Coordinated Planning and Control of Two Mobile Robots for Joint Cargo Transportation

Borna Paro, Jelena Rogina, Luka Petrović, Marija Seder, Ivan Marković*

***Abstract*—Automated cargo transport is a frequent task in intralogistics and industrial scenarios, but is usually carried out by a single robot, thus limited to its payload capabilities. In this paper, we present a method for coordinated motion planning and control of two autonomous mobile robots for the joint transport of cargo that is too large for a single robot. We consider the two robots with a large cargo on top of them as an any-shape single kinematic chain with cargo. We plan an optimal collision free path for the entire differential drive kinematic chain that ensures the transport of the cargo to the desired target location. The implemented algorithm sends control commands to each robot separately, which achieves the coordinated execution of the planned path. Finally, the proposed method is validated in simulated and real-world scenarios.**

***Index Terms*—autonomous mobile robots, ROS, path planning, coordinated motion planning, differential drive, kinematic chain**

I. Introduction

Motion planning in robotics is the process of determining a feasible and optimal path between two points in an environment and calculating the velocities required to follow that path while avoiding obstacles and meeting specific performance criteria, such as minimizing travel time or energy consumption. It is essential for autonomous robots and is widely used in various robotic systems, including mobile robots, drones, and robotic manipulators. In mobile robotics, path planning involves searching the robot's environment and planning a path to the goal point before execution, considering all robot limitations, such as size and kinematic and dynamic constraints. Effective path planning ensures that each robot navigates safely, maintains a clear route, and avoids obstacles.

Path planning is a challenging problem due to the complexity of achieving efficient, safe, and collision-free navigation, especially in dynamic, cluttered, unpredictable, or partially unknown environments [1]. Navigating in such conditions requires constant adaptation and real-time decision-making. Moreover, it often involves optimizing for multiple conflicting criteria, such as minimizing travel time, energy consumption, or path smoothness while ensuring safety [2]. Finding a solution that balances these objectives is computationally intensive, particularly in real-time applications. When multiple robots are involved, the complexity increases further because the robots must coordinate their movements to avoid collisions with each other and obstacles. In scenarios where two robots are jointly carrying cargo, maintaining a fixed distance between the robots is essential to prevent the cargo from falling off. Despite these difficulties, the multi-robot path planning problem is worth addressing because it offers a novel approach to transporting cargo, where cargo of the greater weight can be transported using two smaller, more accessible, and cheaper robots.

Transporting large cargo is a frequent task in industrial and intralogistic settings, where items need to be moved from storage areas to locations where they are needed. Automating this process with robots can improve efficiency and allow human workers to focus on more complex tasks. Moreover, if multiple robots are available that can transport cargo, but the cargo is too large or heavy for a single robot, using two smaller robots collaboratively is more cost-effective than purchasing a single larger, more expensive robot capable of handling the load alone. Several methods have been proposed to address multi-robot cooperative tasks. The authors in [3] presented a method for multi-robot formation control in dynamic environments that avoids collisions with static and moving obstacles. In [4], a method was introduced that allows two wheeled mobile robots—a leader and a follower—to navigate through known environments while cooperatively carrying an object. A motion planning method for multiple mobile robots for cooperative transportation of a large object in a three-dimensional environment was presented in [5]. A distributed leader–helper architecture for teams of two autonomous mobile robots that jointly transport large payloads while avoiding collisions with obstacles (either static or dynamic) was introduced in [6]. Most of these methods utilize some form of leader-follower configuration to solve the problem of multi-robot cargo carrying tasks. However, none of them model the robots and cargo as a tightly-coupled kinematic chain with differential drive kinematics.

In this paper, we propose a motion planning method for an any-shape differential drive kinematic chain that consists of two robots and an any-shape large cargo that they are carrying. In our approach, the two robots and the cargo are considered as a single kinematic structure with differential drive kinematics, where the left wheel velocity of the structure corresponds to the linear velocity of the left robot, and the right wheel velocity corresponds to the linear velocity of the right robot. This modeling allows the robots to move independently, with motion commands for each robot depending only on the desired poses of the kinematic chain structure along the path, ensuring collision-free motion from the start to the goal. Each robot node calculates fewer parameters, resulting

This research has been funded by the H2020 project AIFORS under Grant Agreement No 952275 and supported by the Croatian Science Foundation under contract No DOK-NPOO-2023-10-7766.

* Authors are with the University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics, Croatia. {name.surname}@fer.unizg.hr

in faster computations, and they can operate independently without explicit coordination, relying on the kinematic chain model to maintain the necessary formation. To validate the proposed method, we create a case study involving two Clearpath Husky robots collaboratively transporting a large cargo. Experiments are conducted both in simulation and in a real-world environment. The results showcase the feasibility and advantages of using two smaller robots to transport large cargo, highlighting the potential for cost savings and increased flexibility in industrial applications.

II. The Proposed Method

A. Overview

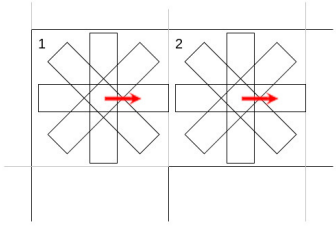
Traditional path planning algorithms often approximate a robot's footprint with a circle to simplify collision checking and pathfinding [7]. This approximation allows obstacles to be inflated by the robot's radius, reducing the robot to a point in the environment, which simplifies the planning to a graph search problem. While computationally efficient, this assumption can be overly conservative for non-circular robots or composite structures, potentially limiting their mobility in confined spaces. In our setup, where two robots carry a large cargo, approximating the entire structure with a circle results in an unnecessarily large footprint. This oversimplification can prevent the planner from finding feasible paths in environments where the structure could navigate with precise maneuvering. Additionally, most motion planning algorithms are designed for single-drive robots and do not consider the coordinated control required when multiple robots are involved in moving a single structure.

Therefore, we need a method that plans for the entire structure while generating individual commands for each robot. Our approach is based on a global planner that operates on any-shape differential drive robots, considering the constraints of differential drive kinematics [8]. The planner computes a path for the entire structure, while our local planner calculates individual velocities for each robot to ensure the structure follows the desired motion.

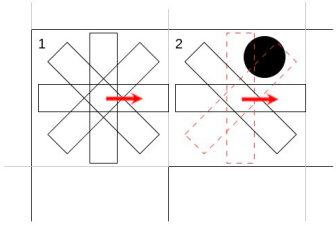
B. Global Path Planning

In our approach, the two robots and the cargo are considered as a single structure. The primary task of the global path planner is to generate a collision-free path from the current pose to the goal pose of the structure within a known environment. Unlike traditional planners that simplify the robot's shape, our planner considers the exact footprint of the structure, which includes the two robots and the cargo. This detailed consideration allows for more accurate path planning in environments with tight spaces and obstacles.

Our global planner first receives a static map of the environment and the structure's footprint. The map is discretized into a grid of cells, each with dimensions defined by a parameter $cell_size$. The choice of $cell_size$ depends on the environment; smaller values are used for maps with narrow passages or dense obstacles to preserve navigable paths. For each cell in the grid, we compute a set of admissible orientations. An admissible orientation is an angle at which the structure can be placed in the center of the cell without colliding with any obstacles. To determine these orientations, we consider a predefined list of possible orientations that the structure can achieve. For each orientation, we perform collision checking by placing the structure's footprint at the cell center and rotating it to the given orientation. If there is no collision, the orientation is added to the cell's list of admissible orientations.



(a) Example without an obstacle



(b) Example with an obstacle

Fig. 1: Admissible orientations for cells in the grid. (a) Without obstacles, the structure can move from cell 1 to cell 2 at orientation $\theta = 0°$ (the $0°$ orientation is defined as the direction indicated by the red arrow). (b) With an obstacle, the structure cannot adopt orientation $\theta = 0°$ in cell two without collision, so movement between cells is not possible. For simplicity, only a few orientations are shown, and the cell size is exaggerated for clarity.

Collision checking is a critical component of our planner. For each orientation, we check if any part of the structure's footprint intersects with obstacles. This is done by tracing lines from points on the footprint to the cell center and verifying that these lines do not pass through occupied areas. If any line intersects an obstacle, the orientation is considered inadmissible for that cell. Fig. 1 illustrates the process of creating the graph and determining admissible movements between cells. Once the admissible orientations are computed for all cells, we construct a search graph. Each cell represents a node in the graph, and edges are defined based on the ability of the structure to move from one cell to a neighboring cell without collision. Specifically, an edge between two cells exists if the current cell and the neighboring cell both have the required orientation in their list of admissible orientations or the structure can rotate in place in the current cell to the required orientation if necessary. Since the structure has differential drive kinematics and cannot move laterally, we consider movements to neighboring cells that are feasible given these constraints. We use Dijkstra's algorithm [9] to

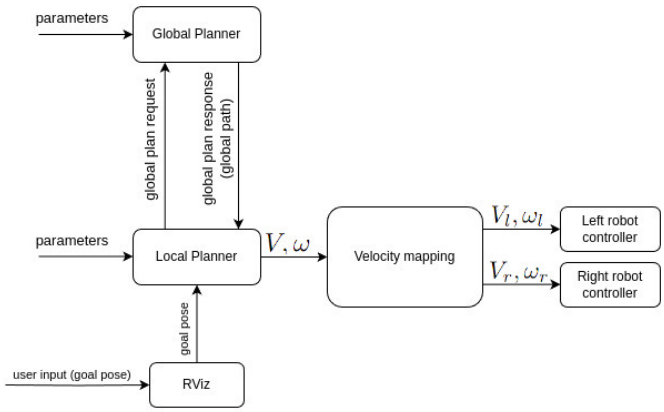


Fig. 2: Abstracted system architecture integrating global and local planners with individual robot control.

search the graph for the shortest path from the start cell to the goal cell. The result is a sequence of cells and orientations that define the global path for the structure.

C. Motion Planning

Based on the global path obtained from the planner, we compute the velocity commands required for the structure to follow the path using the Dynamic Window Approach (DWA) [10], [11]. DWA is a local reactive collision avoidance technique that searches for optimal velocity commands directly in the velocity space, considering the robot's kinematic and dynamic constraints.

We first reduce the search space to velocities that the structure can achieve within the next sampling interval Δt, based on its current velocity (v_c, ω_c) and acceleration limits. Then, for each candidate velocity tuple (v, ω), we simulate the trajectory and check for potential collisions. A velocity is considered safe if the structure can stop along the trajectory before encountering any obstacles. We select the velocity tuple that maximizes an objective function:

$$\Gamma(v,\omega) = \sigma(\alpha \cdot heading(v,\omega) + \beta \cdot dist(v,\omega) + \gamma \cdot vel(v,\omega)) \tag{1}$$

, which balances progress towards the goal, distance to obstacles and forward velocity. This balance allows the structure to navigate efficiently while avoiding collisions. The function σ is responsible for smoothing the weighted sum of the three components and creating trajectories more side-clearenced from the obstacles. For more details please see the original paper [10].

Since DWA is designed for a single differential drive robot, we need to transform the computed velocities (V, ω) for the structure into individual commands for each robot. We model the two robots and the cargo as a single differential drive system, where each robot corresponds to a "wheel" of the structure. Using the differential drive kinematic equations [12]:

$$V = \frac{V_L + V_R}{2} \tag{2}$$

$$\omega = \frac{V_R - V_L}{b}, \tag{3}$$

where V_L and V_R are the linear velocities of the left and right robots, respectively, b is the distance between the two robots, while V and ω are the desired linear and angular velocities of the entire structure. Solving these equations for V_L and V_R, we obtain:

$$V_L = V - \frac{b}{2} \cdot \omega \tag{4}$$

$$V_R = V + \frac{b}{2} \cdot \omega. \tag{5}$$

These velocities are then sent to each robot to execute.

D. System Architecture

The overall system architecture integrates the global and motion planning components, as depicted in Fig. 2. The system receives the environment map, the structure's footprint, the starting and goal poses of the structure, and laser scans from each robot. The global planner then computes the global path based on the input parameters, considering the structure's exact footprint and differential drive constraints. The local planner receives the global path and computes the velocity commands for the structure using DWA. The commands are then transformed into individual velocities for each robot. The individual robots execute the velocity commands, coordinating their movements to transport the cargo along the planned path.

This architecture allows for efficient and coordinated navigation of the two robots carrying a large cargo, enabling them to move through environments that would be challenging for a single robot or methods that use simplified approximations of the robot's footprint. By modeling the robots and cargo as a tightly coupled kinematic chain with differential drive kinematics, our method ensures that the robots can operate independently while maintaining the necessary coordination. Each robot can calculate its own velocities based on the desired motion of the structure, potentially resulting in faster computations and robust performance without explicit communication between the robots.

III. Experimental results

In this section, we present a case study to demonstrate the effectiveness of the proposed method, which models two

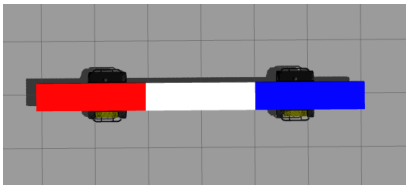


Fig. 3: Top-view of the simulation setup with two Husky robots and cargo.

robots and a cargo as a tightly coupled kinematic structure with differential drive kinematics. The developed methods are implemented using the Robot Operating System (ROS) with Python and C++. We evaluate our approach through both simulation and real-world experiments using two Clearpath Robotics Husky robots.

A. Simulation experiments

The simulation environment is set up using Gazebo and visualized in RViz. The cargo is represented by a long beam placed atop two Husky robots, forming the tightly coupled structure described in our method. A top-view of the simulation setup is shown in Fig. 3. Each robot is equipped with a laser scanner essential for map creation, localization, and navigation. For simplification, we utilize the ground truth poses provided by Gazebo. As depicted in Fig. 4, in RViz we can see the green line that represents the global path calculated by the planner, the blue curve that indicates the trajectory of the structure based on the current linear and angular velocities, and the green rectangle that depicts the footprint of the structure. The robots and cargo are placed in a custom environment designed to test navigation through various scenarios, including narrow corridors and obstacles, as shown in Fig. 5.

The simulation results demonstrate that the structure successfully navigates from the starting point to the goal, maneuvering through complex environments without collisions. A series of snapshots illustrating the structure's movement through the environment is presented in Fig. 5. These results validate the effectiveness of the proposed kinematic model and control algorithms in a simulated setting.

B. Real-world experiments

To validate the proposed method in a practical setting, we conducted real-world experiments using two physical Husky robots. The robots are equipped with Intel RealSense D435i depth cameras, which are utilized for mapping, localization, and navigation. We opted for depth cameras over laser scanners due to their ease of use and flexibility. Since the depth cameras provide point cloud data, we convert these into laser scans compatible with the local planner's requirements. For localization, we employ the Adaptive Monte Carlo Localization (AMCL) package, as ground truth poses are not available in the real-world environment. Due to space constraints on the robots (laptops are mounted on top for control purposes), we use a long metal bar to represent the cargo. This choice

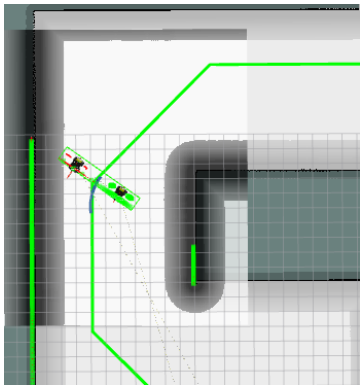


Fig. 4: RViz visualization of the simulation, showing the global path, structure trajectory, and footprint.

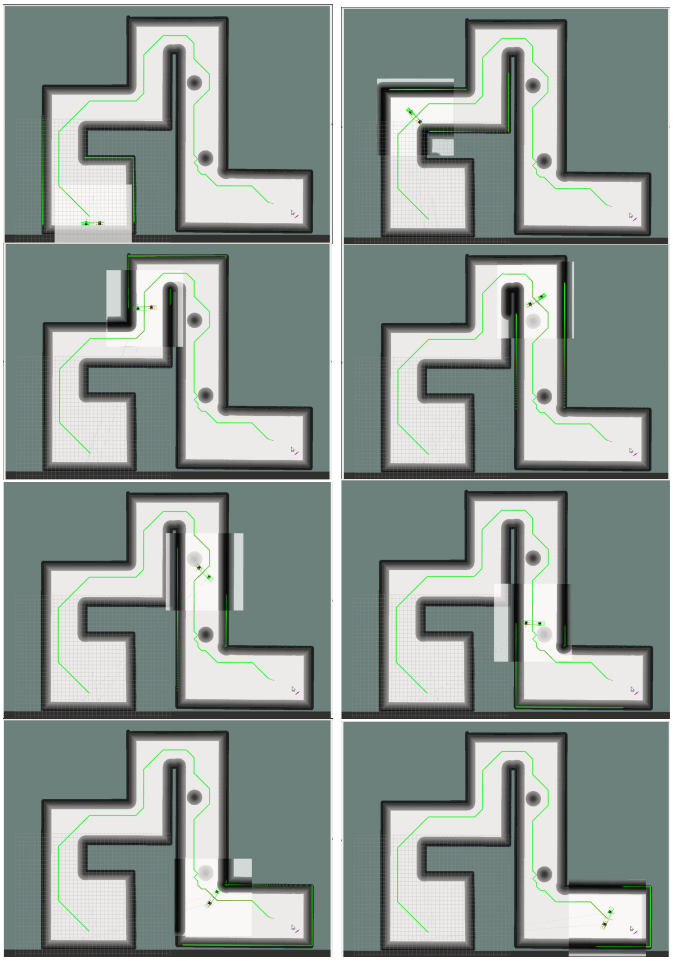
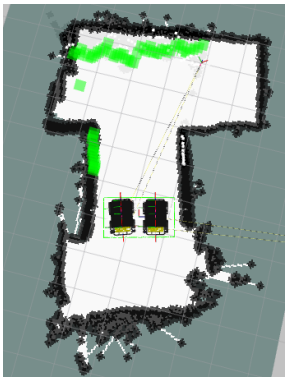


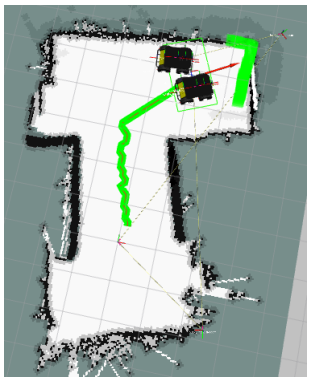
Fig. 5: Sequence of snapshots showing the structure successfully navigating from the start to the goal in simulation.

ensures that the cargo does not obstruct the cameras' field of view while effectively increasing the structure's footprint, similar to having actual cargo on top.

Fig. 6 illustrates the RViz visualizations of the real-world experiment at the start and end positions. In Fig. 6a, the starting pose and environment are depicted, where green dots



(a) RViz environment at the start of the experiment



(b) RViz environment after reaching the goal

Fig. 6: RViz visualizations of the real-world experiment at the start and end positions.



Fig. 7: Sequence of images showing the structure successfully moving from the start to the goal in the real-world environment.

represent laser measurements from the left Husky and white dots represent those from the right Husky. After setting a goal pose, the global planner computes a path, and the local planner successfully navigates the structure to the goal without collisions. The final state upon reaching the goal is shown in Fig. 6b. To further illustrate the effectiveness of the proposed method, Fig. 7 presents a sequence of images showing the structure's movement from the starting point to the goal in the real-world environment. Despite challenges such as sensor noise and environmental uncertainties, the structure successfully navigates to the target pose without collisions, validating the practical applicability of our approach.

IV. Conclusion

In this paper, we have presented a method for coordinating two mobile robots to transport large cargo from a starting position to a goal position while avoiding collisions with the environment. By modeling the two robots and the cargo as an any-shape single kinematic structure with differential drive kinematics, we implemented algorithms for global path planning and motion control. Our approach was validated through a case study with two Husky mobile robots, both in simulation and real-world environment, demonstrating the feasibility and effectiveness of the proposed method.

Each robot is equipped with its own laser scanner, but only the measurements from the right laser are utilized for motion planning. Future work could explore combining the data from both laser scanners and transforming them as if originating from the center of the structure. This would provide the local planner with more comprehensive environmental information, potentially enabling the computation of more optimal paths.

Another limitation is the discretization of the environment for faster global path planning, which can result in paths with sharp turns over short intervals. These abrupt changes can be suboptimal for smooth motion. A future enhancement could involve implementing a smoothing algorithm to refine the generated paths, ensuring both optimality and smoother motion.

References

[1] N. Correll, B. Hayes, C. Heckman, and A. Roncone, *Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms*, 1st ed. MIT Press, Cambridge, MA, 2022.

[2] J. A. Abdulsaheb and D. J. Kadhim, "Classical and heuristic approaches for mobile robot path planning: A survey," *Robotics*, vol. 12, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2218-6581/12/4/93

[3] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017. [Online]. Available: https://doi.org/10.1177/0278364917719333

[4] D. Mohan and A. Vivek, "Navigation of two wheeled mobile robots cooperatively carrying an object," in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, 2017, pp. 1–7.

[5] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 223–237, 2003.

[6] T. Machado, T. Malheiro, S. Monteiro, W. Erlhagen, and E. Bicho, "Multi-constrained joint transportation tasks by teams of autonomous mobile robots using a dynamical systems approach," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3111–3117.

[7] E. G. Tsardoulias, A. Iliakopoulou, A. Kargakos, and L. Petrou, "A review of global path planning methods for occupancy grid maps regardless of obstacle density," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 829–858, Dec 2016. [Online]. Available: https://doi.org/10.1007/s10846-016-0362-z

[8] M. Dakulović, C. Sprunk, L. Spinello, I. Petrovic, and W. Burgard, "Efficient navigation for anyshape holonomic mobile robots in dynamic environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2644–2649.

[9] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *1959 Numerische Mathematik 1*, 1959, pp. 269–271.

[10] 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.

[11] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 1986–1991.

[12] G. Klancar, A. Zdesar, S. Blazic, and I. Skrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*, 1st ed. USA: Butterworth-Heinemann, 2017, ch. 2.2.1.