

Article

Dual-Quaternion-Based SLERP MPC Local Controller for Safe Self-Driving of Robotic Wheelchairs

Daifeng Wang ^{1,*} , Wenjing Cao ² and Atsuo Takanishi ^{1,3}¹ Faculty of Science and Engineering, Waseda University, Tokyo 162-0056, Japan; takanisi@waseda.jp² Department of Engineering and Applied Sciences, Sophia University, Tokyo 1028554, Japan; wenjing-cao@sophia.ac.jp³ Humanoid Robotics Institute, Waseda University, Tokyo 1690056, Japan

* Correspondence: daifeng@fuji.waseda.jp

Abstract: In this work, the motion control of a robotic wheelchair to achieve safe and intelligent movement in an unknown scenario is proposed. The primary objective is to develop a comprehensive framework for a robotic wheelchair that combines a global path planner and a model predictive control (MPC) local controller. The A* algorithm is employed to generate a global path. To ensure safe and directional motion for the wheelchair user, an MPC local controller is implemented taking into account the via points generated by an approach combined with dual quaternions and spherical linear interpolation (SLERP). Dual quaternions are utilized for their simultaneous handling of rotation and translation, while SLERP enables smooth and continuous rotation interpolation by generating intermediate orientations between two specified orientations. The integration of these two methods optimizes navigation performance. The system is built on the Robot Operating System (ROS), with an electric wheelchair equipped with 3D-LiDAR serving as the hardware foundation. The experimental results reveal the effectiveness of the proposed method and demonstrate the ability of the robotic wheelchair to move safely from the initial position to the destination. This work contributes to the development of effective motion control for robotic wheelchairs, focusing on safety and improving the user experience when navigating in unknown environments.

**Citation:** Wang, D.; Cao, W.;

Takanishi, A. Dual-Quaternion-Based SLERP MPC Local Controller for Safe Self-Driving of Robotic Wheelchairs.

Robotics **2023**, *12*, 153. [https://](https://doi.org/10.3390/robotics12060153)doi.org/10.3390/robotics12060153

Academic Editors: Bruno Brito and Giorgos Mamakoukas

Received: 19 October 2023

Revised: 6 November 2023

Accepted: 11 November 2023

Published: 13 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: robotic wheelchair; motion control; dual quaternions; model predictive control (MPC); safe self-driving

1. Introduction

The rapid growth of the global elderly population poses significant challenges. Studies showed that in 2022, people aged over 65 years old accounted for nearly 10% of the world's population, and is expected to reach 16% in 2050 [1]. In Japan, which leads globally in the aging trend, the elderly represented 26% of its population in 2014 [2], with forecasts estimating an increase to 38.8% by 2050 [3]. The aging process brings various mobility impairments due to musculoskeletal issues, neurological issues, chronic conditions, medication impacts, and psychological factors. These impairments can diminish the life quality and daily independence of the elderly, increasing dependency on long-term care, which adds economic strain and contributes to labor shortages. To address these concerns and enhance the quality of life for the aging population, robotic wheelchairs have emerged as a promising assistive innovation, offering potential solutions. While advancements in robotic wheelchair technology have been notable in recent years [4,5], ensuring the safety and autonomous operation of these devices remains a critical challenge. Enhancing wheelchair intelligence to minimize the reliance on manual control is key to improving the quality of life for the elderly who are experiencing mobility impairments.

The early research focused on stability control systems designed for robotic wheelchairs [6]. The primary objective of a robotic wheelchair is to determine an optimal or suitable path

between a starting point and a destination point, considering various constraints and objectives. The navigation task is similar to the common path-planning problem in traffic management, which requires determining the optimal path that satisfies a series of requirements [7]. Over the years, many path-planning methods have been developed for various academic and practical applications [8,9]. Rapidly Exploring Random Trees (RRT) is an algorithm frequently used in dynamic and complex situations. Its primary purpose is to efficiently explore the search space and to generate feasible paths [10]. The utility of RRT was further demonstrated in experiments involving cooperative robotics, where it proved successful in enabling a group of robots to push a box coordinated without collisions [11]. Similarly, the genetic algorithm (GA) has established itself as an effective optimization tool for path planning. Studies have shown its effectiveness in environments with stationary and mobile obstacles [12]. Although the above-mentioned strategy has a wide search range, rapid search speed, and high computational efficiency, the high calculation cost and slow convergence speed are two intractable shortcomings.

Additionally, the graph search algorithm is another widely used methodology for path planning, which has proven to be effective in navigating complex environments. One commonly used graph search algorithm is Dijkstra's algorithm, which guarantees finding the shortest path in terms of distance or cost [13]. Another popular algorithm is the A* algorithm, which extends Dijkstra's algorithm by introducing a heuristic function that estimates the remaining cost from each node to the target. This heuristic helps guide the search towards the target and can significantly improve the search efficiency [14]. The A* algorithm was employed for mobile robots in indoor disaster areas, and the experimental results demonstrated the effectiveness of the approach in enabling the robot to successfully navigate and explore the indoor disaster area [15]. In [16], the simulation results in a Mars-like scenario validated the applicability of the A* algorithm in minimizing the traveling time and energy consumption, while maintaining the safety of exploration robots. In addition to its applications in robotic path planning, Santos J et al. [17] introduced an A*-based planning and scheduling approach designed for the management of multiple automated guided vehicles within an industrial environment. It provided an efficient means for planning routes and schedules, optimizing operations in complex industrial environments. Studies indicate that the A* algorithm can be used to balance optimality and efficiency, with its heuristic-guided search and efficient data structures making it particularly fit for dealing with path-planning problems in extensive environments.

However, the limitation of these path-planning strategies is their inability to consider the dynamic characteristics of the control plant. Model predictive control (MPC) is a highly effective technique employed in motion control across various domains. By formulating an optimization problem over a defined prediction horizon, MPC enables the determination of an optimal sequence of control actions that minimize the given cost function, while considering system dynamics and constraints. The existing literature has shown that MPC is especially suitable for dynamic environments and complex robot dynamics scenarios. Its real-time adaptability and optimality make it a powerful tool for effectively generating optimal paths and successfully navigating complex environments [18,19]. Previous research by Rosmann C et al. [20] has demonstrated the effectiveness of an innovative approach to online motion planning using nonlinear MPC with non-Euclidean rotation groups in various scenarios. This method addresses the challenges posed by environments with non-Euclidean geometry and complex robot dynamics. Farina M conducted a comprehensive investigation on the use of distributed MPC for motion coordination in unicycle autonomous robots. Their research demonstrated the effectiveness of this control strategy in enhancing the performance and efficiency of robots in various tasks and scenarios [21]. In [22], the successful application of dynamic nonlinear MPC for both path planning and tracking control in unmanned transportation systems was highlighted. Meanwhile, Ref. [23] introduced an MPC-based framework that emphasized the safety of robots in unpredictable environments, demonstrating their effectiveness in navigating dynamic obstacles and uncertainties. Bjelonic M et al. [24] also introduced an optimization-based framework uniting offline motion libraries with online MPC to enable robots with legs and

wheels to perform complex locomotion skills. It combined long-time horizon planning and short-time horizon solutions, allowing smooth transitions between diverse motions while ensuring effective feedback control and resilience to disturbances. In conclusion, these results highlight the potential of the MPC scheme for practical applications in motion control for robotic systems. The demonstrated effectiveness of the MPC approach in generating optimized and safe trajectories reveals its viability and applicability in various real-world scenarios.

The development of robot control requires complex mathematical structures to describe their transformation in space. Among many mathematical tools, the dual quaternion as a particularly effective tool has attracted much attention in the field of robotics. It has the ability to represent rotation and translation simultaneously. In [25], the use of dual quaternion to help robots move precisely in position and direction was explored. The study not only introduced the mathematical principle behind this method but also validated its control ability through experiments with real robots and confirmed that it was a promising method for better robot control. Then, a reliable library for robot modeling and control using dual quaternion was further developed, which was popularly used recently [26]. This library capitalizes on the strengths of dual quaternions in representing 3D transformations, offering a robust toolbox for modern robotic applications. A dual-quaternion-based robot forward kinematics algorithm in the Robot Operating System (ROS) was implemented in [27]. The proposed implementation provided an accurate and efficient solution for solving forward kinematics problems in robotic systems, thereby enhancing the capabilities of ROS for robot control and handling. In conclusion, dual quaternions can be used in robotic control to consider both the position and the direction of the action, thus improving the safety of robot operation and realizing directional movement. Therefore, in this work, we make use of the unique characteristics of dual quaternion, combined with an MPC control scheme, to enhance the motion of robotic wheelchairs.

In this work, a robotic wheelchair system based on ROS is built first. Then, a hierarchical motion control strategy for the robotic wheelchair that ensures safe and directional movement in an unknown environment is presented. The proposed strategy includes a global path planner, which determines an optimal path in the overall environment, and an MPC local controller, which takes into account the specific dynamics and constraints of the robotic wheelchair. Particularly, a novel approach that combines a dual-quaternion-based spherical linear interpolation (SLERP) technique with the MPC scheme is used in the local controller. The dual quaternion can effectively capture the complex spatial transformations required for robot control, which enables the robotic wheelchair to achieve reliable and directional navigation. This integration allows for smooth and continuous transitions between different poses of the robotic wheelchair during movement. The dual-quaternion-based SLEPP MPC local controller is employed to address disturbances in the local map, effectively solving the motion control issue of local obstacle avoidance and path optimization. Finally, the experimental results in the ROS simulation environment and real-world scenarios demonstrate the effectiveness of the proposed control strategy in achieving self-driving and safe navigation in unknown environments for robotic wheelchairs. Furthermore, the experimental results revealed the prospective capability of our proposed control strategy in addressing dynamic obstacle avoidance challenges that arise in the future.

The remainder of the paper proceeds as follows. In Section 2, the considered robotic wheelchair is first introduced and formulated by the so-called unicycle model, and then the hierarchical motion control problem is proposed. In Section 3, the comprehensive framework is proposed. The path planning strategy using the A* algorithm to generate a global path is first described. In the following parts, this section intends to achieve the MPC local path planner, introducing a dual-quaternion-based SLERP technique. In Section 4, the validation results and discussions are presented. Finally, the conclusions are summarized in Section 5.

2. System Model

2.1. Robotic Wheelchair Model

In this section, we introduce the kinematic model of the robotic wheelchair, specifically focusing on the modeling of the wheelchair produced by Whill Company. The hardware configuration of our proposed robotic wheelchair system is depicted in Figure 1. The wheelchair's rotation diameter is 76 cm, and its 3D dimensions are 98.5 cm in length, 55.4 cm in width, and 94.5 cm in height. The battery of the wheelchair can cover a travel distance of up to 18 km. A 3D-LiDAR sensor (RS-Helios 5515) is mounted to the left front of the wheelchair, enabling the environmental scanning. The laser beams of the sensor can cover a vertical field of view of 70° with a vertical resolution of 1.33°. The vertical scanning range is depicted on the left side of Figure 1. A battery located under the seat of the wheelchair provides power to the 3D-LiDAR and PC.

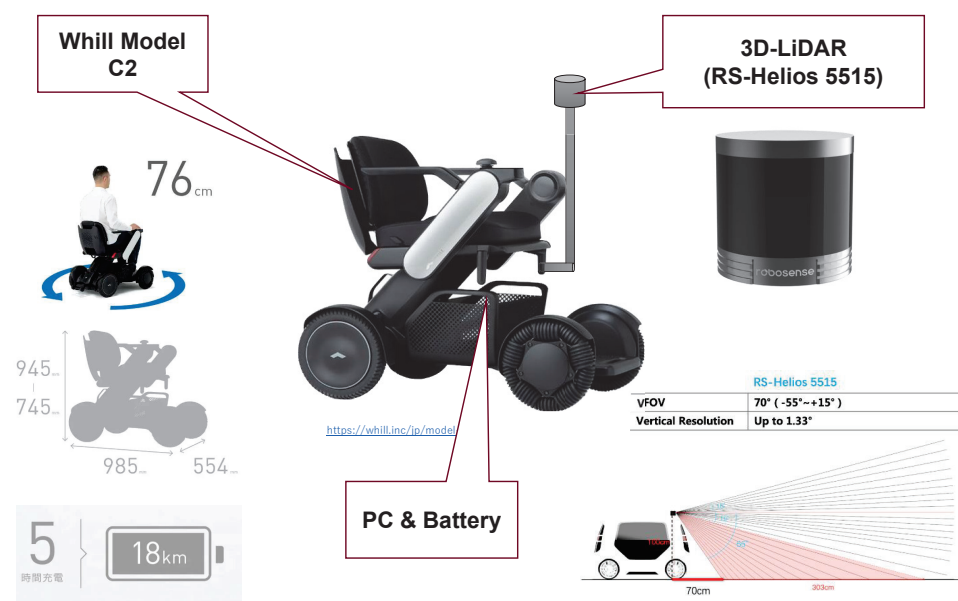


Figure 1. The hardware system of the proposed robotic wheelchair.

The Whill Model C2 wheelchair has four wheels. The two rear wheels are the driving wheels, governing both the translation and rotation movement of the wheelchair. The two front wheels are driven wheels that support the wheelchair since the front wheels are Omni Wheels that ensure the rotation center of the wheelchair is located on the center of the rear axle. The motion of the wheelchair shares similarities with a unicycle-type mobile robot. The abstract model of the wheelchair is depicted in the left subsection of Figure 2. This comparison derives from the wheelchair's drive mechanism and layout. The energy storage unit, located at the center of the rear axle, provides the necessary energy for driving the motors. It is assumed that the model is established in an absolute coordinate system fixed to the ground. The rotation center is denoted as G , and the distance between the front and rear axles is represented by l . The position of the wheelchair is defined as $G = (x, y)$, while v and ω denote the longitudinal and lateral velocities, respectively. The orientation of the wheelchair is represented by θ .

The kinematic model of the wheeled robot establishes a relationship between its control inputs, linear velocity v , and angular velocity ω , with the time derivatives of its position and orientation. The center point of the axle serves as the center of rotation for the angular velocity. The relationship to describe the robotic wheelchair can be expressed as follows [28]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (1)$$

In summary, the presented kinematic model provides a foundation for understanding the motion and control of the robotic wheelchair, considering its structural configuration and kinematic relationships.

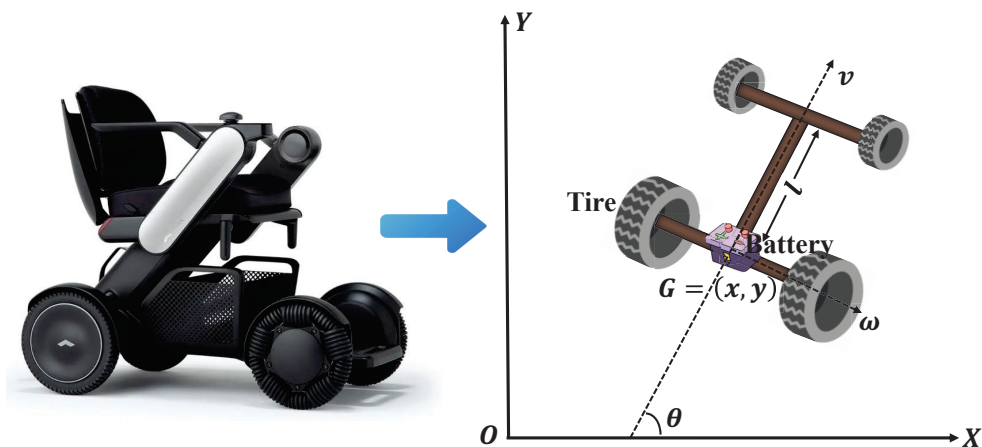


Figure 2. The schematic of the proposed robotic wheelchair.

2.2. Environment Model Description

Accurately simulating the surrounding environment is crucial for solving the mobile robot motion control problem of a mobile robot. With advancements in sensor and LIDAR technologies, relevant information can be extracted from the real environment and transformed into a map model. In this work, simultaneous localization and mapping (SLAM) is used to track the robot's position and construct a map in an unknown environment. It plays a crucial role in enabling autonomous mobile robots to perform various tasks such as navigation, exploration, and mapping. By utilizing sensor data such as laser range finders and cameras, a robot employing SLAM can estimate its position within the environment and create a map that accurately represents the surroundings.

The wheelchair utilizes LiDAR sensors to scan the surrounding environment and gather 3D point cloud information, which is then processed using the SLAM method. This sensor data, combined with odometry information from the wheel, is used to generate a real-time map. The map is represented as 2D occupied grids that accurately depict the wheelchair's environment. Initially, the robot has no prior knowledge of the environment or its own position and relies on the sensor data for an initial estimation. As the robot moves, the SLAM algorithm continually refines its position estimates and updates the map based on newly acquired sensor data. By incorporating the latest sensor measurements, SLAM ensures that the robot's position and the map representation remain accurate and up-to-date throughout its operation. The constructed map using the SLAM method is depicted in Figure 3.

Once the map is built using SLAM, it can be used for robot motion planning and control. The robot can plan a path through the environment by considering the obstacles and other features on the map, such as walls, doors, and furniture.

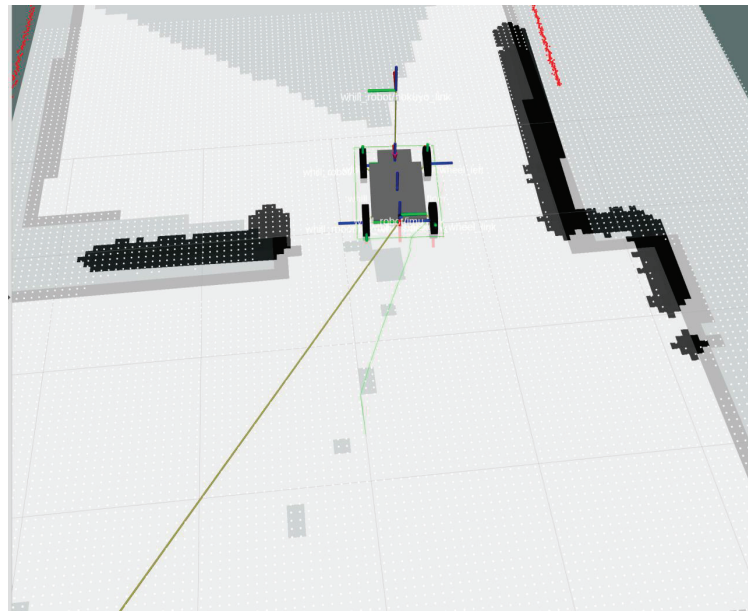


Figure 3. The environment detected by SLAM method.

3. Self-Driving Motion Control

In this section, we present a hierarchical control strategy for ensuring safe and directional movement in unknown scenarios, as shown in Figure 4. The construction of our hierarchical motion control system is depicted within the green-dotted box on the right of the figure. Utilizing the point cloud data from 3d-LiDAR and odometer data from the wheelchair, a PC generates real-time action decisions, enabling the robotic wheelchair's autonomous navigation. The strategy consists of two layers: the upper layer, which involves a global path planner utilizing the A* algorithm to generate efficient paths and avoid obstacles from the start position to the target, and the lower layer, which employs MPC local controller considering wheelchair dynamic constraints to achieve safe and directional motion around obstacles. The developed dual-quaternion-based SLERP MPC local controller serves as a vital component in our proposed system, focusing on optimizing local paths and poses while providing effective motion control. In particular, the use of a dual-quaternion-based SLERP technique enables the robotic wheelchair to handle translation and rotation movement simultaneously in diverse scenarios. In terms of implementation, the output of the global path planner serves as the input for the local path planner, enabling coordinated and obstacle-aware navigation for the wheelchair. The implementation of the proposed MPC local controller is able to handle disruptions within local maps, effectively addressing issues of local obstacle avoidance challenges in motion control. Its rapid response to dynamic changes and disturbances in the local map reveals that our robotic wheelchair can effectively navigate in complex environments, making real-time adjustments to path optimization and obstacle avoidance. This capability enhances the safety and efficiency of self-driving robotic wheelchairs, facilitating their smooth and reliable performance.

3.1. The Global Path Planner Using A* Algorithm

A* algorithm is utilized in this work to generate an optimal trajectory that avoids obstacles and boundaries. The A* algorithm is chosen for the global path planning of our wheelchair due to its optimal pathfinding capability, heuristic efficiency, and adaptability to varied environments. Its deterministic nature ensures the reliable navigation of the wheelchair in the actual environment. The A* algorithm's ability to re-calculate routes in dynamic settings is critical for obstacle avoidance, prioritizing user safety. Furthermore, its scalability and resource efficiency make it practical for the computational limits of onboard systems. It is a heuristic search algorithm that evaluates the estimated value of

each search position using an evaluation function. The algorithm selects the position with the best-estimated value and continues the search until it reaches the target point.

$$f(n) = g(n) + h(n) \quad (2)$$

where $f(n)$ represents the evaluation function of node n from the initial node to the target node, and $g(n)$ denotes the actual cost, indicating the total cost from the initial node to the current node. Moreover, $h(n)$ represents the estimated cost of the optimal path from the current node to the target node, serving as the heuristic function in the A* algorithm. The two terms, $g(n)$ and $h(n)$, play a crucial role in the A* algorithm as they determine the evaluation and selection of nodes during the search process. Since the robot can move in all directions, the Euclidean distance calculation algorithm is suitable for measuring distances between nodes. This can be described as follows:

$$f = g(n) + \sqrt{(x - x_n)^2 + (y - y_n)^2} \quad (3)$$

where x and y , respectively, represent the lateral and longitudinal coordinates of the current node position, and x_n and y_n , respectively, represent the corresponding coordinates of the target node position. It is obvious that the closer the robot is to the target node, the smaller the value of the function h will be. Consequently, this results in a relatively small value of f . Therefore, by employing the A* algorithm with its heuristic search strategy, this section achieves efficient and effective trajectory generating, ensuring safe and obstacle-free navigation for the robotic wheelchair in unknown environments. The details for the use of the A* algorithm in the ROS platform can be found in [29].

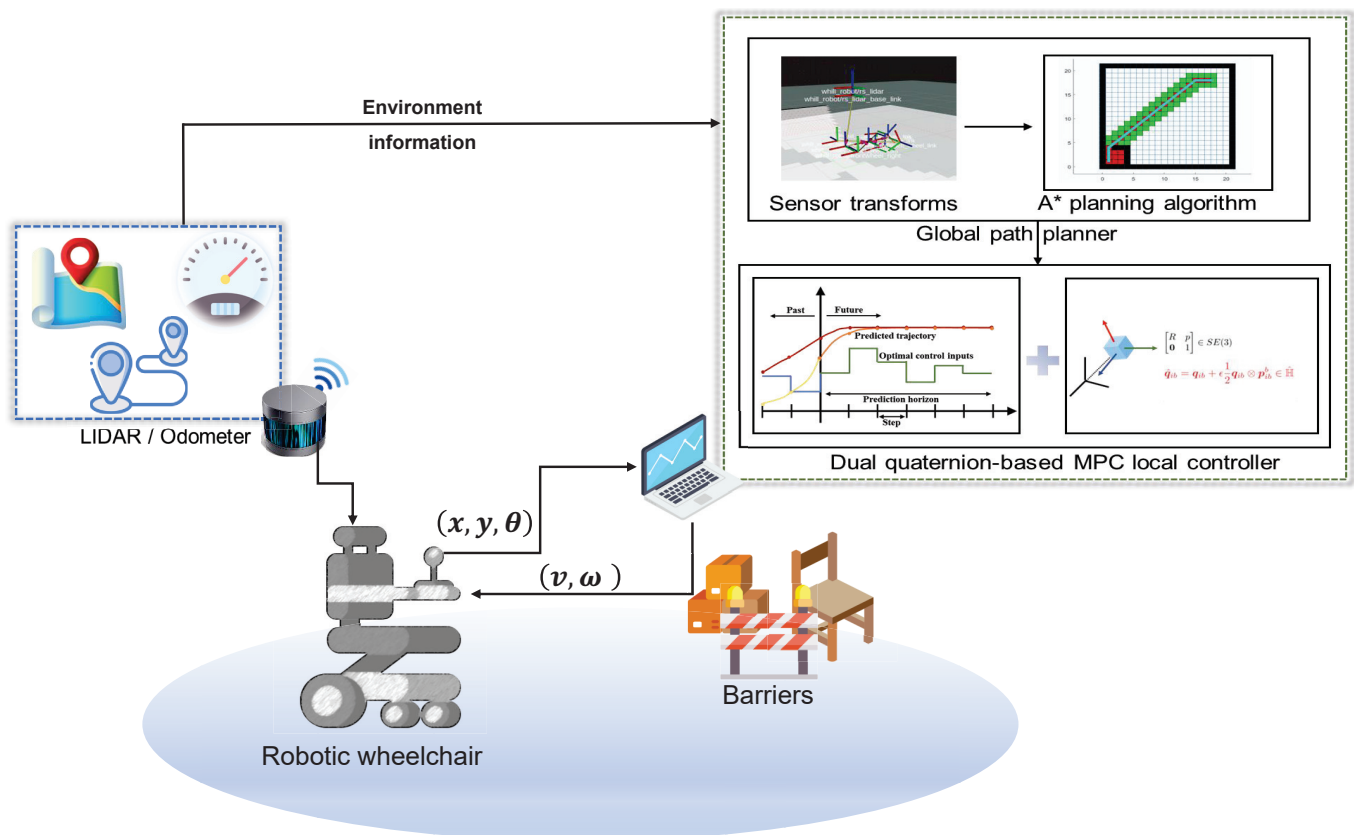


Figure 4. The structure diagram of the MPC-based hierarchical motion control system.

3.2. MPC-Based Local Controller

In real-life scenarios, the upper global path planner is unable to consider the dynamics of the wheelchair, resulting in an unsafe or shaking motion that leads to low driving

comfort. To address this issue, an MPC-based local controller is introduced in this section. The local path planner utilizes a dual-quaternion-based SLERP approach, enabling the smooth and continuous interpolation of poses. By incorporating the MPC local controller, real-time optimal control actions are computed using a predictive optimization scheme.

The motion control problem of robotics is formulated into a discretized optimization problem under the MPC framework. The goal is to minimize predicted tracking errors and control efforts while satisfying constraints on the system dynamics. The formulation of the MPC optimization problem is as follows:

$$\min_{u(k)} J = \sum_{k=t}^{t+N-1} \lambda_1 J_1 + \lambda_2 J_2 + \beta J_3 + \gamma_1 J_4 + \gamma_2 J_5 \quad (4)$$

where $J_1 = [x(k) - x_{ref}(k)]^2$ and $J_2 = [y(k) - y_{ref}(k)]^2$ represent the deviation from the desired path generated by the A* algorithm, $J_3 = v^2(k)$ denotes the control effort, and $J_4 = [x(k) - x_{via}(k)]^2$ and $J_5 = [y(k) - y_{via}(k)]^2$ represent the constraint of movement direction.

With the following constraints:

$$\begin{cases} x(k+1) = p(x(k), u(k)) \\ x_{\min} \leq x(k) \leq x_{\max} \\ y_{\min} \leq y(k) \leq y_{\max} \\ \omega_{\min} \leq \omega(k) \leq \omega_{\max} \\ v_{\min} \leq v(k) \leq v_{\max} \end{cases} \quad (5)$$

where $k = t, t+1, \dots, t+N-1$, N is the control horizon; $x(k)$ is the k -steps ahead state computed based on the current state $x(t)$; function p is defined as state function of the robotic wheelchair dynamic in discretized form in Equation (1); x_{\min} , y_{\min} , x_{\max} , and y_{\max} are the lower and upper bounds on the states; and v_{\min} , ω_{\min} , v_{\max} , and ω_{\max} are the lower and upper bounds on the control inputs, respectively. λ , γ , and β are the weights. The position (x_{ref}, y_{ref}) is the reference path generated by the A* algorithm, while (x_{via}, y_{via}) is provided by the dual-quaternion-based SLERP algorithm that will be introduced in Section 3.3.

In this problem, the cost function incorporates three main components. Firstly, it penalizes the difference between the pose of the robot and the desired pose at the target point. This ensures that the robot follows the intended trajectory closely. Secondly, the cost function includes a term that penalizes the control effort, aiming to achieve smooth and energy-efficient motion. Thirdly, the via points serve as nodes in the graph optimization problem, forming a close graph with the reference trajectory that constrains the movement direction of the wheelchair. For more details, refer to [30]. The control commands generated by the MPC local planner are then applied to the motion system of the robotic wheelchair, resulting in safe motions and directional movements.

The efficacy of the proposed MPC controller in handling local disruptions and navigating around obstacles is rooted in its predictive and optimization-centric nature. MPC is a complex control strategy capable of handling constrained multivariable control problems. The MPC-based controller predicts the future state of the wheelchair based on its dynamic model, current states, and a series of upcoming control inputs. The MPC controller considers the current state, the reference path, the via points, and any potential obstacles within its prediction horizon. It formulates an optimization problem where the control inputs are adjusted to minimize a cost function. When it encounters disruptions or obstacles, the MPC controller updates its predictions at each control cycle, recalculating the optimal control actions required to navigate around the obstacles while still trying to adhere to the constraints. The local map is continuously updated with sensor data, and the MPC controller incorporates this information into its predictions. If an obstacle is detected within the local map, the MPC planner evaluates the need to change the wheelchair's trajectory

by solving the optimization problem with the updated constraints. This ensures that the wheelchair avoids obstacles while maintaining its path toward the destination.

3.3. Dual-Quaternion-Based SLERP Algorithm

Our proposed approach integrates the dual quaternions theory, the SLERP technique, and MPC into the enhanced local controller by generating a control sequence constrained by the via points and reference trajectory. The via points are extracted from the poses generated with the dual-quaternion-based SLERP method. The reference trajectory is generated by the global planner. In this work, a significant advantage of such combinations is their ability to handle translation and orientation simultaneously, ensuring directional movement for the wheelchair.

In our proposed MPC local controller, SLERP, which was first introduced by Shoemaker [31], is effectively employed to ensure the smooth and continuous interpolation of poses. It generates intermediate orientations between two given orientations, ensuring a seamless transition between them. Figure 5 illustrates the schematic diagram comparing the SLERP interpolation method with the traditional linear interpolation method. It can be found that SLERP considers the shortest path on the unit sphere to smoothly rotate from one orientation to another.

The dual quaternion is known for its ability to handle both translation and orientation in a unified manner. A dual quaternion is an extension of a quaternion using the dual number that consists of a real part and a dual part. A dual quaternion can be represented as:

$$\mathbf{D} = \mathbf{q} + \epsilon \mathbf{p}, \quad (6)$$

where \mathbf{q} and \mathbf{p} are quaternions, ϵ is the dual unit, and $\epsilon^2 = 0$. For rigid body transformations, \mathbf{q} represents the rotation, and \mathbf{p} contains the translation information. In particular, \mathbf{q} is a unit quaternion and $\mathbf{p} = \frac{1}{2} \mathbf{t} \mathbf{q}$, where \mathbf{t} is a quaternion representing the translation vector.

In our designed robotic wheelchair controller, the combination of dual quaternions and SLERP is employed in the local path planner to achieve smooth and continuous transition poses between the current pose \mathbf{D}_c and the last pose \mathbf{D}_l in the prediction horizon. This enables the wheelchair to navigate the planned path while maintaining consistent and continuous poses, ensuring safe and directional motion. The interpolation parameter $\tau \in [0, 1]$ is used to calculate the interpolated quaternion $\mathbf{D}(\tau)$, which represents the robot's orientation at a given point along the trajectory. The interpolated quaternion $\mathbf{D}(\tau)$ by the SLERP method can be mathematically described as follows:

$$\mathbf{D}(\tau) = \frac{\mathbf{D}_c (\mathbf{D}_c^{-1} \mathbf{D}_l)^\tau}{\|\mathbf{D}_c \mathbf{D}_c^{-1} \mathbf{D}_l\|} \mathbf{D}_c, \quad (7)$$

where $\|\cdot\|$ represents the norm of a dual quaternion, defined as:

$$\|\mathbf{D}\| = \sqrt{(\mathbf{q} + \epsilon \mathbf{p})(\bar{\mathbf{q}} + \epsilon \bar{\mathbf{p}})} = \|\mathbf{q}\|. \quad (8)$$

In the use of SLERP, we are interested in interpolating between two unit dual quaternions ($\|\mathbf{D}_c\| = \|\mathbf{D}_l\| = 1$). Consequently, the dual-quaternion-based SLERP formula can be simplified as:

$$\mathbf{D}(\tau) = \mathbf{D}_c (\mathbf{D}_c^{-1} \mathbf{D}_l)^\tau. \quad (9)$$

Therefore, to calculate the transition poses between the current pose \mathbf{D}_c and the last pose \mathbf{D}_l in the prediction horizon, we can use the dual-quaternion-based SLERP method described above. For the specific calculation process, please refer to the steps shown in Appendix A. The via point (x_{via}, y_{via}) extracted from the transition pose calculated by (9) served as a node within the graph optimization problem. With the transition poses corresponding to the interpolations at the parameter value τ_1, τ_2, τ_3 , the via points are utilized by the MPC scheme to calculate control commands. By solving the MPC problem

tracking the reference trajectory while considering the via points at each time step, and applying the resulting control inputs, the wheelchair realizes the directional movement tracking the reference trajectory generated by the global planner and the consistent poses generated by dual-quaternion-based SLERP, while adhering to its dynamic constraints.

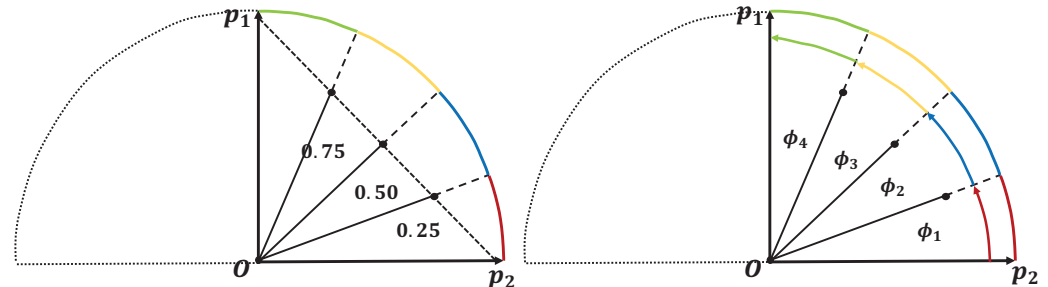


Figure 5. Left side: linear interpolation between p_1 and p_2 , where the length and arc length of the interpolated vectors may not be equal. Right side: SLERP interpolation, where both the length and arc length of the interpolated vectors are equal.

4. Simulation and Experimental Results

In our study, we have developed a robotic wheelchair model and the proposed control system within the ROS platform for validation. This model serves as the basis for both the simulation and the experiment. Through a combination of simulation and real-world experiments, we can clearly validate the effectiveness and our strategy in navigating a robotic wheelchair through complex scenarios. The results reveal that our proposed control system is robust and reliable, confirming its ability to provide safe and efficient navigation solutions for practical applications.

4.1. Experimental Setup

To validate the proposed optimal control strategy, we utilize the ROS platform, an open-source robotics middleware, to create a simulation scenario for the robotic wheelchair. The ROS platform provides a comprehensive set of software libraries and tools specifically for the development of robotic systems, thus simplifying the integration and implementation of robotic applications. It acts as a bridge between the underlying operating system and the robot actuator, enabling communication and coordination between various software components. ROS can serve as an intermediate platform to connect the operating system to the robot's controller, reducing the complexity of developing new control algorithms.

In our study, we build a robotic wheelchair model equipped with 3D-LiDAR within the ROS environment, as illustrated in Figure 6. The simulation model of the robotic wheelchair is constructed with a focus on dimensional accuracy, mirroring the actual size of the physical wheelchair. The simplified model contains the essential components—wheels, chassis, and footrest—ensuring that the virtual representation corresponds closely to its real-world counterpart in terms of size and form factor. To replicate the sensor of the wheelchair, a simulated 3D-LiDAR sensor is positioned in the same location as on the actual robotic wheelchair. By maintaining consistency with the physical dimensions and sensory setup of the actual wheelchair, the simulation model enables us to conduct experiments that denote real-world performance. This level of detail in the simulation model is critical for validating the proposed motion control framework and ensures that the findings of our simulation tests are transferable to practical applications. The red (x-axis), green (y-axis), and blue (z-axis) straight lines in this figure represent the original coordinate system of the simulation environment. In Figure 6b, the white block represents the initial pose of the wheelchair, the green rectangular represents the present footprint of the wheelchair, the yellow arrows are the position vector via points, the three colored coordinates are the via pose generated by the dual-quaternion-based SLERP, the red sphere near the coordinates represents the end of prediction horizon of the proposed MPC controller, the blue curve represents the local path, and the purple arrow represents the target pose. The control

algorithms can be realized in this platform, which is shown in the right diagram in Figure 6. This choice enables us to make use of the capabilities of ROS for simulation, control, and analysis, ensuring a robust and realistic evaluation of the proposed control strategy.

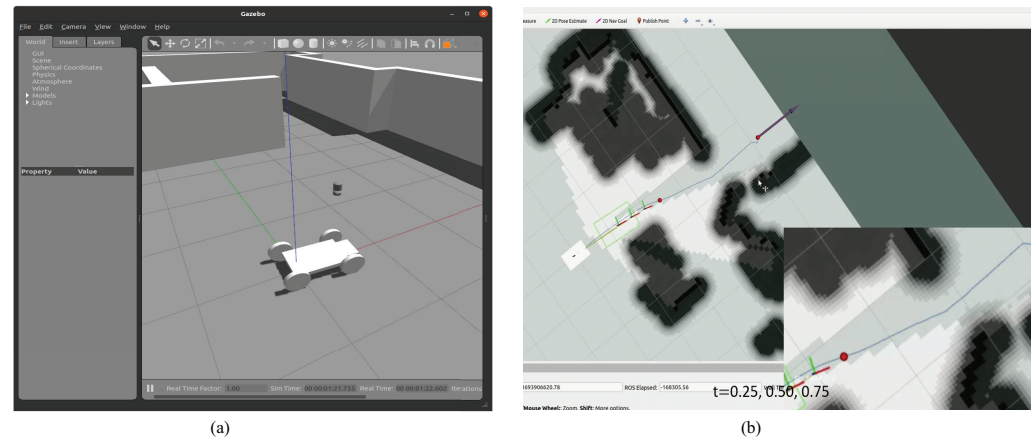


Figure 6. The considered robotic wheelchair model and environment. (a) The simulation robotic wheelchair model in the Gazebo. (b) The visualization of the environment in RViz

4.2. Simulation Results

In this section, the effectiveness of our designed strategy is validated in simulations. The performance of the control strategy was evaluated based on the results of trajectory tracking accuracy, obstacle avoidance ability, and the overall stability of the wheelchair. The selected case study is designed to simulate real-world scenarios that are common in daily life. The limitation constraints for the wheelchair in the optimization are determined as follows: the five terms are set to $v_{\min} = -0.2$ [m/s], $v_{\max} = 0.4$ [m/s], $\omega_{\min} = -0.2$ [rad/s], and $\omega_{\max} = 0.2$ [rad/s]. The simulation was run on a laptop. The CPU is Ryzen 7-4800H, the GPU is RTX 2060, and the MEMORY is 24 GB. The experiment was run with another laptop. In our study, the solver of the optimization problem is the Interior Point Optimizer (Ipopt), which is a popular open-source optimization library. It is a powerful optimization solver used in ROS for solving nonlinear optimization problems.

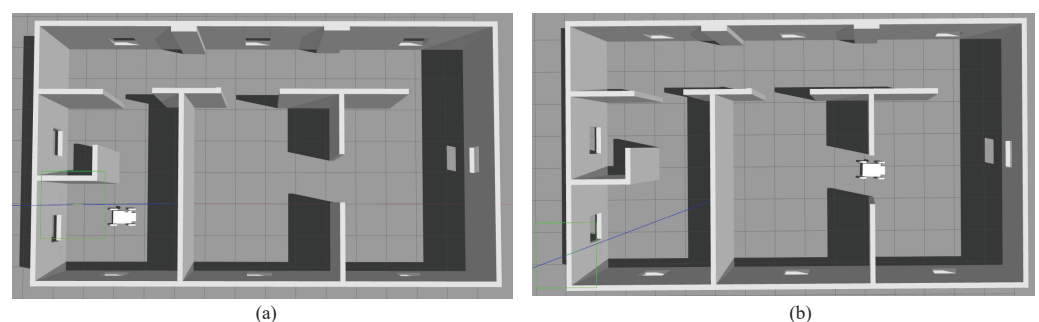


Figure 7. The simulation environment in Gazebo under ROS. (a) The initial pose of the wheelchair. (b) The target pose of the wheelchair.

To validate the performance of the proposed MPC local controller, which utilizes dual-quaternion-based SLERP interpolation, we conducted a comparative simulation case study using the ROS. In this work, we use the timed elastic band (TEB) local planner, a local path planner for mobile robots, as a benchmark. The TEB local planner is a path-planning algorithm for mobile robots to generate dynamically feasible trajectories. It optimizes the robot's path by adjusting a series of timed waypoints to navigate around obstacles while respecting the robot's locomotion capabilities. The TEB planner is integrated with ROS and is one of the common methods for mobile robot trajectory planning. Furthermore,

we use the Gazebo environment shown in Figure 7 for simulation and utilize RViz in Figures 8 and 9 for visualization. Gazebo is a powerful robot simulation software that integrates with the Robot Operating System (ROS). It offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. RViz is an open-source 3D interactive visualization tool in ROS. It is highly configurable and enables developers to visualize sensor inputs like 3D-LiDAR and camera feeds, robot status, and planned trajectories in a real-time interface. Its functionality is extendable through plugins, accommodating a broad spectrum of visualization needs in robotic development. Rviz facilitates the display of 3D robot models and the environments they navigate, offering interactive tools for direct manipulation and monitoring. These features make Rviz a powerful tool for debugging and improving robotic systems, allowing for an intuitive assessment and diagnosis of a robot's performance and issues.

With the aim of enhancing accuracy and reliability in our simulation results, we designed repeated comparison experiments in the same scenario. These results allow us to conduct thorough comparisons and analyses of the outcomes, shown in Table 1. It should be noted that the proposed method was implemented in the same simulation environment. We implemented eight simulation tests in the same scenario, shown in Table 2. For the TEB local planner, 4 out of 8 times the wheelchair could reach the goal pose. For the SLERP MPC local controller, 5 of 8 times the wheelchair could reach the goal pose. Although the success rates are almost the same, the failure states are not the same. In Figure 8a, the red lines on the wall are the simulated lidar beams, the green line represents the global path, the red arrow on the wheelchair represents the moving direction, the red arrow on the right represents the target pose, and the yellow lines represent positions vectors. In Figure 8b, the purple arrow on the right represents the target pose, and the blue line represents the local path generated by TEB local planner. For the TEB local planner, the motion was stopped by the wheelchair hitting the wall, shown in Figure 8b,d. However, with our proposed dual-quaternion-based SLERP MPC local controller, the motion was stopped because the solver failed to solve the MPC optimal problem. During the failure scenario, the wheelchair will stop at a safe area without contacting the walls, as shown in Figure 8a,c. With a new proper target pose published, the wheelchair will continue its motion. With such a character, our designed controller is more suitable for real applications. In the real application, safety is important for the wheelchair and the movement of the wheelchair can be restarted again by publishing a new target pose.

In the following, we provide a detailed analysis and discussion of the comparative simulation results under specific test cases.

The simulation scenario we built is shown in Figure 7, where the initial pose of the robotic wheelchair is shown in subgraph (a), and the target pose is shown in subgraph (b). The initial position is set as $(-2.0, -0.5, 0.0)$, and the initial orientation is $(0, 0, 0, 1)$. The target position is $(4.9, 0.5, 0.0)$, and the target orientation is $(0, 0, 0, 1)$.

Table 1. Simulations for motion control in a specific scenario with TEB local planner.

Scenario	Result	Remark
Test 1	Fail	Hitting the wall
Test 2	Fail	Hitting the wall
Test 3	Success	Reaching the target
Test 4	Success	Reaching the target *
Test 5	Success	Reaching the target
Test 6	Fail	Hitting the wall
Test 7	Fail	Hitting the wall
Test 8	Success	Reaching the target

* Before reaching the target, the wheelchair rotates significantly.

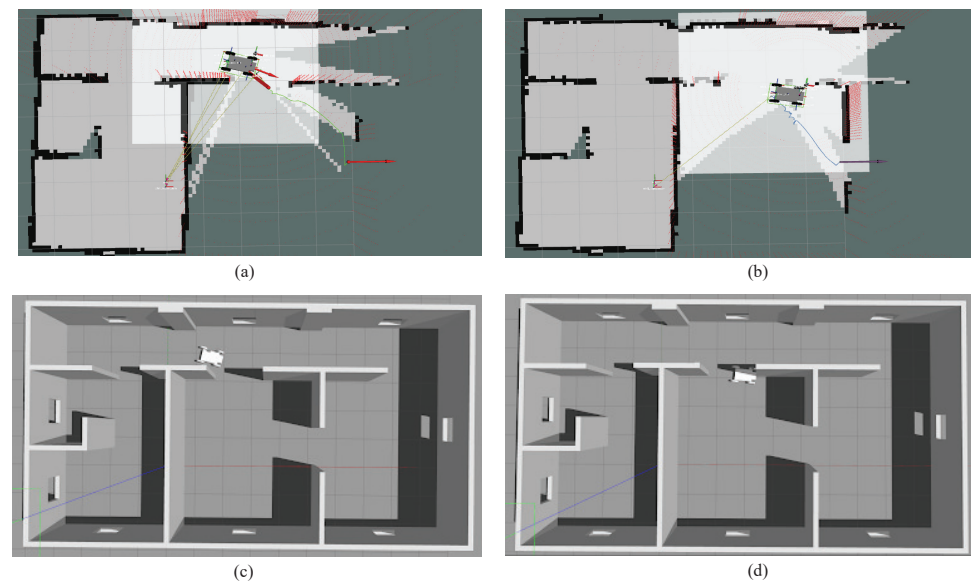


Figure 8. The failure scenarios in the simulation environment. (a) The pausing pose during navigation with our proposed MPC local controller in RViz. (b) The pausing pose during navigation with TEB local planner in RViz. (c) The pausing pose during navigation with our proposed MPC local controller in Gazebo. (d) The pausing pose during navigation with TEB local planner in Gazebo.

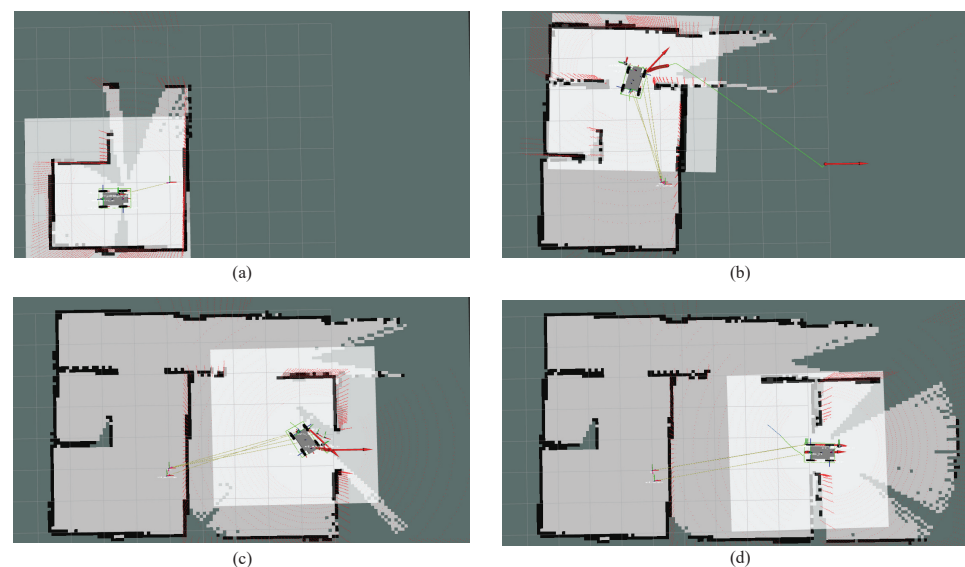


Figure 9. The motion sequence of the wheelchair in the simulation environment with our proposed MPC local controller in Rviz. (a) The wheelchair's initial pose. (b) The wheelchair entering the corridor. (c) The wheelchair approaching the target pose. (d) The wheelchair reaching the target pose successfully.

Table 2. Simulations for motion control in a specific scenario with our proposed MPC local controller.

Scenario	Result	Remark
Test 1	Success	Reaching the target
Test 2	Success	Reaching the target
Test 3	Fail	Pausing at the safe area
Test 4	Success	Reaching the target
Test 5	Fail	Pausing at the safe area
Test 6	Success	Reaching the target
Test 7	Fail	Pausing at the safe area
Test 8	Success	Reaching the target

The corresponding search process is demonstrated in Figure 9, where (a) represents the initial pose of the wheelchair in motion and the navigation process with our proposed MPC local controller, and (b), (c), (d) shows that the wheelchair safely avoids obstacles (walls) with the proposed MPC local controller while plotting the map with SLAM algorithm. It should be noted that the colored lines in Figure 9 have the same meaning as in Figure 8a. Using the control system we designed, the robotic wheelchair has the ability to avoid obstacles, turn around corners, and pass through corridors, while, most importantly, maintaining its motion safety and realizing directional movement.

The comparative simulation results of Case 1 comparing Test 1 in Tables 1 and 2 are illustrated in Figure 10. From top to bottom, these figures represent the wheelchair's orientation, its position along the x-axis, its position along the y-axis during motion, and the linear and angular velocity profiles of the robotic wheelchair. The yellow curve represents the result without the MPC strategy, while the blue curve represents the comparison when using the SLERP MPC local controller that we designed. It can be observed that at the 27-second mark when the robotic wheelchair encounters an obstacle, the robotic wheelchair is unable to realize the obstacle avoidance and pauses without our proposed motion controller, while the wheelchair can avoid the obstacle and continue the motion using our designed hierarchical motion controller. Furthermore, it is worth noting that the utilization of dual-quaternion-based SLERP within the MPC controller accounts for both rotational and translational movement simultaneously; the wheelchair realizes the directional movement without unnecessary orientation and noticeable sharp fluctuations. In summary, the proposed strategy enables the wheelchair to generate safe and directional movement in a complex scenario.

Figure 11 shows the trajectory simulation comparison of the two controllers under case 2 comparing Test 4 in Tables 1 and 2, where the arrow on the curve represents the movement direction of the wheelchair. It can be clearly seen that the yellow trajectory curve, representing the wheelchair using the TEB controller, has trajectory offset and unnecessary rotation when encountering obstacles and approaching the target. In comparison, with the same initial and target pose, the trajectory using the controller we proposed is more stable and realizes directional movement without unnecessary rotation. Then, Figure 12 shows the detailed simulation results for case 2. Although the robotic wheelchair using both controllers can avoid obstacles and finally reach the target position, the wheelchair with the MPC local controller has a smaller orientation deviation and directional movement. Since the controller we designed utilizes the dual-quaternion-based SLERP and has the ability to facilitate rotation and translation simultaneously, it is reasonable for the wheelchair to approach the target more directly without unnecessary rotation.

According to the results, the proposed method can navigate the robot to the target pose safely. Although there may be errors, the simulations have proved that the method works. In general, the combination of dual quaternions and the SLERP method further enhances the control capability of the MPC local controller. The simulation results demonstrate the effectiveness and robustness of the proposed robotic wheelchair control strategy. These results highlight the potential of this control strategy to significantly improve the safety, stability, and overall performance of robotic wheelchairs in practical applications.

4.3. Experimental Results

To further demonstrate the application potential of the proposed control strategy in daily life, we also conducted real-machine experiments. The experimental evaluation is conducted using a robotic wheelchair controlled by the developed MPC-based control system. The computer configuration is as follows, CPU is Intel Core i7-12700H, GPU is RTX 3060, and MEMORY is 32GB. A specific real-world scenario is designed to validate the wheelchair's performance. To show the results clearly, the details for some moments are demonstrated in Figure 13. To better accord with real-life scenarios, we conduct practical experiments in a laboratory scenario. Our experiment begins indoors, as depicted in Figure 13a. The map of the environment is unknown to the wheelchair. The wheelchair

explores and navigates through the unknown environment with the target pose sent by the user in RViz. After successfully navigating through a doorway, the robotic wheelchair safely maneuvers through narrow corridors and avoids obstacles, as evidenced in (b) and (c). Upon reaching the corridor's end and encountering an elevator, the wheelchair automatically executes a stable turn, as captured in (d). Subsequently, the robotic wheelchair safely and reliably makes its way back to the laboratory, as illustrated in (e) and (f). It is important to note that, for practical reasons, we intentionally avoid choosing a clean and unobstructed testing environment.

The experimental results are shown in Figures 14–17. It can be seen that in Figure 14, the initial driving origin of the robotic wheelchair is (0,0), and the red dots are target positions sent by the user. The blue curve is the actual path of the robotic wheelchair with the dual-quaternion-based SLERP MPC controller. It is clearly shown that the robotic wheelchair can drive steadily and avoid obstacles. However, it is worth noting that two target points are outside the margin of reasonable area and have been marked by red arrows (\rightarrow). This is because the experimental scenario is too narrow, and the 3D-LiDAR is unable to obtain the overall information. The target poses sent by the user may located in the walls. In Figure 15, the purple curve represents the real orientation of the robotic wheelchair during the whole movement process. The two error points we mentioned are also indicated by arrows (\rightarrow) in the figure. The real position results can be observed in Figure 16, where the light green curve in the upper diagram is the position along the x-axis and the result in the lower shows the position along the y-axis, respectively.

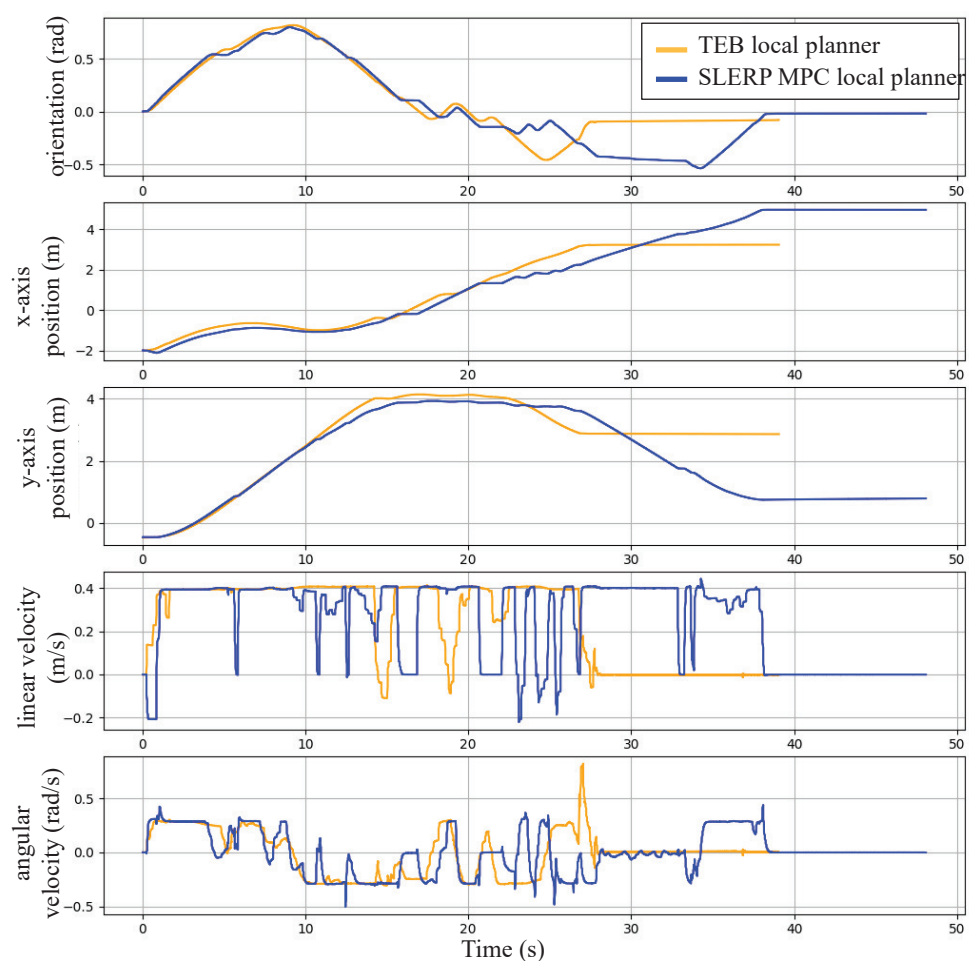


Figure 10. The comparative simulation results of Test 1 in Tables 1 and 2, from top to bottom: the orientation, x-axis position, y-axis position, linear velocity, and angular velocity.

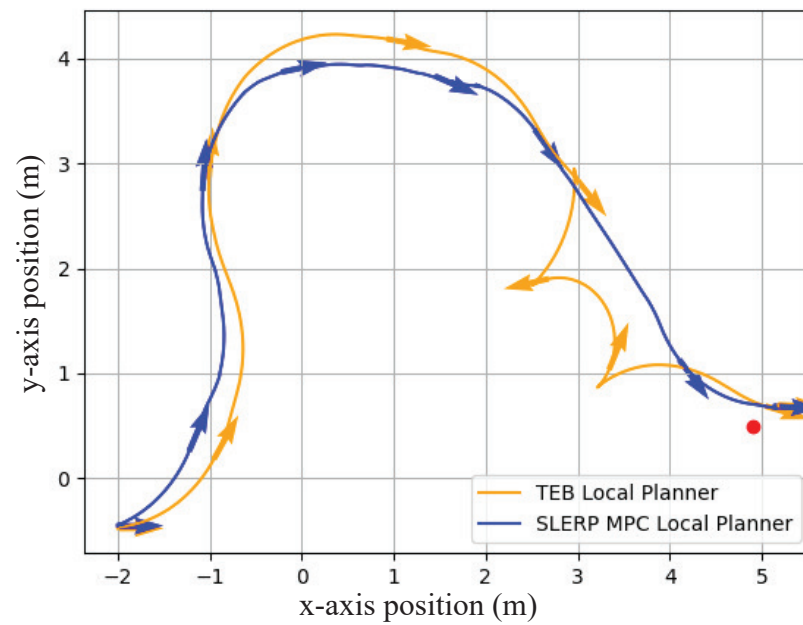


Figure 11. The comparative simulation results of trajectories with two controllers in case 2.

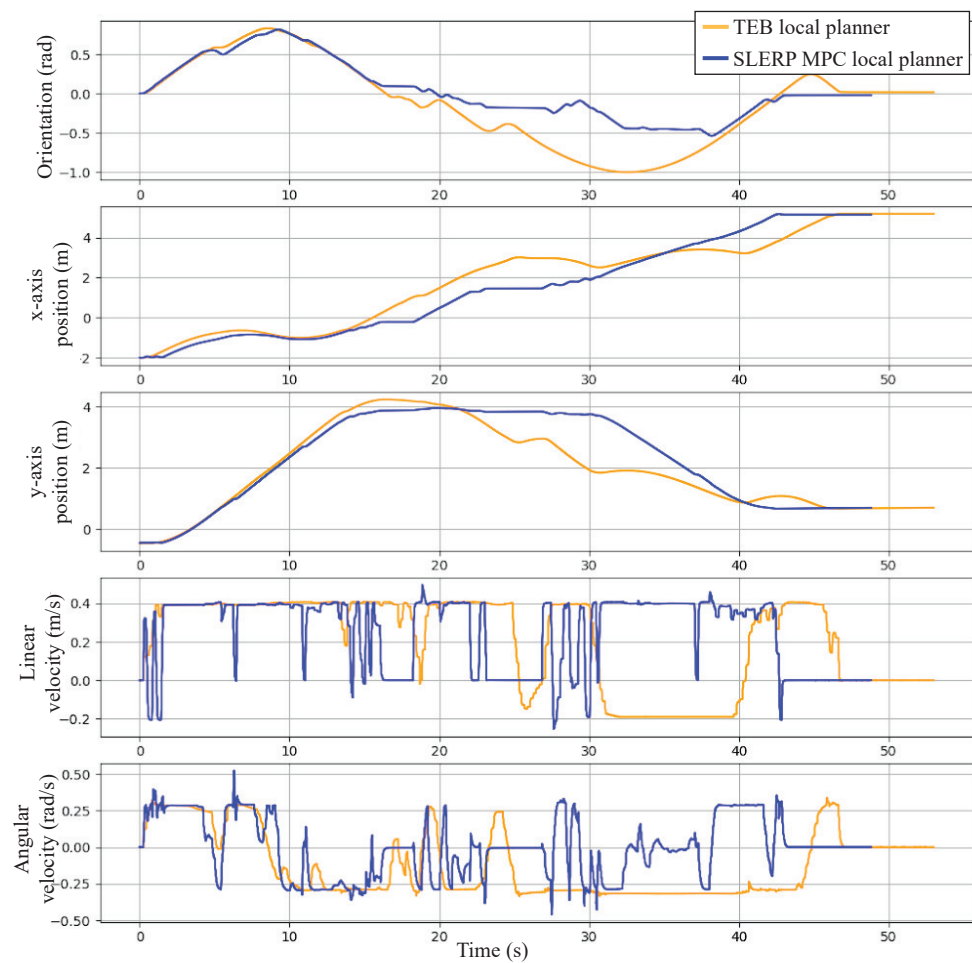


Figure 12. The comparative simulation results with two controllers in case 2, from top to bottom: the orientation, x-axis position, y-axis position, linear velocity, and angular velocity.

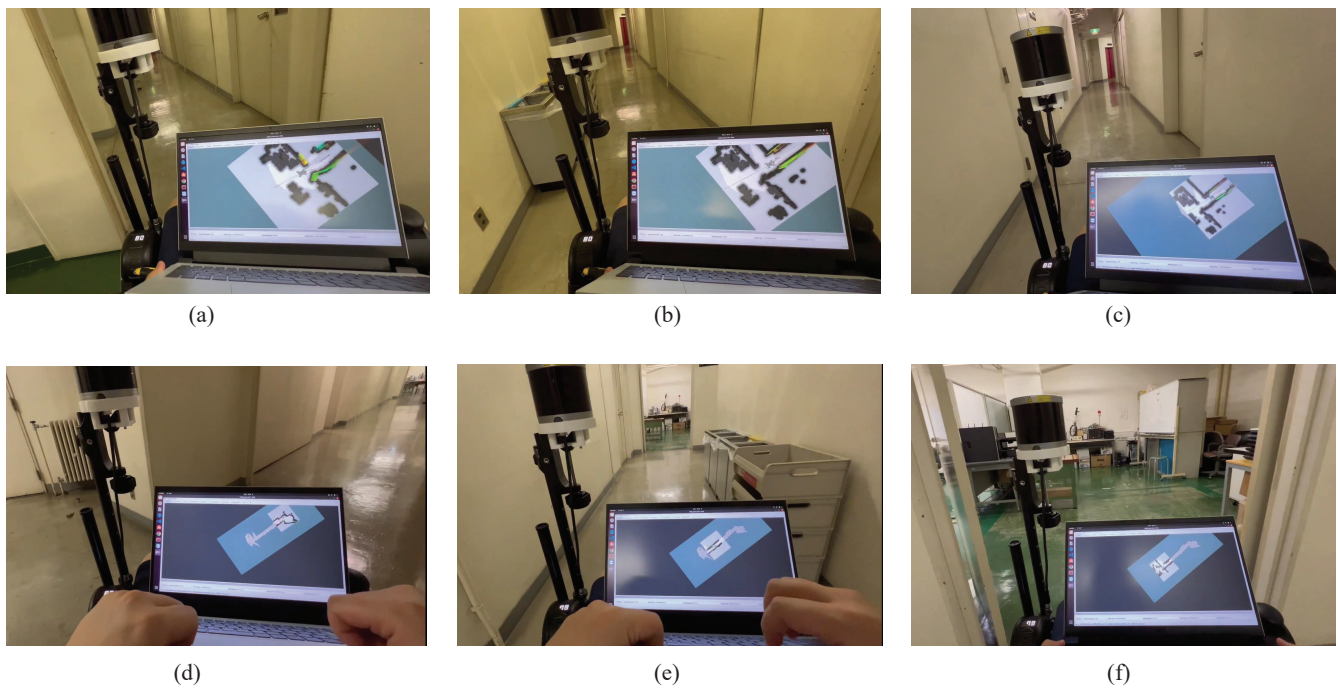


Figure 13. Moments of the real-world experiment using a real robotic wheelchair in a narrow corridor with obstacles with the proposed MPC-based hierarchical system. (a) Exit the lab door. (b) Encounter trash cans. (c) Avoid obstacles and move on. (d) Turn around at the end of the corridor. (e) Avoid obstacles. (f) Move back through the door into the lab.

In the experiment, there is a problem worthy of our observation, that is, when the wheelchair is exiting the lab door, denoting the position in Figure 13a, the wheelchair generates a strong bumpy feeling. In Figure 17, the linear velocity and angular velocity change abruptly. This is mainly because the wheelchair encountered a threshold of the door when passing through the lab door. In other words, there was interference in the local map. This disturbance in the road surface also caused a bumpy feeling for the user during the movement. However, after the wheelchair crosses the threshold, it returns to the original path generated by the global path planner. This also illustrates the effectiveness of the control strategy we proposed in dealing with interference problems.

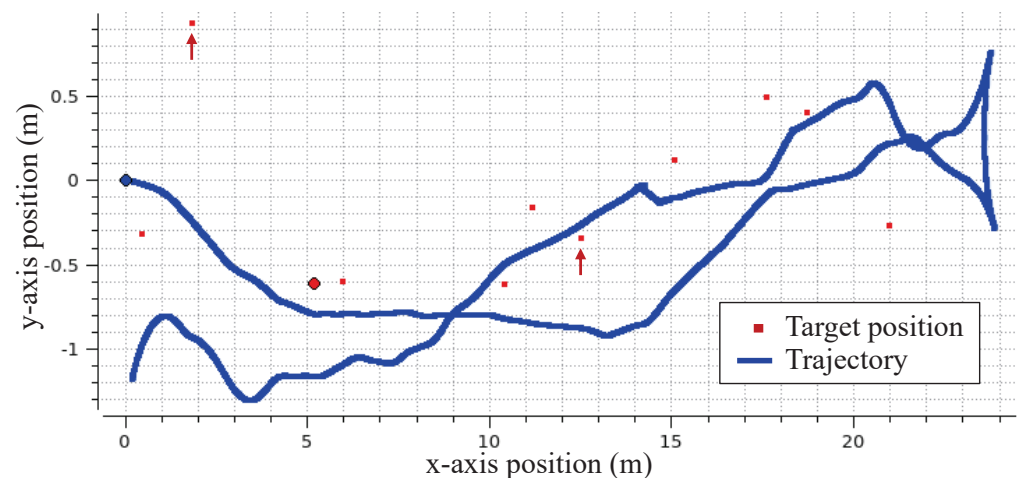


Figure 14. The real movement trajectory of the robotic wheelchair with the MPC-based hierarchical system.

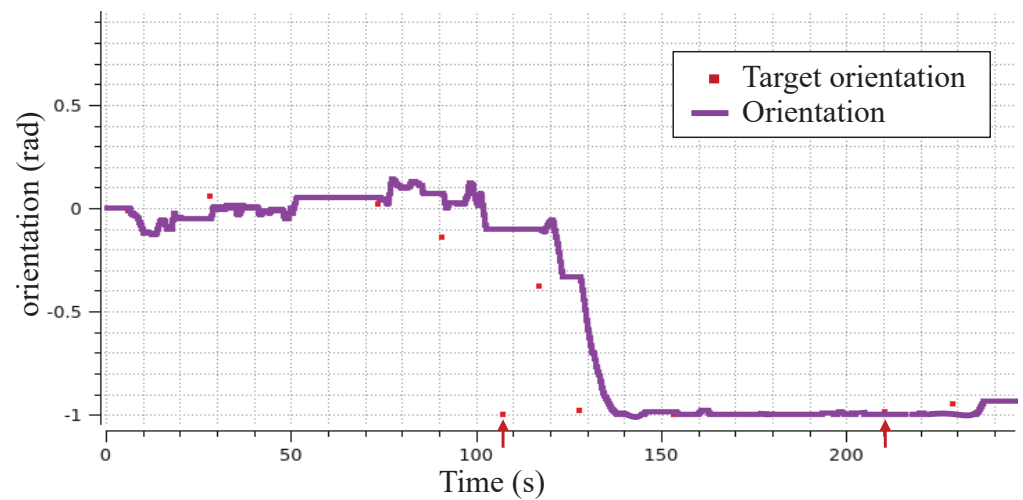


Figure 15. The real orientation of the robotic wheelchair during the whole movement process.

The experimental results show the practicality of our proposed controller. From all of the presented results, it can be observed that for both the simulation and experimental results, the proposed control strategy demonstrated performance. The robotic wheelchair can achieve path planning, successfully avoid obstacles, and maintain stability throughout the whole navigation process. The experiment also shows the adaptability of the proposed control strategy to actual environmental conditions, ensuring safe and efficient wheelchair movement.

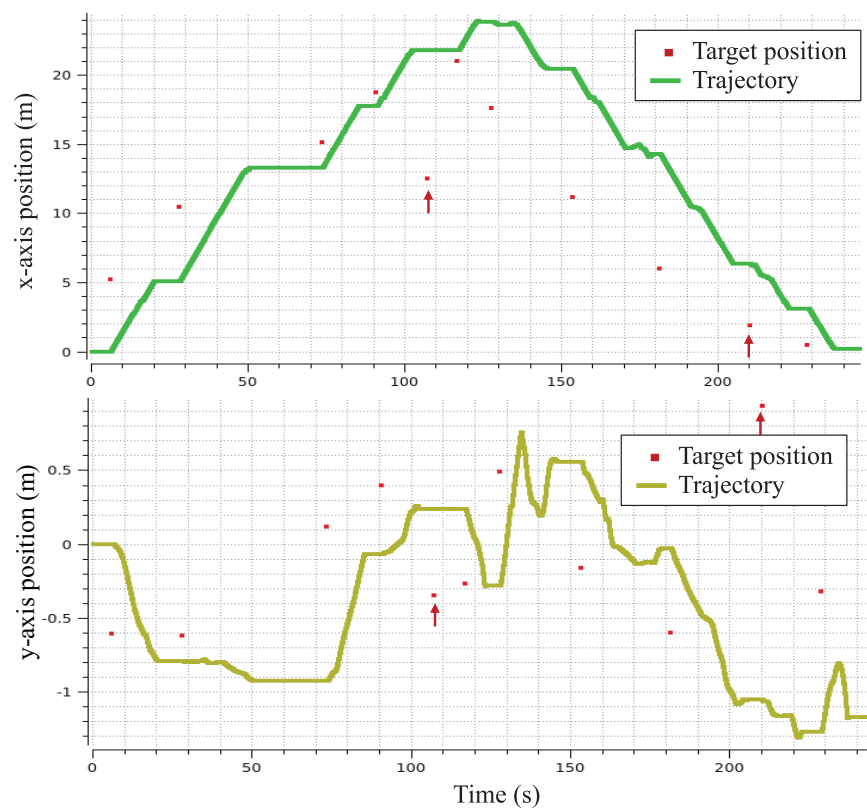


Figure 16. The real position of the robotic wheelchair during the whole movement process; the distance along the x-axis is shown at the top, and the distance along the y-axis is shown at the bottom.

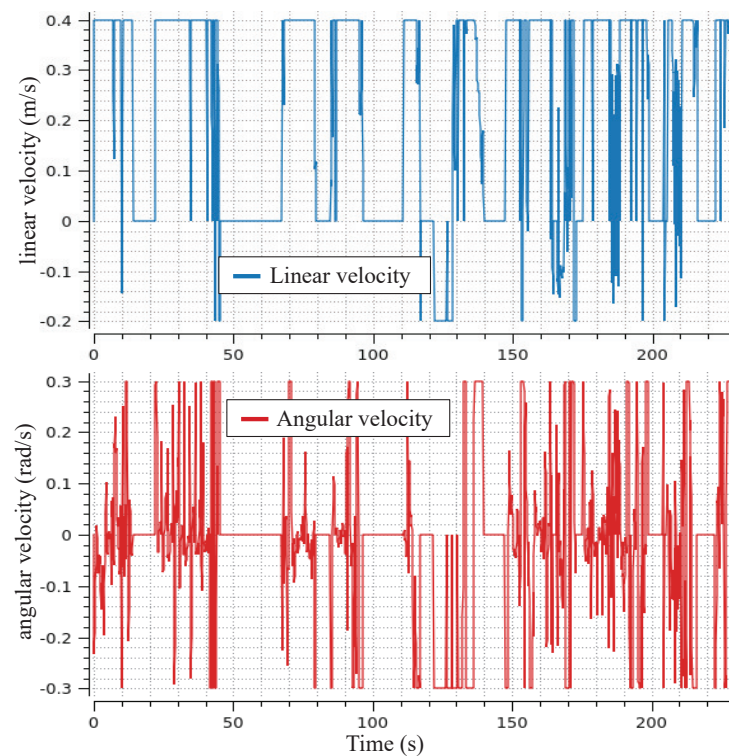


Figure 17. The real velocities of the robotic wheelchair; the linear velocity is shown at the top, and the angular velocity is shown at the bottom.

4.4. Discussion

(i). **Safety and stability for practical application.** The simulation and experimental results clearly highlight that the use of dual-quaternion-based SLERP in robotic wheelchair control with the MPC scheme shows significant advantages and advances in terms of safety, stability, and overall performance. This method exhibits stability when tracking the target path and handling translation and rotation simultaneously, while also being adept at avoiding obstacles in real time. Such capabilities greatly improve the overall safety of robotic wheelchairs. The simulation test results in Tables 1 and 2 show that although the success rates are almost the same, the proposed MPC local controller ensures a higher safety margin as it stops the wheelchair in a safe zone without collision in case of failure. Figure 9 shows the comparative simulation result in specific failure scenarios. It clearly shows that without the MPC strategy, the robotic wheelchair hits the wall and gets stuck. In comparison, when using the proposed controller, the robotic wheelchair stops in a safe area and can be restarted by a new target pose, making it more viable for real-world applications where safety and the ability to resume movement are vital. Furthermore, the dual-quaternion-based SLERP approach used in the local path planner takes advantage of its feature of handling translation and orientation simultaneously and allowing directional interpolation to ensure consistent and continuous orientation when the wheelchair approaches the target. This feature also enhances the stability of wheelchair driving, contributing to the safe and directional movement for wheelchair users.

(ii). **The future potential.** It is important to highlight that the critical element in evaluating the efficacy of our proposed control strategy lies in its potential for future applications. Repetitive simulation comparisons show the safety of our approach: when obstacles cannot be avoided with our controller, the robot wheelchair can stop and maintain a safe standby distance from obstacles. Once a new target pose is given, the robotic wheelchair controlled by our proposed controller can promptly resume self-driving. This addresses the effectiveness of the MPC-based hierarchical motion control system we designed to ensure the robot wheelchair's safe operation. In particular, the experimental outcomes provide further evidence of the robustness of our proposed strategy when encountering the ground surface

of disturbance. Although the wheelchair will produce an uncomfortable fluctuation when passing the threshold, it can immediately return to the desired trajectory after successfully passing the threshold and can continue to perform local searches and self-drive. This highlights the efficacy of our proposed controller in handling temporary local obstacles and disturbance problems. It also reveals the potential of our MPC-based hierarchical motion control system to tackle dynamic obstacle avoidance challenges in the future. Moreover, the use of dual quaternions in the MPC scheme, while taking into account robotic kinematics and constraints, facilitates the real-time calculation of optimal control actions. Its simultaneous calculation of rotation and translation optimizes the wheelchair's navigation performance, facilitating safe and directional self-driving in unknown environments. For future practical applications, the robotic wheelchairs are expected to assist elderly users with mobility issues in different scenarios, such as facing toward the table for meals or approaching beds for sleep. The directional movement realized by the proposed controller can give intuitive feedback to the users and improve their life quality.

5. Conclusions

In conclusion, this work presents a comprehensive framework for the motion control of robotic wheelchairs in unknown scenarios. The proposed approach, combining global path planning and MPC-based control with the integration of dual quaternions and SLERP, demonstrates effective unknown environment exploration and safe obstacle avoidance. A global path is generated by the A* algorithm and serves as the reference trajectory for the MPC-based local path planner. Moreover, the utilization of dual-quaternion-based SLERP in the local path planner allows for the continuous interpolation of poses, ensuring stability and directional motion for the wheelchair. The dual-quaternion-based SLERP MPC-based control strategy, considering the dynamics and constraints of the wheelchair, leads to safe navigation and directional movement for the robotic wheelchair. The implementations of the proposed framework in the simulation environment under ROS and the real world further demonstrate its practical applicability and feasibility. This work contributes to the advancement of assistive robotics, showing promising potential for real-world applications and improving the mobility and quality of life of elderly individuals with mobility issues. Currently, the system may stop movement in narrow spaces and has not incorporated dynamic obstacles into the scenario, presenting a limitation in adaptability to environments with moving entities. Future research will focus on enhancing the wheelchair's interactive capabilities, considering user preferences for a more personalized experience, and aim to refine the system to navigate nimbly in confined spaces and to react intelligently to dynamic obstacles.

Author Contributions: Methodology, W.C.; software, D.W.; validation, D.W.; formal analysis, D.W. and W.C.; investigation, D.W.; resources, D.W.; data curation, D.W.; writing—original draft preparation, D.W.; writing—review and editing, D.W., W.C. and A.T.; supervision, A.T.; and project administration, W.C. All of the authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data presented in this study are available from corresponding authors upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

To calculate the mid-pose of two dual quaternions, we use the spherical linear interpolation (SLERP) method. The mid-pose corresponds to the interpolation at the parameter value $t = 0.5$. Given two dual quaternions \mathbf{D}_0 and \mathbf{D}_1 , we can calculate their mid-pose \mathbf{D}_{mid} using the following steps:

1. Compute the conjugate of \mathbf{D}_0 , denoted by \mathbf{D}_0^{-1} .
2. Compute the product of the conjugate and \mathbf{D}_1 , denoted by $\mathbf{D}_{con} = \mathbf{D}_0^{-1}\mathbf{D}_1$.
3. Apply the SLERP with derived formula (9) with $t = 0.5$, which is $\mathbf{D}_{mid} = \mathbf{D}_0(\mathbf{D}_{con})^{0.5}$.
4. To compute the power of a dual quaternion, first decompose \mathbf{D}_{con} into its scalar and vector parts: $\mathbf{D}_{con} = (s_r + \mathbf{v}_r) + \epsilon(s_t + \mathbf{v}_t)$.
5. Compute the scalar and vector parts of the resulting dual quaternion: $(s_{mid} + \mathbf{v}_{mid}) + \epsilon(s_{t,mid} + \mathbf{v}_{t,mid}) = \mathbf{D}_0(\mathbf{D}_{con})^{0.5}$.
6. For the real part, apply the power operation to a unit quaternion using the formula: $(s_{mid} + \mathbf{v}_{mid}) = \cos(0.5 \cdot \theta) + (\sin(0.5 \cdot \theta) \cdot \frac{\mathbf{v}_r}{|\mathbf{v}_r|})$, where θ is the angle between \mathbf{D}_0 and \mathbf{D}_1 .
7. For the dual part, use the formula: $(s_{t,mid} + \mathbf{v}_{t,mid}) = \frac{1}{2}(s_t + \mathbf{v}_t)(s_{mid} + \mathbf{v}_{mid})$.
8. Combine the real and dual parts to obtain the mid-pose dual quaternion: $\mathbf{D}_{mid} = (s_{mid} + \mathbf{v}_{mid}) + \epsilon(s_{t,mid} + \mathbf{v}_{t,mid})$.

Now, we have the mid-pose dual quaternion \mathbf{D}_{mid} of the two given dual quaternions \mathbf{D}_0 and \mathbf{D}_1 .

References

1. Pablo, A. Charted: The World's Aging Population from 1950 to 2100. *Visual Capitalist*, 29 May 2023.
2. KYODO NEWS. Elderly citizens accounted for record 28.4% of Japan's population in 2018, data show. *The Japan Times*, 15 September 2019.
3. Jack, D. The issue of Japan's aging population. In *Law School International Immersion Program Papers*; University of Chicago Law School: Chicago, IL, USA, 2016; p. 8.
4. Watanabe, S.; Mukai, M. Optimal trajectory generation of a drone for wheelchair tracking using mixed-integer programming. *Artif. Life Robot.* **2022**, *27*, 159–164. [\[CrossRef\]](#)
5. Shamseldin, M.A.; Khaled, E.; Youssef, A.; Mohamed, D.; Ahmed, S.; Hesham, A.; Elkodama, A.; Badran, M. A new design identification and control based on GA optimization for an autonomous wheelchair. *Robotics* **2022**, *11*, 101. [\[CrossRef\]](#)
6. Nikpour, M. A Novel Stability Control System for Two-Wheeled ROBOTIC wheelchairs. Doctoral Theses, Auckland University of Technology, Auckland, New Zealand, June 2021.
7. Lu, E.H.C.; Chen, H.S.; Tseng, V.S. An efficient framework for multirequest route planning in urban environments. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 869–879. [\[CrossRef\]](#)
8. Campbell, S.; O'Mahony, N.; Carvalho, A.; Krpalkova, L.; Riordan, D.; Walsh, J. Path planning techniques for mobile robots a review. In Proceedings of the 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE), Barcelona, Spain, 12–15 February 2020; pp. 12–16.
9. Psotka, M.; Duchoň, F.; Roman, M.; Michal, T.; Michal, D. Global Path Planning Method Based on a Modification of the Wavefront Algorithm for Ground Mobile Robots. *Robotics* **2023**, *12*, 25. [\[CrossRef\]](#)
10. Yuan, C.; Liu, G.; Zhang, W.; Pan, X. An efficient RRT cache method in dynamic environments for path planning. *Robot. Auton. Syst.* **2020**, *131*, 103595 [\[CrossRef\]](#)
11. Otani, T.; Koshino, M. Applying a path planner based on RRT to cooperative multirobot box-pushing. *Artif. Life Robot.* **2009**, *13*, 418–422. [\[CrossRef\]](#)
12. Kyprianou, G.; Doitsidis, L.; Chatzichristofis, S.A. Towards the achievement of path planning with multi-robot systems in dynamic environments. *J. Intell. Robot. Syst.* **2022**, *104*, 15. [\[CrossRef\]](#)
13. Prasad, N.L.; Ramkumar, B. 3-D Deployment and Trajectory Planning for Relay Based UAV Assisted Cooperative Communication for Emergency Scenarios Using Dijkstra's Algorithm. *IEEE Trans. Veh. Technol.* **2022**, *72*, 5049–5063. [\[CrossRef\]](#)
14. Guernane, R.; Achour, N. Generating optimized paths for motion planning. *Robot. Auton. Syst.* **2011**, *59*, 789–800. [\[CrossRef\]](#)
15. Kuswadi, S.; Santoso, J.W.; Tamara, M.N.; Nuh, M. Application SLAM and path planning using A-star algorithm for mobile robot in indoor disaster area. In Proceedings of the 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), Bali, Indonesia, 29–30 October 2011; pp. 270–274.
16. Jeddisaravi, K.; Alitappeh, R.J.; Guimares, F.G. Multi-objective mobile robot path planning based on A search. In Proceedings of the 2016 6th International Conference on Computer and Knowledge Engineering (ICCCKE), Mashhad, Iran, 20 October 2016; pp. 7–12.
17. Kusuma, M.; Machbub, C. Humanoid robot path planning and rerouting using A-Star search algorithm. In Proceedings of the 2019 IEEE International Conference on Signals and Systems (ICSigSys), Bandung, Indonesia, 16–18 July 2019; pp. 110–115.
18. GMurayama, T. Distributed model predictive consensus control for robotic swarm system: Local subsystem regulator approach. *Artif. Life Robot.* **2018**, *23*, 628–635. [\[CrossRef\]](#)
19. Lafmejani, A.S.; Berman, S. Nonlinear MPC for collision-free and deadlock-free navigation of multiple nonholonomic mobile robots. *Robot. Auton. Syst.* **2021**, *141*, 103774. [\[CrossRef\]](#)

20. Rosmann, C.; Makarow, A.; Bertram, T. Online motion planning based on nonlinear model predictive control with non-euclidean rotation groups. In Proceedings of the 2021 European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021; pp. 1583–1590.
21. Farina, M.; Perizzato, A.; Scattolini, R. Application of distributed predictive control to motion and coordination problems for unicycle autonomous robots. *Robot. Auton. Syst.* **2015**, *72*, 248–260. [[CrossRef](#)]
22. Wu, Y.; Li, S.; Zhang, Q.; Sun-Woo, K.; Yan, L. Route planning and tracking control of an intelligent automatic unmanned transportation system based on dynamic nonlinear model predictive control. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16576–16589. [[CrossRef](#)]
23. Eckhoff, M.; Kirschner, R.J.; Kern, E.; Abdolshah, S.; Haddadin, S. An MPC Framework For Planning Safe & Trustworthy Robot Motions. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4737–4742.
24. Bjelonic, M.; Grandia, R.; Geilinger, M.; Harley, O.; Medeiros, V.S.; Pajovic, V.; Jelavic, E.; Coros, S.; Hutter, M. Offline motion libraries and online MPC for advanced mobility skills. *Int. J. Robot. Res.* **2023**, *41*, 903–924. [[CrossRef](#)]
25. Pham, H.L.; Perdureau, V.; Adorno, B.V.; Fraisse, P. Position and orientation control of robot manipulators using dual quaternion feedback. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 658–663.
26. Adorno, B.V.; Marinho, M.M. Dq robotics: A library for robot modeling and control. *IEEE Robot. Autom. Mag.* **2020**, *28*, 102–116. [[CrossRef](#)]
27. Zivkovic, N.; Vidakovic, J.; Mitrovic, S.; Lazarevic, M. Implementation of dual quaternion-based robot forward kinematics algorithm in ROS. In Proceedings of the 11th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 7–10 June 2022; pp. 1–4.
28. Brockett, R.W.M. Asymptotic stability and feedback stabilization. Differential geometric control theory. *Differ. Geom. Control. Theory* **1983**, *27*, 181–191.
29. de Assis Brasil, P.M.; Pereira, F.U.; Cuadros, M.A.D.S.L.; Cukla, A.R.; Gamarra, D.F.T. A study on global path planners algorithms for the simulated turtlebot 3 robot in ros. In Proceedings of the 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE), Natal, Brazil, 9–13 November 2020; pp. 1–6.
30. Kummerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g 2 o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3607–3613.
31. Shoemake, K. Animating rotation with quaternion curves. In Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, San Francisco, CA, USA, 22–26 July 1985; pp. 245–254.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.