A Novel, RRT* Based Approach to the Coordination of Multiple Planetary Rovers

Sarah Swinton
James Watt School of Engineering
University of Glasgow
Glasgow, Scotland

Euan McGookin
James Watt School of Engineering
University of Glasgow
Glasgow, Scotland

***Abstract*—The use of multiple autonomous rovers could significantly extend the scientific capabilities of future planetary exploration missions. Central to this is the ability to safely coordinate the planned paths of such robots. Previous work has shown that RRT* is a method well suited to single-agent path planning. The aim of this work is to evaluate the use of a RRT* path planner first for a single planetary rover, then applied to different coordination strategies for a small team of rovers as a multi-agent method of path planning. Two centralised coordination methodologies are applied: fixed-path coordination and prioritised planning. An evaluation is carried out, based on a comparison of the number of collisions and the time taken to plan paths, where the prioritised planning coordination algorithm is selected as the most appropriate for this application. The results drawn from this work suggest that the combined use of prioritised planning and RRT* could be appropriate for use in the guidance, navigation and control systems of planetary rovers.**

***Index Terms*—RRT*, path-planning, coordination, planetary robotics**

I. Introduction

Since 1997, planetary rovers have been used to study the history, geology, and climate of Mars, as well as to provide data that may aid future manned missions [1]. So far, each mission has utilised a single rover per launch with every rover generation becoming more complex. However, NASA's recent Mars 2020 mission sets a precedent for multi-vehicle missions by being the first mission to utilise two vehicles: the Perseverance Rover and the Ingenuity Helicopter [2].

Using a team of rovers would increase the effective sensor footprint available for exploration and therefore allow larger regions to be investigated than is currently possible using a single rover. For example, a group of rovers could split up to investigate particular sites of scientific interest, or to attain maximum coverage of a region of interest. This work will focus on the former of these two mission scenarios. The paths of each member of the rover team must be coordinated such that no collisions occur between them as they traverse paths towards their respective targets. As planetary exploration rovers operate in such remote and hazardous environments, collisions could cause the loss of the rovers involved and severe degradation to the group's data collection capabilities.

The proposed solution to this coordination problem is a combined path planner and coordination system as part of a guidance, navigation and control system for a small group of planetary exploration rovers. The path planner algorithm is based on the Rapidly-exploring Random Trees (RRT) algorithm first introduced by LaValle [3], designed to tackle path planning problems with nonholonomic constraints. RRT is a sampling-based algorithm i.e., the configuration space is represented as a road-map of sampled configurations. This algorithm searches a configuration space for a target point by creating a tree of nodes constructed of randomly defined nodes within the free space. These nodes are then connected to the closest pre-existing node on the tree of nodes. Once the target has been acquired by a node, the RRT algorithm searches backwards, from the target to the start point, in order to find a least cost path. However, this is not an optimal solution.

Karaman and Frazzoli [4] extended RRT to RRT* - an asymptotically optimal version of RRT. This provable optimality improved upon RRT by using a nearest neighbours approach to reduce path cost. However, these additional steps can reduce the performance of the RRT* algorithm by increasing the computational time. In order to minimise this, the number of nodes generated can be kept low, or the algorithm can stop generating new nodes once the target point has been found. A comparison by Noreen et al [5], showed that RRT* significantly improved paths generated by RRT.

While extensive study has been carried out on the use of RRT as a path planning method for a single-agent system there remains a gap in the literature regarding the efficacy of RRT as applied to multi-agent systems. By creating paths that are more 'random', the likelihood that any two rovers will attempt to follow exactly the same path is reduced - making this method potentially well suited to multi-agent path planning by delivering a large sensor footprint for the rover group and reducing the likelihood of rover collisions.

Many approaches to autonomous coordination have been studied in recent years, namely centralised and decentralised paradigms [6]. Within a centralised approach, a global planner is used to plan paths and coordinate the motion of all robots within the group. Centralised approaches offer completeness i.e., a solution will be found if one exists. However, a centralised approach can also incur a large computational time, meaning that it may not scale well to large robot groups [6]. This large computational time can be reduced if the system is decoupled. Two decoupled coordination methods are implemented and compared during testing, to evaluate which is more effective in this context. The first is fixed path coordination. Fixed path coordination is a decoupled approach to multi-robot coordination, where the global planner finds paths for each robot separately, and velocities along the path are adjusted to ensure that no robot collisions occur [6]. The second is prioritised planning. Prioritised planning is another decoupled approach, where individual paths are planned and conflicts between these paths are solved during the planning stage [7]. Each robot in the group is assigned a priority, with the highest priority robot's path being planned first. All subsequent robot paths will treat the higher priority robots as moving obstacles.

Section II covers the system modelling for the four-wheel rover, its environment and its guidance, navigation and control system. Section III details the path planning carried out for a single rover. Section IV sets out the methodology and testing for multi-rover path planning and coordination. Section V holds a discussion following these results. Finally, section VI outlines the conclusions drawn from this work.

II. System Modelling

A. Rover Model

The mathematical model of a simple, 4-wheel symmetrical rover, shown in Fig. 1, was developed and validated at the University of Glasgow [8]. This model meets the required functionality for this project.

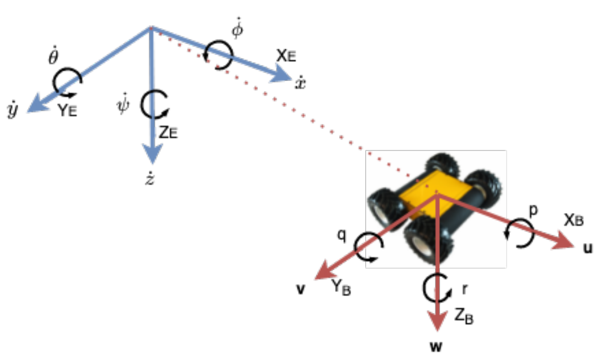

Fig. 1. Earth Fixed And Rover Body Axes

The rover's equations of motion, with reference to the rover body frame and earth fixed frame, can be described by the matrix relationship shown in Equation (1).

$$\begin{bmatrix} \dot{v} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \frac{-(C(v)+D(v))}{M} & \frac{-g(v)}{M} \\ J(\eta) & 0 \end{bmatrix} \begin{bmatrix} v \\ \eta \end{bmatrix} + \begin{bmatrix} \frac{-\tau'}{M} \\ 0 \end{bmatrix} \tag{1}$$

Here, vectors v and η represent the velocity and displacement variables. M is the mass and inertia matrix, C(v) is the Coriolis matrix, D(v) is the damping matrix, g(v) represents the gravitational forces and moments, J(η) is an Euler matrix representing the trigonometric transformation from the body fixed reference frame to the earth fixed reference frame, and the τ vector represents the control inputs. This model is fully described in [8] and [9].

B. Environment Model

The rovers traverse paths through a simulated environment, which includes different terrain types common on the Martian surface, designed to provide a sufficient path planning and traversal challenge to both individual rovers and the group. There are three layers of information within the environmental model: impassable areas, terrain classes, and small obstacles. Impassable areas mimic areas such as craters, cliffs and ditches. Path planning will not allow any traversal through these areas in order to minimise danger to each rover.

Small obstacles are defined within the rock field, of large enough size that they have to be avoided by the rovers. The location of these small obstacles is not available to the path planner and rovers must adjust their planned paths online. The obstacle avoidance cameras ("hazcams") of the simulated rover are modelled on those employed by past NASA Mars rovers [10] - vision-based cameras with a range of 3m and field of view of 120°.

A terrain classification map has been designed to identify the potential risk associated with traversing a particular segment of the map. Commonly occurring terrain classes have been selected [11] i.e. smooth regolith, rock field and steep slopes. Smooth regolith is defined as a firm and smooth surface, without any obstacles or hazards, where the rover should experience minimal slip. Similarly, the rock field should not cause the rover to experience substantial slip. However, the rover must reduce its forward velocity. Steep slopes border the impassable areas, presenting a high risk for rover slip during operation.

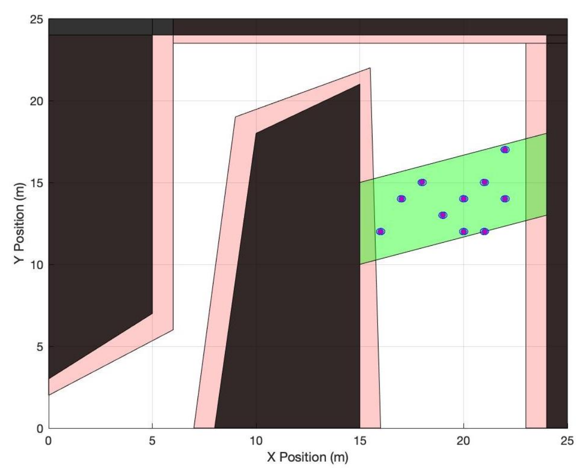

Fig. 2. 2D Environment Model

Fig. 2 shows the environmental model; where black regions are impassable, pink regions represent steep slopes, green regions represent a rough regolith rock field, magenta points represent small obstacles surrounded by a blue safety radius, and white areas represent smooth regolith.

C. Guidance, Navigation and Control System

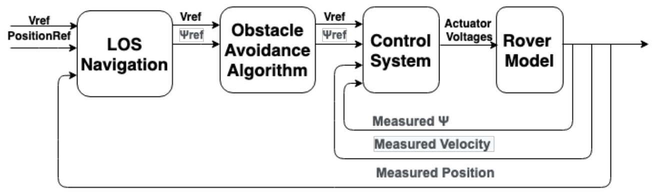

Fig. 3. Guidance, Navigation and Control Architecture

Each rover is equipped with a guidance system providing control, navigation and obstacle avoidance (Fig. 3). The control system consists of two PID controllers for heading and velocity, respectively. Equations (2) and (3) show the error calculations for using the reference and measured values of heading, ψ, and velocity, v, at time, t. These error values are passed to their respective PID controllers.

$$e_{\psi}(t) = \psi_{ref} - \psi_{measured} \tag{2}$$

$$e_{v}(t) = v_{ref} - v_{measured} \tag{3}$$

The navigation algorithm is a Line-of-Sight algorithm [12], allowing each rover to navigate towards its current waypoint. A simple obstacle avoidance algorithm adjusts the heading of the rover in response to the detection of a static obstacle [9]. If an object is identified as visible and is deemed to be a hazard (i.e., if it is within 1m of the rover and it intercepts the rover path), the waypoint following algorithm is adjusted to prioritise manoeuvring around the obstacle. Only once this manoeuvre is complete will the rover return to following its target waypoints.

III. Single Rover RRT* Path Planning

A. Method

Before being applied to a group of rovers, an RRT* algorithm has been developed and applied to a single rover. Fig. 4 sets out the four key stages of this algorithm.

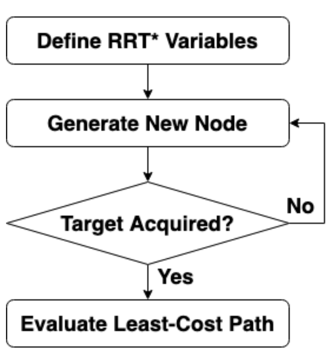

Fig. 4. RRT* Algorithm Stages

In the first stage, the algorithm's operating variables are defined. This includes geographic variables such as the boundaries of the search area, the location of any impassable regions, the rover's start point and target point. Further environmental data is passed to the planner, such as the locations of different terrain areas. This environmental data is used to determine the cost of travelling between nodes, where each terrain areas is allocated a heuristic cost based on the risk associated with it. For example, traversing 1m across a steep slope incurs a greater cost than traversing 1m across smooth regolith. Two variables which affect the computational load of the algorithm are defined: the max number of nodes, q_{max}, and the max distance between nodes, d_{max}. For both of these variables, a design trade-off exists. The larger the maximum number of nodes, the more points can be searched before the algorithm is terminated - however this greatly increases the computational load. If the distance between nodes is increased a larger area can be searched, but the path generated will be less smooth.

Stage two focuses on the generation of nodes in order to search for the target. To create a new node, a random point is generated within the configuration space. A proximity search is then carried out in order to determine the nearest neighbours of the random point. Once the nearest node, q_{near}, has been determined, a new node, q_{new}, can be created on the line between the random point and q_{near}, a distance of d_{max} from q_{near}. Fig. 5 illustrates the node generation process.

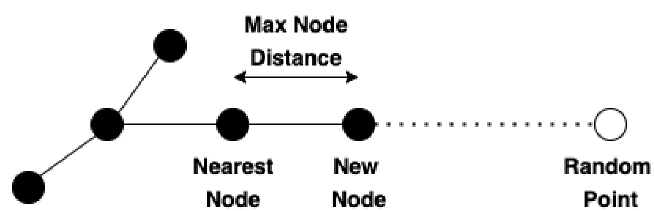

Fig. 5. Random Node Creation Using RRT*

Before moving to stage three, two checks are carried out on q_{new} to ensure it is the most optimal solution for this step. First, the algorithm checks whether the new point, and the branch $q_{near|new}$, do not enter an impassable area. If it does, the algorithm replaces q_{near} with the next nearest neighbour and re-attempts the node generation process. Second, a proximity search is carried out to find the nearest neighbouring nodes to q_{new} and the cost of traversal for $q_{near|new}$ is compared to that of $q_{neighbour|new}$. If a lower cost path is found, $q_{neighbour}$ replaces q_{near}. If both criteria pass, q_{new} is added to the tree of nodes and the algorithm moves to stage three.

In the third stage, a simple check is carried out to evaluate if either of two search termination criteria have been met. The first of these checks is whether any node is within d_{max} of the target. The second check is whether q_{max} has been reached. If either of these criteria is met, the algorithm moves to the final stage.

In the final stage, the algorithm finds the cost from each node to the target point. The least-cost path is found by searching backwards from the target point. Since the start point has no parent node, the algorithm searches for points to add to the least cost path until a node is added whose parent node is equal to zero. The least cost path is then generated and displayed. As the least cost path is created from the target point to the start point, the waypoint array generated must be flipped when passed to the rover's guidance system. Fig. 6 shows a typical tree structure obtained when RRT* performs a search. In this case the path's start point is marked in blue, and its target point is marked in pink. Blue crosses represent points which have been randomly generated. Black lines represent the node tree created by the RRT* planner. The least cost path between the start point and the target point is marked in red.

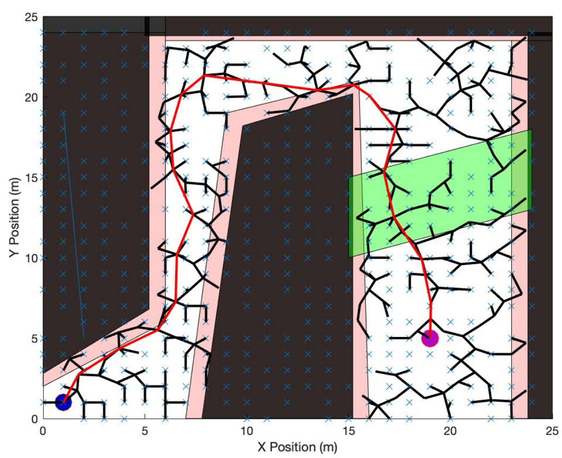

Fig. 6. Example of RRT* Path Generation

IV. Multi-Rover Coordination Using RRT*

A. Method

A simulation has been developed representing five rovers. In this study, a larger group of rovers has not been considered as financial and mass budgetary constraints severely limit the number of rovers which could be transported to a mission location. Here, the team of rovers is directed by a central planner which issues guidance commands and carries out coordination for the rover team. This central planner could be contained within a lander, orbiter or a larger 'parent' rover.

A centralised, decoupled architecture has been selected for this research. By decoupling, each rover is treated as an individual component [6]. Decoupled methods have lower complexity and are more scalable, at the cost of the optimality of the solution. This architecture facilitates the use of one central planner and multiple rovers. In order to make a multi-rover exploration mission feasible, the cost and computational load of each rover should be minimised. A centralised approach to the rover group's coordination and monitoring reduces the expense of separate coordination systems for each individual rover. Furthermore, no communication takes place between rovers, all communications are sent via the central planner. Two centralised, decoupled methods of multi-robot coordination are implemented: *fixed path coordination* and *prioritised planning*.

1) Fixed Path Coordination: Using fixed path coordination, each rover's path is initially planned using RRT* - without any consideration of the other rovers. During operation, collisions are avoided by treating rovers as dynamic obstacles. The algorithm is designed such that rovers are given an arbitrary operational priority, where lower priority rovers slow down to give way to higher priority rovers if a collision risk is detected.

Equation (4) shows the desired velocity calculation, V_{desired}, for any given rover. $V_{\text{environment}}$ is the velocity assigned to the current terrain, e.g., the environmental velocity of the rock field is less than that of the smooth regolith, to allow rovers to safely avoid obstacles. Reduction Factor represents the velocity reduction required in order to give way to a higher priority rover.

$$V_{desired} = V_{environment} - ReductionFactor.(0.025) \quad (4)$$

For each higher priority rover within 1.2m of a given rover, the reduction factor of the given rover is incremented by 1. This multi-level velocity reduction approach ensures that coordination can be achieved in cases where more than two rovers are at risk of colliding. A second stage of velocity reduction is implemented in order to further reduce collisions in high-risk situations, where rovers are within 0.8m of each other. In this case, for each higher priority rover within 0.8m of a given rover, the reduction factor of the lower priority rover is incremented by 2. The addition of this second velocity reduction stage was found to significantly reduce the number of collision occurrences between rovers. If re-coordination is successful and no collisions occur, rovers return to normal operation with a reduction factor of zero. However, if any rovers are measured to be within 0.35m (a rover length) of each other, they are deemed to have crashed. Crashed rovers are halted and added to an array to be treated as static rover obstacles.

2) Prioritised Planning: Using the prioritised planning algorithm, each rover is given an arbitrary priority. The highest priority rover's path is planned first. The algorithm then attempts to plan a path for the second rover, comparing the positions of both rovers at each timestep to check for potential collisions. If collisions are detected, another path planning attempt is made for the second rover. This process repeats until the second rover's path is deemed 'safe'. All subsequent rover paths are then calculated one at a time, in order of descending priority and treating higher priority robots as obstacles at each time step, until all five rover paths are deemed ‘safe’. Once all paths have been planned and evaluated as safe from collisions, the full rover simulation can be run. During the full simulation run, rovers may deviate from their planned paths in order to avoid static obstacles. Any rovers which experience a collision are halted.

B. Results

The primary factors used to evaluate the coordination methods are the length of time taken to plan the group’s paths and the number of collision occurrences. Path planning time is given less weight during evaluation, as coordination which delivers ‘safer’ operation but takes longer to calculate is still a preferable outcome. Collisions which occur during a test run can be either static (between a moving rover and a static object) or dynamic (between two or more moving rovers). To ensure a fair comparison of the two coordination methods, certain factors are kept consistent throughout each of the test sets i.e. environmental factors, such as the position of static obstacles, and the environmental velocity for each terrain type. Each rover retained its same, distinct start and goal waypoints. Similarly, all five rovers were utilised for each of the test runs. Each of the three test sets involved running the path planning and coordination algorithm ten times.

Fig. 7, 8 and 9 each show a set of five rover paths, planned using the RRT* algorithm, representing an average outcome of their respective test sets. Each waypoint is marked by a cross, each rover’s planned path is marked in black and each rover’s measured path is marked as a red line. Collisions between rovers are marked by a blue circle at the crash location.

A control test set was first run, where each rover path was planned without the use of a coordination algorithm. Similarly, no coordinating action was taken during path traversal. This data acts as a baseline for comparison of the coordination method tests. Fig. 7 shows an example test run from this test set, where all rovers experience dynamic collisions (marked by blue circles). The average total path planning time for the uncoordinated rovers was 32.65 seconds.

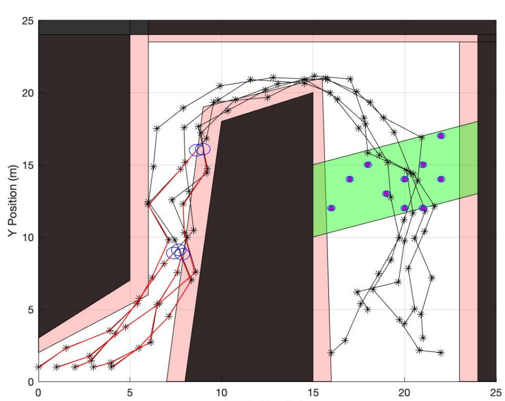

Fig. 7. Rover Paths Using No Coordination

The first coordination method to be tested was fixed path coordination. The average total path planning time for this algorithm was 42.11 seconds. The average number of collisions per test run was 1.9. Fig. 8 shows an example test run, where two rovers experience a dynamic collision and one rover experiences a static collision.

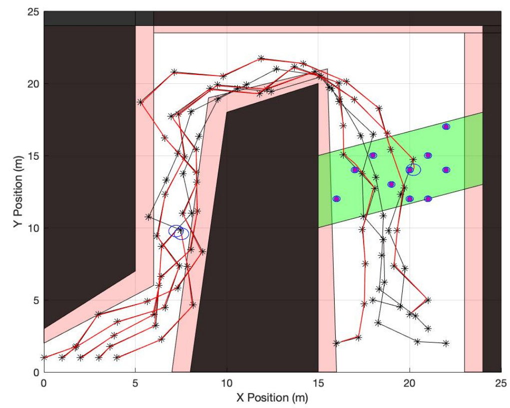

Fig. 8. Fixed Path Coordination

During the prioritised planning test set, the average path planning time was 406.08 seconds. The average number of collisions per test run was 0.6, where each collision was with a static obstacle i.e. no dynamic collisions took place. Due to the nature of the prioritised planning coordination method, the possibility of dynamic collisions between rovers on their ideal paths is essentially eliminated. As collisions are not tolerable in such a scenario, better obstacle avoidance algorithms are required to mitigate static collisions. The number of path planning attempts required before a safe path is found for each rover increased linearly with the number of rovers in the group. This suggests that the algorithm may not be scalable to a large number of rovers. Fig. 9 shows an example test run.

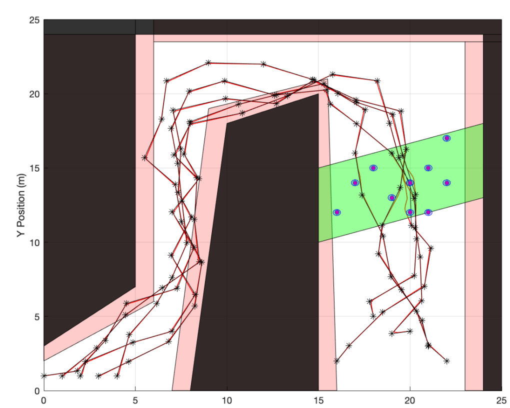

Fig. 9. Prioritised Planning

V. Discussion

Comparing the average path planning time of each coordination method to that of the control tests, fixed path coordination performs similarly, while prioritised planning takes significantly longer. The similarity between the control and fixed path coordination methods is expected as neither method carries out coordination during the path planning stage. The time taken by the prioritised planning method can be attributed to the path planning attempts required in order to successfully coordinate paths.

Fig. 10 shows the collision data gathered from the three test sets. The control tests exhibited the fewest static collisions. However, this can be attributed to a large proportion of rovers experiencing dynamic collisions early in the simulation. There was no significant difference between the number of static collisions incurred by each of the two coordination methods as both utilise the same static obstacle avoidance algorithm. The largest performance difference between the three tests sets can be seen in the average number of dynamic collisions: 4.1 in control tests, 1.1 in fixed path coordination, 0.0 in prioritised planning. These results show that both coordination methods show a significant increase in the performance of the rover group when compared to the control tests. Prioritised planning has been selected as the best performing algorithm due to its measured reduction of rover collisions. The time taken to plan paths using this method has been deemed acceptable.

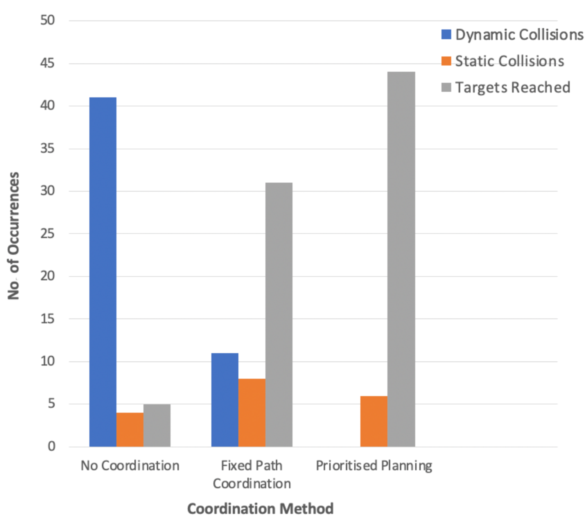

Fig. 10. Comparison Of Coordination Outcomes

VI. Conclusions

The use of multiple rovers could significantly extend the capabilities of a planetary exploration mission. The ability to safely coordinate such rover teams in remote and hazardous locations is critical. To achieve this, a path planner and coordination system for a small group of planetary exploration rovers has been proposed within this work.

A RRT* path planner has been implemented, which allows the system to search an environmental model for a rover's target point and find a least cost path to that point. This algorithm was first applied to a single rover, where it allowed the rover to avoid high risk terrain regions and evaluate a low-cost path to its target.

RRT* has also been found to be appropriate for application to a multi-rover group, as the random nature of the algorithm provides rovers with distinct paths, reducing the risk of inter-rover collisions. Two coordination algorithms have been developed and evaluated: fixed path coordination and prioritised planning. Fixed path coordination carries out all coordination during group operations, whereas prioritised planning carries out all coordination during the planning stage. A comparison of the two algorithms shows that, while the prioritised planning algorithm takes significantly longer to evaluate safe paths for each rover, it incurs far fewer inter-rover collisions. As minimising the loss of rovers is the prime objective of this coordination system, prioritised planning has been selected as the more appropriate algorithm for this application.

A coordination algorithm is essential for multi-vehicle planetary exploration systems as uncoordinated rovers could cause collisions which incur large consequences, both in terms of missions success and financial loss. In this work, the use of a central planner which carries out path planning, using RRT*, and coordination, using prioritised planning, has been found to significantly reduce the risk of inter-rover collisions for a small group of rovers in a simulated environment.

References

[1] Muirhead B.K. Mars Rovers, Past and Future. *IEEE Aerospace Conference Proceedings*. 2004;1:134

[2] Farley K, Williford K, Stack K, Chen A, Torre M, Hand K, et al. Mars 2020 Mission Overview. *Space Science Reviews*. 2020;216:142

[3] LaValle S.M. Rapidly-exploring random trees : a new tool for path planning. *The Annual Research Report*. 1998

[4] Karaman S, and Frazzoli E. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*. 2011;30:846–894

[5] Noreen I, Khan A, and Habib Z. A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms. *UCSNS*. 2016;16(10):20-27

[6] Parker L. Path Planning and Motion Coordination in Multiple Mobile Robot Teams. *Encyclopedia of Complexity and System Science*. 2009

[7] Čáp M, Novák P, Kleiner A, and Selecký M. Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots. *IEEE Transactions on Automation Science and Engineering*. 2015;12(3):835-849

[8] Worrall K.J. Guidance and Search Algorithms For Mobile Robots: Application and Analysis Within The Context of Urban Search and Rescue [PhD Thesis]. Glasgow, Scotland. *University of Glasgow*. 2008

[9] Swinton S. Fault Tolerant Coordination of Multiple Rovers For Planetary Exploration [MEng Dissertation]. Glasgow, Scotland. *University of Glasgow*. 2021

[10] Hallock Smith G, Hagerott E.C., Scherr L.M., Herkenhoff K.E., Bell III J.f. Optical Designs For The Mars '03 Rover Cameras. In Proceedings of SPIE 4441, Current Developments in Lens Design and Optical Engineering II, San Diego, USA; 2001

[11] Rothrock B, Kennedy R, Cunningham C, Papon J, Heverly M, and Ono M. SPOC: Deep Learning-based Terrain Classification for Mars Rover Missions. *AIAA Space*. 2016

[12] Breivik M. Nonlinear Maneuvering Control of Underactuated Ships [Dissertation]. Trondheim, Norway. *Norwegian University of Science and Technology*. 2003