGVs offer a distinct advantage due to the straightforward and consistent nature of their paths. In contrast, AMRs are a more sophisticated, flexible, and cost-effective technology. They perform a variety of tasks at different locations with simple software adjustments and adapt effortlessly to changes in the environment.

The question that arises is the feasibility and effectiveness of implementing these robots on a large scale, working together in the same space. Coordinating the movement of a fleet of robots in limited areas is a complex task and its study is important to guarantee the growth of performance and

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within project GreenAuto, with reference 54.

efficiency of these vehicles in industrial environments.

This study aims to provide a comparative analysis of the characteristics and operational advantages and disadvantages of AMR and AGV robot fleets, with the objective of gaining critical insights into the operational efficiency, flexibility, scalability, and implementation challenges of these two types of fleets.

An essential element of this comparison is to evaluate the impact of path planning for mobile robots on their efficiency and effectiveness in a controlled and simulated environment. The path planning algorithm, Time Enhanced A* (TEA*) [5], was used for computing the path for the AGV fleet. TEA* is a graph-based algorithm that incorporates a third dimension, time, to the traditional A* algorithm. It calculates the minimum path over the temporal layers, considering the robots' positions and the changes in the environment.

In the case of the AMRs, the ROS Navigation Stack was employed. The ROS package move_base [6] is an integral component of the navigation stack and is responsible for guiding the robot to a designated goal while avoiding obstacles. The move_base node integrates a global and local planner to achieve its global navigation objective. Additionally, it maintains two cost maps, one for the global planner and one for the local planner. The global planner is responsible for generating a path from start to goal. However, it does not consider obstacles that were not previously mapped. The local planner ensures obstacle avoidance and smooth path execution, providing velocity commands for the robot.

II. Local Planning Algorithms

As previously stated, the move_base ROS package integrates a global and a local planner plugin. These plugins must adhere to an interface defined by the ROS node to be used [7]. As part of this study, an examination and comparison of the three local planners compliant with the defined interface is conducted. The objective of this study is to identify the optimal local planner for implementation in the AMR fleet. In the comparative analysis, the Global Planner ROS package was employed as the global planner algorithm.

This study examines the Base Local Planner, Dynamic Window Approach (DWA), and Timed Elastic Band (TEB). These planners compute a feasible trajectory for the robot, updating it dynamically based on environmental feedback. Modifying specific parameters allows these planners to adjust the robot's behavior, such as speed and safety margins. The computed local trajectory guides the robot toward its intended destination while continuously detecting and avoiding obstacles in real time, ensuring smooth and efficient navigation even in dynamic environments.

A. Dynamic Window Algorithm

The Dynamic Window Approach is a collision avoidance strategy that considers a robot's dynamics and constraints on velocity and acceleration [8]. It generates multiple velocity sets and simulates their trajectories for a short period of time. The robot then selects a trajectory that maximizes translational velocity and distance to obstacles while minimizing the angle to its goal, optimizing an objective function.

In ROS, the Base Local Planner package implements the DWA and Trajectory Rollout algorithms. It provides a controller that drives a mobile base, managing communication between high-level planning and low-level motion control. The planner generates a 2D grid map, assigning each cell a "cost" value based on traversability. The controller uses this cost information to determine local controls, guiding the robot toward its target [9]. The DWA Local Planner ROS package offers a modular DWA implementation and a more flexible y-axis variable for holonomic robots than the Base Local Planner's DWA.

B. Trajectory Rollout Algorithm

The Trajectory Rollout algorithm works similarly to the DWA algorithm, sampling linear and angular velocities, simulating trajectories at those velocities, and checking for collisions. The main difference between DWA and Trajectory Rollout lies in their respective methodologies for sampling the robot's control space for optimal velocity commands. Trajectory Rollout samples velocities over the entire simulation period, while DWA samples at just one step, making it more computationally efficient [10]. However, for robots with low acceleration limits needing smooth velocity transitions, Trajectory Rollout may perform better.

C. Timed Elastic Band Algorithm

The TEB Local Planner package implements the approach known as Timed Elastic Band, which augments the Elastic Band method with temporal information. This allows for the consideration of the robot's dynamic motion constraints and direct modification of trajectories, rather than paths [11].

The Timed Elastic Band (TEB) algorithm for mobile robot path planning is a multi-objective optimization approach that balances several constraints, including global path following, obstacle avoidance, velocity and acceleration limits, and non-holonomic kinematic restrictions. TEB optimizes trajectories to minimize execution time, maintain safe distances from obstacles, and comply with robot motion dynamics [12]. The algorithm uses a sparse optimization model, applying graph optimization techniques to solve this problem efficiently, forming a hyper-graph that connects robot states and time intervals as nodes, with constraints and objectives as edges.

To identify the most suitable local planner for the given scenario, a series of tests were performed, focusing on extracting two key metrics: mission duration and distance traveled. The mission involved navigating a path composed of alternating forward movements and turns, providing a comprehensive test for planner evaluation. In these tests, the TEB Local Planner outperformed the others, completing the mission 7.97% faster than the DWA Local Planner and 9.10% faster than the Base Local Planner. In terms of distance, the difference was less pronounced, with TEB following a 0.58% shorter trajectory than the DWA Local Planner and 1.58% shorter than the Base Local Planner.

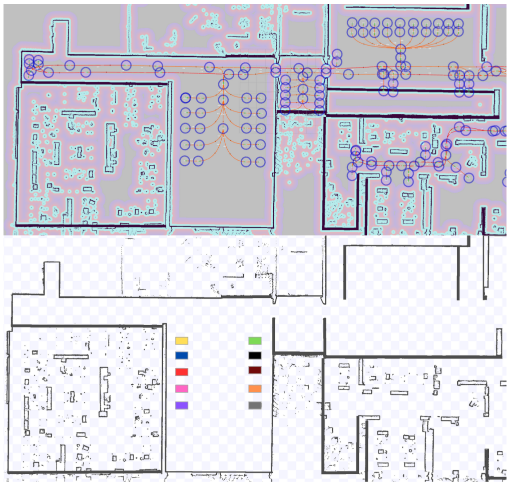

Fig. 1. Simulation environment. The top figure shows Rviz with the costmaps enabled, and the bottom figure displays the ROS Stage environment.

The TEB local planner is also highly recommended for dynamic environments. Its ability to optimize trajectories in real-time makes it well-suited for handling moving obstacles. This capability is especially relevant in the context of this study, which compares the performance of fleets with five and ten robots. Additionally, the TEB local planner offers a wide range of parameters for trajectory, obstacle handling, and optimization, providing enhanced adaptability to specific scenarios. For these reasons, the TEB local planner was selected as the local planner algorithm to be implemented in the AMRs for testing.

III. Experimental Evaluation of Robot Fleets

In order to facilitate a comparative analysis of the two fleets, a simulation environment was constructed. The Rviz and Stage simulation tools were utilized to visualize the representation of the environment. These tools enable the visualization of robot models, laser data, TF, and the map of the environment. Fig. 1 illustrates the simulation environment. The top image shows the Rviz interface, displaying the trajectories in a graph format with nodes and edges, while the costmap generated by the ROS Navigation package is highlighted in blue and pink. The bottom image presents the Stage environment, where the robots are depicted as colored rectangles in their initial positions.

The comparative tests between the AGV and AMR fleets will be carried out in this simulated environment, with both types of robots configured identically. Both AGVs and AMRs are differential-wheeled and equipped with two 270º field-of-view lasers. The primary difference lies in their navigation strategies. For the AGVs, paths are pre-defined using nodes and vertices generated by the TEA* algorithm. Conversely, the AMRs use the ROS navigation stack, particularly the move_base package. Their trajectories are dynamically computed using a global planner combined with the TEB local planner, enabling them to navigate autonomously without relying on pre-defined routes.

To compare the two fleets, a series of tests were conducted to assess the performance of AGVs and AMRs in navigation tasks. The tests focused on analyzing how well each type of robot followed its designated path and performed under different conditions, highlighting differences in behavior and strengths between trajectory-following AGVs and free-path navigating AMRs.

The Key Performance Indicators (KPIs) used for evaluating each mission were as follows:

- **Time:** The time required for the robot to travel from its start position to its mission endpoint;
- **Success rate:** The percentage of missions where robots successfully reached the destination;
- **Collision rate:** The proportion of missions where collisions occurred;
- **Average number of collisions:** The average number of robots involved in collisions during the mission.

The last two KPIs, collision rate and average number of collisions, were only assessed for fleets consisting of five and ten robots.

A. Comparison between one AGV and one AMR

Initial evaluations focused on individual assessments of both AGVs and AMRs. The goal was to observe and document their behavior in isolation, evaluating their performance without interference or disruption from other robots in the environment. These preliminary tests provided baseline data on how each type of robot navigated, responded to commands, and handled obstacles under controlled conditions. This step was critical to understanding their core functionality before introducing more complex scenarios involving multiple robots.

The tests included three different mission lengths, classified as short, medium, and long-distance missions, in which the robot moved from point A to point B. The start and end points and the trajectories taken by the robots are shown in Fig. 4.

For each robot, each mission was repeated three times, and the time it took to complete each trial was recorded. Table I shows the average time for each mission for both AGV and AMR. The task was straightforward for the robots, which achieved a 100% success rate in all three missions.

B. Comparison between a fleet of five AGVs and a fleet of five AMRs

In the second phase of testing, the number of robots was increased to five, forming a fleet. The tests involved a simple mission in which all robots were tasked with navigating from

TABLE I
Simulation time results for the three distance missions, for the AMR and AGV.

Missions	Average ToA for AMR	Average ToA for AGV
Short	00:35,41	00:40,38
Medium	00:46,38	01:02,70
Long	01:06,63	01:13,95

point A to point B. Navigation commands were issued to the entire fleet simultaneously. Both the AGV and AMR fleets were analyzed to determine how well the robots performed and behaved in a dynamic obstacle-filled environment.

The tests were conducted separately for the AGV fleet and the AMR fleet, with each fleet undergoing ten trials to ensure the reliability of the results. Neighboring goals were set in a linear array, maintaining a consistent distance from start to end for all robots, since it was not possible for all robots to reach the same goal at the same time.

Tables II and III display the results for the fleet of five AGVs and the fleet of five AMRs, respectively. In these tables, the shortest arrival time is highlighted in green, while the longest arrival time is highlighted in red.

The average duration of the mission for the AGVs was 02:38,2, while the duration of the mission for the AMRs was 01:36,5, indicating a difference of one minute between these two fleets to complete the mission. The AGVs did not clash, but the AMRs collided 4 times during the ten simulations. The success rate was 100% for both fleets.

In the AGV simulation, the arrival time of the last robot varied significantly, ranging from 04:09.44 in Simulation 1 to 04:38.62 in Simulation 3. This variation is due to the fact that mission goals were sent to all robots simultaneously. This overload caused ROS to process tasks at different times, resulting in random order arrivals to the TEA* algorithm. Since this algorithm prioritizes orders based on their arrival time, the priorities differed in each test, affecting the overall completion times.

In contrast, the AMR fleet showed remarkably consistent arrival times. During the simulations, it was observed that the same robot consistently arrived first, as indicated by its similar time of arrival. Furthermore, the arrival time of the last robot exhibited minimal variation, further confirming the overall consistency of the fleet.

C. Comparison between a fleet of ten AGVs and a fleet of ten AMRs

For the final phase of testing, a fleet of ten robots was deployed. The mission objective remained the same, requiring navigation from point A to point B. The map was expanded to accommodate the increased number of robots, increasing the number of adjacent goals to ten while maintaining a linear formation.

The simulation results for the AGV fleet are displayed in Table IV and the results for the AMR fleet are displayed in Table V. The shortest arrival time is highlighted in green and the longest arrival time is highlighted in red.

On average, the AGV fleet completed the mission in 04:23.2 seconds, while the AMR fleet completed it in 02:00.9 seconds, resulting in a significant time difference of 02:22.4. However, the AMR fleet had a 90% success rate due to the inability to compute a new trajectory after a collision, compared to the AGV fleet's perfect 100% success rate. The AMR fleet averaged 1.8 collisions per mission, with 80% of missions involving at least one collision. In contrast, the AGV fleet experienced no collisions.

IV. Discussion of Results

After thoroughly testing both fleets in different scenarios and extracting data based on the defined KPIs, the data can be analyzed to assess its impact on key performance areas such as operational efficiency, flexibility, and scalability of the AGV and AMR fleets.

Regarding time efficiency, AMRs consistently outperformed AGVs, demonstrating their superiority in dynamic environments by autonomously avoiding obstacles and adjusting their paths in real time. In contrast, AGVs are required to follow predefined routes that are not always the shortest path to the destination, resulting in longer execution times.

In the single-robot tests, particularly for the medium-distance mission, where a significant time difference between the AMR and the AGV can be seen in the third row of Table I, the results confirm the previous statement. The AMR takes a direct path to the goal, which is located behind its starting point, while the AGV follows the predefined route, first moving forward and then turning towards the goal, taking more time to complete the mission. This behavior can be seen in Fig. 4 in the second image.

During the test with a single robot, we observed its behavior when faced with an obstacle. The AGV effectively detected the obstacle and stopped to avoid a collision, but it could not

TABLE II
SIMULATION RESULTS FOR THE FLEET OF FIVE GUIDED NAVIGATION ROBOTS.

Simulation	First robot's ToA	Last robot's ToA	Number of Collisions
1	01:23,15	02:38,38	0
2	01:29,83	**02:48,10**	0
3	**01:30,04**	02:40,86	0
4	01:15,67	02:44,52	0
5	**01:15,24**	02:30,35	0
6	01:16,34	02:41,02	0
7	01:16,23	02:36,44	0
8	01:17,42	**02:27,86**	0
9	01:16,86	02:30,82	0
10	01:24,35	02:43,96	0

TABLE III
SIMULATION RESULTS FOR THE FLEET OF FIVE FREE NAVIGATION ROBOTS.

Simulation	First robot's ToA	Last robot's ToA	Number of collisions
1	01:07,67	01:36,89	0
2	01:07,81	01:37,10	0
3	01:07,77	01:34,58	0
4	01:07,60	01:36,86	0
5	01:08,22	**01:41,55**	1
6	01:07,89	01:35,61	0
7	01:08,23	01:39,32	1
8	**01:08,36**	**01:31,74**	0
9	**01:07,18**	01:37,48	1
10	01:07,56	01:33,59	1

carry out its task while the obstacle was present. In contrast, the AMR was able to detect the obstacle and dynamically re-route itself to avoid it.

After introducing fleets of five and ten robots into the experimental environments, a significant issue with collisions among the robots was encountered. This problem was particularly noticeable with the AMRs, where the number of collisions increased almost fivefold when the fleet size was increased from five to ten robots.

Another important observation is that the trajectory between the robot's start and end points passes through a critical choke point on the map, posing a significant challenge. This issue becomes even more pronounced when simulating fleets of five and ten robots, where effective management of this choke point is essential to avoid congestion. The localization of the choke points and the robots' behavior in navigating around them are illustrated in Fig. 2.

With AGVs, this problem is effectively solved. With a single entry and exit point, the system ensures that only one robot passes through at a time, preventing congestion. In addition, the AGV fleet is coordinated by fleet management software that, after receiving the planned path from the TEA* algorithm, continuously monitors the robots' trajectories, establishing priorities and preventing collisions to maintain an orderly flow.

However, the AMRs that rely on the ROS navigation stack do not have any capability for fleet management. As a result, multiple robots attempt to traverse the choke point at the same time. While their collision avoidance systems prevent direct collisions, they constantly maneuver around each other, causing delays and, in the worst cases, collisions.

To further evaluate the performance of the fleet, the environment map was modified to remove any choke points between the start and end points as seen in Fig. 3, resulting in a more open space for navigation. The same mission was performed with fleets of five and ten robots, and the results are summarized in Table VI.

The trajectories of the AGVs were left unchanged, maintaining the same graph structure, while the goal was to observe how the AMRs performed in the absence of choke points. For the fleet of five robots, the average mission

Fig. 2. Fleet of 5 AMRs (1) and 10 AMRs (2) navigating through a narrow corridor, creating congestion as they approach the entry of a choke point.

time improved from 01:36.5 to 01:09.1, a reduction of 27.3 seconds. Similarly, for the fleet of ten robots, the average mission time decreased from 02:00.9 to 01:26.2, a difference of 34.6 seconds. Overall, this represents a 28% improvement in mission duration for both fleets, with no collisions observed during this evaluation.

These tests revealed that choke points on the map significantly affect the performance of an AMR fleet, particularly in task completion times. In narrow corridors, congestion becomes a major issue, leading to delays. Without an optimized fleet management system to regulate traffic flow, the robots struggle to navigate efficiently. In contrast, in open areas, the AMRs experience minimal space competition, allowing them to move freely toward their objectives with fewer obstacles and delays.

V. CONCLUSIONS

This study provides a comparative analysis between a fleet of Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs) across various scenarios, focusing on operational efficiency, flexibility, and scalability. The AGVs used the TEA* algorithm for path planning, while the AMRs

TABLE IV
SIMULATION RESULTS FOR THE FLEET OF TEN GUIDED NAVIGATION ROBOTS.

Simulation	First robot's ToA	Last robot's ToA	Number of Collisions
1	01:21,85	**04:09,44**	0
2	01:24,64	04:16,58	0
3	01:26,34	**04:38,62**	0
4	**01:29,04**	04:35,75	0
5	**01:18,06**	04:15,07	0
6	01:27,07	04:11,01	0
7	01:22,54	04:21,08	0
8	01:23,79	04:33,83	0
9	01:26,72	04:27,80	0
10	01:21,30	04:23,26	0

TABLE V
SIMULATION RESULTS FOR THE FLEET OF TEN FREE NAVIGATION ROBOTS.

Simulation	First robot's ToA	Last robot's ToA	Number of Collisions
1	00:54,64	**02:10,57**	1
2	**00:55,88**	01:57,94	1
3	**00:53,23**	01:57,51	3
4	00:55,74	01:59,37	1
5	00:54,16	**01:52,45**	0
6	00:54,39	01:57,55	0
7	00:54,16	02:01,77	2
8	00:55,31	02:03,68	2
9	00:55,33	02:05,83	3
10	00:54,51	02:02,01	5

Fig. 3. Modified map for the tests with AMRs in an open environment.

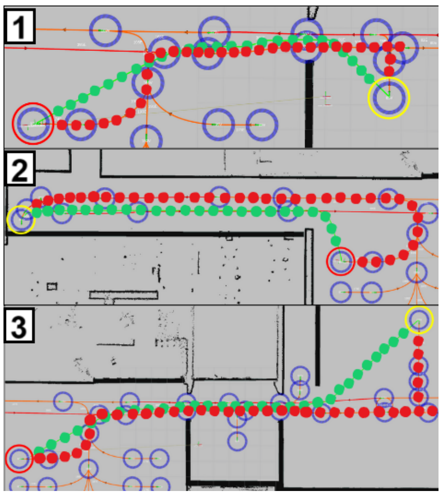

Fig. 4. Illustration of the trajectory followed by the robot in the short (1), medium (2) and long (3) missions. In green the trajectory computed by the TEB Local Planner and in red the trajectory from TEA* algorithm.

used the global planner in conjunction with the TEB local planner from the ROS navigation stack's move_base package.

A preliminary study of the local planning algorithms provided by the move_base ROS package was conducted, in order to choose the best option considering the environment. The individual robot tests established a baseline for subsequent experiments with fleets of five and ten robots.

Results revealed that AMRs consistently exhibited superior time efficiency in dynamic environments compared to AGVs, which followed predefined routes, leading to longer execution times. Specifically, AMRs demonstrated a 39% improvement in mission duration for a fleet of five robots and a 54% improvement for a fleet of ten robots. Notably, AGVs successfully avoided collisions, emphasizing the advantages of their fleet management system, whereas AMRs faced increasing collision rates as fleet size grew.

TABLE VI
SIMULATION RESULTS FOR A FLEET OF FIVE AND TEN AMRS IN AN OPEN ENVIRONMENT

AMR Fleet Size	First Robot's average ToA	Last robot's average ToA	Number of Collisions
5	00:58,95	01:09,13	0
10	00:51,77	01:26,23	0

Additionally, AGVs effectively managed choke points with fleet management software to prevent congestion, whereas AMRs struggled without coordination, causing delays. Removing choke points significantly improved AMR performance, highlighting that open spaces enhance navigation efficiency for AMRs.

The findings of this study serve to reinforce existing knowledge regarding the operational efficiencies of AGVs and AMRs, while simultaneously providing new insights into fleet management practices. The capacity of AGVs to avoid collisions and sustain an orderly navigation flow is indicative of the efficacy of their fleet management system, thereby suggesting a potential area for improvement in AMR coordination strategies.

These insights are crucial for guiding the design of future robotic systems and highlight the importance of tailored navigation solutions that consider the specific environments in which these robots operate. Ultimately, this study lays the groundwork for future research aimed at enhancing the performance of AMRs in congested spaces and exploring hybrid systems that can leverage the strengths of both AGVs and AMRs.

REFERENCES

[1] MiR, "Agv vs amr - what's the difference?" mobile-industrial-robots.com. Accessed: July 18, 2024. [Online.] Available: https://mobile-industrial-robots.com/blog/agv-vs-amr-whats-the-difference.

[2] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721000217

[3] X. Zhao and T. Chidambareswaran, "Autonomous mobile robots in manufacturing operations," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–7.

[4] G. Ullrich and T. Albrecht, *History of Automated Guided Vehicle Systems*. Wiesbaden: Springer Fachmedien Wiesbaden, 2023, pp. 1–26. [Online]. Available: https://doi.org/10.1007/978-3-658-35387-2_1

[5] J. Santos, P. Costa, L. F. Rocha, A. P. Moreira, and G. Veiga, "Time enhanced a*: Towards the development of a new approach for multi-robot coordination," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 3314–3319.

[6] "move_base," wiki.ros.org. Accessed: July 17, 2024. [Online.] Available: http://wiki.ros.org/move_base.

[7] "nav_core," wiki.ros.org. Accessed: July 18, 2024. [Online.] Available: http://wiki.ros.org/nav_core.

[8] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[9] B. P. Gerkey, "Planning and control in unstructured terrain," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:12848641

[10] "base_local_planner," wiki.ros.org. Accessed: July 18, 2024. [Online.] Available: http://wiki.ros.org/base_local_planner.

[11] C. Roesmann, W. Feiten, T. Woesch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.

[12] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *2013 European Conference on Mobile Robots*, 2013, pp. 138–143.