



Path Planning for a Reconfigurable Robot in Extreme Environments

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Cheah, W., Garcia Nathan, T. B., Groves, K., Watson, S., & Lennox, B. (Accepted/In press). *Path Planning for a Reconfigurable Robot in Extreme Environments*. Paper presented at IEEE ICRA 2021, Xi'an, China.

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



Path Planning for a Reconfigurable Robot in Extreme Environments

Wei Cheah Tomas B. Garcia-Nathan Keir Groves Simon Watson Barry Lennox

Abstract—In recent years, the inspection of extreme environments using mobile robots has gained traction, as robots are able to mitigate the risk placed on humans and at times achieve what humans are unable to. In some scenarios, the robot is required to operate in cluttered environments with highly restricted access through 150 mm diameter ports. The MIRRAX robot has been designed to meet these challenges with the capability of reconfiguring itself to both access environments and navigate through tightly spaced obstacles. The joints used for reconfiguration of the robot introduce additional challenges for path planning due to the significant changes that can occur between adjacent poses. This paper presents a global path planner for MIRRAX. A Voronoi diagram is first used to generate a sparse graph to represent the topology of the environment, which allows for fast, coarse path planning. The coarse path is then refined via a heuristic pose fitting routine to ensure that the path is both collision-free and reduce unnecessary joint angle changes. The planner has been evaluated in simulation, demonstrating the feasibility of generating collision-free paths through narrow pathways for a reconfigurable robot.

I. INTRODUCTION

The need for inspection, monitoring and intervention in extreme environments has grown considerably following the Fukushima incident and the continued aging of nuclear facilities [1]. These tasks are non-trivial, requiring substantial cost, and where human operators are involved, there is an element of risk to their health when operating in these environments. Faced with these challenges, the use of mobile robots has gained interest as risks to human health can be mitigated and robots can in some cases perform tasks that humans are unable to [2]. Even so, challenges remain in deploying and operating mobile robots in these environments. One such challenge is navigating through constrained and cluttered areas and another is that access points may be restricted, for example, many areas in legacy nuclear facilities can only be accessed through a 150 mm access port [3].

Reconfigurable terrestrial robots with movable joints offer a solution to these challenges. A typical robot setup would comprise two or more rigid bodies that are connected by movable joints, with locomotion provided by wheels or tracks. The range of configurations allow such robots to access areas through small ports and to traverse through highly constrained environments such as inside pipes, or ducts or between tightly spaced obstacles. In a recent case study [4], Sarcos' Guardian S reconfigurable robot was used

to inspect a dust extraction system, reducing both risk to employees and manufacturing downtime. Another example was developed by Tokyo Electric Power Company (TEPCO) [3]. Their robot was designed to take a tubular form, traverse a narrow pipe, and once clear of the pipe expand into a U shape for stability. The algorithms developed in this paper are designed to be applicable to any reconfigurable terrestrial robot and are tested on MIRRAX [5], shown in Fig. 1. MIRRAX is a holonomic three link reconfigurable robot that can fit through a 150 mm access port and was deployed to map the inside of a sealed, legacy nuclear facility on the Sellafield site in the UK (see Fig. 1b for result of the mapping). This deployment highlighted the importance of the robot's reconfigurable capabilities, as debris on the floor resulted in a cluttered and partially confined environment, that the robot needed to navigate through.

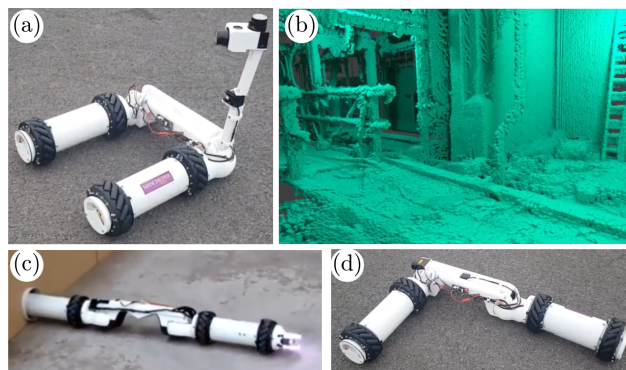


Fig. 1. The MIRRAX reconfigurable robot (a) Dual LIDAR set up for mapping (b) 3D point cloud reconstruction of Sellafield facility (c) Egress from a 150 mm access port (d) Example of reconfigurable shape.

When undertaking missions with a reconfigurable ground robot, there is significant stress on the operator because the robot's pose and joint angles must be constantly adjusted to fit between obstacles without getting stuck. In the case of MIRRAX, the two additional joints expands the search-space to a five dimensional problem, which is non-trivial, compared to a three dimensional problem for a fixed configuration robot in planar space. Usually there is also time pressure on the operator. This could be due to battery life constraints, to minimise the radiation dose received by the robot or when searching for an injured person in a rescue scenario.

Having a high-level planner to find a feasible path with collision-free poses for the robot would reduce the burden on the operator and allow the mission to be completed faster than using manual control. One particular motion planning approach is the combination of map representation using

This work was supported by UK Research and Innovation through the Engineering and Physical Science Research Council under grant number EP/P01366X/1 and EP/R026084/1

The authors are all with the Department of Electrical and Electronic Engineering, University of Manchester, UK; contact wei.cheah@manchester.ac.uk

Voronoi diagram and Signed Distance Fields (SDFs) which allows for fast planning and collision checking, outperforming sampling-based approaches especially when dealing with narrow pathways, which is the focus of this paper [6], [7]. The Voronoi diagram may at times be insufficient to guarantee a collision-free path, requiring the addition of heuristics [8], gradient-based trajectory optimization [9] or sampling-based methods [10] to avoid the collision. However, the robots used in the published research are either fixed configuration or have no constraints in their configurations.

The motion planning problem for MIRRAX introduces new challenges compared to fixed and other reconfigurable robots. The literature on motion planning for reconfigurable robots that is closely similar to MIRRAX often concerns modular type robots with detachable units, as opposed to MIRRAX's joints which are not detachable [11], [12]. Moving using the U-shape configuration as opposed to other shapes provide better controllability hence is preferred where the environment allows [13], [14]. Furthermore, navigating from the U-shape to an almost straight shape configuration for the ingress of narrow pathways requires the transition to start at a certain distance away from the entry. Using existing approaches would result in collision as the robot would be too near the entry to make significant change in its pose.

This paper proposes a global path planner for the MIRRAX reconfigurable robot to address this gap in the literature. The approach here extends upon the work of [7] which was used for a fixed configuration robot. The sparse graph is first used for coarse path planning, followed by a heuristic-based pose fitting approach for ensuring collision-free poses along the path, and at the same time avoids both singular configurations and unnecessary pose changes. The contribution of this work is a global path planner for a reconfigurable robot that is capable of generating collision-free poses along a path.

II. ROBOT AND PLANNER OVERVIEW

The annotations for the MIRRAX robot, along with its dimension and reference frame, is shown in Fig. 2. The robot has two leg links attached to its base link via joints q_1 and q_2 , both of which has a range of 0 to π rad, where the pose shown in Fig. 2 have both joints at 0 rad. The pose of the robot is defined as $\mathbf{p} = (p_x, p_y, \theta, q_1, q_2)$ where the first three terms are the standard 2D pose x , y and rotation, followed by the two joints on MIRRAX.

An overview of the planner is shown in Fig. 3. The Euclidean Signed Distance Field (ESDF), Voronoi diagram and sparse graph follows the approach of [6] and [7], though any other approaches to generate the first two map representations would also work. The Sparse Graph tree is a truncated version of the Voronoi diagram and is represented in a graph form. Path finding then becomes a task of traversing the sparse graph using algorithms such as Dijkstra, A* or shortest-path. The resulting path consists of a list of vertices with x - y position (2D planning in this study), with large gaps between them due to the sparsity of the graph in representing

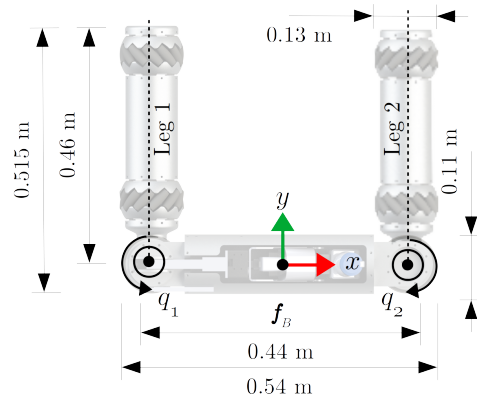


Fig. 2. MIRRAX robot dimension and frame of references.

the topology of the space. Hence, the path is discretised into smaller uniform segments for the subsequent steps.

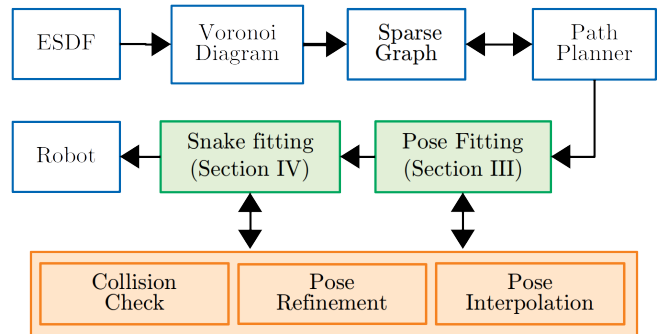


Fig. 3. Path planner overview.

The Pose Fitting (Section III) first finds a coarse pose, refines it, and then ensures that the transition between adjacent poses are collision-free. In the event that the interpolation fails, the Snake Fitting approach (Section IV) is used which would handle difficult motions such as the ingress and egress of narrow pathways. Similar to before, the pose generated by Snake Fitting is refined and interpolation between adjacent poses is used to check for collision. It should be noted that the planner explicitly takes advantage on the small number of links that the robot has.

III. POSE FITTING

The pose fitting approach for the MIRRAX reconfigurable robot does this through two steps, both of which utilises the SDF in different manner. First, a coarse pose fit based on the SDF with respect to the base frame. Second, a fine fitting approach using the SDF at various locations on the link to find the magnitude and direction to rotate away from obstacles.

A. Coarse Pose Fitting

First, the base rotation, θ , is computed using the tangent of the path via linear regression centered at the robot's frame, f_b , similar to [6]. Second, a symmetric pose based on the SDF at f_b is used to offset the robot so that its geometrical

center lies on the path itself. For fixed robots, the base or robot frame is chosen to coincide with the geometrical center of the robot and is a fixed point. In the case of MIRRAX, the geometrical center is not fixed and is a function of the leg's joint angles, $\mathbf{p}_{gc} = f(q_1, q_2)$. The geometrical center is calculated using four points placed at the edges of the robot, as shown in Fig. 4a. Depending on the leg's joint angles, the points are switched so that they correspond to the far edges of the robot (see Fig. 4b). The geometrical center then simply takes the mean between the extreme end points on the x- and y-axis among the available points.

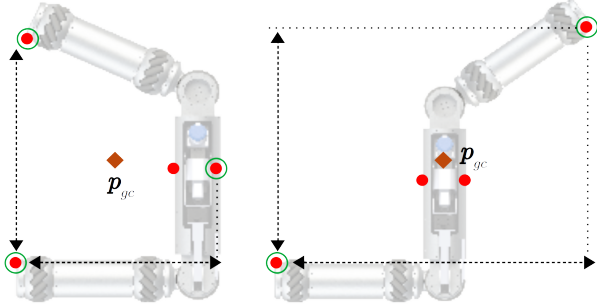


Fig. 4. Geometrical center of robot. The point on the link is switched when it exceeds the x-axis of the base frame.

The symmetric pose is a linear interpolation between the narrowest and widest configuration the robot can achieve. The collision-free distance from the robot's geometric centre to the nearest obstacle for these two configurations are set beforehand, thus the linear interpolation for the joint angles, q , can be described by

$$q = \begin{cases} q_{min}, & d < d_{min} \\ \frac{d_{max} - d}{d_{max} - d_{min}} \cdot q_{min}, & d_{min} \leq d \leq d_{max} \\ 0, & d > d_{max} \end{cases} \quad (1)$$

where d_{max} and d_{min} corresponds to the minimum permissible distance for the widest and narrowest configuration measured from the robot's base frame to the obstacle, q_{min} is the joint angle at the narrow configuration and d is the SDF at f_b from the path. The condition $d < d_{min}$ is required for scenarios where the path width is narrower than the ideal minimum path width. This scenario can occur due to discretization in the map representation, or in practice due to sensor noise.

At this point, the full pose, although a coarse one, is used to offset the robot's position so that its geometrical center lies on the path. This is calculated using a simple 2D transformation, $T(\theta)\mathbf{p}_{gc}$.

B. Pose Refinement

The next phase is concerned with refining the coarse pose-fitting in the previous step and ensures that all the links are collision free. The refinement is done sequentially, first for the base link followed by the leg links. The approach used

here is to rotate the base and leg links using geometrical calculations that are functions of SDF values at the collision points, coupled with an iterative scheme.

The pseudo-code for calculating the rotation for a link to move from collision to collision-free is given in Alg. 1. While the details of the approach below uses the base link, the same approach applies for the leg link as well. The robot's link have a vertical pair of points distributed across its length, shown in Fig. 5a, for the base link.

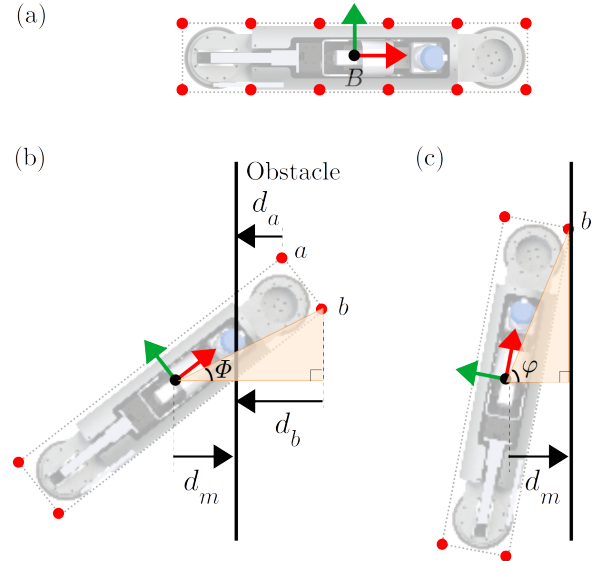


Fig. 5. (a) Collision points for base link (b) Collision pair used for identifying rotation magnitude and direction (c) Collision free SOMETHING.

Algorithm 1 computeRot

Require: link, j , θ

Ensure: c_{free}, θ_{new}

```

1:  $c_{free} \leftarrow \text{true}$ 
2:  $\text{updatePose}(\theta)$ 
3:  $\mathbf{c}_a, \mathbf{c}_b \leftarrow \text{getCollisionPointsPair}(\text{link}, j)$ 
4:  $d_a, d_b \leftarrow \text{getSDF}(\mathbf{c}_a, \mathbf{c}_b)$ 
5: if ( $d_a$  or  $d_b$ ) <  $d_{thres}$  then
6:    $c_{free} \leftarrow \text{false}$ 
7:   if  $d_a < d_b$  then
8:      $d \leftarrow d_a$ 
9:      $r \leftarrow -1$ 
10:  else if  $d_b < d_a$  then
11:     $d \leftarrow d_b$ 
12:     $r \leftarrow 1$ 
13:  end if
14:   $\theta_{rot} \leftarrow r \cdot \Delta\text{Rot}(d, d_m, d_{hpt})$  ▷ see Eq. 2
15:  if  $\text{isnan}(\theta_{rot})$  then
16:     $\theta_{rot} \leftarrow r \cdot \theta_{off}$ 
17:  end if
18:   $\theta_{new} \leftarrow \theta + \theta_{rot} + r \cdot \theta_{off}$ 
19: end if

```

The aim if to find a rotation, θ_{rot} , that would rotate the link from its initial orientation, Φ , to a collision-free orientation,

φ (see Fig. 5b-c). The magnitude, $\theta_{rot} = \Phi - \varphi$ and direction, r , to rotate the link to a collision free pose is calculated based on the collision point that has the lower SDF value (point b for Fig. 5b). Upon extracting the SDF values for the collision pairs (line 4), the direction and point to use for calculating the magnitude can be determined via a simple check on their respective SDF magnitude (lines 7 to 12). Following this, θ_{rot} can be calculated as follows (line 14):

$$\theta_{rot} = \cos^{-1} \left(\frac{d_m}{d_{hpt}} \right) - \cos^{-1} \left(\frac{d_m + d}{d_{hpt}} \right) \quad (2)$$

where, d is d_a or d_b , as appropriate, and d_{hpt} is the distance from the centre of rotation to the lower valued of point a or b , i.e the hypotenuse of the right angled triangle in Fig. 5b-c.

Discretization error from the SDF can cause error in the both \cos^{-1} in (2) and the actual distance between the collision point to the obstacle. Both these issues are addressed through the use of a constant offset in the desired rotation direction (lines 15 to 17).

Algorithm 1 describes the rotation for only a single pair of points. The pseudo-code in Alg. 2 describes how multiple pairs of points on a link is used to rotate the link out of collision. The rotation described in Alg. 1 is iterated for each collision pair points, starting from the pair nearest to the base frame and finishing at the edge of the link (line 6). The set of collision points are then further iterated until a collision-free pose is found or terminated after a number of iteration, n_i (line 3). Both these bounded iterative approaches ensure that the entire link is collision free, as compared to using only a single pair of collision points for the entire link, and also accounts for scenarios where some collision points may be in a local minimums.

Algorithm 2 Base Rotation Refinement

```

1:  $\theta_{new} \leftarrow \theta$ 
2:  $c_{free} \leftarrow false$ 
3: while  $c_{free} = false \ \& \ i < n_i$  do
4:    $i \leftarrow i + 1$ 
5:    $c_{free} \leftarrow true$ 
6:   for  $j < n_{cp}$  do
7:      $c_{free}, \theta_{new} \leftarrow computeRot(base_{front}, j, \theta_{new})$ 
8:      $c_{free}, \theta_{new} \leftarrow computeRot(base_{rear}, j, \theta_{new})$ 
9:   end for
10: end while

```

Under certain circumstances, Alg. 2 cannot guarantee a collision-free yaw since the base link is constrained to its x - y position. However, since the base link is largely centered on the Voronoi Diagram, it is expected that the proposed approach is sufficient to find a collision-free yaw.

Algorithm 2 has been adapted to find collision-free joint angles for the leg link, with two minor modifications. First, the collision points only consists of a frontal part since the point of rotation is located at the end of the link as opposed to the center in the case of the base link. Second, an additional check has been introduced to ensure that the joint angles are within their physical range of motion (lines 8 and 9). This is

achieved by flipping the rotation if the joint angles are out of range and then relying on the iterative scheme to move it to a collision free pose. These are described in Algorithm 3.

Algorithm 3 Leg i Joint Angle Search

```

1:  $q_{new} \leftarrow q_i$ 
2:  $c_{free} \leftarrow false$ 
3: while  $c_{free} = false \ \& \ i < n_i$  do
4:    $i \leftarrow i + 1$ 
5:    $c_{free} \leftarrow true$ 
6:   for  $j < n_{cp}$  do
7:      $c_{free}, q_{new} \leftarrow computeRot(leg_i, j, q_{new})$ 
8:     if  $q_{new} > q_{i,limit}$  then
9:        $q_{new} \leftarrow -q_{new}$ 
10:    end if
11:  end for
12: end while

```

C. Pose Interpolation

Collision checking on the linear interpolation between adjacent poses is used to determine if the transition from one pose to another is feasible. The robot's collision is represented by a set of evenly distributed points around the robot, similar to [15]. This simplifies the collision check to that of comparing the collision points SDF values to a threshold.

D. Reducing Pose Change

The outcome of the Voronoi diagram can introduce small changes in position for adjacent vertices caused by the topology of the map. These small changes in position are passed from the Voronoi diagram to the sparse graph and generate paths with zig-zags where a straight path could be fitted. The non-straight paths add unnecessary complications in the pose fitting of the robot, so an extra computation step has been applied to the sparse graph to address this problem.

An alignment process is attempted for each pair of connected vertices. The x -coordinate of the first vertex is changed to the same value as the x -coordinate of the second vertex in order to align them. The modified vertex is then made permanent if it is collision free, otherwise the vertex is returned to its original position. This process is repeated for the y -coordinate and then for the second vertex. The result of this alignment process is a graph with more straight lines compared to the Voronoi Diagram.

IV. SNAKE FITTING

The ingress and egress of MIRRAX from a narrow to wide configuration, or vice versa, requires a different pose-fitting approach than the one suggested above. This is due to the significant difference between adjacent poses, which would result in collision during the interpolation step. To this end, the approach of aligning the robot along its longitudinal axis with the tangent of path is used here, similar to [10]. This fitting approach is only used when the interpolation between two adjacent poses (detailed in Section III-C) fails.

The snake fitting consists of a segment, starting and finishing at fixed intervals away from where the interpolation fails, as shown in Fig. 6a. These intervals have been introduced to provide sufficient clearance space to transition from the original pose to the straight configuration and vice versa. The pose associated with the path within this segment will be replaced with the approach here, along with the rest of the points until the end of the segment.

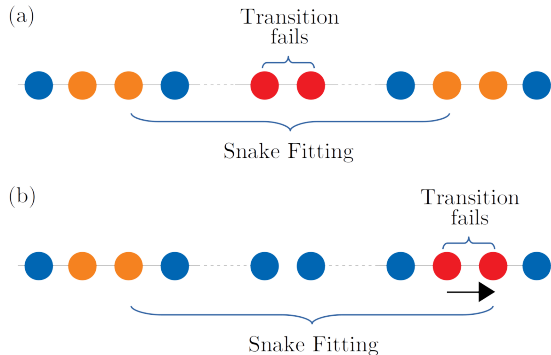


Fig. 6. Discretised points along the path showing the pairs (orange) used for checking interpolation. The red points are where interpolation fails. (a) The original segment used for snake fitting. (b) The extended segment for snake fitting since the interpolation failed.

The start and end point of the snake fitting segment is subsequently interpolated with their adjacent point, to ensure a collision-free transition (see Fig. 6a where the orange point pair are). If the interpolation fails for either end, as shown in Fig. 6b, the snake fitting is extended for that end to the adjacent point. This extension is repeated until either there is a successful interpolation or the process is terminated at the end of the path.

As the robot approaches the end of the path, its front half would exceed the path, resulting in no further points that can be used to determine the path tangent for fitting the robot using the approach here. Using the tangent at the end of the path can cause incorrect pose fitting if the path terminates towards the end of a corner. To address this, the Voronoi diagram that continues on from the end of the path is used for the pose fitting approach here.

A drawback of the snake-fitting configuration is that the robot can be fitted with a straight pose ($q_1 = \pi/2$, $q_2 = -q_1$), which is a singular configuration in which the robot is not fully controllable. In practice, this configuration would result in the robot rolling. This problem is addressed by replacing the singular configuration, when it has been detected, with the narrow configuration (slight bend at both joints). The robot's base frame is also offset slightly as detailed earlier in Section III-A to centralise the robot along the path.

V. EVALUATION

The proposed motion planner has been verified in simulation. This section describes the setup used for verifying the planner and the results of the simulation evaluated across a number of start and goal positions.

A. Simulation Setup

A 2D CAD generated environment with multiple instances of narrow pathways has been created to evaluate the proposed motion planner. The ESDF map was generated from the CAD mesh using the `voxblox_ground_truth` ROS package¹, followed by the generation of the Voronoi diagram using the `dynamicvoronoi` ROS package² [6]. The simulated environment, along with the one-voxel thick Voronoi diagram and sparse graph is shown in Fig. 7. A voxel size of 0.05 m is used. The minimum path width found in the map is 0.4 m. The selection of this path width will be discussed further shortly. To generate the sparse graph the approach taken by [6] was used, plus the reduction of unnecessary pose changes between adjacent vertices. The path planner was evaluated across three different start and end positions (1-2, 1-3, 1-4), marked in Fig. 7.

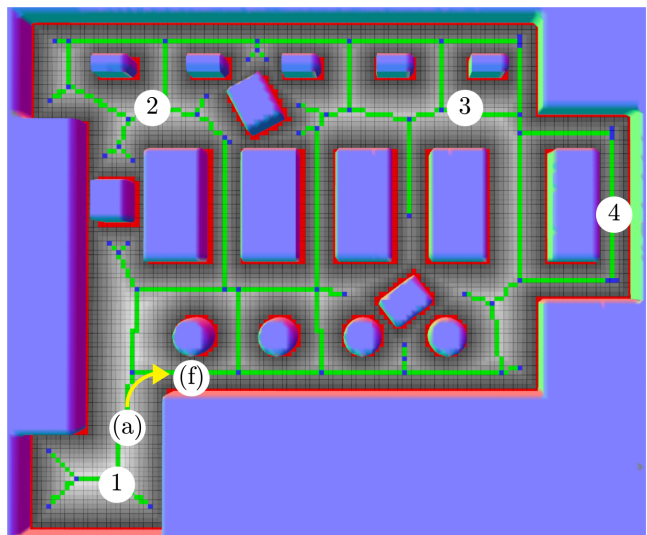


Fig. 7. Simulated environment showing the SDF (grayscale shade), sparse graph (vertices in blue while edges are in green), and the positions used for path planning.

B. Results and Discussion

The proposed path planner succeeded in finding collision-free paths for the three different start and goal positions. A snapshot of MIRRAX navigating from a large area into a narrow pathway (the segment a-f in Fig. 7 is shown in Fig. 8). It starts with the default configuration and then transitions to the snake fitting pose (Fig. 8b) to minimise its lateral width for ingressing the narrow pathway (Fig. 8c-e). In Fig. 8f, MIRRAX would have been in a singular configuration if not for the singular-avoidance approach in Section IV, resulting in the configuration observed. A video showing the simulation results is available in the supplementary material.

The pose fitting for the base link used in the snake-fitting approach has been based on the heuristic proposed by [8],

$$w_p = 2\sqrt{\left(\frac{r_w}{2}\right)^2 + \left(\frac{r_l}{4}\right)^2} \quad (3)$$

¹https://github.com/ethz-asl/voxblox_ground_truth

²<https://github.com/frontw/dynamicvoronoi>

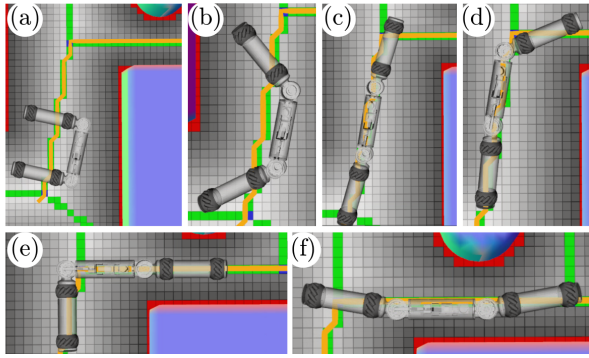


Fig. 8. Snapshot sequence of MIRRAX navigating along a planned path.

where r_w and r_l is the robot's width and length respectively.

For the case of MIRRAX which consists of two unique links (since leg 1 and 2 are of the same design), the base link result in a larger minimum path width compared to the leg links, $w_p \approx 0.3$ m. This is approximately the same corner path width that MIRRAX can manually navigate through physically, although with occasional contacts with the surrounding obstacles. However, the planner was unable to find collision-free poses for MIRRAX when the path width is less than 0.4 m. This is partly due to the approach used here which addresses the robot pose fitting sequentially instead of simultaneously, thus limiting the free-space that the robot is able to explore. Although limiting in this sense, there are applications in the solution obtained here such as using it as an initial guess for gradient-based trajectory optimization approaches or for the operator in manual control operations.

The path planner used here is the shortest-path approach, i.e. the least number of vertices required to connect the start and goal position. The result of this is much more evident in the second simulation where the path passes through the bottom horizontal instead of the middle horizontal. Indeed, there are alternative paths that can be selected to reach the goal position by way of selecting appropriate graph-search algorithms and defining objective functions to improve on the quality of the path. However, this is beyond the scope of this study in demonstrating the feasibility of path planning for a reconfigurable robot, which this paper has achieved.

There were instances where the Voronoi diagram, and the resulting sparse graph, did not lie directly between obstacles. This is due to discretization error and depends on the location of the origin. An example of this can be seen in the vertical narrow paths connecting the top and bottom half of the map whereby the graph edge lie closer to the left obstacle compared to the right. The result, in this case, are SDF values smaller than the expected 0.2 m for the cells lying on the graph edge. The first condition in (1) is thus required for scenarios such as these.

VI. CONCLUSION

This paper proposed an approach for global path planning for reconfigurable ground robots that are connected by joints.

The approach uses a combination of known methods, namely Voronoi Diagrams, SDF maps, and a set of newly developed heuristics. The heuristics used ensure that transition between poses is collision-free especially during the ingress or egress from narrow pathways. The approach has evaluated in simulation, demonstrating the capability of finding collision-free paths through narrow pathways by modifying the robot pose to fit within these confined environments.

For future work, this planner will be evaluated experimentally on the physical platform to identify the practical limitations of the proposed planner. Uncertainties such as sensor noise, state-estimation, and control system, all of which will have an effect on the robot being able to track the path generated by the proposed planner, will be investigated. Further areas of development include extending the planner with objective functions and using it as an initial-guess for gradient-based trajectory optimization.

REFERENCES

- [1] F. Zhao, Y. Ma, and Y. Sun, "Application and standardization trend of maintenance and inspection robot (mir) in nuclear power station," *DEStech Transactions on Engineering and Technology Research*, 2017.
- [2] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Progress in Nuclear Energy*, vol. 111, pp. 109–124, 2019.
- [3] Tepco, "Application of Robot Technology," <https://www.tepco.co.jp/en/decommission/principles/robot/index-e.html>, 2017, accessed: 2020-10-13.
- [4] Sarcos, "Guardian S Case Study: General Electric (GE)," <https://www.sarcos.com/products/guardian-s/case-study/ge/>, accessed: 24-10-2020.
- [5] H. Martin, S. Watson, B. Lennox, and X. Poteau, "Miniature Inspection Robot for Restricted Access eXploration (MIRRAX)," in *WM Symposia*, 2018.
- [6] B. Lau, C. Sprunk, and W. Burgard, "Efficient grid-based spatial representations for robot navigation in dynamic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1116–1130, 2013.
- [7] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Sparse 3D Topological Graphs for Micro-Aerial Vehicle Planning," in *IEEE International Conference on Intelligent Robots and Systems*, 2018, pp. 8478–8485.
- [8] R. J. Takahashi, Osamu and Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Transactions on robotics and automation*, vol. 5, no. 2, pp. 143–150, 1989.
- [9] E. Oleynikova, Helen; Burri, Michael; Taylor, Zachary; Nieto, Juan; Siegwart, Roland; Galceran, "Continuous-time trajectory optimization for online UAV replanning," in *IEEE International Conference on Intelligent Robots and Systems*, 2016, pp. 5332–5339.
- [10] M. Foskey, M. Garber, M. C. Lin, and D. Manocha, "A voronoi-based hybrid motion planner," in *IEEE International Conference on Intelligent Robots and Systems*, 2001, pp. 55–60.
- [11] D. Brandt, "Comparison of a and rrt-connect motion planning techniques for self-reconfiguration planning," in *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 892–897.
- [12] A. V. Le, V. Prabhakaran, 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*, vol. 18, no. 8, 2018.
- [13] C. He, D. Wu, K. Chen, F. Liu, and N. Fan, "Analysis of the Mecanum wheel arrangement of an omnidirectional vehicle," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 233, no. 15, pp. 5329–5340, 2019.
- [14] Y. Li, S. Dai, L. Zhao, X. Yan, and Y. Shi, "Topological design methods for mecanum wheel configurations of an omnidirectional mobile robot," *Symmetry*, vol. 11, no. 10, 2019.
- [15] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian Optimization for Motion Planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.