




Article

Autonomous Alignment and Docking Control for a Self-Reconfigurable Modular Mobile Robotic System

Shumin Feng , Yujiong Liu , Isaac Pressgrove and Pinhas Ben-Tzvi * 

Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24060, USA; shuminf@vt.edu (S.F.); yjliu@vt.edu (Y.L.); ipressgr@vt.edu (I.P.)

* Correspondence: bentzvi@vt.edu

Abstract: This paper presents the path planning and motion control of a self-reconfigurable mobile robot system, focusing on module-to-module autonomous docking and alignment tasks. STORM, which stands for Self-configurable and Transformable Omni-Directional Robotic Modules, features a unique mode-switching ability and novel docking mechanism design. This enables the modules that make up STORM to dock with each other and form a variety configurations in or to perform a large array of tasks. The path planning and motion control presented here consists of two parallel schemes. A Lyapunov function-based precision controller is proposed to align the target docking mechanisms in a small range of the target position. Then, an optimization-based path planning algorithm is proposed to help find the fastest path and determine when to switch its locomotion mode in a much larger range. Both numerical simulations and real-world experiments were carried out to validate these proposed controllers.

Keywords: modular self-reconfigurable robotic system; mobile robot stabilization; switching kinematics; trajectory optimization



Citation: Feng, S.; Liu, Y.; Pressgrove, I.; Ben-Tzvi, P. Autonomous Alignment and Docking Control for a Self-reconfigurable Modular Mobile Robotic System. *Robotics* **2024**, *13*, 81. <https://doi.org/10.3390/robotics13050081>

Academic Editors: Bruno Brito and Giorgos Mamakoukas

Received: 5 April 2024

Revised: 6 May 2024

Accepted: 13 May 2024

Published: 20 May 2024



Copyright: © 2024 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/>).

1. Introduction

Over the past decades, numerous multi-robot architectures [1–5] have been created and examined to fulfill diverse missions. These multi-robot systems were designed to leverage the collective capabilities of multiple robots, allowing them to undertake tasks that surpass the inherent limitations of a single robot. The objective is to enhance overall system performance, efficiency, and task accomplishment in comparison to individual robotic agents. While multi-robot systems offer these advantages, there are instances where they fall short. A single robot such as Matilda [6], Packbot [7], TAROS [8], TALON [9], SherpaTT [10], and Ralter [11], may have diverse manipulation capabilities and additional functions provided by an extensive configuration, but such capabilities often necessitate a larger structure.

In order to bridge the gap between small multi-robot systems and large single-robot systems to better respond to the challenges of complex tasks and environments, modular robots with reconfigurable capabilities have been proposed. Various modular reconfigurable robot systems have been designed and studied with primary interests in robotic mobility, docking interfaces [12–14], and reconfiguration strategies [15,16].

A novel modular robotic system with a self-reconfiguration capability that offers greater versatility and better functionalities over the traditional fixed-structure robots or the typical multiple mobile robot systems is proposed in this paper. During the reconfiguration process, the target coupling mechanism of a multi-directional locomotion module with switching dynamics is required to align with the docking mechanism of another robot module, called the load carrier. Therefore, a well-defined controller that can maneuver the locomotion module of the STORM (Self-configurable and Transformable Omni-Directional Robotic Module) system to the desired pose is required for this process.

The locomotion module is considered a nonholonomic system that is drift-free, under-actuated, and nonlinear. According to Brockett's theory [17], these systems cannot be stabilized to a desired configuration using a time-invariant state feedback law, which makes it more challenging to develop a motion controller of the locomotion module for the autonomous alignment task. Previously, researchers developed both smooth and nonsmooth time-varying controllers for these systems. In general, the nonsmooth controller behaves better because it achieves an exponential rate of convergence. However, its convergence rate critically depends on the available gains. The oscillatory behavior of the mobile robots during the approach to the goal, which makes the motion rather erratic, is an intrinsic characteristic of time-varying control laws [18]. The polar coordinate transformation-based controller in [19] converges fast and generates smooth paths with a weak dependence on the choice of the control gain parameters. However, it is based on a unicycle model and cannot be generalized to more complicated robot models. The dynamic feedback linearization method [20] also has a fast convergence rate and is able to generate natural curves. However, the parameter tuning process requires a very large feasible set of PD gains to choose from. In this study, a Lyapunov function-based [21–26] pose stabilization controller was developed for the locomotion module to accomplish the autonomous alignment task.

In order to bring the robot module close enough to the target module to perform the aforementioned autonomous alignment tasks, a novel path optimization method is also proposed. The new path planning method takes the special mode switch capability (track or wheel) of the locomotion module into consideration and uses numerical methods to find the optimal path.

Therefore, the complete robot module control consists of two stages: the trajectory optimization stage and the motion control stage. The trajectory optimization stage aims to find an optimal path for the robot module when the robot module is away from its target module and the motion control stage aims to perform the module-to-module autonomous docking and alignment tasks when the robot module moves close enough to its target module.

The main contributions of this work are summarized as follows: (1) the development and initial validation of a pose stabilization controller for a novel self-reconfigurable modular mobile robot to achieve autonomous alignment and docking and (2) the development of a direct collocation-based trajectory optimization method to account for the switching mechanism of the locomotion module and thus improve the autonomous docking process.

The rest of this paper is organized as follows. Section 2 introduces the STORM robotic system and analyzes its functionalities in simulations. Section 3 presents the pose stabilization controller for the autonomous alignment and docking processes. Section 4 presents the direct collocation-based path planning algorithm for the improvement of the autonomous alignment control. Section 5 concludes the research and presents future work.

2. Overview of the STORM Robotic System

The proposed robotic system mainly consists of two major types of robotic modules—the load carrier module and the locomotion module—as shown in Figure 1. Each module is capable of navigating through and interacting with the environment independently. In addition, the robotic system is proposed to be able to reassemble autonomously into various configurations, including a humanoid setup, facilitating the system's ability to traverse obstacles such as ditches. For the autonomous reassembly process, a non-backdrivable active docking mechanism with two Degrees of Freedom (DOFs) was developed to integrate into the tracked units of the robot modules. The quantity and placement of the docking mechanism are flexible, allowing for customization to meet diverse mission and environmental requirements.

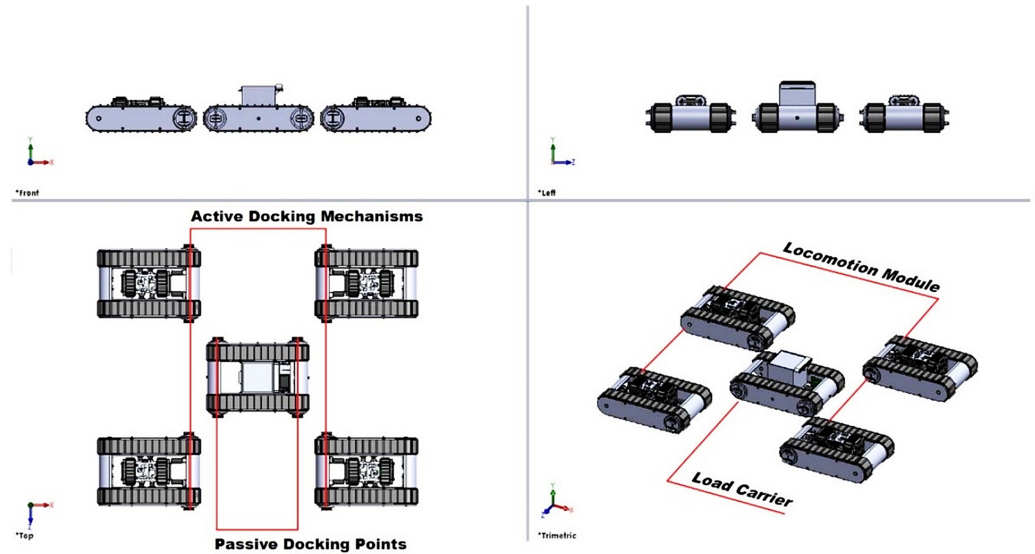


Figure 1. STORM robotic system: load carrier and locomotion modules.

The load carrier was designed for carrying and transporting packages. It incorporates four passive docking points and four cameras strategically positioned onboard to facilitate autonomous docking.

The locomotion module (Figure 2) consists of two longitudinal tracked units, two lateral tracked units, and a Vertical Translational Mechanism (VTM) that switches between the two modes of locomotion. An alternative setup for lateral motion involves replacing the tracks with wheels. In the mechanical design of the locomotion module, special attention was given to achieving multi-directional mobility, enhancing spatial alignment capabilities for autonomous docking functionality.

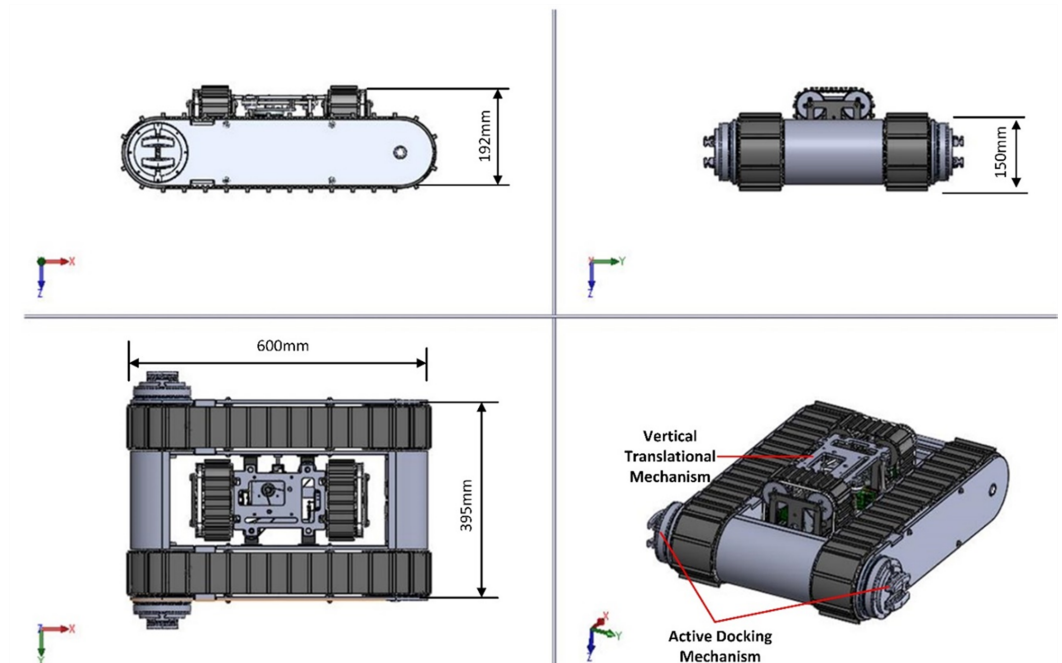


Figure 2. A 3D model of the locomotion module.

The active docking mechanism is essential for enabling the reconfigurability of the STORM system. The version of this docking mechanism implemented on the locomotion module was updated from those previously developed and discussed in [27]. Figure 3 demonstrates the worm gear assembly drives for the relative rotation between the docking

mechanism and the stationary robot frame. The worm drive was chosen for its high torque capability and non-backdriveable nature, allowing the safe rotation of the locomotion module for different configurations. A symmetric translation of the clamping profiles is achieved with a constant lead cam, selected for its compact design and non-backdriveable feature.

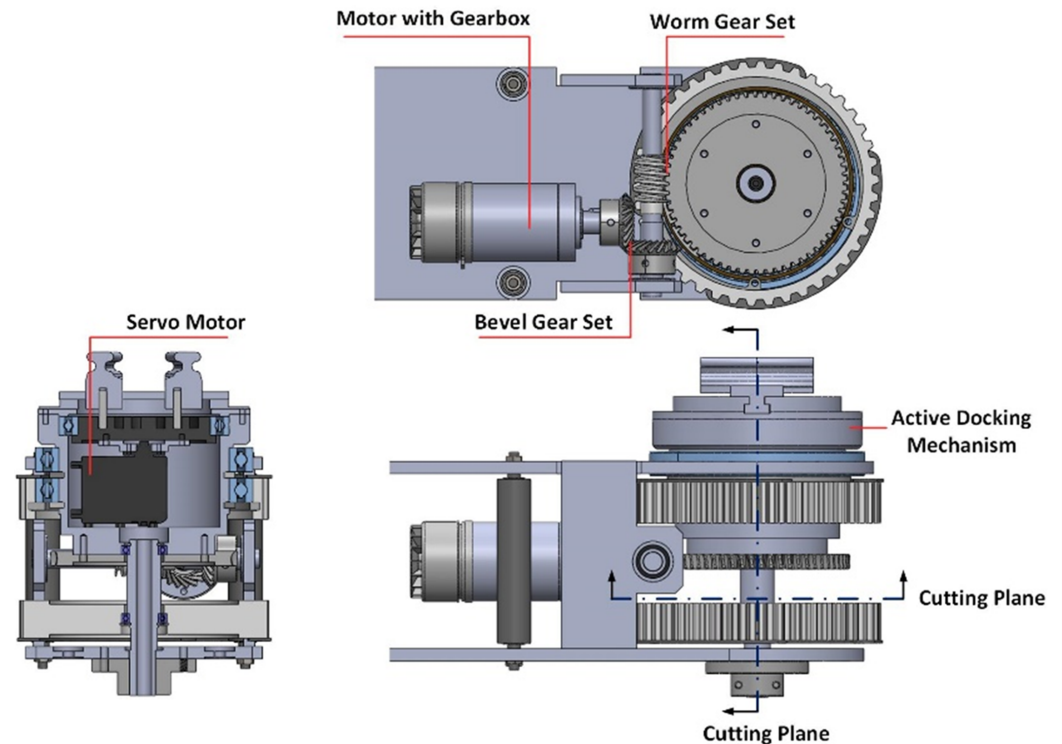


Figure 3. Active docking mechanism.

Simulations of the STORM System

To evaluate the overall functionalities and feasibility of the STORM system, simulations were conducted using the CoppeliaSim robot simulator [28]. The test environment included a variety of terrain types such as muddy terrain, ditches, rocky terrain, high hills, and narrow corridors. A load carrier module and two locomotion modules were used in the simulations to evaluate the performances of the robot modules in different configurations.

Preliminary tests involving three different configurations on diverse terrains were conducted, demonstrating that the proposed robotic modules successfully navigated through these terrains with distinct configurations. Ongoing tests will further analyze the robots' performance and limitations. Figure 4 illustrates various configurations designed for different terrain types. The robots adopted a diagonal line configuration to navigate rocky terrain, enhancing balance and preventing unforeseen issues like rollovers or getting stuck on large rocks (Figure 4a). For crossing a ditch, the robot module assumed a crab-like shape to maximize length, ensuring that the center of gravity (COG) falls on the side of the two locomotion modules to prevent rollover (Figure 4b). In the simulation of muddy terrain with a low coefficient of friction, a closed-form configuration was selected to evenly distribute traction and reduce slipping (Figure 4c). A triangular configuration was employed for ascending and descending hills to enhance stability and traction (Figure 4d). The large configuration was decoupled and oriented toward the narrow corridor, as shown in Figure 4). To enhance autonomy and prevent collisions, a local planner [29] based on reinforcement learning was developed.

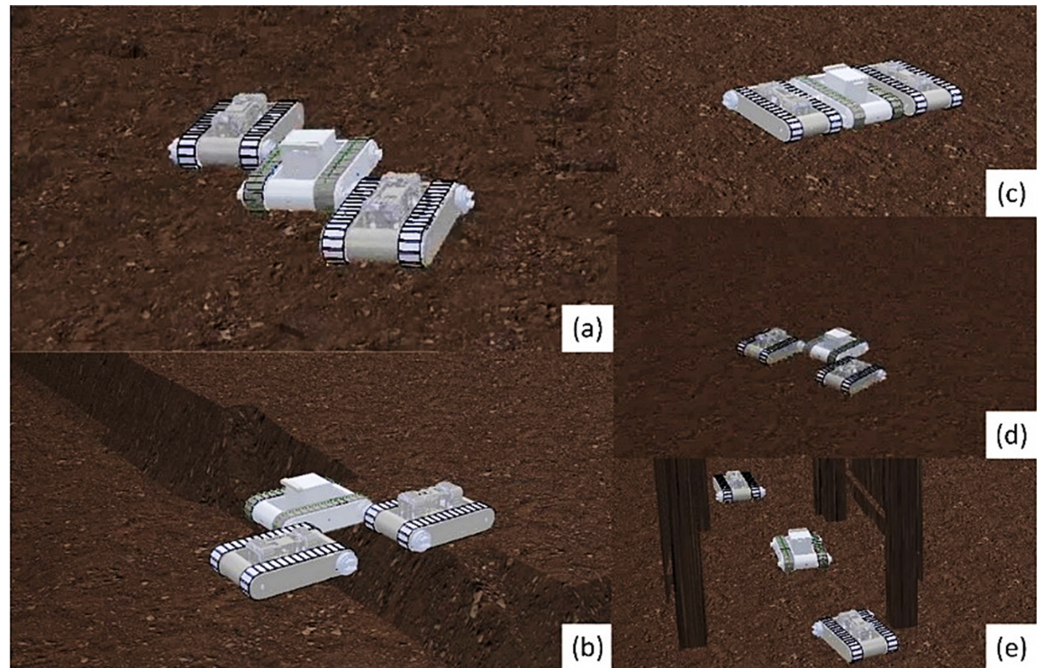


Figure 4. Different configurations for different terrains. (a) Diagonal form to pass the rocky terrain; (b) crab-like form to cross a ditch; (c) closed form to pass the muddy terrain; (d) triangle form to drag up the load carrier; (e) decoupled form to pass a narrow corridor.

3. Motion Control of the STORM Module toward Self-Reconfiguration

3.1. Problem Statement

To achieve autonomous docking, it is essential to develop efficient algorithms and controllers for the locomotion module. These systems must enable the module to approach the final pose while adapting to the unique characteristics of multi-directional locomotion and accommodating specific task requirements.

It is assumed that the load carrier will be stationary during the docking process. One of the locomotion modules in the lateral tracked mode should gradually approach the desired pose to align the two docking mechanisms. As shown in Figure 5, the determination of a desired pose for the locomotion module was based on the pose of the load carrier, ensuring that the two docking mechanisms align and couple together after the locomotion module reaches the target position. This task is formulated as a pose stabilization problem: the proposed controller takes the differences in pose between the current position and the desired position of the locomotion module as input and generates appropriate velocity commands for the module.

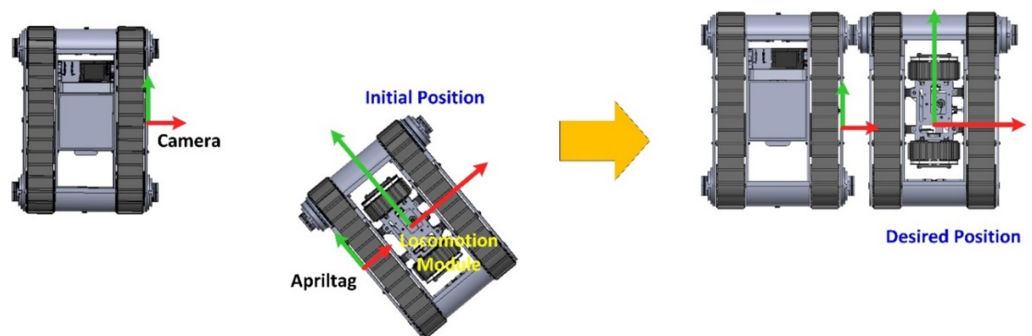


Figure 5. Initial and desired positions of the locomotion module.

3.2. Controller Design

Addressing the complexity of minimizing the error between an initial pose and the desired pose, which includes orientation, is challenging when dealing with a target system

characterized as nonlinear, nonholonomic, and under-actuated. Finding a smooth, time-invariant controller capable of globally asymptotically stabilizing the system under these conditions proves to be difficult [18,30]. In response to this challenge, we propose a solution: a Lyapunov function-based controller with a transit goal generator.

Studying robot kinematics is essential to establish the connection between the velocities of the robot and the speed of each track, facilitating the translation of commands from high-level controllers to low-level controllers. The load carrier shares the same kinematics as the locomotion module in the longitudinal tracked mode, as expressed by the following equation:

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

where v and ω are the linear and angular velocities of the locomotion module, respectively. (x, y) is the current position of the robot (defined at the center of the robot) with respect to the world coordinate frame. θ is the angle between the heading of the robot and the x -axis of the world frame. $\mathbf{x} = [x \ y \ \theta]^T$ is the state vector, and $\mathbf{u} = [v \ \omega]^T$ is the control input vector. Therefore, the primary objective of the controller is to find a control input $\mathbf{u}(t)$ that makes $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{x}_d$, where \mathbf{x}_d is the desired pose.

To prevent the scenario where the robot directly reaches the goal position without considering its heading orientation, a transient goal is generated for the robot. The transient goal generator provides alternative target positions until the robot can navigate to the goal position with the desired orientation. The overall control process is shown in Figure 6.

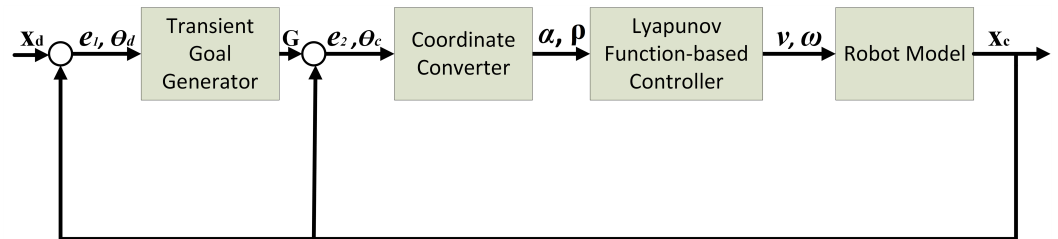


Figure 6. Control diagram of the proposed controller.

The transient goal generator creates a temporary goal according to the desired state $\mathbf{x}_d = [x_d \ y_d \ \theta_d]^T$ and the current pose $\mathbf{x}_c = [x_c \ y_c \ \theta_c]^T$ of the robot following the equation below:

$$\mathbf{G} = \begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \|\mathbf{e}_1\| \cdot \begin{bmatrix} \cos(\beta + \gamma) \\ \sin(\beta + \gamma) \end{bmatrix}, \quad (2)$$

where $\mathbf{e}_1 = [x_d - x_c \ y_d - y_c]^T$, $\beta = \tan^{-1}\left(\frac{y_d - y_c}{x_d - x_c}\right)$, and $\gamma = \beta - \theta_d$. Figure 7 shows the location of the transient goal; as the robot gets closer to the desired position, the transient goal tends to coincide with the desired position. Throughout this process, the locomotion module operates in lateral tracked mode, and the robot's heading is denoted by the yellow arrow in the figure. This heading has a 90-degree offset compared to the longitudinal tracked mode. The difference between the coordinates of the transient goal and the current position of the robot, $\mathbf{e}_2 = [x_g - x_c \ y_g - y_c]^T$, as well as the heading angle θ_c of the robot, will be fed to a coordinate converter that aims to represent the robot poses in the Polar coordinate system as $\rho = \|\mathbf{e}_2\|$ and $\alpha = \varphi - \theta_c$, where $\varphi = \tan^{-1}\left(\frac{y_g - y_c}{x_g - x_c}\right)$. The outputs ρ and α are the inputs to the Lyapunov function-based controller. The kinematics of the robot are first converted into polar coordinates to develop the control laws:

$$\begin{cases} \dot{\rho} = -v \cos \alpha \\ \dot{\varphi} = v \frac{\sin \alpha}{\rho} \\ \dot{\theta}_c = \omega \end{cases} \quad (3)$$

Then, the control laws are designed to be

$$\begin{aligned} v &= k_1 \rho \cos \alpha \\ \omega &= k_1 \sin \alpha \cos \alpha + k_2 \alpha \end{aligned} \quad (4)$$

where k_1 and k_2 are the gains to be analyzed and tuned to obtain a stable system. The control laws come from a positive definite quadratic form of the Lyapunov function: $V = V_1 + V_2 = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2$. The time derivative can be calculated as follows:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = \rho\dot{\rho} + \alpha\dot{\alpha} = \rho(-v \cos \alpha) + \alpha(v \frac{\sin \alpha}{\rho} - \omega) \quad (5)$$

If a continuous function $V : \mathbf{U} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ is positive definite such that $V(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$ and its derivative $-\dot{V}$ is positive definite, then the origin of the system is globally asymptotically stable. In order to obtain a stable system, $-\dot{V}$ must be positive definite. Let $V_1 = -k_1\rho^2\cos^2\alpha$ and $v = k_1\rho \cos \alpha$, where $k_1 > 0$. Substitute v into Equation (5) and obtain $\dot{V}_2 = \alpha(k_1 \cos \alpha \sin \alpha - \omega)$, where $\omega = k_1 \sin \alpha \cos \alpha + k_2\alpha$ and $k_2 > 0$. Therefore, the system is proved to be globally asymptotically stable at the origin.

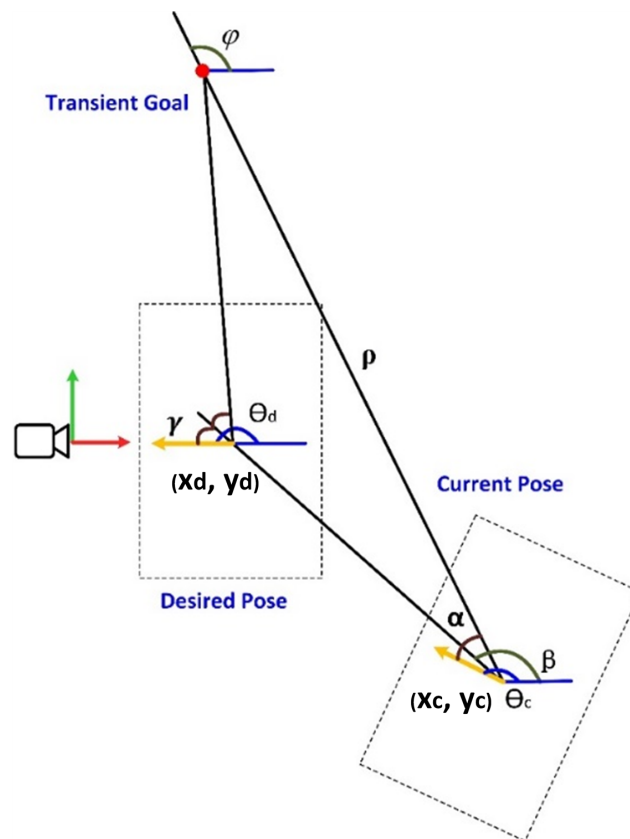


Figure 7. The robot poses and parameters.

After receiving the linear velocity v and angular velocity ω from the controller, a skid-steer robot model calculates the velocities of the right and left tracks by estimating the Instantaneous Center of Rotation (ICR) for the left and the right units [31,32], $C_2 = (x_{c2}, y_{c2})$

and $C_3 = (x_{c3}, y_{c3})$, as shown in Figure 8. Then, the velocities of the left and right tracks v_L and v_R can be calculated as follows:

$$\begin{aligned} v_L &= -|x_{c2}|\omega + v \\ v_R &= |x_{c3}|\omega + v \end{aligned} \tag{6}$$

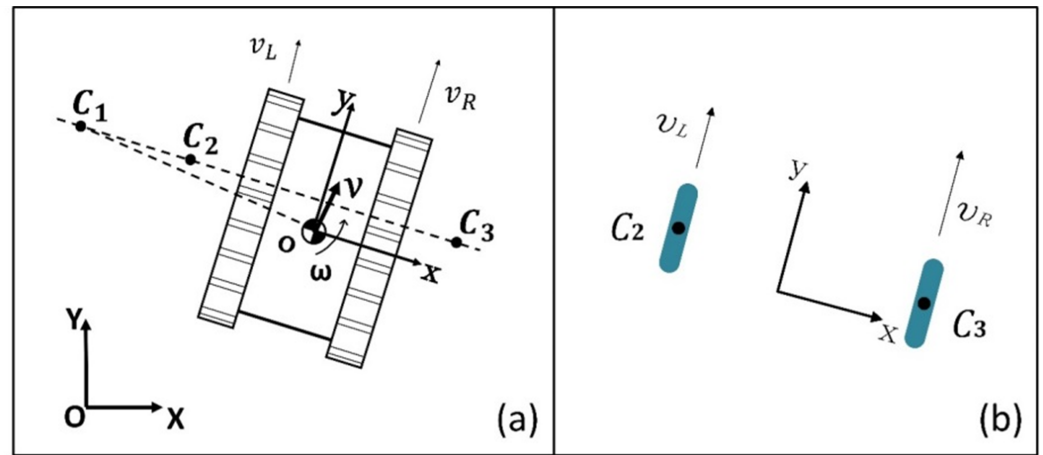


Figure 8. Kinematic model for (a) skid-steering robots: converting linear and angular velocities to track velocities (b) equivalent wheel contact points of differential drive robot.

3.3. Simulation Validation

The pose stabilization controller was implemented using Simulink in MATLAB to test the feasibility and tune the parameters of the controller. The Simulink module is presented in Figure 9. The simulation’s stop time was configured to 30 s with a time step of 0.1.

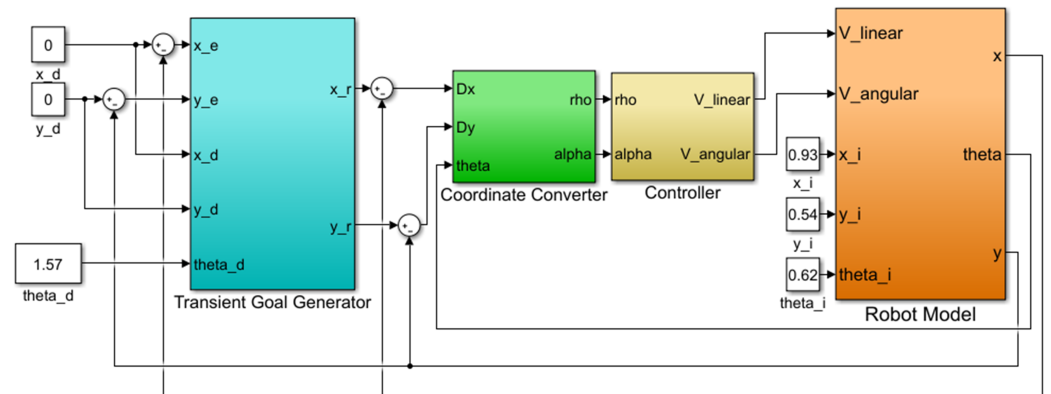


Figure 9. Simulink model of the overall control process.

Four trajectories of the robot are plotted and presented in Figure 10 to demonstrate the performance of the controller. The desired position is set at the origin of the world frame. Four random initial positions, selected from different quadrants within 1.5 m of the target, were used to evaluate the controller. The time intervals between adjacent triangle marks on the trajectories are uniform. The results demonstrate that the robot successfully reaches the desired pose in each run. The desired position is located at the origin of the world frame. Four random initial positions were selected from different quadrants within 1.2 m of the target. It should be pointed out that the controller aims to steer the robot to a transient goal until the robot’s heading is the same as the desired pose. Since the robot allows backward motion, the desired heading of the robot should be determined according to the initial position of the robot. From the results, the robot successfully achieved the desired pose in each run.

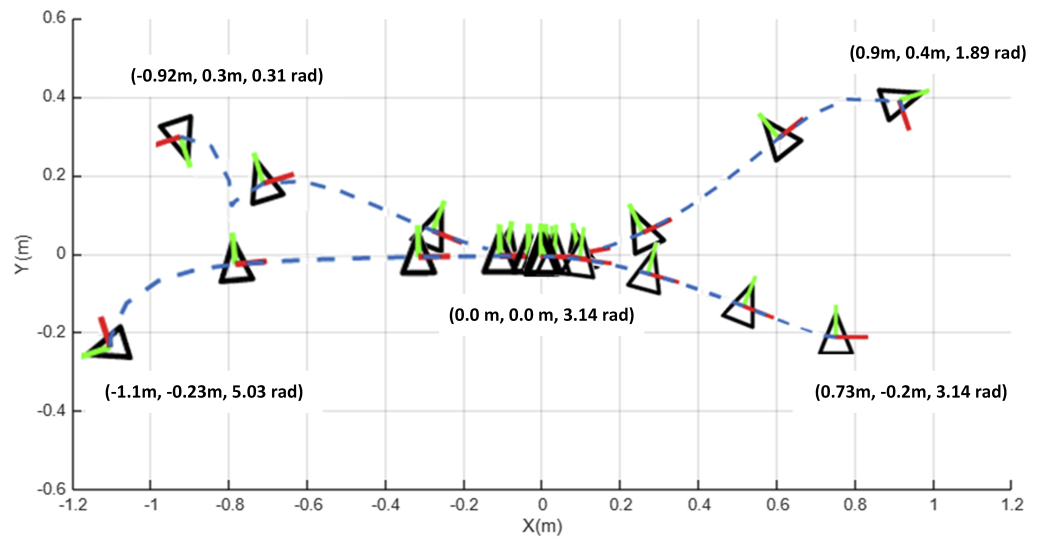


Figure 10. Trajectories of the robot model from the simulation in MATLAB.

3.4. Real-World Validation

A real-world experiment was carried out to validate the feasibility of the proposed controller and the docking process. The essential requirement for this autonomous alignment is a dependable sensing system used to accurately track the poses of the locomotion module. Apriltags [33] were utilized for localizing the locomotion module during the experiment. As shown in Figure 11, an Apriltag was positioned on the track frame of a prototype locomotion module, and a USB camera was configured to track the tag.

The experiment was designed to align the passive docking mechanism on the locomotion module with the active docking mechanism next to the camera. Subsequently, it sought to demonstrate that the docking mechanism could securely hold and rotate the locomotion module to fulfill various task requirements (Figure 11).

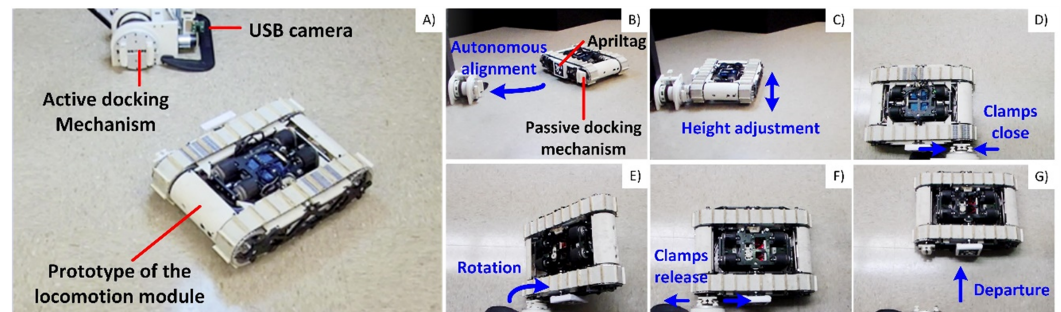


Figure 11. Experimental setup and the overall process.

The ROS package *apriltag_ros* was utilized to publish transformations between the camera frame and the tag frame. To facilitate kinematic analysis, the output relative pose was converted from the camera frame’s relative pose to the local frame of the locomotion module. In ensuring continuous visibility, the Apriltag should remain within the camera’s field of view during autonomous alignment. Upon reaching the desired position, the Vertical Translational Mechanism (VTM) adjusted the height of the passive docking mechanism for alignment along the z-axis. Subsequently, the clamps of the active docking mechanism closed, and the locomotion module rotated 180 degrees. Following the rotation of the locomotion module by 180 degrees through the active docking mechanism, as depicted in Figure 12e, the subsequent Figure 12f shows the module positioned upside down. Given the symmetrical structure of the locomotion module, its vertical translational mechanism can adjust the lateral wheels’ position to ensure that the longitudinal tracked units maintain contact with the ground, thereby enabling longitudinal motions. While the frame remains

consistent, the control commands are recalculated to take into account the information from the IMU. The robot then departed following the release of the clamps. However, it is worth noting that this study did not address the motion control problem after the docking mechanism was released. This post-releasing control will be the focus of our future work, particularly concerning aspects such as formation control.

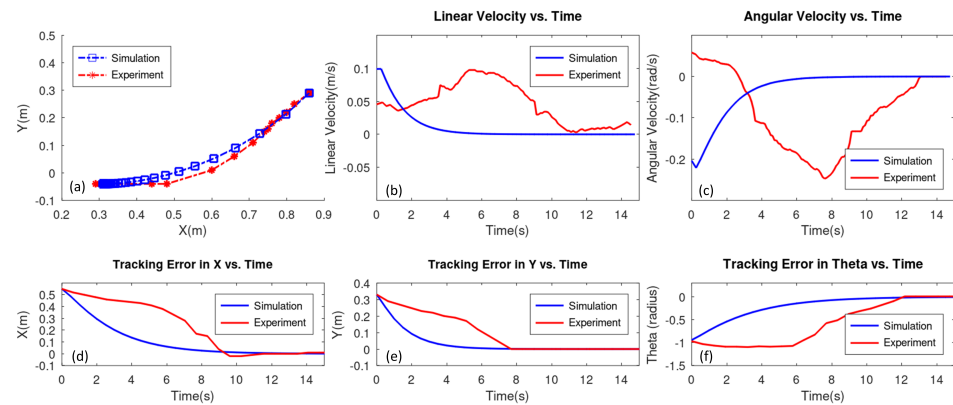


Figure 12. Comparison between the simulation and real–world experiment: (a) paths, (b) linear velocities, (c) angular velocities, (d) tracking error in x coordinates, (e) tracking error in y coordinates, and (f) tracking error in theta.

The recorded trajectory of the autonomous alignment process was compared with that of simulation results, as shown in Figure 12. The experimental and simulation findings emphasize the controller’s proficiency in minimizing error between the current and desired pose, encompassing both position and orientation. Discrepancies between simulation and real-world outcomes can be ascribed to various factors acting as disturbances. Despite facing challenges like localization measurement errors, robot slippage, and response delay, the real-world experiment showcased the controller’s effective disturbance rejection performance. This underscores the controller’s ability to navigate unforeseen disruptions while maintaining precise pose control.

4. Trajectory Optimization and Tracking of the STORM Module to Improve the Autonomous Alignment Process

The pose stabilization controller presented in the previous section successfully maneuvered the locomotion module to the desired pose with allowable misalignments for completing the reconfiguration task. However, the prerequisite of this controller is that the locomotion module has already been at a close point to the final destination and is on the wheeled/lateral locomotion mode to perform the autonomous alignment task. Therefore, decisions on how to reach this close point and when to switch from the tracked mode to the wheeled mode are still required for the autonomous alignment task. This section presents the solution to making these decisions.

Similar to other path planning algorithms, the direct collocation method was used to find the fastest path in this study. However, the uniqueness of the solution here is that it takes the special mode-switching ability of the locomotion module into consideration and plans not only the fastest path but also the optimal switching time, which is essentially a numerical solution for the time-optimal problem of a switching system [34].

4.1. Problem Statement

In our cases, the hybrid tracked-wheeled feature of the locomotion module introduces discontinuous system dynamics. For example, the longitudinal locomotion mode, lateral locomotion mode, and the motion of the VTM are intrinsically discrete. Therefore, previous motion planning for the autonomous docking task could be divided into several sub-tasks as follows:

1. Rough navigation to the desired position using tracks;
2. Fine adjustment of the robot poses in lateral locomotion mode;
3. Fine adjustment of the height of the VTM to align the docking mechanism in the Z direction;
4. Fine adjustment of the clamps' positions.

To find an optimal solution for the locomotion module to approach the target and choose the best position to switch the dynamics, the problem can be defined as a switched optimal control problem, and a direct collocation-based optimization method is proposed to solve this problem. To apply the direct collocation method, the continuous-time problem is converted into a nonlinear program by discretizing the trajectory into collocation points. The system dynamics are then converted into a set of constraints that apply to the state and control at the collocation points. In this way, both the state and the control variables become a finite set of decision variables.

The proposed trajectory optimization methods should plan an optimal path from the initial pose to the desired pose and determine the position for the locomotion module to switch the dynamics simultaneously. The locomotion module approaches the desired position for reconfiguration in the longitudinal tracked mode. After it arrives at the point determined by the proposed direct collocation-based path planner, the VTM will lower down the lateral tracked unit to change the locomotion mode.

Two different approaches were proposed to determine the position to switch the dynamics. When optimizing the trajectories, the mobile robot is considered an ideal agent moving on flat ground, and there is no slippage between the robot and the ground. The robot dynamics are fully described by the following two dynamics, which correspond to the track and wheel modes. The disturbances from uneven terrain and the robot's slipperiness are taken into consideration when tracking the trajectory.

4.2. Problem Formulation

The trajectory optimization aims to discover an optimal path according to an objective function that mathematically describes what the best solution should be. In general, an objective function can be defined in Bolza form:

$$J(t_0, t_f, \mathbf{x}(t), \mathbf{u}(t)) = \int_{t_0}^{t_f} l(t, \mathbf{x}(t), \mathbf{u}(t)) dt + g(t_0, t_f, \mathbf{x}(t_0), \mathbf{x}(t_f)), \quad (7)$$

where the state $\mathbf{x} \in \mathbb{R}^n$, control $\mathbf{u} \in \mathbb{R}^m$, and the initial and final time points (t_0, t_f) are considered the decision variables. The integral of $l : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ defines the running cost, and the function $g : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ defines the boundary objective. If $l \equiv 0$, the problem is said to be in Mayer form. If $g \equiv 0$, it becomes a Lagrange problem. In our case, the best solution is defined as the fastest path from the initial point to the final position, and the objective function is in Lagrange form:

$$J = \int_{t_0}^{t_f} 1 dt, \quad (8)$$

The locomotion module is a hybrid system with discontinuous dynamics. Generally, these control systems have two types of switching behaviors [34,35]. One is the autonomous or uncontrolled switch without the influence from a separate switching mechanism. The second type of switch can be controlled directly, as the switch between the longitudinal tracked mode and vertical tracked mode through the VTM of the locomotion module. In our case, the switch of the system will happen only once during the entire autonomous alignment process. The state variables for the hybrid system are $\mathbf{x} = [x \ y \ \theta]^T$, where x and y represent the position of the locomotion module in the $\{XOY\}$ plane, and θ is the direction of the heading of the locomotion module. The control inputs are $\mathbf{u} = [v \ \omega]^T$, where v and ω are the linear and angular speeds of the locomotion module, respectively. Before switching,

the system dynamics of the locomotion module in the longitudinal tracked mode can be formulated as follows:

$$f_1 = \dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (9)$$

After switching, the system dynamics of the locomotion module in the lateral tracked mode become

$$f_2 = \dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta + \frac{\pi}{2}) & 0 \\ \sin(\theta + \frac{\pi}{2}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (10)$$

Then, the problem can be formulated as follows:

minimize J

subject to:

$$\text{dynamics: } \dot{\mathbf{x}}(t) = \begin{cases} f_1(\mathbf{x}(t), \mathbf{u}(t)) & t \leq t_s \\ f_2(\mathbf{x}(t), \mathbf{u}(t)) & t > t_s \end{cases} \quad (11)$$

boundary constraints: $\mathbf{x}(0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{x}_f$

path constraints: $-\mathbf{x}_{\max} \leq \mathbf{x} \leq \mathbf{x}_{\max}, -\mathbf{u}_{\max} \leq \mathbf{u} \leq \mathbf{u}_{\max}$

4.3. Proposed Solution

The first step to solve the optimization problem is to define the collocation points by discretizing the trajectory into a finite set of decision variables according to the following steps:

1. Partition the time interval T into M+N subintervals to compute a piecewise constant control sequence $\{\mathbf{u}_{1,1}, \mathbf{u}_{1,2}, \dots, \mathbf{u}_{1,M}\}$ and $\{\mathbf{u}_{2,M+1}, \mathbf{u}_{2,M+2}, \dots, \mathbf{u}_{2,M+N}\}$; then, the corresponding states at the collocation points are defined as $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{M+N}\}$.
2. Convert the system dynamics into collocation constraints, as follows:

- (a) The dynamics can be constructed in integral form:

$$\begin{cases} \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}} dt = \int_{t_k}^{t_{k+1}} f_1 dt & k = \{1, 2, \dots, M-1\} \\ \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}} dt = \int_{t_k}^{t_{k+1}} f_2 dt & k = \{M, \dots, M+N-1\} \end{cases}, \quad (12)$$

- (b) Then, the discretized system dynamics can be approximated using trapezoidal quadrature. The approximation is applied between every pair of collocation points as follows:

$$\begin{cases} \mathbf{x}_{k+1} - \mathbf{x}_k = \frac{1}{2} h_k (f_{1,k+1} + f_{1,k}) & k = \{1, 2, \dots, M-1\} \\ \mathbf{x}_{k+1} - \mathbf{x}_k = \frac{1}{2} h_k (f_{2,k+1} + f_{2,k}) & k = \{M, \dots, M+N-1\} \end{cases}, \quad (13)$$

where $h_k = t_{k+1} - t_k$ and t_k is the time at collocation point k . Then, the system dynamics at each collocation point become

$$\begin{cases} f_{1,k} = f_1(t_k, \mathbf{x}_k, \mathbf{u}_k) & k = \{1, 2, \dots, M\} \\ f_{2,k} = f_2(t_k, \mathbf{x}_k, \mathbf{u}_k) & k = \{M+1, \dots, M+N\} \end{cases}, \quad (14)$$

where $t_k, x_k,$ and u_k are the decision variables in the nonlinear program.

3. Enforce the path constraints at each collocation point— $-\mathbf{x}_{\max} \leq \mathbf{x}_k \leq \mathbf{x}_{\max}, -\mathbf{u}_{\max} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}$ —and the boundary constraints should be enforced at the first and the last collocation points.

4. Find the approximate objective function using the trapezoid rule as follows:

$$\int_{t_0}^{t_f} l(t, \mathbf{x}(t), \mathbf{u}(t)) \approx \sum_0^{M+N-1} \frac{1}{2} h_k (l_k + l_{k+1}) \quad (15)$$

where l_k and l_{k+1} are the costs at points k and $k + 1$. In following the procedures above, the switching dynamics are enforced at the specific collocation point M . However, a further analysis of the timing to change the dynamics is required. Two different methods were utilized to determine the switching point. In the first method, a fixed time interval is set, the switching point at different collocation points is gradually chosen, and the nonlinear program with different switching points to obtain $M + N - 1$ paths is solved. Then, the fastest path is selected as the final solution. For convenience, this approach is referred to as the “discrete method” in the following discussion since it mainly optimizes on the discrete collocation points. The second method is to define different time intervals Δt_1 and Δt_2 for the two different dynamics. This approach enables us to determine the optimal timing of the switching point by solving the nonlinear programming only once. This method is referred to as the “continuous method” since it mainly optimizes on the continuous time variable. In this method, the objective function becomes

$$\int_{t_0}^{t_f} l(t, \mathbf{x}(t), \mathbf{u}(t)) \approx \sum_0^{M-1} \frac{1}{2} \Delta t_1 (l_i + l_{i+1}) + \sum_M^{M+N-1} \frac{1}{2} \Delta t_2 (l_j + l_{j+1}) \quad (16)$$

where $M \in \mathbb{Z} : 0 \leq M \leq N - 2$, and $N \in \mathbb{Z} : N \geq M + 1$ are predefined constant values. l_i and l_{i+1} represent the costs at point i and $i + 1$ for the longitudinal tracked mode, while l_j and l_{j+1} denote the costs at point j and $j + 1$ for the lateral locomotion mode.

4.4. Numerical Results

The proposed direct collocation-based method for solving an optimal path with a proper switching position was implemented in Matlab using the *fmincon* solver. As mentioned in Section 4.3, two different approaches were developed to solve the problems. Figures 13–15 show the results from the first iterative method (the “discrete method”) in which fifteen collocation points were defined and the switching point was enforced at the first collocation point to the fourteenth points successively. This was to ensure that the robot module will be in the lateral tracked mode in the end.

Different sets of boundary and path constraints were chosen to assess the proposed method. In Figure 13a, all optimal paths are presented under uniform boundary conditions: $\mathbf{x}_0 = (0 \text{ m}, 0 \text{ m}, 0 \text{ rad})$ and $\mathbf{x}_f = (1 \text{ m}, 1 \text{ m}, 0 \text{ rad})$, control constraints $\mathbf{u}_{\max} = (0.5 \text{ m/s}, 0.5 \text{ rad/s})$, and consistent dynamic constraints. The switching points are marked with red dots. Figure 13b illustrates the fastest path, while Figure 13c demonstrates the control signals required for achieving the optimal trajectory. Similarly, Figure 14 displays results with the initial position set to $\mathbf{x}_0 = (3 \text{ m}, 0 \text{ m}, \pi \text{ rad})$, final pose at $\mathbf{x}_f = (1 \text{ m}, 1 \text{ m}, 0 \text{ rad})$, and boundary control inputs $\mathbf{u}_{\max} = (0.1 \text{ m/s}, 0.1 \text{ rad/s})$. The boundary conditions for the results shown in Figure 15 are $\mathbf{x}_0 = (1.6 \text{ m}, -0.5 \text{ m}, \pi \text{ rad})$, $\mathbf{x}_f = (0.26 \text{ m}, 0 \text{ m}, 0.5\pi \text{ rad})$, and $\mathbf{u}_{\max} = (0.1 \text{ m/s}, 0.1 \text{ rad/s})$. In summary, this study shows that the proposed method effectively handles different boundary and path constraints for trajectory optimization.

Additionally, this study evaluated the results obtained from the second method (the “continuous method”), as described in Section 4.3, which formulates the objective function using different time intervals for the two dynamic constraints, under the same boundary conditions as those utilized in the first method. The outcomes from each method were compared, as illustrated in Figure 16. Despite slight variations, the comparison reveals that the optimal trajectories from different methods largely align. In conclusion, the findings across all tests successfully identify the optimal trajectory that satisfies all constraints, including a switching point to adapt dynamics. This underscores the practicality and efficiency of the proposed direct collocation-based optimization approach for path planning

in hybrid systems. Moreover, the novel trajectory optimization method demonstrated its feasibility in planning a time-optimal path and determining the optimal timing for switching robot dynamics. Future real-world tests will be conducted to further analyze the effectiveness of the planner.

It is worth noting that the control inputs obtained from both methods show bang-bang control characteristics, which are commonly found in optimal control problems, especially in minimum time problems (or shortest path problems with a bounded speed range). One similar example is the classic Dubins shortest path problem [36,37] wherein the optimal path is found to consist of line segments and arcs (which are generated by saturating its steering control, i.e., fully steering to the left or the right).

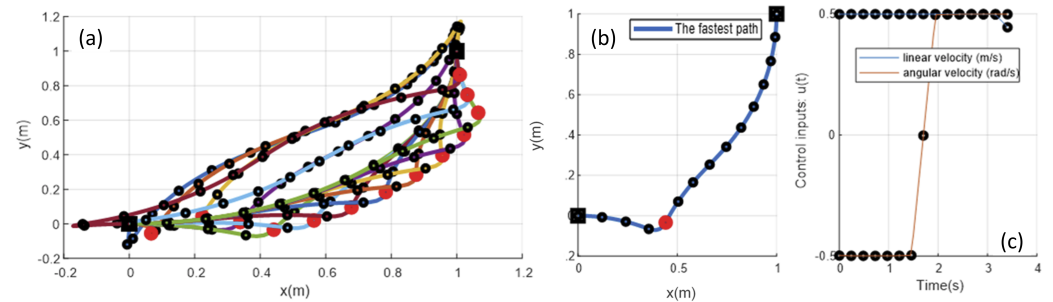


Figure 13. Solutions with boundary constraints $x_0 = (0 \text{ m}, 0 \text{ m}, 0 \text{ rad})$ and $x_f = (1 \text{ m}, 1 \text{ m}, 0 \text{ rad})$ and path control constraints $u_{\max} = (0.5 \text{ m/s}, 0.5 \text{ rad/s})$: (a) all the planned paths with switching points (red dots); (b) the fastest path; and (c) control signals presented.

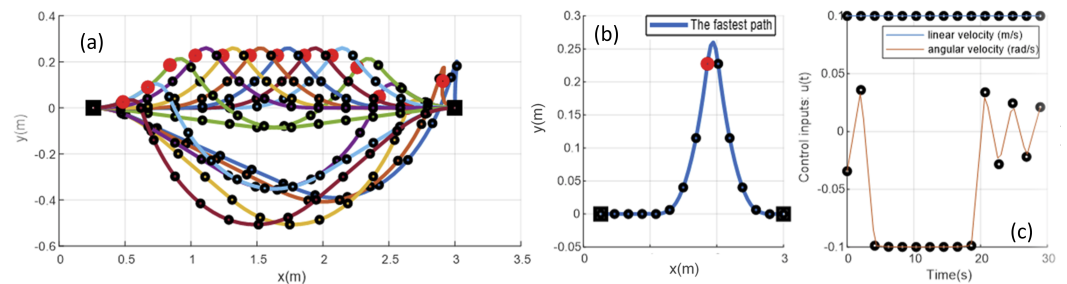


Figure 14. Solutions with boundary constraints $x_0 = (3 \text{ m}, 0 \text{ m}, \pi \text{ rad})$ and $x_f = (0.26 \text{ m}, 0 \text{ m}, 0.5\pi \text{ rad})$ and path control constraints $u_{\max} = (0.1 \text{ m/s}, 0.1 \text{ rad/s})$: (a) all the planned paths with switching points (red dots); (b) the fastest path; and (c) control signals presented.

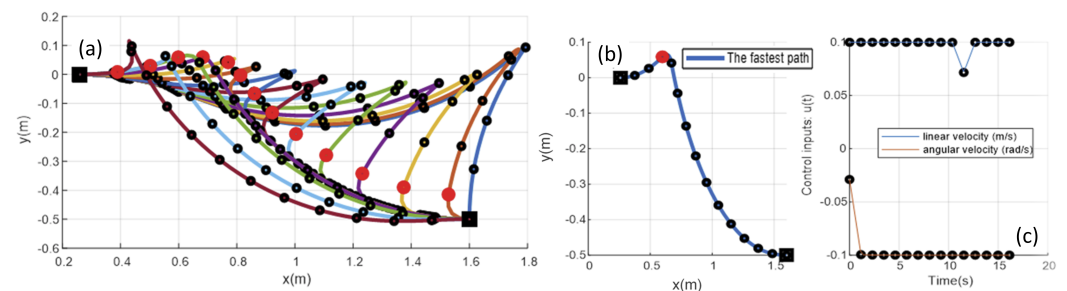


Figure 15. Solutions with boundary constraints $x_0 = (1.6 \text{ m}, -0.5 \text{ m}, \pi \text{ rad})$ and $x_f = (0.26 \text{ m}, 0 \text{ m}, 0.5\pi \text{ rad})$ and path control constraints $u_{\max} = (0.1 \text{ m/s}, 0.1 \text{ rad/s})$: (a) all the planned paths with switching points (red dots); (b) the fastest path; and (c) control signals presented.

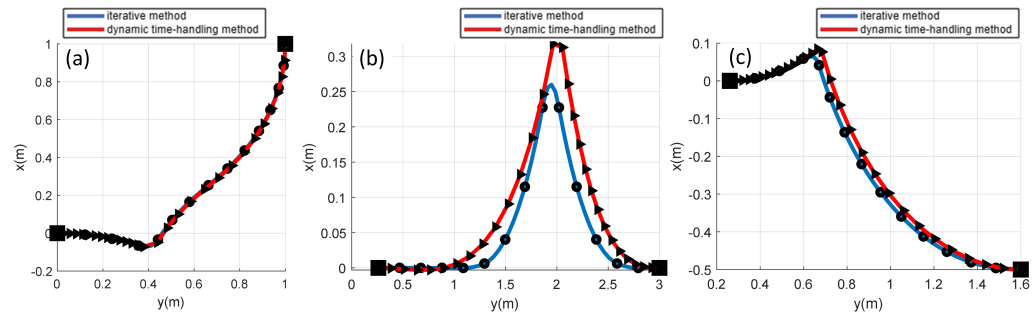


Figure 16. Comparisons between results from different methods. The blue line represents the fastest path from the first method, and the red path is the solution of the second method: (a) $\mathbf{x}_0 = (0 \text{ m}, 0 \text{ m}, 0 \text{ rad})$, $\mathbf{x}_f = (1 \text{ m}, 1 \text{ m}, 0 \text{ rad})$, and $\mathbf{u}_{\max} = (0.5 \text{ m/s}, 0.5 \text{ rad/s})$; (b) $\mathbf{x}_0 = (3 \text{ m}, 0 \text{ m}, \pi \text{ rad})$, $\mathbf{x}_f = (0.26 \text{ m}, 0 \text{ m}, 0.5\pi \text{ rad})$, and $\mathbf{u}_{\max} = (0.1 \text{ m/s}, 0.1 \text{ rad/s})$; (c) $\mathbf{x}_0 = (1.6 \text{ m}, -0.5 \text{ m}, \pi \text{ rad})$, $\mathbf{x}_f = (0.26 \text{ m}, 0 \text{ m}, 0.5 \pi \text{ rad})$, and $\mathbf{u}_{\max} = (0.1 \text{ m/s}, 0.1 \text{ rad/s})$.

5. Conclusion and Future Work

This paper introduced autonomous alignment and docking control for a self-reconfigurable modular robotic system. An overview of the proposed robotic system is provided, and simulations in CoppeliaSim validated its adaptability to various terrains through reconfiguration. A Lyapunov function-based pose stabilization controller was developed and verified in MATLAB simulations for autonomous alignment. Experimental validation was conducted, showcasing the locomotion module's ability to locate, dock, flip, and drive away from a standalone docking mechanism.

In taking into account the unique mode-switching ability of the modular mobile robot, an optimization approach was introduced to assist the locomotion module in determining the optimal position for switching dynamics and achieving an optimal trajectory for autonomous alignment.

Future work will focus on further analysis and validation. Mechanical design improvements, such as enhanced packaging space, smooth operation, and the increased durability of the active docking mechanism, are envisioned. Prototypes will be manufactured for testing. On the software side, algorithms and controllers will be extended for multiple-robot architectures and implemented on prototype robot modules for real-world experiments.

The current locomotion module can adjust its pose during autonomous alignment, but there are limitations in roll direction adjustments. Experiments assumed reconfiguration on relatively flat terrain within allowable misalignment limits. Future improvements may involve designing a Vertical Translational Mechanism (VTM) capable of rotating tracks to adjust the roll, enhancing the reconfiguration capabilities for challenging terrains.

Author Contributions: Conceptualization, S.F., Y.L. and P.B.-T.; methodology, S.F. and Y.L.; software, S.F.; hardware, S.F. and I.P.; validation, S.F.; formal analysis, S.F.; investigation, S.F.; resources, S.F. and P.B.-T.; data curation, S.F.; writing—original draft preparation, S.F.; writing—review and editing, S.F., Y.L. and P.B.-T.; visualization, S.F.; supervision, P.B.-T.; project administration, P.B.-T.; funding acquisition, P.B.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Defense Advanced Research Projects Agency under grant number FA8100-19-P-0022.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author (accurately indicate status).

Acknowledgments: The authors would like to acknowledge Ruichang Chen's contributions to the simulations in CoppeliaSim. This material is based on work supported by (while serving at) the National Science Foundation.

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

References

1. Matarić, M.J. Issues and approaches in the design of collective autonomous agents. *Robot. Auton. Syst.* **1995**, *16*, 321–331. [[CrossRef](#)]
2. Parker, L. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Trans. Robot. Autom.* **1998**, *14*, 220–240. [[CrossRef](#)]
3. Parker, L.E. Lifelong Adaptation in Heterogeneous Multi-Robot Teams: Response to Continual Variation in Individual Robot Performance. *Auton. Robot.* **2000**, *8*, 239–267. [[CrossRef](#)]
4. Bogue, R. Growth in e-commerce boosts innovation in the warehouse robot market. *Ind. Robot. Int. J.* **2016**, *43*, 583–587. [[CrossRef](#)]
5. Baker, P.; Canessa, M. Warehouse design: A structured approach. *Eur. J. Oper. Res.* **2009**, *193*, 425–436. [[CrossRef](#)]
6. Munkeby, S.H.; Jones, D.; Bugg, G.; Smith, K. Applications for the MATILDA robotic platform. In *Unmanned Ground Vehicle Technology IV*; SPIE: Bellingham, WA, USA, 2002; Volume 4715. [[CrossRef](#)]
7. Yamauchi, B.M. PackBot: A versatile platform for military robotics. In *Unmanned Ground Vehicle Technology VI*; SPIE: Bellingham, WA, USA, 2004; Volume 5422. [[CrossRef](#)]
8. Kot, T.; Novak, P. Application of virtual reality in teleoperation of the military mobile robotic system TAROS. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 172988141775154. [[CrossRef](#)]
9. Wells, P.; Deguire, D. TALON: A universal unmanned ground vehicle platform, enabling the mission to be the focus. In *Unmanned Ground Vehicle Technology VII*; SPIE: Bellingham, WA, USA, 2005; Volume 5804. [[CrossRef](#)]
10. Dimastrogiovanni, M.; Cordes, F.; Reina, G. Terrain Estimation for Planetary Exploration Robots. *Appl. Sci.* **2020**, *31*, 6044. [[CrossRef](#)]
11. Purvis, J.; Klarer, P. RATLER: Robotic All-Terrain Lunar Exploration Rover. In Proceedings of the 6th Annual Workshop on Space Operations Applications and Research (SOAR 1992), Washington, DC, USA, 4–6 August 1993; Volume 2, pp. 174–179.
12. Moubarak, P.; Ben-Tzvi, P. Modular and reconfigurable mobile robotics. *Robot. Auton. Syst.* **2012**, *60*, 1648–1663. [[CrossRef](#)]
13. Saab, W.; Racioppo, P.; Ben-Tzvi, P. A review of coupling mechanism designs for modular reconfigurable robots. *Robotica* **2019**, *37*, 378–403. [[CrossRef](#)]
14. Hossain, K.M.; Nelson, C.A.; Dasgupta, P. Enumeration of Configurations and Their Kinematics for ModRED II Modular Robots. *J. Mech. Robot.* **2017**, *9*, 054501. [[CrossRef](#)]
15. Yao, M.; Xiao, X.; Belke, C.H.; Cui, H.; Paik, J. Optimal Distribution of Active Modules in Reconfiguration Planning of Modular Robots. *J. Mech. Robot.* **2019**, *11*, 011017. [[CrossRef](#)]
16. Sohal, S.S.; Sebastian, B.; Ben-Tzvi, P. Autonomous Docking of Hybrid-Wheeled Modular Robots With an Integrated Active Genderless Docking Mechanism. *J. Mech. Robot.* **2022**, *14*, 011010. [[CrossRef](#)]
17. Brockett, R. Asymptotic stability and feedback stabilization. *Differ. Geom. Control Theory* **1983**, *27*, 181–191.
18. De Luca, A.; Oriolo, G.; Vendittelli, M. Control of Wheeled Mobile Robots: An Experimental Overview. *RAMSETE Articul. Mob. Robot. Serv. Technol.* **2002**, *25*, 181–226. [[CrossRef](#)]
19. Aicardi, M.; Casalino, G.; Bicchi, A.; Balestrino, A. Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robot. Autom. Mag.* **1995**, *2*, 27–35. [[CrossRef](#)]
20. De Luca, A.; Oriolo, G.; Vendittelli, M. Stabilization of the Unicycle Via Dynamic Feedback Linearization. *IFAC Proc. Vol.* **2000**, *33*, 687–692. [[CrossRef](#)]
21. Clarke, F.H.; Ledyae, Y.S.; Stern, R.J. Asymptotic Stability and Smooth Lyapunov Functions. *J. Differ. Equ.* **1998**, *149*, 69–114. [[CrossRef](#)]
22. Tran, T.H.; Phung, M.D.; Nguyen, T.T.V.; Tran, Q.V. Stabilization control of the differential mobile robot using Lyapunov function and extended Kalman filter. *Vietnam. J. Online* **2012**, 441–452. [[CrossRef](#)]
23. Li, Z.; Sastry, S.S. Lyapunov stability theory. In *Extensions of Linear-Quadratic Control Theory*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 39–59. [[CrossRef](#)]
24. Hespanha, J. Internal or Lyapunov stability. In *Linear Systems Theory*; Princeton University Press: Princeton, NJ, USA, 2018; pp. 87–107. [[CrossRef](#)]
25. Hoang, T.T.; Duong, P.M.; Van, N.T.T.; Vinh, T.Q. Stabilization Control of the Differential Mobile Robot Using Lyapunov Function and Extended Kalman Filter. *arXiv* **2017**, arXiv:1707.05458.
26. Schiller, J.D.; Muntwiler, S.; Köhler, J.; Zeilinger, M.N.; Müller, M.A. A Lyapunov Function for Robust Stability of Moving Horizon Estimation. *IEEE Trans. Autom. Control* **2023**, *68*, 7466–7481. [[CrossRef](#)]
27. Saab, W.; Ben-Tzvi, P. A Genderless Coupling Mechanism With Six-Degrees-of-Freedom Misalignment Capability for Modular Self-Reconfigurable Robots. *J. Mech. Robot.* **2016**, *8*, 061014. [[CrossRef](#)]
28. Rohmer, E.; Singh, S.P.; Freese, M. CoppeliaSim (formerly V-REP): A versatile and scalable robot simulation framework. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1321–1326.
29. Feng, S.; Sebastian, B.; Ben-Tzvi, P. A Collision Avoidance Method Based on Deep Reinforcement Learning. *Robotics* **2021**, *10*, 73. [[CrossRef](#)]
30. d’Andréa Novel, B.; Campion, G.; Bastin, G. Control of Nonholonomic Wheeled Mobile Robots by State Feedback Linearization. *Int. J. Robot. Res.* **1995**, *14*, 543–559. [[CrossRef](#)]

31. Martínez, J.L.; Mandow, A.; Morales, J.; Pedraza, S.; García-Cerezo, A. Approximating kinematics for tracked mobile robots. *Int. J. Robot. Res.* **2005**, *24*, 867–878. [[CrossRef](#)]
32. Mandow, A.; Martínez, J.L.; Morales, J.; Blanco, J.L.; García-Cerezo, A.; González, J. Experimental kinematics for wheeled skid-steer mobile robots. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 1222–1227. [[CrossRef](#)]
33. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3400–3407. [[CrossRef](#)]
34. Zhu, F.; Antsaklis, P.J. Optimal control of hybrid switched systems: A brief survey. *Discret. Event Dyn. Syst. Theory Appl.* **2015**, *25*, 345–364. [[CrossRef](#)]
35. Bock, H.G.; Kirches, C.; Meyer, A.; Potschka, A. Numerical solution of optimal control problems with explicit and implicit switches. *Optim. Methods Softw.* **2018**, *33*, 450–474. [[CrossRef](#)]
36. Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [[CrossRef](#)]
37. Boissonnat, J.D.; Cérézo, A.; Leblond, J. Shortest paths of bounded curvature in the plane. *J. Intell. Robot. Syst.* **1994**, *11*, 5–20. [[CrossRef](#)]

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.