Robotica (2025), **43**, pp. 2121–2142
doi:10.1017/S0263574725000645

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Inter-reconfigurable robot by modified informed sampling-based shortest path planning in cleaning and maintenance

Anh Vu Le[1], Cong Hien Dinh[2], Vinu Sivanantham[3], Prabakaran Veerajagadheswar[3], Do Quang Huy[4], Bui Vu Minh[5], Guangming Chen[6] and Rajesh Elara Mohan[3]

[1]Advanced Intelligent Technology Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[2]Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[3]ROAR Lab, Engineering Product Development, Singapore University of Technology and Design, Singapore
[4]Computational Mechanics, Institute of Mechanics Faculty of Mechanical Engineering, Otto von Guericke University, Germany
[5]Faculty of Engineering and Technology, Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam
[6]College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
Corresponding author. Anh Vu Le; Email: leanhvu@tdtu.edu.vn

Received: 14 July 2024; **Accepted:** 27 March 2025; **First published online:** 19 May 2025

Keywords: inter-reconfigurable robotics; reconfigurable mechanism; cleaning and maintenance; shortest path planning; greedy informed rapidly-exploring random tree

Abstract

Connecting individual robots to form an inter-reconfigurable system with a flexible base size enhances the ability to access and cover areas for cleaning and maintenance tasks. Given that increased configuration complexity expands the search space dimension, an optimal routing solution ensuring efficiency is essential. In this paper, we present an inter-reconfigurable multi-robot system capable of adjusting the bases of its two units, along with an optimal path planning approach for confined spaces based on a modified informed rapidly-exploring random tree algorithm by a greedy set (RIRRT*). We validate the navigation of the proposed inter-reconfigurable platform using RIRRT* for four informed dimensional search spaces as a case study in both simulated and real-world environments. The proposed path planning method for the inter-reconfigurable system outperformed conventional strategies, achieving significant reduction in both execution time and energy utilization.

1. Introduction

Recently, multi-robots cooperation for cleaning and maintenance tasks in both domestic and industrial fields has been the topic of great interest. The world's leading technology firms, especially those providing services in cleaning, maintenance, and logistic applications, have invested heavily in developing their robot systems. The need for the multi-robot operation platform is crucial in enhancing automation aspects in industry 4.0 implementation [1]. Hence, there are a number of researches in domestic as well as industrial usages to develop self-operate multi-robot systems such as AGV's self-propelled pallet trucks [2], logistic robots in industrial production line [3], robots participated in the assembly heavy industrial automobile product lines [4]. Among these applications, robot systems for autonomous cleaning and maintenance tasks have been applied to operate in real environments. As a result, they have proven their efficiency and accuracy in wide and complex environments, gradually reducing operation costs and human labor resources. Regarding robots for area coverage tasks, millions of devices have been sold on the market at all prices [5]. However, these robot versions also have some limitations because of their fixed-shape designs, which limit their ability to access the space needed to cover. Therefore, they need to be upgraded regarding platform design and the advanced software algorithm. Such solutions

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

of finding the optimal plan of trajectories to complete coverage of the complex area while avoiding obstacles and finding the shortest path to access the confined area are highly desirable.

Therefore, it is necessary to have fully self-operated devices to handle these manual workloads. Numerous areas apply the shortest path planning method, such as exploration, security [6], lawn mowing, and floor cleaning [7, 8]. To meet those demands, the shapeshifting robot's shortest path planning and self-coverage algorithms are indispensable. The novel path planning algorithms help a robot get these features: the ability to define ways for smarter coverage with lower energy needed, avoid obstacles in complex environments, ability to be autonomous of shapeshifting, and self-adapting in complex environments.

In complete coverage tasks such as cleaning, maintenance, and inspection, the bigger the robot base, the faster the predefined areas are covered, as well as the less energy is spent. On the other hand, the robot should be able to shift adaptively between the big to the smaller base to access the restricted area. For example, when cleaning and maintenance in confined areas, robots often have to find the shortest path to the locations that need to be cleaned, so robots prefer large base sizes to speed up the time of covering the defined area. As a result of shapeshifting to various widths, robots cover more complex areas and can navigate through narrow spaces to discover confined areas. Therefore, in practical application, the large-footprint robot systems maintain the bigger size and only need to shift their shape to a smaller size when needed, which poses challenges in determining the optimal paths. Furthermore, since the system added more degree of freedom in configuration, dealing with the higher dimensional search space while finding the optimal routing solution for inter-reconfigurable systems is also to be addressed. In this paper, besides proposing the novel inter-reconfigurable system for cleaning and maintenance applications, we modeled the global shortest path planning of the inter-reconfigurable robot for cleaning and maintenance applications as geometric planning in high-dimensional state spaces. We leverage the anytime sampling-based planning algorithms to find asymptotically optimal solutions in obstacle-rich environments with narrow constrained regions. To this end, we enhance the computational relax of the direct informed sampling technique of the IRRT* algorithm [9], which relies more on the heuristics regardless of the current solution path cost. Specifically, the proposed greedy informed sampling approach is demonstrated for inter-reconfigurable multi-robots systems called RIRRT*, which is analytically shown to have a better convergence rate than existing sampling based path planning techniques, especially in higher dimensional state spaces of inter-reconfigurable robots system. Experiments with an in-house built-in inter-reconfigurable robot system in both simulated and real-world scenarios show that our approach is capable of finding better solution paths in four-dimensional search spaces with a faster convergence rate than existing methods. The proposed method is validated for reconfigurable multi-purpose service robots in simulation environments and on real hardware, therefore increasing the feasibility of commercializing the real products.

The main contributions are summarized as follows:

- A novel inter-reconfigurable system, which can change its shape to access narrow spaces and enhance the area coverage, is presented.
- By modeling the global shortest path planning of an inter-reconfigurable system in high dimensional search space using modified IRRT* by greedy set called RIRRT*, the algorithm improves the convergence rate and path's qualities.
- The inter-reconfigurable platform with a path planning approach was validated in both large simulated and real-world environments with narrow constrained areas scenarios.

The remain of the paper is organized as follows: Section 2 presents the related works. Section 3 describes the design of an inter-reconfigurable multi-robots system. Section 4 defines software architecture with inter-reconfigurable mechanism. The demonstration of shortest path planning with a RIRRT* algorithm for inter-reconfigurable robots is described in Section 5. Section 6 demonstrates RIRRT* with an inter-reconfigurable system in simulated and real-world experiments. Finally, section 7 concludes the paper with a discussion and future works.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

2. Related works

In recent years, the use of reconfigurable robots in constrained and obstacle-cluttered environments has gained traction as they provide rich sets of morphologies by reconfiguring their bases and can navigate tight spaces. Reconfigurable robots offer tremendous potential in advancing performance by assuming multiple morphologies concerning their environment. Because of their multifunctionality and adaptability, self-reconfigurable robots have been progressively used in many applications like space and pavement cleaning [10], ship cleaning [11, 12], floor, and stairace cleaning [13].

Reconfigurable robots were introduced in the late 1980s and have been extensively studied. Reconfigurable robots are classified into three major categories; namely, intra-reconfigurable [14,15], inter-reconfigurable [15], and nested-reconfigurable [16] robots. Intra-reconfiguration is where an individual robot can change its morphology from one form to another. Some examples to this end include Scorpio [17], a bio-inspired robot capable of switching between rolling, crawling, and wall climbing forms, and RoboMods, a reconfigurable, under-actuated legged robot that could generate distinct walking patterns [18]. Inter reconfiguration is where multiple similar robots come together to form global morphologies by undergoing assembly and disassembly processes. Some examples include Transbot [19], Panthera [20, 21], hTrihex [22], hTrihex [23], and Stetro [24, 25]. The third category is nested robots, which are capable of performing both inter and intra- reconfigurations. One such example is a hinged tetro robot that can change its morphology on its own and can form global morphology by attaching and detaching with a team of other hinged-tetro robots [26]. These robots have the capacity to change themselves into various shapes, subsequently productively navigating through compelled areas, which may not be workable for regular fixed-shape robots. However, an efficient path planning by navigation framework with collision-free for reconfigurable robots in higher configurable state dimensions is computationally more expensive than conventional fixed-shaped robots.

Robot navigation in complex environments poses research and development challenges. Simultaneous localization and mapping (SLAM) [27] is a significant open problem in mobile robotics. To move precisely, a mobile robot must have an accurate environment map; however, to build an accurate map, the mobile robot sensing locations must be known. In 1990, Smith R., Self M., and Cheeseman P. first proposed using the Kalman filter, extended Kalman filter (EKF) [28] for incrementally estimating the posterior distribution over robot pose along with the positions of the landmarks. Starting from the unknown location of the unknown environment, the robot locates its position and attitude through repeated observation of environmental features in the movement process. It then builds an incremental map of the surrounding environment according to its position to achieve simultaneous positioning and map purposes. Sensor fusions can use an inertial sensor such as IMU to assist 2D SLAM [29] to improve the mapping accuracy by compensating for the drifting of the wheel encoder. Further, it incorporates the time dimension into the mapping process to enable a robot to maintain an accurate map while operating in dynamic environments.

Global path planners in robot navigation determine the possible path to reach the goal position. The choice of a global path-planning algorithm depends on the application of the robot. For example, in the case of area coverage robots, grid-based graph search methods for complete coverage path planning have been demonstrated in ref. [30], where they split the map into grid cells for efficient and faster navigation. Another work proposed by Le et al. [16] demonstrated the idea of adapting polyomino tiling theory with path planning on a Tetris-inspired self-reconfigurable floor cleaning robot. They further validated area coverage on the robot using A-star based zigzag planner to maximize coverage on accessible space areas. Regarding logistics transportation, obstacle avoidance and the shortest path are the most preferred criteria while adopting a path planning technique. One approach for estimating the shortest path uses the roadmap method to randomly generate n number of curves connecting the start and end points and choose the best curve as a path to the goal point. Bhattacharya et al. [31] presented a path planning algorithm based on Voronoi diagram in the presence of polygonal obstacles in the environment. Similarly, Djikstra [32] algorithm also uses the roadmap method and solves the problem by assigning cost value to each region of the grid cell map and estimating the path with minimum cost.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Another approach to real-time planning on estimating the best path includes heuristic search algorithms, most popularly the A-star search algorithm [33] to find optimal paths. The Dhouib-Matrix Shortest Path Problem (DM-SPP) is an efficient algorithm for solving shortest path problems with a time complexity of $O(n + m)$ [34]. It is particularly useful for mobile robot path planning in static grid-based environments. Two variants, DM-SPP-24 and DM-SPP-4, enhance its adaptability. DM-SPP-24 allows movement in 24 directions for greater precision, while DM-SPP-4 restricts movement to four primary directions, reducing computational complexity for real-time applications [35, 36].

Sampling-based planners are categorized into multiple-query and single-query approaches. Multiple-query planners construct a roadmap—a graph of collision-free paths—allowing efficient shortest-path computation for various start-goal pairs. In contrast, single-query planners solve specific motion planning instances. Among the most widely used are probabilistic roadmaps [37] for multiple-query scenarios and rapidly-exploring random trees (RRTs) [38] for single-query tasks. These methods are well-suited for high-dimensional spaces, as they avoid exhaustive obstacle representation required by deterministic grid-based graph-search algorithms like Dijkstra's algorithm [39] or A* [33]. Given sufficient time, they ensure probabilistic completeness, guaranteeing a solution if one exists. However, while they generate feasible (collision-free) paths, they do not inherently optimize them.

Several extensions enhance efficiency by incorporating anytime capabilities, such as anytime motion planning with RRT* [40], which refines solutions incrementally while maintaining probabilistic completeness and achieving asymptotic optimality. To improve solution quality, RRT* [40] iteratively rewires its tree structure for asymptotic optimality. However, its uniform random sampling slows convergence, particularly in high-dimensional problems, requiring longer computation times for near-optimal solutions.

Recent research has focused on enhancing RRT*'s sampling strategies to improve convergence while preserving its theoretical optimal solution guarantees. Techniques such as sample elimination, biased sampling, graph pruning, and direct informed sampling address these limitations [9]. Informed RRT* [9] leverages direct informed sampling by initially operating as RRT* until a solution is found, then restricting sampling to an L^2-informed subset of the state space containing potentially lower-cost paths accelerating convergence by directing exploration toward promising regions. BIT* [41] integrates graph search with sampling-based planning, leveraging heuristics for accelerated convergence. While it offers strong asymptotic optimality and efficiency, its requirement for precomputing the entire planning space limits applicability in dynamic environments or localized tasks.

The efficiency in addressing complex robotic motion planning challenges is a potential research topic. Shahabi et al. [42] conducted a comparative analysis of various sample-based motion planning methods applied to redundant robotic manipulators, highlighting their strengths and limitations in high-dimensional configurations, which provides critical insights for optimizing manipulator performance. Dong et al. [43] proposed a novel modification RRT* path planning algorithm and obstacle-crossing motion planning strategy for cable trench inspection robots, demonstrating the effectiveness of specialized algorithms in constrained environments. Furthermore, Gao et al. [44] explored path planning by sampling-based technology for multi-section continuum robots in confined multiobstacle environments, presenting strategies that enhance maneuverability in complex operational settings. These studies emphasize the growing importance of customized sampling-based methods to meet the diverse demands of modern robotic applications, particularly in challenging and reconfigurable robotic scenarios.

Using the inter-reconfiguration in cleaning and maintaining tasks where multi-robots can connect together, this paper resolves the effective shortest path planning issue by a greedy informed set called RIRRT∗; the proposed shortest path planning enhances the algorithm's convergence rate and the solution path's qualities.

3. Design inter-reconfigurable robots system

In this section, we present a single reconfigurable unit design as in Figure 1 and a multi-robots inter-reconfigurable system for robot shapes with vertical and horizontal configurations as in Figure 2.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

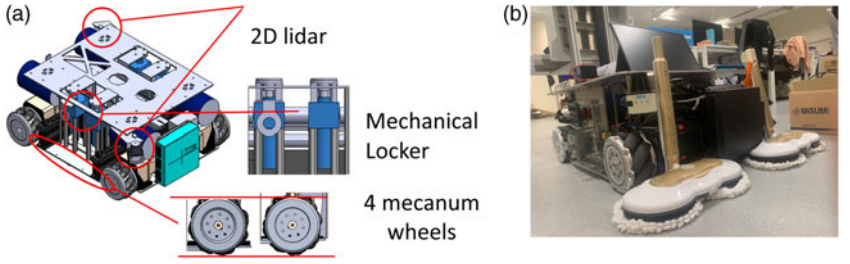


Figure 1. *Design of reconfigurable robot. (a) Robot with the hardware components, (b) robot with cleaning payload.*

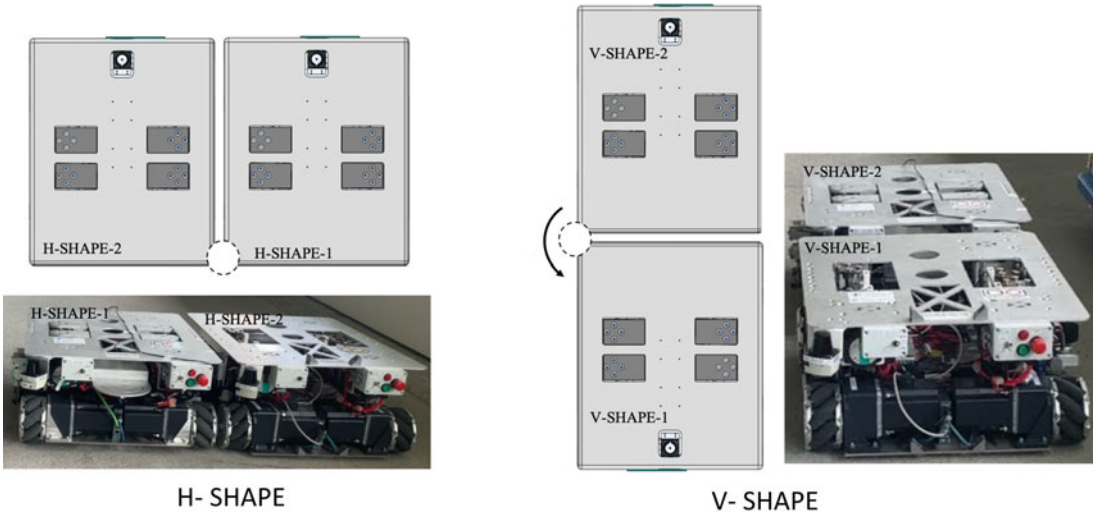


Figure 2. *Robots shapes with vertical and horizontal configuration.*

Depending on the environment's layouts, the inter-reconfigurable multi robots framework enables the physical expansion of the robot base footprint of system to access confined spaces. The multi-robot system is inter-reconfigurability in which the two robots in the framework operate together to fulfill the cleaning and maintenance tasks. As a result, rather than having an independent system for each robot and redundant utilization of similar equipment units on each robot, this multi-robots system requires less hardware, perception sensors, and cleaning payloads. The idea of the inter-reconfigurable system owns the advantages as follows:

- Reduce the complexity of hardware and computation by the master and client scheme in multi-robots cooperation where the task is implemented in the master robot with advanced computational hardware component and the client agent with only manipulators and sensors.
- Efficiency to implement the global shortest path by time and energy criteria to conduct the accessing the confined environment with complex layout setting.

The inter-reconfigurable multi-robot system is designed with two distinct morphologies to enable adaptive locomotion and efficient operation: the H-shape configuration, representing a horizontal form, and the V-shape configuration, representing a vertical form. The V-shape configuration reduces the base footprint by half compared to the H-shape, making it suitable for navigating narrow passages (Figure 2). In this configuration, the two robots are virtually connected to maintain coordination. For most tasks, such as cleaning, the H-shape configuration is used to maximize the cleaning area. In this mode, the two robots are physically connected via passive hinges mounted at their midpoints, as shown in Figure 1a.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Each robot is equipped with a mechanical base featuring additional internal space to house cleaning modules, as depicted in Figure 1b.

The locomotion of each reconfigurable robot is enabled by a holonomic drive system, utilizing a four-wheeled mecanum structure. The free-moving rollers on the mecanum wheels are positioned at a 45-degree angle along the wheel edges, providing omnidirectional movement. Thick, hard-rubber mecanum wheels are employed at the four corners of each robot to minimize slipping on surfaces. This mechanism allows the robots to move freely in the X_r and Y_r planes and rotate in θ_r orientation without requiring a center of rotation radius.

Holonomic locomotion proves especially advantageous when the two robots are virtually connected or performing shape-shifting maneuvers. This feature allows the robots to move longitudinally along their edges, turn in place, and shift positions without altering their heading orientation. During the transition between configurations, the static rotation of the robots is achieved through holonomic kinematics, with the center of instantaneous rotation set at the hinge. This facilitates seamless shape changes between the horizontal H-shape and vertical V-shape configurations, guided by the path-planning algorithm. The holonomic drive system also enhances the robot's ability to access confined spaces and reduces path-tracking errors while following global path planning. These capabilities ensure that the robots can adapt effectively to various operational scenarios, from cleaning open areas in the H-shape configuration to navigating tight spaces in the V-shape configuration.

Specifically, the Figure 3 depicts the wheels setup and robot locomotion where the configuration setup V_{fl}, V_{fr}, V_{br}, V_{bl} determine the corresponding speed of wheel. Parameters a and b are the separations from the robot's middle in the comparing x and $y-axis$ of the robot outline. Following the model in Taheri et al., [45] forward kinematic conditions are formulated as (1–4), and reverse kinematic conditions of the robot can be composed as (5–7):

$$\omega_{fl} = \frac{1}{r}(vel_y - vel_x + (a+b)\omega_\theta) \tag{1}$$

$$\omega_{fr} = \frac{1}{r}(-vel_y + vel_x + (a+b)\omega_\theta) \tag{2}$$

$$\omega_{br} = \frac{1}{r}(vel_y + vel_x + (a+b)\omega_\theta) \tag{3}$$

$$\omega_{bl} = \frac{1}{r}(-vel_y - vel_x + (a+b)\omega_\theta) \tag{4}$$

where $\omega_{fl}, \omega_{fr}, \omega_{br}, \omega_{bl}$ represents the corresponding angular velocities of each wheel calculated based on the command velocities on x, y, and orientation θ in the workspace $vel_y, vel_x, \omega_\theta$ received by the robot and r is the radius of each wheel.

$$V_x = 0.25 * (V_{fl} - V_{fr} + V_{br} - V_{bl}) \tag{5}$$

$$V_y = 0.25 * (-V_{fl} + V_{fr} + V_{br} - V_{bl}) \tag{6}$$

$$V_\theta = 0.25/(a+b) * (V_{fl} + V_{fr} + V_{br} + V_{bl}) \tag{7}$$

After calculating the velocity of the robot based on inverse kinematics equations, Eqs. (8), (9), (10) ate used to estimate the position of the robot relative to the robot environment where δ_t the represents the time step or sampling interval.

$$\dot{x} = (V_x * \cos(-\dot{\theta}) - V_y * \sin(-\dot{\theta})) * \delta_t \tag{8}$$

$$\dot{y} = (V_x * \sin(-\dot{\theta}) - V_y * \cos(-\dot{\theta})) * \delta_t \tag{9}$$

$$\dot{\theta} = V_\theta * \delta_t \tag{10}$$

To facilitate the transmission of control data to the robot hardware, the robot operating system (ROS) [46] serves as the foundational middleware framework. The robot's estimated position and velocity are published as wheel odometry messages on dedicated ROS topics, ensuring efficient communication and data synchronization.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

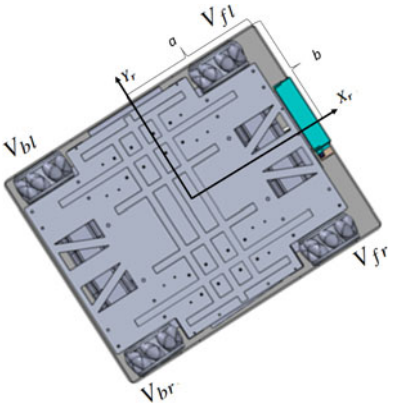


Figure 3. *Reconfigurable robot: kinematic motion model.*

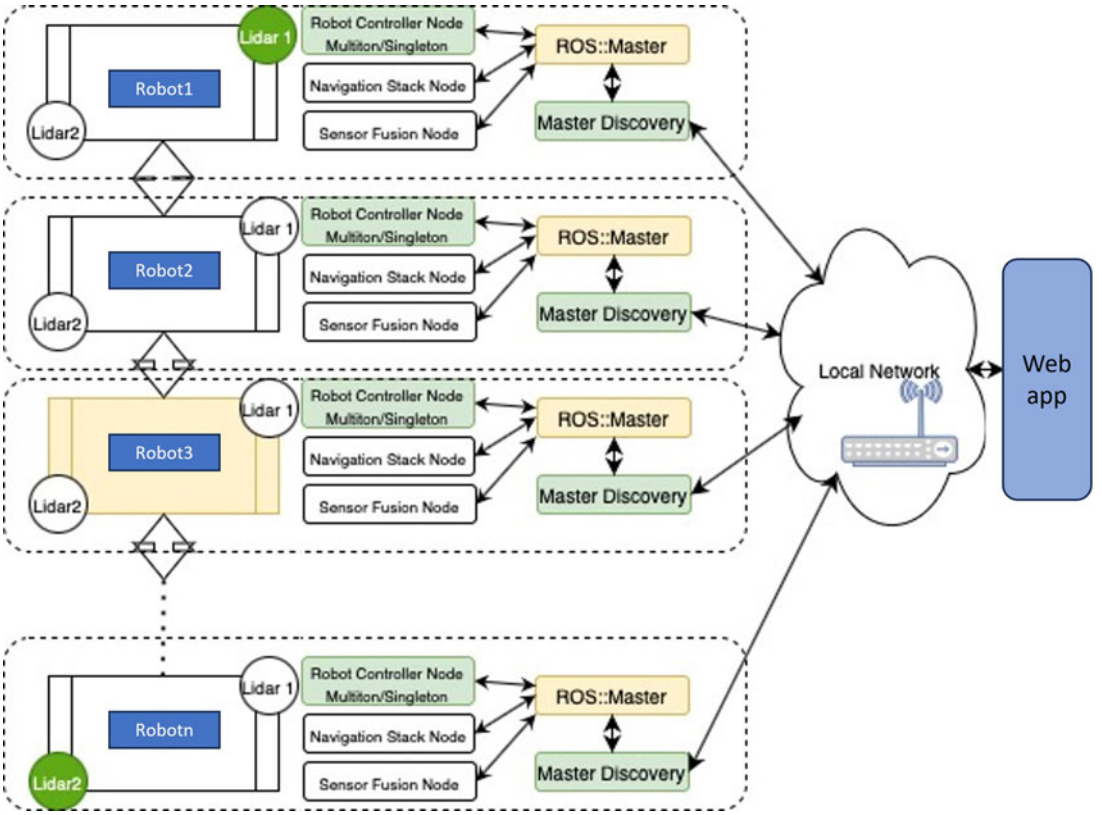


Figure 4. *Software architecture of inter-reconfiguration configuration.*

4. Inter-reconfiguration framework

The software framework of the inter-reconfigurable multi-robots system is displayed in Figure 4. The installed Nvidia Jetson Xavier PC is 8 dedicated core CPUs and 8 GB RAM to fulfill the path planning computational load and handle the streaming sensors. The environment perception and localization sensors comprise industrial IMU sensor, wheel odom encoders, and 2D Lidar. In multi-robot operation, robot localization is essential for stable navigation and robot interactions within the known prebuilt map and known other agents' locations. The sensor fusion technique uses an EKF method incorporating the

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

information of IMU, wheel encoders, and SLAM to get the accurate robot location. During implementation, the robot realizes itself and another robot in the system to define the path from the source to the goal positions. A local planner is created to avoid obstacles while tracking the global path. In the next section, we will describe the fundamental inter-reconfigurable framework that enables our multi-robots system's inter-operation.

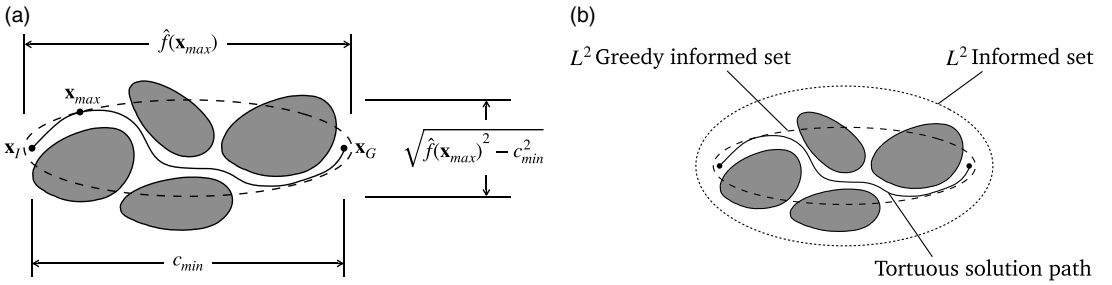
For the inter-reconfiguration state of the robot, a centralized local network configuration is used to establish communication between each robot. As mentioned in the robot system architecture, the software system is developed on ROS framework, which hosts ROS master in a specific local IP address of the robot. ROS master enables the communication between multiple nodes by topics and services within the robot system. In reconfigurable system, we mainly use a robot controller node to handle the robot's overall control. A sensor fusion node is used to fuse and compute data from 2D lidars used. Moreover, a navigation stack node is used to perform autonomous capability on the robot based on the sensor feedback computed from the robot controller node and sensor fusion node. During the inter-reconfiguration of the robot, multiple ROS masters are multicasted over a locally connected network using a master discovery node. Master discovery node remotely connects ROS masters of 'n' number of robots and registers topics and services to all locally connected robots. Figure 4 describes the inter-reconfiguration framework for 'n' number of interconnected robots. The proposed framework chooses any one robot based on the battery percentage to handle the autonomous tasks given to the robot. For example, If system 3 is chosen to handle the autonomous tasks in Inter reconfiguration mode, as highlighted in yellow in Figure 4 where system 3 handles the navigation stack node and sensor fusion node. The sensor fusion node merges the Lidar sensors highlighted in green color to achieve the maximum field of view for autonomous navigation tasks. The robot controller node in all the interconnected robots works in Inter-reconfiguration mode where the velocity commands generated from the navigation stack node in system 3 will be recomputed to velocity commands for individual wheels of all the interconnected robots. This approach minimizes the computational complexity of each robot with an increasing number of interconnected robots.

5. Global path planning for inter-reconfigurable robot

Since the inter-reconfigurable system can enlarge the footprint by double the size of an individual robot, in theory, it can cover the empty space at double speed. However, to enhance the area coverage of the inter-reconfigurable multi-robots system, the shortest path planning in a confined environment is essential to improving access to the confined hard-to-access area; as a result, an increase in the percentage of area cleaned. The shortest path planning methods for single platform [47, 48], appear differently in relation to the proper cleaning and maintenance proposed on the inter-reconfigurable multi-robots cleaning and maintenance frameworks. In this section, by modeling the global path planning of the flexible generic self-reconfigurable robots as geometric planning in high dimensional state spaces, we leverage the anytime sampling-based planning algorithms to find asymptotically optimal solutions in obstacle-rich environments with narrow constrained regions. We define the geometric path planning problem for general self-reconfigurable robots with n-dimensional joints similar to [40]. List of acronyms for shortest path planning are mentioned in the Appendix A.

In the general inter-reconfigurable multi-robot system, let $\mathcal{W} = \mathbb{R}^n$ be the n-dimensional environment consisting of obstacles, $\mathcal{O} \subset \mathcal{W}$. Let $\mathcal{X}$ be the state space of a self-reconfigurable robot with n-dimensional joints and $\mathbf{x} \in \mathcal{X}$ denotes the state of a robot in which $\mathbf{x} = (x, y, z, roll, pitch, yaw, j_0, j_1, \ldots, j_{n-1})$, where $(x, y, z, roll, pitch, yaw)$ denotes the 6 variants *Special Euclidean group in SE(3) 3D space* representation and $(j_0, j_1, \ldots, j_{n-1})$ are the positions of n joints in $\mathbb{R}^n$ space. Let $\mathcal{X}\text{obs} \subseteq \mathcal{X}$ denote the set of all states, $\mathbf{x}$, that are obstructed due to reconfiguration, $\mathcal{O}$, and let $\mathcal{X}\text{free} = \mathcal{X} \setminus \mathcal{X}\text{obs}$ represent the corresponding set of states free from collisions. Define $\mathbf{x}I \in \mathcal{X}\text{free}$ as the starting state and $\mathbf{x}G \in \mathcal{X}\text{free}$ as the target state. Let $\pi{:}[0, 1] \rightarrow \mathcal{X}\text{free}$ be a continuous and valid path. A safe path on a map refers to a collision-free route that a robot can follow while avoiding obstacles and considering environmental constraints. It ensures that the robot does not enter restricted areas, maintains sufficient clearance

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press



L^2 greedy informed set L^2 greedy informed set versus L^2 informed set

Figure 5. *Illustration of the L^2 greedy informed set and its advancement with safe path.*

from obstacles, and adheres to constraints such as narrow passage navigation and energy saving. The objective of the optimal path planning problem is to identify an admissible path π^*, as described in (11), connecting $\mathbf{x}_I$ to $\mathbf{x}G$ through collision-free space, while minimizing the path cost function, $c: \Sigma \rightarrow \mathbb{R} \geq 0$.

$$\pi^* = \underset{\pi \in \Sigma}{\arg\min} \{c(\pi) \mid \pi(0) = \mathbf{x}_I,\ \pi(1) = \mathbf{x}_G, \forall s \in [0, 1],\ \pi(s) \in \mathcal{X}_{\text{free}}\}, \tag{11}$$

where Σ denotes the set of all non-trivial paths and $\mathbb{R}_{\geq 0}$ is the set of nonnegative real numbers.

We validate the optimal global path planning algorithm with a proposed inter-reconfigurable multi-robot system as the 2D shortest path planning study case. Note that each platform has four independent steering units designed for omnidirectional locomotion, and two system body frames can also be expanded to H- shape and compressed to V- shape by independently controlling the additional passive physical connected hinge to traverse through tight spaces. Hence, the state of the multi-robots system in the world frame can then be represented as 4-dimensional space, $\{x, y, \theta, j_0\}$, where $\{x, y, \theta\}$ denotes the location state space and $\{j_0 \in (0, \pi)\}$ are the controllable join of reconfigurable robots. The angles for each steering unit of the robot are taken into account during the planning process, as the focus is on geometric path planning within a 2D environment. The paths generated by the proposed global planner comprise waypoints that represent the state space of the multi-robot system, while an external local trajectory controller ensures the execution of these waypoints following.

5.1. The RIRRT for reconfigurable system*

Among the sampling-based path planning, IRRT * as recommended in ref. [9] tends to speed up the RRT and RRT* convergence *'s rate by focusing the search space using heuristics based on a cost function. Since the calculation's union rate corresponds to the size of the hyperellipsoid, we improved IRRT* more by contracting the size of its n-dimensional hyperellipsoid. Rather than utilizing the current arrangement cost as the hyperellipsoid's cross-over width, this paper suggests utilizing the current arrangement's greatest heuristic expense as the cross-over measurement, bringing down the hyperellipsoid's size and upgrading the assembly rate.

Building upon IRRT*, the RIRRT* algorithm (Algorithm 1) incrementally grows a tree, $G = (V, E)$, where V represents a set of vertices and E denotes edges in the collision-free space. This process aims to improve the solution path iteratively. Similar to IRRT*, RIRRT* rewires the graph with newly added vertices to reduce the cost-to-come of nearby vertices, optimizing the overall solution (Algorithm 1, Lines 21–28).

Once an initial solution is available, the algorithm biases random sampling toward a reduced subset of the hyperellipsoid called the *greedy informed set* (Algorithm 1, Lines 18, 16). This subset includes states capable of producing improved solutions relative to the maximum heuristic cost from the current solution path (Algorithm 1, Lines 7–11). The greedy informed set focuses on states with a cost-to-come $g(\mathbf{x})$, cost-to-go $h(\mathbf{x})$, and total cost $f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$ balancing exploitation and exploration through

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

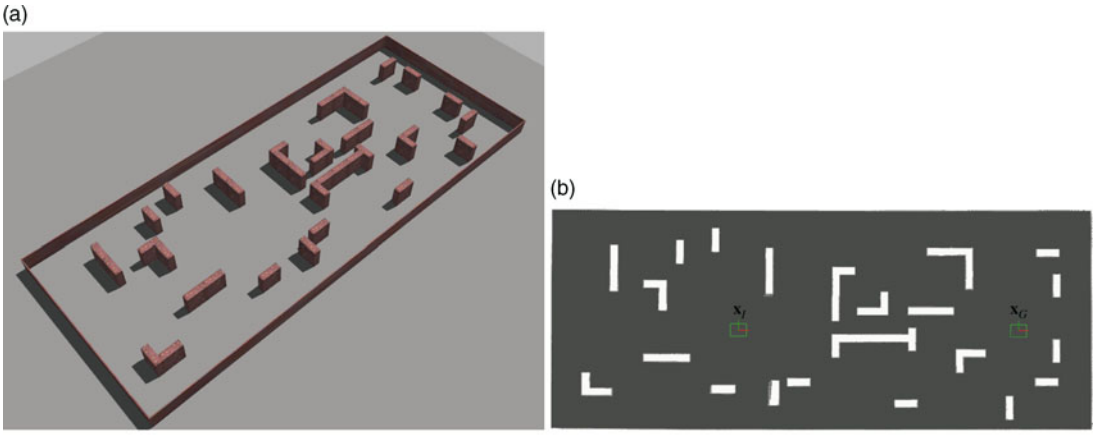


Figure 6. *Illustrations of the large workspace set up for the simulation experiments.*

the *greedy biasing ratio*, ϵ (Algorithm 1, Line 9). For $\epsilon = 1$, sampling is confined to the greedy informed set; for $\epsilon = 0$, uniform sampling is applied. Adjusting ϵ optimizes exploration and exploitation, leading to faster convergence.

To navigate narrow passages, RIRRT* incorporates a dynamic mechanism for adjusting the robot's base width. Initially, edges are validated using the maximum base width; if unsuccessful, the width is reduced incrementally until a valid configuration is found or the edge is deemed invalid. This approach enables adaptability while preserving the robot's larger default size when feasible. At each iteration, the algorithm selects the goal vertex $\mathbf{x}_i$ with the lowest cost-to-come $g(\mathbf{x})$ (Algorithm 1, Line 5) and determines the maximum heuristic cost from the current solution path ($c_{\max}$) for sampling (Algorithm 1, Lines 6–11). If no solution path exists, sampling spans the entire state space (Algorithm 1, Line 6-11). RIRRT* preserves the rewiring radius r_{rewire} from IRRT* (Algorithm 1, Line 23), ensuring efficient tree refinement. Additionally, vertices within the goal region are incorporated into the solution set X_{soln} when their cost-to-come is minimal (Algorithm 1, Lines 29–30).

The algorithm ensures probabilistic completeness and asymptotic optimality (Appendices B, C) by adhering to the core principles of RRT* and IRRT*. RIRRT* guarantees probabilistic completeness (Appendix B) by ensuring each sample has a nonzero selection probability, progressively covering the feasible space. Meanwhile, its adaptive rewiring mechanism (Algorithm 1, Lines 4–14) maintains asymptotic optimality by iteratively refining solution quality as iterations increase.

Specifically, to further enhance convergence, RIRRT* integrates cost-to-come, maximum heuristic cost, and greedy informed sampling with parameter ϵ, balancing global exploration and local exploitation. An adaptive cost-to-go heuristic refines sampling weights, guiding the planner toward lower-cost solutions more effectively. As demonstrated in Appendix, the algorithm improves convergence by reducing the effective problem dimensionality through greedy informed sampling, increasing the likelihood of selecting optimal path states. Theoretical analysis, including the worst-case sample count impact theorem, quantifies the trade-off between focused sampling and probabilistic optimality, demonstrating efficiency gains without compromising solution quality. These refinements strengthen the exploration-exploitation balance, enhancing solution quality while reducing computational overhead, particularly in high-dimensional and constrained planning of inter-reconfigurable robots.

□

6. Experimental results and discussion

The RIRRT* algorithm for inter-reconfigurable multi-robot system was evaluated in both simulation environment (Section 6.2) and real-world experiments (Section 6.3). Simulations were run in Ubuntu

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Algorithm 1. RIRRT* for reconfigurable robot $(\mathbf{x}_I, \mathbf{x}_G, \mathcal{X}_{\text{free}}, \epsilon)$

$V \leftarrow \{\mathbf{x}_I\}; E \leftarrow \emptyset; G = (V, E)$;
$c_i \leftarrow \infty$;
$r_{\text{rewire}} \leftarrow$ Rewiring radius;
BaseWidth $\leftarrow$ Maximum configuration;
$X_{\text{soln}} \leftarrow \emptyset$;
for $i = 1, \ldots, n$ **do**
  if $|X_{soln}| > 0$ **then**
    $\mathbf{x}_i \leftarrow \underset{\mathbf{x} \in X_{\text{soln}}}{\text{argmin}}\{g(\mathbf{x})\}$;
    if $\epsilon > U([0, 1])$ **then**
      $\mathbf{x}_{\text{best}} \leftarrow \mathbf{x}_i; c_{\max} \leftarrow -\infty$;
      while *Parent*$(\mathbf{x}_{best}) \neq$ *null* **do**
        $c_{\text{best}} \leftarrow \hat{f}(\mathbf{x}_{\text{best}})$;
        if $c_{best} > c_{max}$ **then**
          $c_{\max} \leftarrow c_{\text{best}}$;
        $\mathbf{x}_{\text{best}} \leftarrow \text{Parent}(\mathbf{x}_{\text{best}})$;
      $c_i \leftarrow c_{\max}$;
    else
      $c_i \leftarrow g(\mathbf{x}_i)$;
  $\mathbf{x}_{\text{rand}} \leftarrow \text{Sample}(\mathbf{x}_I, \mathbf{x}_G, c_i)$;
  $\mathbf{x}_{\text{nearest}} \leftarrow \text{Nearest}(G, \mathbf{x}_{\text{rand}})$;
  $\mathbf{x}_{\text{new}} \leftarrow \text{Steer}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}})$;
  while *ObstacleFree* $(\mathbf{x}_{nearest}, \mathbf{x}_{new})$ == *False and BaseWidth > Minimum configuration* **do**
    Adjust BaseWidth to a smaller configuration;
    $\mathbf{x}_{\text{new}} \leftarrow \text{Steer}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}})$;
  if *ObstacleFree* $(\mathbf{x}_{nearest}, \mathbf{x}_{new})$ **then**
    $V \leftarrow V \cup \{\mathbf{x}_{\text{new}}\}$;
    $X_{\text{near}} \leftarrow \text{Near}(G, \mathbf{x}_{\text{new}}, r_{\text{rewire}})$;
    $\mathbf{x}_{\min} \leftarrow \mathbf{x}_{\text{nearest}}$;
    foreach $\mathbf{x}_{near} \in X_{near}$ **do**
      if *Cost*$(\mathbf{x}_{near})$ < *Cost*$(\mathbf{x}_{min})$ **then**
        $\mathbf{x}_{\min} \leftarrow \mathbf{x}_{\text{near}}$;
    $E \leftarrow E \cup \{(\mathbf{x}_{\min}, \mathbf{x}_{\text{new}})\}$;
    foreach $\mathbf{x}_{near} \in X_{near}$ **do**
      if $g(\mathbf{x}_{new}) + d(\mathbf{x}_{new}, \mathbf{x}_{near}) < g(\mathbf{x}_{near})$ **then**
        if *ObstacleFree* $(\mathbf{x}_{new}, \mathbf{x}_{near})$ **then**
          $E \leftarrow E \setminus \{(\text{Parent}(\mathbf{x}_{\text{near}}), \mathbf{x}_{\text{near}})\}$;
          $E \leftarrow E \cup \{(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{near}})\}$;
    if *InsideGoalRadius*$(\mathbf{x}_{new}, r_{goal})$ **then**
      $X_{\text{soln}} \leftarrow X_{\text{soln}} \cup \{\mathbf{x}_{\text{new}}\}$;
return $G = (V, E)$;

20.04 on a workstation with 16 GB of RAM and Ryzen 5 3600 6-core processor. All the experiments use the same hyperparameters and are tuned in faith to perform better. The maximum edge length of the tree is set to 2.5 for all the planners, and IRRT∗-like planners share the same rewiring factor constant of 4.5. Simulation experiments were run 50 trials with different pseudorandom seeds.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

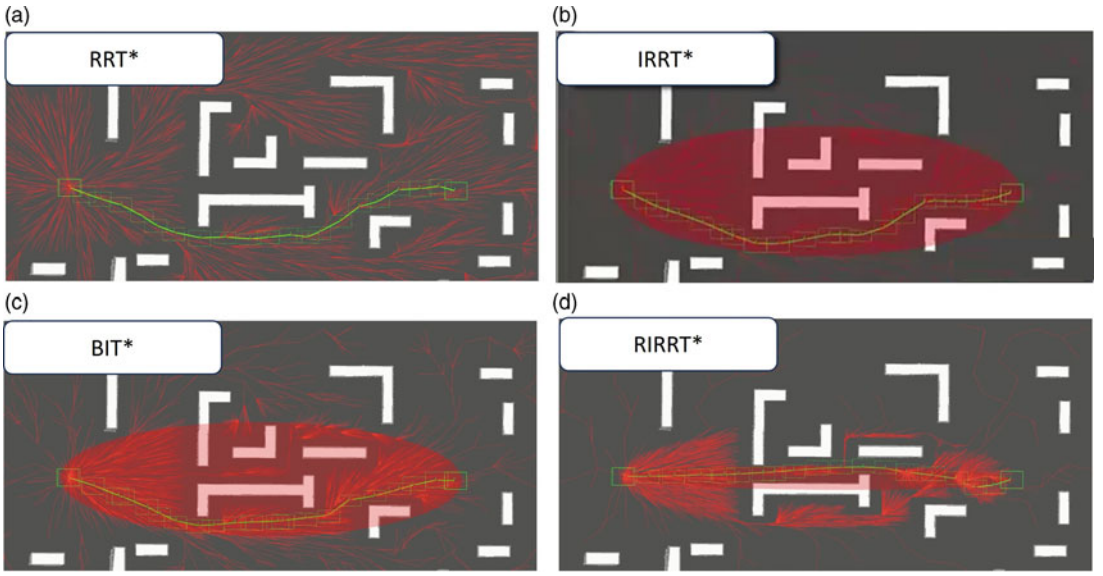


Figure 7. *An example of solution paths found by the planners after* 30 *seconds of planning time. (a) RRT* with path length, (b) Inform RRT* with path length 2.38, (c) BIT* with path length 2.31, (d) RIRRT* with a path length of 2.18.*

6.1. Large simulation environment with shortcuts

We demonstrate the ability of RIRRT* to find optimal solutions with a faster convergence rate in a large simulation environment with various obstacles, depicted in Figure 6(a). The environment is 29 m wide and 70.5 m long, and consists of several obstacles and narrow passage shortcuts for the reconfigurable robot to traverse through it. Since reconfigurable robot navigates only on a planar surface, a 2-dimensional occupancy grid map [Figure 6(b)] is used during planning by projecting a map of the simulation environment onto 2D plane based on robot's height. The $\mathbf{x}_I$ and $\mathbf{x}_G$ in Figure 6(b) represent the initial and final states of the reconfigurable robot footprint, and the planner has to find a solution path from $\mathbf{x}_I$ to $\mathbf{x}_G$, where both states are in the fully expanded footprints. This task is challenging as several solutions exist, and only one unique shortcut to the goal is available, which can only be achieved traversing through by reconfiguring each reconfigurable robot's gait. Therefore, we investigate the performance of the tested planners in a simulation environment where the cost objective minimizes the distance through a four-dimensional search space. Note that the distance cost function is the summation of the Euclidean distance for $\mathbb{R}^n$ state spaces and angular distance for SO(n) state spaces. All planners RRT*, IRRT*, BIT*, and proposed RIRRT* were given 300 s of planning time, and the median solution costs were calculated from the total 50 trials by recording the solution cost of each planner at every 100 ms from a separate thread and interpolating each trial. If no solution has been discovered during the recording process, the solution cost is assigned a value of infinity to facilitate the calculation of the median. In order to accelerate finding a narrow passage shortcut, we intentionally set a fixed value to the SO(2) component of 4-dimensional state space for half the planning time and used uniform random sampling for the rest.

An example of the solution paths obtained by the planners in this simulation environment is illustrated in Figure 7, along with the reconfigurable robot's footprint. The edges generated by the planners are highlighted in red, while the current best solution paths, including the reconfigurable robot's footprint, are depicted in green. The red ellipsoidal regions represent the informed sets of each planner. Notably, the proposed RIRRT* efficiently identifies a shortcut path through the narrow passage and commands the system to transition from an H-Shape to a V-Shape to navigate through the constrained space. In contrast, other planners have not yet discovered this shortcut within the given planning time.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

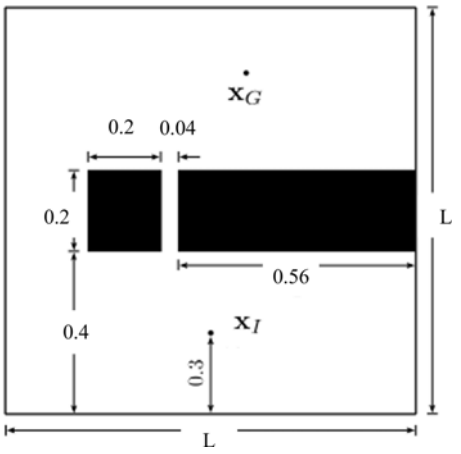


Figure 8. *Simulation workspace includes complex planning problems containing narrow passage gap with $\boldsymbol{x}_I$ and $\boldsymbol{x}_G$ represent the start and goal states.*

When reconfiguration is enabled [Figure 7(d)], RIRRT* consistently finds the lowest-cost path through narrow passages, demonstrating the advantage of dynamic reconfigurability. Meanwhile, IRRT* and BIT* require significantly more planning time to explore these constrained areas effectively. Despite allowing reconfiguration, IRRT* and BIT* struggle to find paths through the narrow passage within the allocated planning time, highlighting their slower convergence in highly constrained environments. As a result, RIRRT* outperforms these approaches by quickly identifying viable paths and optimizing traversal through confined spaces. Overall, the results in Figure 7 confirm that RIRRT* not only discovers better-quality solution paths but also converges faster toward an optimal solution compared to existing state-of-the-art planners. Its ability to dynamically adjust the robot's configuration enhances both efficiency and adaptability in complex environments.

6.2. Planning on narrow passage problems

The planners were also tested on simulated planning problems in $\mathbb{R}^2$, $\mathbb{R}^4$, and $\mathbb{R}^8$, as illustrated in Figure 8. The test environment includes two distinct homotopy classes, featuring a narrow passage with a width of 0.04. The initial and goal states are positioned at $[-0.3, 0.0, \ldots, 0.0]^T$ and $[0.3, 0.0, \ldots, 0.0]^T$, respectively. To ensure a comprehensive evaluation, the allocated planning time varied according to the problem complexity and the dimensionality of the state space. Each planner was executed for 50 trials with different pseudorandom seeds. The solution cost was recorded every 10^{-4} seconds via a separate thread. If a solution had not yet been found at the time of recording, its cost was set to infinity for median computation purposes.

Table I quantifies the a statistical comparison of IRRT*, BIT*, and RIRRT* performance with and without reconfiguration scenarios in terms of path length and energy cost at different iteration steps (2000, 5000, and 10,000). To ensure statistical reliability, the results include standard deviations (SD) and 95% confidence intervals (CIs), providing insights into the consistency and robustness of each planner.

As these results, highlighted in Table I, RIRRT* demonstrates a significant advantage in reconfiguration scenarios, efficiently adapting to narrow passage constraints. While BIT* improves upon IRRT* in terms of path length and energy cost, RIRRT* consistently outperforms both across all iteration levels. At 10,000 iterations with reconfiguration, RIRRT* achieves the shortest path length (2.18) and lowest energy cost (4.30), surpassing BIT* (2.31, 4.50) and IRRT* (2.38, 4.55). At higher iteration counts, RIRRT* significantly reduces energy cost, demonstrating a favorable trade-off between computational effort and solution quality. Even without reconfiguration, where performance differences are less obvious, RIRRT* maintains its lead, achieving the shortest path (2.28) and lowest energy cost (5.43) at 10,000

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Table I. *Average comparison results from 50 trials of planners in terms of with (w) configuration and without (wo) reconfiguration on a narrow passage gap of large simulation environment problem, including standard deviations (SD) and 95% confidence intervals (CI).*

		w		wo	
Planners Cost	**Iterations**	**Path Length**	**Energy Cost**	**Path Length**	**Energy**
IRRT*	2000	2.50 (0.12)	4.80 (0.15)	2.69 (0.10)	5.97 (0.12)
	5000	2.43 (0.10)	4.65 (0.13)	2.66 (0.09)	5.89 (0.11)
	10,000	2.38 (0.09)	4.55 (0.10)	2.64 (0.08)	5.78 (0.09)
BIT*	2000	2.40 (0.13)	4.72 (0.12)	2.53 (0.11)	5.80 (0.13)
	5000	2.35 (0.11)	4.60 (0.10)	2.47 (0.09)	5.73 (0.10)
	10,000	2.31 (0.09)	4.50 (0.08)	2.42 (0.08)	5.62 (0.08)
RIRRT*	2000	2.30 (0.11)	4.50 (0.14)	2.41 (0.10)	5.67 (0.13)
	5000	2.25 (0.10)	4.40 (0.11)	2.35 (0.09)	5.55 (0.11)
	10,000	2.18 (0.08)	4.30 (0.09)	2.28 (0.07)	5.43 (0.06)

iterations. This underscores its ability to generate efficient solutions across diverse planning environments, reinforcing its adaptability and effectiveness. Additionally, RIRRT* exhibits the lowest standard deviation, reinforcing its stability and reliability across multiple trials. For path length, RIRRT* has the lowest SD = 0.07 (wo reconfiguration) and SD = 0.08 (w reconfiguration). For energy cost, RIRRT* also has the lowest SD = 0.06 (wo) and SD = 0.09 (w).

Figure 9 illustrates the success rate (top row of each subfigure) and solution cost (bottom row of each subfigure) as a function of computation time for three path planning algorithms—IRRT* (red), BIT* (purple), and RIRRT* (green)—in different dimensional spaces: $\mathbb{R}^2$, $\mathbb{R}^4$, and $\mathbb{R}^8$. The shaded regions in the plots represent the confidence intervals and standard deviations, providing a clear measure of variability in success rate and cost. Additionally, significance tests are performed to validate the statistical differences between the planners. The success rate consistently increases over time across all dimensions, with RIRRT* achieving faster convergence compared to the other planners. For example, in $\mathbb{R}^2$, the success rate of RIRRT* is approximately 9.12% higher within the first 10^{-2} seconds compared to IRRT*. In $\mathbb{R}^4$, RIRRT* achieves a 15.26% improvement in success rate over BIT* within the same time frame. In higher-dimensional space $\mathbb{R}^8$, RIRRT* outperforms both planners with a gain of nearly 22.68% in success rate at early computation times.

In terms of solution cost, RIRRT* consistently produces better results. In $\mathbb{R}^2$, the solution cost is reduced by approximately 10.56% compared to IRRT*. Similarly, in $\mathbb{R}^4$ and $\mathbb{R}^8$, the cost achieved by RIRRT* shows improvements of 16.15% and 19.12%, respectively. The proposed RIRRT* achieved better performance with gains of 9.8% in path length and 8.5% in energy cost in the without reconfiguration case and gains of 17.79% in path length and 15.26% in energy cost in the reconfiguration case. Overall, RIRRT* demonstrates superior performance in both success rate and solution cost, particularly in higher dimensions, with significant gains of up to 25% over competing planners.

6.3. Planning of reconfigurable robot platforms in real-world environment

To evaluate the efficiency of saving the energy and time spent by the proposed inter-reconfigurable system with global path generated by RIRRT* in the real-world environments, we integrated the RIRRT* to ROS navigation package as the global path planning plugin and compare its efficiency with other existing planning plugins. We set the predefined source and destination in real-world environments for both the reconfiguration four-dimensional state space (consisted of 2d position: x, y, orientation and hinge angle) and without reconfigurable three-dimensional state space (consisted of 2d position: x, y, *orientation*). Note that we select the solutions of planners after 180 s planning time to conduct the navigation

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

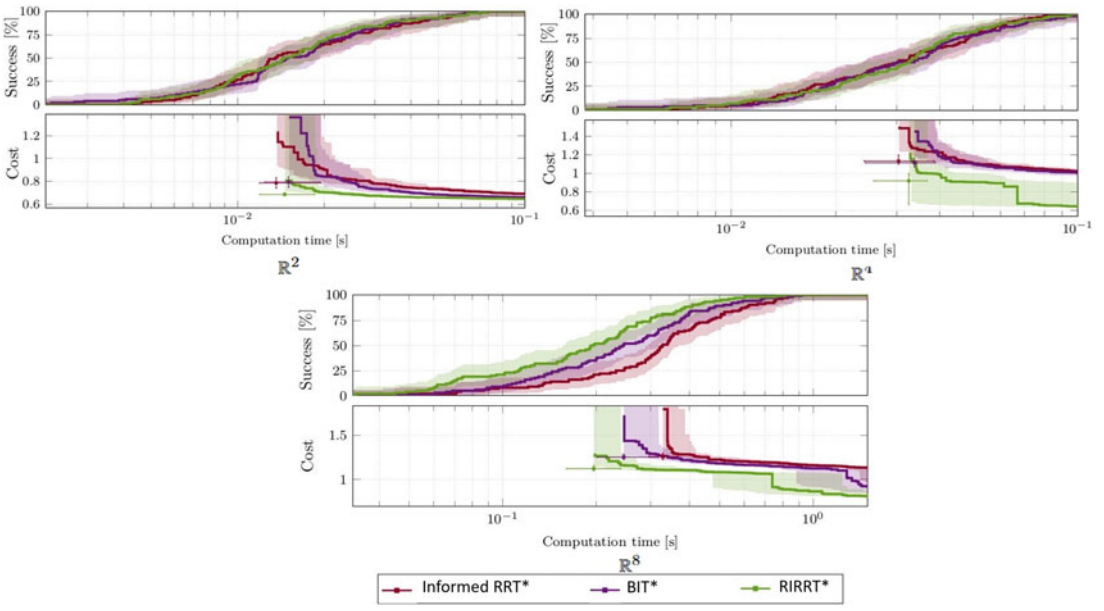


Figure 9. *Results from the* 50 *different trials of the planners.*

experiment. Meanwhile, A* and D* as the default global path planner integrated into ROS navigation stack [46]. The workspace scenario is mapped using SLAM [49] with 2D laser sensor. Once the global paths are generated for the built map to connect these waypoints, the inter-reconfigurable multi-robots system follows the path tracking from the velocity commands issued by a local trajectory controller to visit the waypoints orderly. The spent time and the consumed energy were recorded from reconfigurable robot's battery management system after each navigation. The reported runtime includes both the path planning time and the time taken by the robot to complete navigation along the planned path. This comprehensive measurement ensures a realistic evaluation of the system's performance in real-world conditions. The runtime is influenced by factors such as the complexity of the environment, the need for reconfiguration in narrow passages, and the dynamic execution of path adjustments during navigation.

In the first scenario as shown in Figure 10, RIIRT with (w) and wo (without) reconfigurable is compared against the conventional global path planning methods A*, D*, IRRT*, and BIT*. The workspace consists of possible options (Paths 1 and 2) with different path lengths to connect the defined source to destination waypoints. The example path of each tested method with the platform actions while navigating through the narrow space is shown in Figure 11.

Based on the software architecture described in Section 4.1, the transformation between the V-shape and H-shape configurations is executed seamlessly while navigating narrow passages, with minimal delay (typically less than 100 ms). In each iteration, the transformation process remains consistent, ensuring that the robot's positional identity is maintained. Furthermore, the transformation does not introduce any positional offset between the two reconfigurable robot platforms, preserving spatial alignment for effective inter-reconfiguration.

To validate the effectiveness of RIRRT* with other advanced path planners in more complex environments, the second scenario featuring a multi-room layout with obstacles and varied terrain is presented in Figure 12. The test map spans approximately 90 m by 80 m. In this experiment, the robot is equipped with cleaning modules to perform maintenance tasks. These applications demand a comprehensive path planning algorithm to ensure that the robot moves from one location to another location separated by multiple narrow constrained spaces to thorough cleaning in areas requiring attention.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

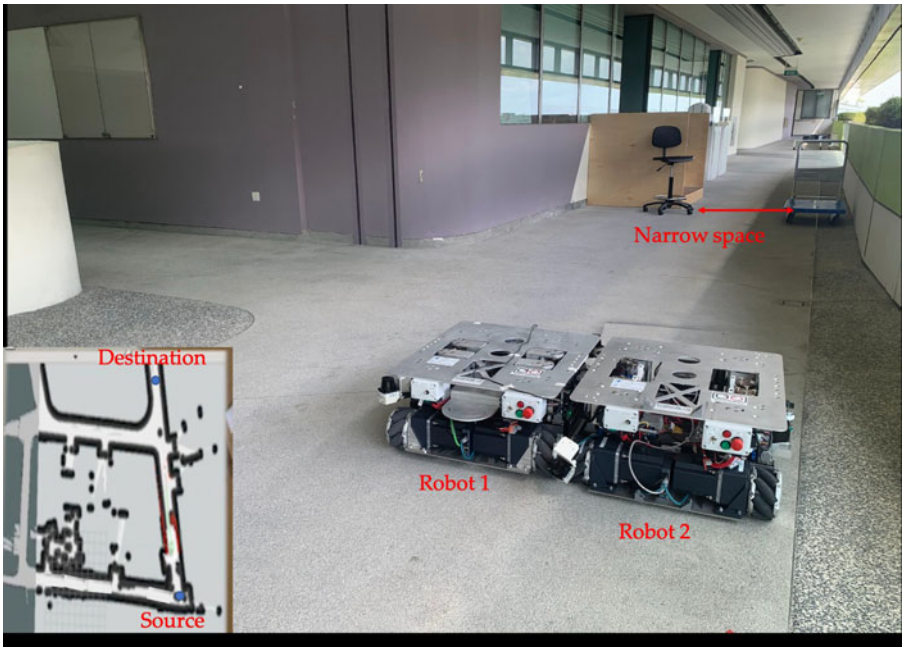


Figure 10. *The workspace setup of the real environment with narrow space.*

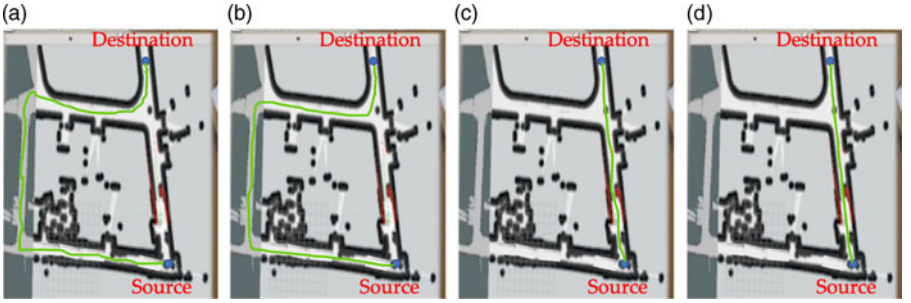


Figure 11. *The robot navigation trajectories (green color) in the 2D map of the real testbed workspace. (a) The A* without reconfiguration, (b) D* without reconfiguration, (c) IRRT* with reconfiguration, and (d) RIRRT* with reconfiguration.*

We can realize that the RIRRT* outperformed the IRRT* in terms of energy and time use, and the outperform margin increases significantly when the search space increases. Although the A* and D* methods perform better than the random path, they consume considerably more energy and time than both RIRRT* and the IRRT* methods, which have reconfiguration implemented to shapeshift through the narrow passage. When comparing the inter-reconfigurable and non-reconfigurable setups of RIRRT*, the inter-reconfigurable configuration achieves shorter and more efficient paths. Additionally, the RIRRT* method demonstrates smoother trajectories with fewer turns in the reconfigurable setup, outperforming IRRT* in terms of path smoothness and efficiency.

The results clearly demonstrate that RIRRT* outperforms the other planners in both execution time and energy consumption under all tested conditions. The SD and CI results strengthen the reliability and robustness of these findings. While IRRT*, BIT*, and D* deliver competitive performance, RIRRT* consistently leads in efficiency, making it the optimal choice for applications requiring both rapid execution

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

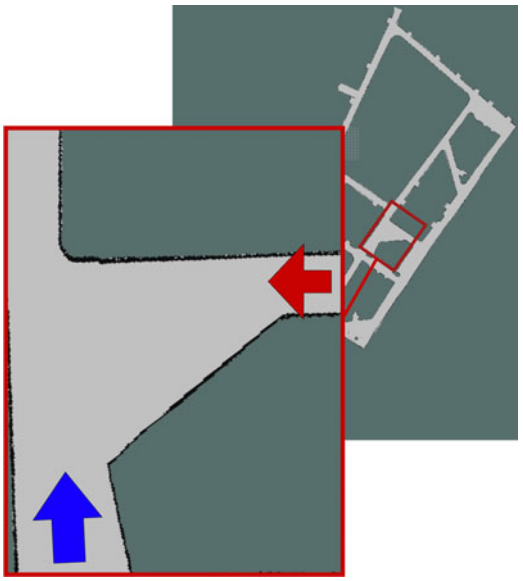


Figure 12. *The map with complex multi-room layouts. The red and blue arrows annotate the perspective of location and orientation toward the actual view of the test environment.*

and low energy usage. The statistical significance of these results confirms RIRRT∗'s advancements over the other planners, solidifying its position as the most effective solution in this context.

6.4. Implementation details discussion

Since the informed sampling technique relies on L^2 heuristics, it is only limited to real-vector state spaces in $\mathbb{R}^n$ with path-minimizing objectives. However, in the case of self-reconfigurable robots with n-dimensional joints, the robot's state space needs to consider both real vector and angular quantities. In order to achieve this and incorporate additional uninformed angular state spaces such as SO(2) or SO(3) during sampling, we only employ uniform random sampling from the whole problem space for these uninformed state spaces. Consequently, this might take significant computation time to find a solution path through narrow passages in planning problems with constraint spaces. We are currently investigating techniques further to incorporate these uninformed state spaces in informed sampling to improve the planning time of the anytime sampling-based algorithms. Moreover, we employ the 2D occupancy grid maps for collision checking, utilizing the two-dimensional polygonal footprints of the self-reconfigurable robots operating on planar surfaces. This can be extended to 3D bounding-box collision checking with 3D-space represented map types such as OctoMaps [50].

7. Conclusion and future works

This paper presented a novel inter-reconfigurable multi-robot system and validated the proposed sampling-based shortest path planning approach, demonstrating its capability to enhance system accessibility in confined environments. The inter-reconfigurable framework effectively supports navigation tasks by enabling two platforms to connect and collaborate seamlessly. The proposed approach leverages greedy anytime sampling-based planning algorithms, which operate in high-dimensional state spaces without explicitly constructing high-dimensional occupancy obstacles. Instead, these algorithms rely on random sampling to asymptotically achieve optimal solutions. By employing greedy-based sampling tailored for reconfigurable systems, the proposed algorithm improves convergence rates, particularly in

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Table II. *Execution time and energy consumption with standard deviations (SD) and confidence intervals (CI) of the planners in terms of with (w) configuration and without (wo) reconfiguration for the real-world testbed environment.*

planners	w		wo	
	running time (s) (SD) (CI)	**energy (kJ) (SD) (CI)**	**running time (s) (SD) (CI)**	**energy (kJ) (SD) (CI)**
A*	672.58 (34.25) CI: [630.42, 714.74]	274.61 (14.32) CI: [260.23, 288.99]	491.32 (32.51) CI: [459.17, 523.47]	239.56 (15.47) CI: [228.12, 250.00]
D*	641.37 (31.21) CI: [609.22, 673.52]	263.45 (13.75) CI: [249.70, 277.20]	469.81 (30.02) CI: [439.77, 499.85]	229.32 (14.11) CI: [215.65, 242.99]
IRRT*	593.41 (28.18) CI: [565.23, 621.59]	269.90 (12.91) CI: [256.99, 282.81]	472.19 (30.71) CI: [441.21, 503.17]	221.21 (13.92) CI: [207.91, 234.51]
BIT*	581.21 (25.09) CI: [556.62, 605.80]	258.69 (10.45) CI: [248.24, 269.14]	453.74 (29.12) CI: [424.17, 483.31]	214.71 (11.88) CI: [202.56, 226.87]
RIRRT*	571.39 (21.67) CI: [549.79, 593.99]	251.25 (9.87) CI: [240.22, 262.28]	412.79 (27.81) CI: [387.73, 437.85]	196.32 (10.43) CI: [185.83, 206.81]

high-dimensional planning problems involving narrow, constrained regions, outperforming sample and grid based state-of-the-art methods.

For future work, the framework will be extended to accommodate generic self-reconfigurable robots operating in 3D workspaces with dynamic obstacles and complex layouts. This extension requires integrating both real-vector and angular state spaces to fully represent the robot's configurations. Additionally, for planning problems with path-minimizing objectives, the real-vector state spaces will be further explored using the greedy informed sampling process to enhance computational efficiency and solution quality. These advancements will enable broader applicability of the proposed framework in dynamic and complex environments. While the current work focuses on single-inter reconfigurable robot scenarios, the value of extending the framework to multi-robot systems for distributed tasks is also the potential direction for future research.

Author contributions. Anh Vu Le, Vinu Sivanantham, and Cong Hien Dinh conceived and designed the paper. They also conducted the paper gathering, along with Do Quang Huy Prabakaran Veerajagadheswar and Guangming Chen. Anh Vu Le, Vinu Sivanantham, Cong Hien Dinh, Do Quang Huy, Bui Vu Minh, Prabakaran Veerajagadheswar, Guangming Chen, and Rajesh Elara Mohan wrote the article.

Financial support. This research is supported by the National Robotics Programme under its National Robotics Programme BAU, Ermine III: Deployable Reconfigurable Robots, Award No. M22NBK0054 and also supported by A∗STAR under its "RIE2025 IAF-PP Advanced ROS2-native Platform Technologies for Cross sectorial Robotics Adoption (M21K1a0104)" programme.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Conflicts of interest. We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval. None.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/S0263574725000645.

References

[1] M. Javaid, A. Haleem, R. P. Singh and R. Suman, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," *Cogn. Robot.* **1**, 58–75 (2021).

[2] R.-J. Hong, Y.-R. Li, M.-H. Hung, J.-W. Chang and J. C. Hung. "Integrating Object Detection and Semantic Segmentation into Automated Pallet Forking and Picking System in AGV." **In:** *International Conference on Frontier Computing* (Springer, 2022) pp. 121–129.

[3] I. Karabegović, E. Karabegović, M. Mahmić and E. Husak, "The application of service robots for logistics in manufacturing processes," *Adv. Prod. Eng. Manage.* **10**(4), 185–194 (2015).

[4] A. A. Tubis and H. Poturaj, "Challenges in the implementation of autonomous robots in the process of feeding materials on the production line as part of logistics 4.0," *Logforum 17*(3), 411–423 (2021).

[5] Cleaning Robot Market by Type Global Forecast to 2023, 2018. https://www.marketsandmarkets.com/Market-Reports/cleaning-robot-market-22726569.html.

[6] A. V. Le, B. Ramalingam, B. F. Gomez, R. E. Mohan, T. H. Q. Minh and V. Sivanantham, "Social density monitoring toward selective cleaning by human support robot with 3D based perception system," *IEEE Access* **9**, 41407–41416 (2021).

[7] A. V. Le, P. Veerajagadheswar, P. T. Kyaw, M. R. Elara and N. H. K. Nhan, "Coverage path planning using reinforcement learning-based tsp for htetran—a polyabolo-inspired self-reconfigurable tiling robot," *Sensors* **21**(8), 2577 (2021).

[8] A. V. Le, R. Parween, P. T. Kyaw, R. E. Mohan, T. H. Q. Minh and C. S. C. S. Borusu, "Reinforcement learning-based energy-aware area coverage for reconfigurable hrombo tiling robot," *IEEE Access* **8**, 209750–209761 (2020).

[9] J. D. Gammell, T. D. Barfoot and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Trans. Robot.* **34**(4), 966–984 (2018).

[10] L. Yi, A. V. Le, A. A. Hayat, C. S. C. S. Borusu, R. E. Mohan, N. H. K. Nhan and P. Kandasamy, "Reconfiguration during locomotion by pavement sweeping robot with feedback control from vision system," *IEEE Access* **8**, 113355–113370 (2020).

[11] M. A. Muthugala, A. V. Le, E. S. Cruz, M. R. Elara, P. Veerajagadheswar and M. Kumar, "A self-organizing fuzzy logic classifier for benchmarking robot-aided blasting of ship hulls," *Sensors* **20**(11), 3215 (2020).

[12] V. Prabakaran, A. V. Le, P. T. Kyaw, R. E. Mohan, P. Kandasamy, T. N. Nguyen and M. Kannan, "Hornbill: A self-evaluating hydro-blasting reconfigurable robot for ship hull maintenance," *IEEE Access* **8**, 193790–193800 (2020).

[13] A. V. Le, P. T. Kyaw, R. E. Mohan, S. H. M. Swe, A. Rajendran, K. Boopathi and N. H. K. Nhan, "Autonomous floor and staircase cleaning framework by reconfigurable stetro robot with perception sensors," *J. Intell. Robot. Syst.* **101**(1), 1–19 (2021).

[14] K. P. Cheng, R. E. Mohan, N. H. K. Nhan and A. V. Le, "Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot," *IEEE Access* **8**, 121267–121284 (2020).

[15] A. V. Le, A. A. Hayat, M. R. Elara, N. H. K. Nhan and K. Prathap, "Reconfigurable pavement sweeping robot and pedestrian cohabitant framework by vision techniques," *IEEE Access* **7**, 159402–159414 (2019).

[16] A. V. Le, V. Prabakaran, V. Sivanantham and R. E. Mohan, "Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor," *Sensors* **18**(8), 2585 (2018).

[17] N. Tan, R. E. Mohan and K. Elangovan, "Scorpio: A biomimetic reconfigurable rolling–crawling robot," *Int. J. Adv. Robot. Syst.* **13**(5), 1729881416658180 (2016).

[18] S. Nansai, N. Rojas, M. R. Elara and R. Sosa. "Exploration of Adaptive Gait Patterns with a Reconfigurable Linkage Mechanism." **In:** *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013) pp. 4661–4668.

[19] D. T. Vo, A. V. Le, T. D. Ta, M. Tran, P. Van Duc, M. B. Vu and N. H. K. Nhan, "Toward complete coverage planning using deep reinforcement learning by trapezoid-based transformable robot," *Eng. Appl. Artif. Intel.* **122**, 105999 (2023).

[20] L. Yi, A. Y. S. Wan, A. V. Le, Q. R. T. Abdullah Aamir Hayat and R. E. Mohan, "Complete coverage path planning for reconfigurable omni-directional mobile robots with varying width using gbnn (n)," *Expert Syst. Appl.* **228**, 120349 (2023).

[21] H. Do, A. V. Le, L. Yi, J. C. C. Hoong, M. Tran, P. Van Duc, M. B. Vu, O. Weeger and R. E. Mohan, "Heat conduction combined grid-based optimization method for reconfigurable pavement sweeping robot path planning," *Robot. Auton. Syst.* **152**, 104063 (2022).

[22] A. V. Le, D. T. Vo, N. T. Dat, M. B. Vu and M. R. Elara, "Complete coverage planning using deep reinforcement learning for polyiamonds-based reconfigurable robot," *Eng. Appl. Artif. Intel.* **138**, 109424 (2024).

[23] A. V. Le, R. Parween, R. E. Mohan, N. H. K. Nhan and R. Enjikalayil, "Optimization complete area coverage by reconfigurable htrihex tiling robot," *Sensors* **20**(11), 3170 (2020).

[24] V. Prabakaran, A. V. Le, P. T. Kyaw, P. Kandasamy, A. Paing and R. E. Mohan, "sTetro-D: A deep learning based autonomous descending-stair cleaning robot," *Eng. Appl. Artif. Intel.* **120**, 105844 (2023).

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

[25] A. V. Le, T. L. A. Pamela, A. A. Hayat, B. R. Nair, P. T. Kyaw, M. B. Vu, D. T. Vo and M. R. Elara, "Towards staircase navigation and maintenance using self-reconfigurable service robot," *Expert Syst. Appl.* **266**, 125969 (2025).

[26] V. Kee, N. Rojas, M. R. Elara and R. Sosa. "Hinged-tetro: A Self-reconfigurable Module for Nested Reconfiguration." **In:** *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (IEEE, 2014) pp. 1539–1546.

[27] A. R. Khairuddin, M. S. Talib and H. Haron. "Review on Simultaneous Localization and Mapping (slam)." **In:** *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)* (IEEE, 2015) pp. 85–90.

[28] R. Smith, M. Self and P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics," **In:** *Autonomous Robot Vehicles* (Springer, 1990) pp. 167–193.

[29] C. Li, S. Wang, Y. Zhuang and F. Yan, "Deep sensor fusion between 2D laser scanner and imu for mobile robot localization," *IEEE Sens. J.* **21**(6), 8501–8509 (2019).

[30] T.-K. Lee, S.-H. Baek, Y.-H. Choi and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robot. Auton. Syst.* **59**(10), 801–812 (2011).

[31] P. Bhattacharya and M. L. Gavrilova. "Voronoi Diagram in Optimal Path Planning." **In:** *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)* (IEEE, 2007) pp. 38–47.

[32] S. Skiena. Dijkstra's algorithm. **In:** *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica* (Addison-Wesley, Reading, MA, 1990) pp. 225–227.

[33] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.* **4**(2), 100–107 (1968).

[34] S. Dhouib, "An optimal method for the shortest path problem: The dhouib-matrix-spp (dm-spp)," *Results Control Optimiz.* **12**, 100269 (2023).

[35] S. Dhouib, "Enhanced Path Planning with dm-spp-24 and dm-spp-4: A Comparative Study," **In:** *Advances in Transdisciplinary Engineering* (2023).

[36] S. Dhouib, "Faster than Dijkstra and a* Methods for the Mobile Robot Path Planning Problem Using Four Movement Directions: The Dhouib-Matrix-spp-4," **In:** *Mechatronics and Automation Technology* (IOS Press, 2024) pp. 284–290.

[37] L. E. Kavraki, J.-C. L. Petr Svestka and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996).

[38] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *Int J. Robot. Res.* **20**(5), 378–400 (2001).

[39] W. Edsger and etal Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.* **1**(1), 269–271 (1959).

[40] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int J. Robot. Res.* **30**(7), 846–894 (2011).

[41] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot. "Batch Informed Trees (bit*): Sampling-based Optimal Planning Via the Heuristically Guided Search of Implicit Random Geometric Graphs." **In:** *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015) pp. 3067–3074.

[42] M. Shahabi, H. Ghariblu and M. Beschi, "Comparison of different sample-based motion planning methods in redundant robotic manipulators," *Robotica* **40**(9), 3104–3119 (2022).

[43] L. Dong, R. Zhang, J. Wang, J. Li, S. Wang and X. Wang, "Research on the path planning algorithm and obstacle-crossing motion planning strategy for a cable trench inspection robot," *Robotica*, 1–16 (2024).

[44] G. Gao, D. Li, K. Liu, Y. Ge and C. Song, "A study on path-planning algorithm for a multi-section continuum robot in confined multi-obstacle environments," *Robotica* **42**(10), 1–24 (2024).

[45] H. Taheri, B. Qiao and N. Ghaeminezhad, "Kinematic model of a four mecanum wheeled mobile robot," *Int. J. Comput. Appl.* **113**(3), 6–9 (2015).

[46] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, "Ros: An Open-Source Robot Operating System," In: *ICRA Workshop on Open Source Software*. vol. **3** (2009) pp. 5.

[47] P. T. Kyaw, A. V. Le, P. Veerajagadheswar, M. R. Elara, T. T. Thu, N. H. K. Nhan, P. V. Duc, and M. B. Vu, "Energy-Efficient path planning of reconfigurable robots in complex environments," *IEEE Trans. Robot.* **38**(4), 2481-2494 (2022).

[48] K. Sriniketh, A. V. Le, R. E. Mohan, B. J. Sheu, V. D. Tung, P. Van Duc and M. B. Vu, "Robot-aided human evacuation optimal path planning for fire drill in buildings," *J. Build. Eng.* **72**, 106512 (2023).

[49] K. Koide, J. Miura and E. Menegatti, "A portable 3d lidar-based system for long-term and wide-area people behavior measurement," *IEEE Trans. Hum. Mach. Syst.* (2018).

[50] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robot.* **34**(3), 189–206 (2013). http://github.com/OctoMap/octomap.

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

Appendix A: List of Acronyms

Acronym	Full Form
RIRRT*	Reconfigurable Informed Rapidly-exploring Random Tree Star
IRRT*	Informed Rapidly-exploring Random Tree Star
RRT*	Rapidly-exploring Random Tree Star
BIT*	Batch Informed Trees
PRM	Probabilistic Roadmap
A*	A-star Algorithm
$G = (V, E)$	Graph Representation
$g(\mathbf{x})$	Cost-to-Come
$h(\mathbf{x})$	Cost-to-Go
$(\mathbf{x}_I)$	Start state
$(\mathbf{x}_I)$	Goal state
$f(\mathbf{x})$	Total Cost
$c_{\max}$	Maximum Heuristic Cost
X_{soln}	Solution Set
r_{rewire}	Rewiring Radius
$\mathcal{X}_{\text{free}}$	Free Configuration Space
$\mathcal{X}_{\hat{f}}$	Informed Subset
$\mathcal{X}_{\hat{f}_{\text{greedy}}}$	Greedy Informed Subset
ϵ	Greedy Biasing Ratio
ROS	Robot Operating System

Appendix B: Probabilistic Completeness

A path planning algorithm is *robustly feasible* if a solution path exists with strong δ-clearance (i.e., a distance $\delta > 0$ from obstacles in $\mathcal{X} \in \mathbb{R}^n$) [40]. An algorithm guarantees *probabilistic completeness* if, as the number of samples approaches infinity, a start vertex $\mathbf{x}_I$ becomes connected to a goal vertex $\mathbf{x}_G$ in the graph.

RIRRT* grows a tree inheriting the probabilistic completeness of classical RRT. During alternating extensions, new vertices are added to V_a and V_b, where $V_k = V_a \cup V_b$ approaches $\mathcal{X}$ in probability as k increases. Theoretical guarantees on probabilistic completeness remain valid since every sample has a nonzero probability of being selected under the alternating extension strategy. The greedy heuristic aids $\mathcal{X}_{\text{free}}$ coverage without compromising convergence.

Appendix C: Asymptotic Optimality

For a feasible planning problem $(\mathcal{X}_{\text{free}}, \mathbf{x}_I, \mathbf{x}_G)$, the objective is to identify a path π^* that minimizes a cost function $c : \Sigma \to \mathbb{R}_{\geq 0}$, where Σ represents the set of feasible paths. A path π^* is considered *asymptotically optimal* if:

$$P\left(\limsup_{n\to\infty} Y_n = c^*\right) = 1, \tag{C1}$$

where $Y_n \geq c^*$ denotes the cost of the best solution in graph G after n iterations, and c^* is the optimal cost.

The RIRRT* algorithm guarantees asymptotic optimality by performing rewiring operations within an n-dimensional unit ball of radius:

$$r_{\text{rewire}} = \min\{\eta, r_{\text{RRT}^*}\}, \tag{C2}$$

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press

where $\eta > 1$ represents the maximum edge length, and:

$$r_{\mathrm{RRT}^*} := \left(2\left(1+\frac{1}{n}\right)\frac{\lambda(\mathcal{X}_{\mathrm{free}})}{\zeta_n}\left(\frac{\log(|V|)}{|V|}\right)\right)^{\frac{1}{n}}, \tag{C3}$$

with $\lambda(\cdot)$ denoting the Lebesgue measure, ζ_n the measure of the unit ball, and $|V|$ the number of vertices in the graph.

To enhance convergence efficiency, we refine the informed sampling technique by incorporating a dual-layer selection approach: (1) global exploration within $\mathcal{X}_{\hat{f}}$ and (2) local exploitation within $\mathcal{X}_{\hat{f}_{\mathrm{greedy}}}$ to accelerate refinement in promising regions. The RIRRT* algorithm further refines this process by sampling within a greedy informed subset $\mathcal{X}_{\hat{f}_{\mathrm{greedy}}}$, a subset of $\mathcal{X}_{\hat{f}}$, which increases the sampling density in regions that are more likely to contain optimal paths.

This dual-layer selection improves asymptotic convergence by reducing inefficient exploration while maintaining sufficient coverage in less promising areas. The impact of sampling with the greedy informed subset on the worst-case number of samples required for probabilistic optimality

Cite this article: A. V. Le, C. H. Dinh, V. Sivanantham, P. Veerajagadheswar, D. Q. Huy, B. V. Minh, G. Chen and R. E. Mohan (2025). "Inter-reconfigurable robot by modified informed sampling-based shortest path planning in cleaning and maintenance", Robotica **43**, 2121–2142. https://doi.org/10.1017/S0263574725000645

https://doi.org/10.1017/S0263574725000645 Published online by Cambridge University Press