Terrain-based Path Planning and Following for an Experimental Mars Rover

Steven Potiris[1], **Anthony Tompkins**[2], **Ali Haydar Göktoğan**[3]
Australian Centre for Field Robotics[3]
The School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney, Australia
[spot2503[1], atom7176[2]]@uni.sydney.edu.au, a.goktogan@acfr.usyd.edu.au[3]

Abstract

Experimental Mars Rover (EMR)-Mawson is a 6 wheeled rover designed and operated on the Mars Yard of the Powerhouse Museum in Sydney, Australia. The Mars Yard terrain is a physical analogue carefully constructed to simulate the surface of Mars. EMR-Mawson is teleoperated by researchers and students from distant locations. There are times that intended missions need autonomous motion of the rover in difficult terrain. This paper presents the path planning, localisation and control system developed and validated for the rover, and preliminary results of an investigation into terrain-roughness-based control. Experimental results demonstrate accurate and repeatable control using different control parameters on the most challenging regions of the Mars Yard that include traversing wheel-sized rocks, and steep slopes with loose terrain in a crater.

Figure 1: Experimental Mars Rover-Mawson in the Mars Yard.

1 Introduction

In this paper we present a supervised autonomy system called Rover Autonomy System (RAS) and conduct a preliminary investigation in to the relation between path tracking accuracy and terrain roughness. RAS is a supporting system for a nonholonomic, wheeled rovers and is designed to deal with the path-planning and path following aspects of Experimental Mars Rover (EMR) operations. It is being used onboard the EMR-Mawson operating in the Mars Yard (Fig. 1).

EMR-Mawson can be operated either locally from the laboratory section of Mars Yard or teleoperated from remote locations via internet. Both the Mars Yard and EMR-Mawson are being actively used by university researchers. Under supervision of the expert operators, young students from primary and secondary schools also use these resources for their science, technology, engineering and mathematics (STEM) education by teleoperating EMR-Mawson from their classrooms.

In planetary rover missions, scientifically important features are often found in difficult to reach and dangerous geological formations. Teleoperation of rovers on such regions is associated with high risk of mission failure. The presented system increases the operational safety of EMR-Mawson by accurately guiding it over desired complex paths that avoid untraversable terrain and minimise exposure to unfavourable terrain, thus offers greater flexibility for operators to access hard to reach but scientifically interesting locations in the Mars Yard. We define "untraversable terrain" to be terrain with features that are higher than one wheel diameter (15cm), and "unfavourable terrain" to be terrain with slopes more than $\pm 10^{\circ}$ or features with height between one-fifth and one wheel diameter (3cm-15cm). EMR-Mawson is capable of being controlled by issuing primitive motion commands such as DRIVE and TURN through a teleoperator interface to affect position and heading respectively. Additionally, low-level actuation to affect an individual actuator can be commanded via STEER and DRIVE commands when specifying an individual actuator ID number. The presented system extends the rover's capabilities by adding a layer of autonomy to the rover, providing easily configurable Guidance, Navigation and Control (GNC) subsystems.

RAS uses an *a priori* point cloud and occupancy grid to first characterise the terrain and then plan a path using a terrain-based cost function. The occupancy grid ensures EMR-Mawson does not collide with large, static obstacles and the terrain-based cost function ensures that the rover takes the path that minimises the rover's exposure to unfavourable terrain. The rover actuates itself along the path using Ackermann steering guided by a vision-inertial localisation system providing pose measurements. Simultaneously, rover's control system can track a nominal speed which provides additional robustness against becoming immobile from traversing rocks or sinkage in to loose terrain.

The system can safely determine and accurately track a path through an ordered set of waypoint while keeping within 1cm/s of a nominal velocity even when overcoming slopes and wheel-sized rocks. The positional and angular error between the rover and the path is evaluated across three test paths chosen to demonstrate the performance of the path follower over different aspects of the terrain.

This paper first briefly reviews previous works in terrain-based path planning and control for planetary rovers in 2, then introduces EMR-Mawson and the Mars Yard in Section 3. The autonomous capabilities that RAS introduces to EMR-Mawson is stated in Section 4. The terrain-based path planning methods are described in detail in Section 5. We then introduce the visual-inertial localisation system onboard EMR-Mawson in Section 6. In Section 7 the control laws and strategies used to guide EMR-Mawson along a desired path is detailed. The results of path-following experiments are presented and discussed in Section 8, and finally in Section 9 conclusions are drawn about the effectiveness of RAS.

2 Previous Works

Digital Terrain Maps (DTMs) have been utilised in previous works to characterise terrain geometry in terms of roughness [Hoffman and Krotkov, 1989]. More recent works have used DTMs to create coarse grid-based [Tahirovic *et al.*, 2013] and cell-based [Rekleitis *et al.*, 2007] roughness maps for path planning for planetary rovers. In these works, high-resolution DTMs have a resolution of 1 point per m^2 and are discretised over a grid.

[Morales *et al.*, 2013] demonstrate a pure pursuit path tracker for a nonholonomic vehicle on a homogeneous flat, hard terrain. Control effort and positional error is evaluated in tracking a path, but no evaluation on angular and speed errors is given. The authors report their lowest positional error to be 10.21cm when travelling at 10cm/s over homogeneous terrain. [Martin and Corke, 2011] present a similar method of path planning using a dynamic model and inferred terrain characteristics, and conclude that terrain shape and ground properties are important when planning high-speed (5 m/s) paths.

We adapt and expand on our prior work in guidance and control for a Mars Rover [Potiris, Tompkins and Göktoğan, 2013] in which we used a holonomic steering controller to follow a trajectory rather than a path.

[**?**] presents a method of rough terrain control for a 4-wheeled planetary rover using a terramechanics solution.

Using these previous works as motivation we present a terrain-based path planner and follower for a Mars rover that takes advantage of a high resolution point cloud to accurately evaluate the 3D terrain properties over a simplified 2D plane.

We evaluate the planned paths against user-generated paths by tracking them with a pure-pursuit path follower that simultaneously tracks a nominal velocity over heterogeneous terrain in the presence of obstacles, in order to determine if there is a relationship between terrain roughness, pursuit distance and path tracking accuracy. The results will be used to motivate future work.

3 The Experimental Mars Rover and The Mars Yard

The Mars Lab is an active laboratory in a public space inside of the Powerhouse Museum, Sydney, Australia. It has two main sections: the Mars Yard, where EMR-Mawson is being operated to perform research and education missions, and the control room, which houses the network infrastructure and is where researchers and students can observe the rover. In addition EMR-Mawson can be teleoperated from an internet connection via onboard and ceiling-mounted cameras.

3.1 The Mars Yard

The Mars Yard is a simulated Martian terrain that contains similar obstacles and hazards that would be found on the Martian surface such as traversable and untraversable rocks, a crater, hills and loose rocks. It also has features of scientific interest including a meteorite, a number of hematites, a gypsum deposit in the crater as well as modern and ancient stromatolite samples. A topographical map of the Mars Yard

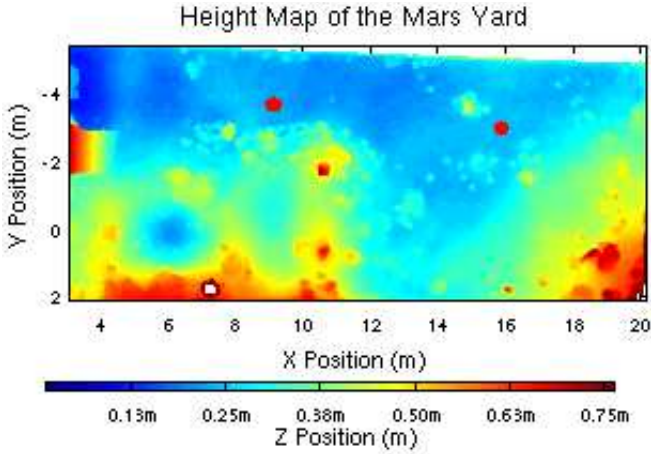


Figure 2: Topographical map of the Mars Yard generated from a point cloud.

terrain is presented in Fig. 2 . As it is shown in this map, the

Mars Yard has relatively flat regions in which EMR-Mawson can be operated without concern for the stability and saftey of the rover. However it also has untraversable terrain that is blocked by large rocks, as well as unfavourable terrain such as the steep slopes of the crater. Unfavourable terrain should be avoided as the wheels of the rover may slip causing it to dig into the ground and become stuck; such a situation caused NASA's Spirit rover to become immobile in 2009 while it was passing by a similarly sized crater.

3.2 EMR-Mawson

EMR-Mawson (Fig. 1), which is named after Australian Antartic explorer Sir Douglas Mawson , is designed and developed to be operated as a robotic explorer in the Mars Yard.

EMR-Mawson carries a variety of sensors for research and mobility purposes: a stereo-vision camera pair, an inertial measurement unit and a localisation sensor, as well as multiple monocular cameras for teleoperation. The rover has a footprint of approximately 66×78cm and weighs almost 30kg. Its light weight allows it to travel over small obstacles and slopes easily. Similar to NASA's Mars Exploration Rovers (MERs) , EMR-Mawson is a 6 wheeled platform with rocker-bogie mechanism. This mechanism assures all wheels in contact with the ground even when the rover drives over obstacles. The rocker-bogie mechanisms on both sides of the rover are linked with each other via a differential gear system in the body. This passive system improves the rovers stability by minimising the change in pitch angle when traversing over rough terrain. Using the rocker-bogie mechanism, EMR-Mawson can traverse terrain up to 25° in slope and obstacles up to 12cm high.

Each wheel on EMR-Mawson has its own driving and steering actutors, enabling them to be steered and driven independently. Independent wheel control offers greater maneuverability for the rover by allowing it to be driven forward, backwards, sideways or diagonally.

When EMR-Mawson is being teleoperated it can be driven at speeds of up to 45cm/s, and when being autonomously controlled at speeds of up to 15cm/s; in comparison the NASA MERs have a top speed of 4.6cm/s and are autonomously controlled at less than 1cm/s.

4 Autonomous Driving Modes

RAS gives EMR-Mawson two autonomous capabilities: *directed driving*, in which the rover drives through an ordered, user-defined set of waypoints, and *guided driving*, in which the rover calculates and follows a path from the robot position to each successive waypoint.

In both directed and guided driving modes the rover tries to achieve a user-defined nominal speed. When the rover is autonomously driving the system adjusts the wheel speeds to compensate for additional forces that are hard to model such as wheel slip and driving over traversable obstacles.

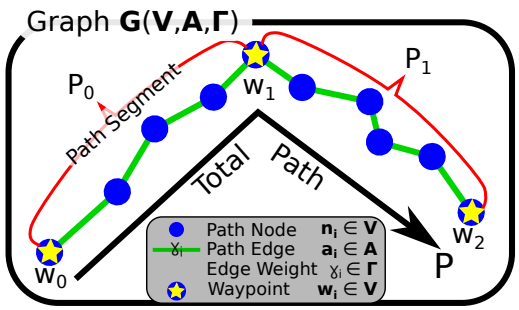


Figure 3: The total path is devided into path segments that lie between waypoint and are composed of multiple path nodes that are joined by edges.

Users can interact with the autonomy system via a text-based interface hosted onboard the EMR-Mawson computer by issuing a PLAN, GOTO or GDRIVE (guided-drive) command. The GOTO and PLAN commands accept an ordered list of comma-delimited X-Y coordinates where each coordinate pair corresponds to a waypoint. After the waypoints have been loaded, and the path has been calculated, the path-follower can be started with the GDRIVE command. Parameters or inputs to the path follower such as the nominal speed, wheel steering speed, actuator update rate and calculated path can be queried or specified using the GET and SET commands with the appropriate parameter name and value. In addition, the interface offers access to the sensor readings including the pose calculated from the localisation system.

5 Path Planning

In order to control the rover to pass through a set of user-defined position waypoints $\mathbf{W} = \{\mathbf{w_1}, \mathbf{w_2}, ...\}$ there must be a safe, collision-free path calculated from its initial position $\mathbf{w_0}$, to each successive waypoint.

To find the best path for the EMR to follow we use a graph representation [Bollobas, 1998] of points called *nodes* which associate terrain information with positions in the continuous R^2 X-Y plane spanned over the Mars Yard. Each node may be connected to multiple other nodes by directed edges, which indicate traversibility between two nodes in the direction specified by the edge. Each edge also has an edge weight which quantifies the cost from moving across the edge. A path node $\mathbf{n_i}$ associates a position $\mathbf{p_i}$ with terrain characteristics c_i^Q through the subscript index i, where the superscript Q can refer to various quantities such as S or R, respectively the slope or roughness of the terrain. c_i^Q is calculated offline and is used in the cost function (5) . A waypoint $\mathbf{w_j}$ is a path node except that is user-specified and its terrain characteristics are calculated at run-time.

A directed graph $\mathbf{G}(\mathbf{V}, \mathbf{A}, \mathbf{\Gamma})$ is created that will be solved to get the desired path. The elements in the adjacency list $\mathbf{A} = \{\mathbf{a_0}, \mathbf{a_1}, ...\}$ are the directed edges formed from a given node $\mathbf{n_j}$ to $\mathbf{n_k}$ and denoted $\mathbf{a_i} = (\mathbf{n_j}, \mathbf{n_k})$. The table $\mathbf{\Gamma}(\mathbf{a_i})$

maps the edge $\mathbf{a_i}$ to the edge weight γ_i.

For a guided drive, a path segment lies between each successive waypoint and consists of multiple path nodes. The ordered sequence of path segments is the path $\mathbf{P} = \{\mathbf{P_0}, \mathbf{P_1}, \mathbf{P_2}, ...\}$, where $\mathbf{P_i}$ is the path segment from $\mathbf{w_i}$ to $\mathbf{w_{i+1}}$. Fig. 3 illustrates the structure and relationship between the path, waypoints and the path segments that join them.

The directed drive command takes as input a user-generated path that consists of only waypoints and has no intermediate path nodes between waypoints; this method of path planning relies on the user to provide valid, safe waypoints and does not consider the terrain characteristics. The guided drive command calculates a path using a Probabilistic Roadmap (PRM) based A* graph-solver to generate path segments between each user-specified waypoint by minimising the terrain-based cost function; this reduces the chance of rover failure by avoiding steep slopes and untraversable rocks.

5.1 Offline Planner Initialisation

This section describes the calculation of a path passing through each element of $\mathbf{W}$ in order, using terrain characteristics to minimise the traversal of undesireable terrain and obstacle collisions. Although great care is taken to keep the Mars Yard terrain as static as possible after each experiment, there is significant discrepancy between the *a priori* data and the physical state of the terrain.

We take advantage of the *a priori* maps by doing most of the calculations offline; however the elements of $\mathbf{W}$ cannot be known *a priori* therefore some of the calculation of the $\mathbf{P}$ is performed at run time in the following section.

Firstly, the set of nodes $\mathbf{V}$ are randomly generated across the unoccupied areas of the Mars Yard, then and edges $\mathbf{A}$ are formed by connecting traversible paths. The terrain data is then used to calculate the known cost between connected edges. In our implementation we use $|\mathbf{V}| = \mathbf{500}$ and $k = 10$. The *a priori* occupancy grid is queried by the function $D(\mathbf{p_q})$ that returns whether or not the position $\mathbf{p_q}$ is occupied by an untraversable obstacle.

A PRM is constructed by generating a number of randomly positioned nodes across the unoccupied space in the Mars Yard. The position $\mathbf{p_q}$ is randomly generated for each candidate node $\mathbf{n_q}$ and the occupancy grid is queried by evaluating $\mathbf{D(p_q)}$; if $\mathbf{p_q}$ is unoccupied then $\mathbf{n_q}$ is a in a valid position and is added to the set of nodes $\mathbf{V}$. This process is repeated until the $\mathbf{V}$ holds the desired amount of nodes.

Each node in $\mathbf{V}$ is then connected to a maximum of k candidate neighbours by an edge. If the edge joining the node to any of its candidate neighbours does not pass through any occupied areas then they are considered adjacent and the edge is added to the adjacency list $\mathbf{A}$. A sample PRM is shown in Fig. 4 with a reduced number of nodes for illustration. Next the terrain characteristics for each node is determined;

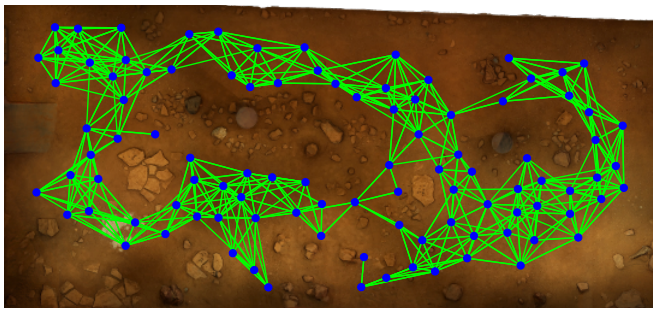


Figure 4: A sample PRM of the Mars Yard with 100 nodes and a maximum of 5 edges per node. Our implementation uses $|\mathbf{V}| = \mathbf{500}$ and $k = 10$.

this is accomplished by fitting a plane to a set of points local to the node position [Hoffman and Krotkov, 1989]. For each node $\mathbf{n_i}$, the points that are enveloped by a circle on the $X-Y$ plane centered at $\mathbf{p_i}$ are used to calculate the local terrain characteristics of $\mathbf{n_i}$. The unordered set of points within this circle is stored in matrix $\mathbf{S_i}$ of size $[N \times 3]$, where N is the amount of points in $\mathbf{S_i}$. $\bar{\mathbf{r}}_\mathbf{i}$ is the centroid of $\mathbf{S_i}$ and is calculated as the mean of each Euclidean component of $\mathbf{S_i}$. The height of the terrain local to $\mathbf{p_i}$ is then given by the Z component of $\bar{\mathbf{r}}_\mathbf{i}$. Using orthogonal residual regression, a plane $\hat{\mathbf{S}}_\mathbf{i}$ that passes through the centroid $\bar{\mathbf{r}}_\mathbf{i}$ is fit to the points in $\mathbf{S_i}$. The equation of the local plane passing through $\bar{\mathbf{r}}_\mathbf{i}$ is represented by (1) , where $\hat{\mathbf{n}}_\mathbf{i}$ is the unknown normal vector of the plane, and $\mathbf{v}$ is a point in $\mathbf{S_i}$. The residuals between the points in $\mathbf{S_i}$ and the centroid $\bar{\mathbf{r}}_\mathbf{i}$ are given by equation (2) where $\mathbf{S_i} \ominus \bar{\mathbf{r}}_\mathbf{i}$ is the subtraction of $\bar{\mathbf{r}}_\mathbf{i}$ from all columns of $\mathbf{S_i}$. $\hat{\mathbf{n}}_\mathbf{i}$ is given by equation (3) and is the value of $\hat{\mathbf{n}}_\mathbf{q}$ that minimises the sum of squares of the residuals for all $\mathbf{v}$ in $\hat{\mathbf{S}}_\mathbf{i}$. $\hat{\mathbf{n}}_\mathbf{i}$ can be found by solving equation (4) for the normalised eigenvector $\bar{\mathbf{v}}$ that corresponds to the smallest strictly positive eigenvalue λ.

$$(\mathbf{v} - \bar{\mathbf{r}}_\mathbf{i}) \cdot \hat{\mathbf{n}}_\mathbf{i} = \mathbf{0} \tag{1}$$

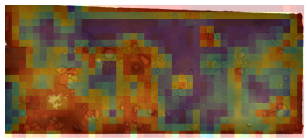
$$\bar{\mathbf{S}}_\mathbf{i} = \mathbf{S_i} \ominus \bar{\mathbf{r}}_\mathbf{i} \tag{2}$$

$$\hat{\mathbf{n}}_\mathbf{i} = \min_{\hat{\mathbf{n}}_\mathbf{q}} \sum_{\mathbf{v} \in \mathbf{S_i}} (\mathbf{v}^\top \mathbf{v}) \cdot \hat{\mathbf{n}}_\mathbf{q} \tag{3}$$

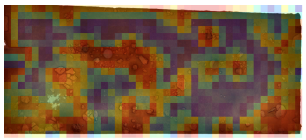
$$\bar{\mathbf{S}}_\mathbf{i}^\top \bar{\mathbf{S}}_\mathbf{i} \bar{\mathbf{v}} = \lambda \bar{\mathbf{v}} \tag{4}$$

$$g(\mathbf{n_i}, \mathbf{n_j}) = ||\mathbf{p_j} - \mathbf{p_i}||(\frac{\mathbf{1}}{\alpha} + {}^{60}\mathbf{c}_\mathbf{j}^\mathbf{S}\, {}^{60}\mathbf{c}_\mathbf{j}^\mathbf{R}\, {}^{40}\mathbf{c}_\mathbf{j}^\mathbf{R}) \tag{5}$$

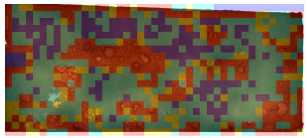
$\hat{\mathbf{n}}_\mathbf{i}$ can alternatively be determined by using singular value decomposition [Golub and Kahan, 1965] [Mandel, 1982] of $\bar{\mathbf{S}}_\mathbf{i}^\top \bar{\mathbf{S}}_\mathbf{i}$ and choosing the singular vector with the smallest singular value. The first and second components of $\hat{\mathbf{n}}_\mathbf{i}$ are the slope of the local plane in the x and y directions. The edge weights $\mathbf{\Gamma}(\mathbf{a_i})$ are evaluated using the cost metric (5) , where α is the function sensitivity to terrain characteristics, ${}^{60}c_j^S$ is the slope of the terrain evaluated in a 60cm diameter circle centered at the position $\mathbf{p_j}$ and similarly ${}^{60}c_j^R$, ${}^{40}c_j^R$ are the roughness of the terrain evaluated in 60cm and 40cm diameter circles respectively. Our implementation uses a terrain



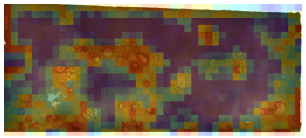
(a) Normalised slope calculated over 60 × 60cm grid



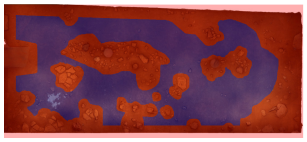
(b) Normalised roughness calculated over 60 × 60cm grid



(c) Normalised roughness calculated over 40 × 40cm grid



(d) Combined cost calculated over 60 × 60cm grid



(e) Occupancy map over 1×1cm grid

Figure 5: Visual representation of the cost maps involved in the path planning: red areas indicate high cost of traversibility, blue areas indicate less cost; the occupancy grid is a boolean map that represents untraversibility in red.

sensitivity of $\alpha = 3$; a high value of α signifies that the planner should take longer routes to avoid slopes and rough terrain. To visualise the various quantities in the cost function Fig. 5 shows a grid-based representation of the quantities evaluated over the Mars Yard.

5.2 Online Path Planning

The online calculations involved in the path planning process enable the path planner to reach the waypoints in $\mathbf{W}$ that are not in the initialised set $\mathbf{V}$ by temporarily adding the waypoints to $\mathbf{V}$, adding the appropriate edges to $\mathbf{A}$ and calculating the edge weights for the new edges using the terrain characterisation methods detailed in Section 5.1 and the cost function in equation (5) . The minimum cost path over graph $\mathbf{G}(\mathbf{V}, \mathbf{A}, \mathbf{\Gamma})$ is solved between each succesive waypoint in $\mathbf{W}$ using a directed, weighted-graph A-STAR (A*) solver. The results are concatenated into the ordered list $\mathbf{P}$, the path to be followed by the control system path follower.

This entire process is outlined in Algorithm 1, where $knn(\mathbf{w}, \mathbf{V}, \mathbf{k})$ is a k-nearest-neighbours algorithim [Clarkson, 1997] that finds at most k nodes in $\mathbf{V}$ with position nearest to $\mathbf{w}$, and $AStar(\mathbf{G}(\mathbf{V}, \mathbf{A}, \mathbf{\Gamma}))$ is the A* solution of the graph $\mathbf{G}$ with vertices $\mathbf{V}$, adjacency list $\mathbf{A}$ and edge weights $\mathbf{\Gamma}$. The $AStar(\mathbf{G}(\mathbf{V}, \mathbf{A}, \mathbf{\Gamma}))$ algorithm returns an ordered list of nodes that compose the desired lowest-cost path.

Algorithm 1 Run-time Graph Update and Path Planning

function PLANPATH($\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{\Gamma}$)
  Append $\mathbf{w_0}$ to $\mathbf{V}$
  for $i = 0 \ldots (|\mathbf{W}| - \mathbf{1})$ **do**
    $\mathbf{Q} = \mathbf{knn}(\mathbf{w_i}, \mathbf{V}, \mathbf{k})$
    $\mathbf{R} = \mathbf{knn}(\mathbf{w_{i+1}}, \mathbf{V}, \mathbf{k})$
    Append $\mathbf{w_{i+1}}$ to $\mathbf{V}$
    for all $q \in \mathbf{Q} \notin \mathbf{W}$ **do**
      $\mathbf{a_q} \leftarrow (\mathbf{w_i}, \mathbf{q})$
      Append $\mathbf{a_q}$ to $\mathbf{A}$
      $\mathbf{\Gamma}(\mathbf{a_q}) = \mathbf{g}(\mathbf{w_i}, \mathbf{q})$
    end for
    for all $r \in \mathbf{R} \notin \mathbf{W}$ **do**
      $\mathbf{a_r} \leftarrow (\mathbf{r}, \mathbf{w_{i+1}})$
      Append $\mathbf{a_r}$ to $\mathbf{A}$
      $\mathbf{\Gamma}(\mathbf{a_r}) = \mathbf{g}(\mathbf{r}, \mathbf{w_{i+1}})$
    end for
    $\mathbf{P_i} \leftarrow AStar(\mathbf{G}(\mathbf{V}, \mathbf{A}, \mathbf{\Gamma}))$
    Append $\mathbf{P_i}$ to $\mathbf{P}$
  end for
  return $\mathbf{P}$
end function

6 Visual-Inertial Localisation

The localisation system used was developed in our previous work [Tompkins, Potiris and Göktoğan, 2013] and is summarised in this section. The localisation system uses three sensors: a monocular RGB camera and an IMU consisting of a 3-axis accelerometer and rate gyroscope. The camera is used to observe a constellation of fiducials on the ceiling, providing accurate, precise and low-frequency pose estimates of the rover with respect to the ground frame. Simultaneously, the IMU provides high-frequency inertial measurements that are subject to bias, causing drift in the pose estimate. Fusing the inertial measurements with the vision estimates provides a more accurate, drift-free and frequent pose estimate. The measurements from both devices are processed on board the EMR to extract pose information expressed in a ground frame defined in the Mars Yard. The vision system performs a series of operations Fig. 6 that enable the system to detect and localise a fiducial within a camera frame. In addition the system can extract from each fiducial image a unique identifier that is visually encoded into it. Each identifier is used to register a fiducial between observed 2D image-frame coordinates and an *a priori* model of the 3D ground frame. When at least 4 fiducials have been successfully detected and registered into the ground frame the position and orientation of the camera can be determined by using the intrinsic parameters of the camera and the registered coordinates to solve the PnP problem [Lepetit *et al.*, 2013]. Once the camera pose is determined the transformation between camera-frame and body-frame is used to determine the pose of the body with expressed in the ground frame. This process runs at 10Hz,

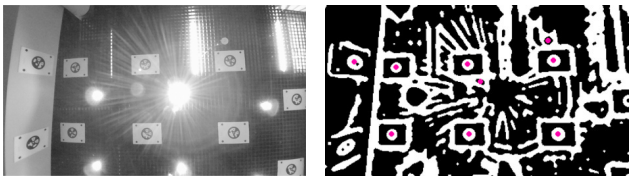


Figure 6: The beginning and end of the vision processing. (a) The image as captured from the ceiling-facing camera; barrel distortion and lens flare is apparent. (b) The processed image with the effects of the distortion and lens flare removed; the center of each fiducial is marked with a red dot to indicate a potential region of interest for identifier extraction.

providing highly accurate measurements on the EMR pose. When combined with the inertial system the pose update rate increases to 40Hz and increases both the accuracy of the system and its robustness to error due to fiducial occlusion.

7 Control System

Once a path has been computed it must be followed accurately through all the waypoints to the goal position. To achieve this we use a control strategy that aims to steer the rover along the path while keeping the rover heading tangential to the path, similar to how a car steers to follow a road. The control system uses the pose updates from the navigation system to steer the rover using an Ackermann steering controller to drive towards a pursuit point that is always ahead of the EMR and progressing along the path. Using the discrete derivative of the pose determined by the localisation system, the speed of the rover is estimated and compared to the nominal speed. The speed error is used in a proportional and integral (PI) controller to scale the wheel speeds. The loop-closure of velocity helps the system traverse over slopes, obstacles and reduces the risk of wheel sinkage. The various stages of the control system is shown in Fig. 7 .

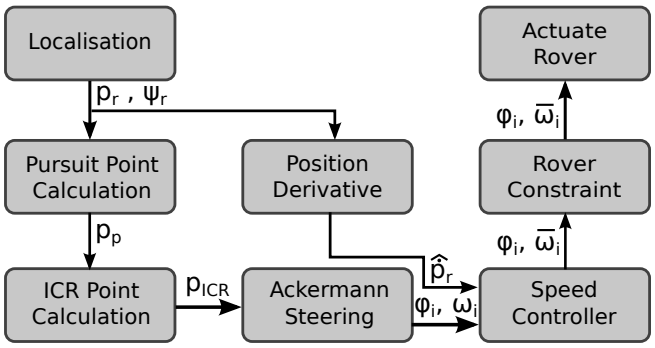


Figure 7: The individual stages involved in the control system.

7.1 Pursuit Point Selection

A pure pursuit [Morales *et al.*, 2013] strategy is used to identify a pursuit point $\mathbf{p_p}$ which the rover should drive towards at each control timestep; this is illustrated in Fig. 8 .

$\mathbf{p_c}$ is the point on the path that is closest to the measured rover position $\mathbf{p_r}$. Because $\mathbf{p_c}$ is found by the finding the minimum distance between each path edge and the rover position, overlapping path segments can cause the tracker to skip ahead or loop path segments. Tracking the EMRs progress through the path is essential to avoid this behaviour.

In order to determine where to place the pursuit point, the point on the path that is closest to the rover, $\mathbf{p_c}$, must first be found. Finding $\mathbf{p_c}$ involves a search through all consecutive line segments defined by a line starting at $\mathbf{P[i]}$ and ending at $\mathbf{P[i+1]}$, where $\mathbf{P[i]}$ is the position of the ith node of the ordered list of nodes $\mathbf{P}$ and $i = \{0 \cdots (|\mathbf{P}| - \mathbf{2})\}$. To keep track of the EMRs progress along the path we introduce an index Z indicating that the EMR is currently on the path edge defined by the line segment starting at $\mathbf{P[Z]}$ and ending at $\mathbf{P[Z+1]}$. Z has an initial value of 0 and at path completion will have a value of $|\mathbf{P}|-\mathbf{1}$. We also introduce the search window half-size s to reduce the search space of the algorithm from the line segments beginning at index $\{0 \cdots (|\mathbf{P}| - \mathbf{1})\}$ to $\{Z-s \cdots Z+s\}$. Searching for $\mathbf{p_c}$ in only a small window of nodes around the tracked index Z enforces a smooth progression along the path and allows $\mathbf{p_c}$ to recover in the case of noise in $\mathbf{p_r}$.

For each line segment starting at $\mathbf{P[i]}$ and ending at $\mathbf{P[i+1]}$, $i = \{Z-s, \cdots, Z+s\}$:

1. $\mathbf{r_{r,i}} = \mathbf{p_r} - \mathbf{P[i]}$ is the position of the EMR with respect to the start position of the line segment.
2. $\mathbf{r_{l,i}} = \mathbf{P[i+1]} - \mathbf{P[i]}$ is the position of the end of the line segment with respect to the start of the line segment.
3. t_i is the distance ratio along $\mathbf{r_{l,i}}$ on which $\mathbf{p_{c,i}}$ will lie, evaluated using equation (6) .
4. If $t \notin [0 \cdots 1]$ then the point that minimises the distance between the line and $\mathbf{p_r}$ does not lie on the line segment and the $d_{p,i} = \infty$. The remaining steps are not processed on this line segment.
5. $\mathbf{p_{c,i}}$ is calculated using equation (7) .
6. The square of the perpendicular distance $d^2_{p,i}$ is calculated using equation (8) .

The index that has the minimum value of $d^2_{p,i}$ is assigned to Z, which now holds the updated index corresponding to the closest path edge to the robot. $\mathbf{p_c}$ is updated with the value of $\mathbf{p_{c,Z}}$. The path tangent heading Ψ_c is then the heading of the vector $\mathbf{r_{l,Z}}$. Both $\mathbf{p_c}$ and Ψ_c are stored for the current timestep to be used in error calculations.

$$t_i = (\mathbf{r_{r,i}} \cdot \mathbf{r_{l,i}})/(\mathbf{r_{l,i}} \cdot \mathbf{r_{l,i}}) \tag{6}$$

$$\mathbf{p_{c,i}} = \mathbf{p_r} - \mathbf{r_{r,i}} + \mathbf{t_i r_{l,i}} \tag{7}$$

$$d_{p,i}^2 = (\mathbf{p_{c,i}} - \mathbf{p_{r,i}}) \cdot (\mathbf{p_{c,i}} - \mathbf{p_{r,i}}) \quad (8)$$

After $\mathbf{p_c}$ has been found the pursuit point $\mathbf{p_p}$ is calculated by advancing a distance d_p through the path from $\mathbf{p_c}$, where d_p is the pursuit distance. This is illustrated in Fig. 8 . Algorithm 2 outlines our implementation of this, which accounts for edge lengths shorter than d_p.

Algorithm 2 Pursuit Point Selection

```
function PURSUITPOINT(p_c, d_p, Z, P[0 ··· N])
    d ← d_p; i ← Z; p_p ← p_c
    while d > 0.00 do
        if i < N then
            u ← P[i + 1] − p_p
            U ← |u|
            if U ≥ d then
                p_p ← p_p + u d/U
                d ← 0.00
            else
                p_p ← P[i + 1]
                i ← i + 1
                d ← d − U
            end if
        else
            d ← 0.00
            p ← P[N]
        end if
    end while
    return p_p
end function
```

7.2 ICR Point Selection

In order to impose nonholonomic constrains on EMR-Mawson, we constrain the ICR point $\mathbf{p_{ICR}}$ to be along the middle wheel axle. A circle $\mathbf{C}$ is fit that intersects $\mathbf{p_r}$ and $\mathbf{p_p}$ and that is tangent to ψ_r. The center of $\mathbf{C}$ is $\mathbf{p_{ICR}}$ and is expressed in the body frame by equation (10) and illustrated in Fig. 8 . The ICR radius r_{ICR} is calculated using equation (9) and is the distance between the $\mathbf{p_r}$ and $\mathbf{p_g}$ in the y direction of the body frame. The minor arc of C from $\mathbf{p_r}$ to $\mathbf{p_p}$ is the arc which the rover will begin to actuate in the given timestep. When the rover has both small positional error with respect to $\mathbf{p_c}$, and heading error with respect to the path tanget at $\mathbf{p_c}$, then the ICR radius becomes asymptotically large causing the rover to drive in a straight line along the path tangent.

$$\mathbf{r_{ICR}} = \frac{\mathbf{d_p^2}}{\mathbf{-2}}(\cos(-\psi_\mathbf{r})(\mathbf{p_{g,y}} - \mathbf{p_{r,y}}) + \sin(-\psi_\mathbf{r})(\mathbf{p_{g,x}} - \mathbf{p_{r,x}})) \quad (9)$$

$$\mathbf{P_{ICR}} = \begin{pmatrix} 0 \\ r_{ICR} \end{pmatrix} \quad (10)$$

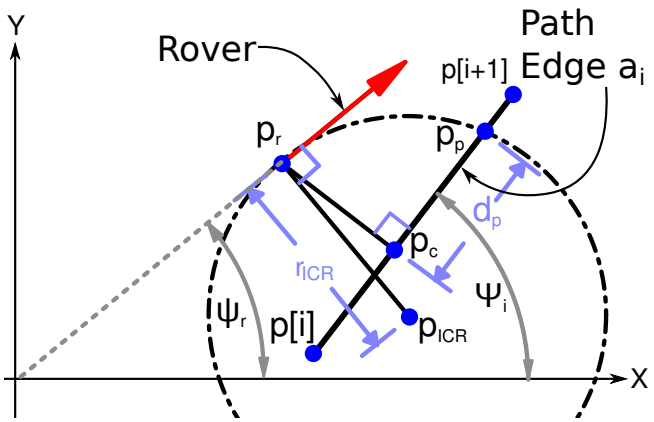


Figure 8: Pursuit point and ICR point selection geometry.

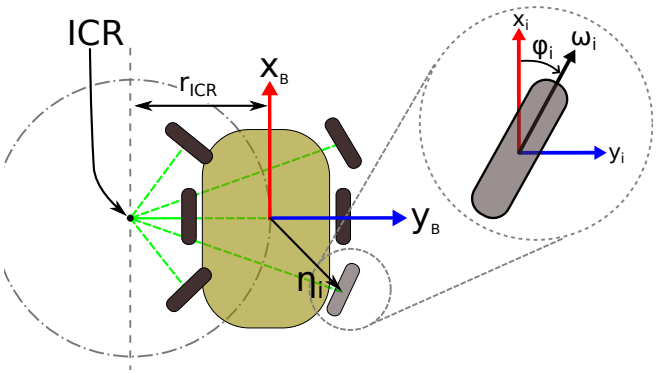


Figure 9: The ICR of EMR-Mawson for a sample wheel configuration. One of the wheel steering frames is shown to illustrate the steering angle and resultant wheel velocity. The offset of the ith wheel steering frame from the body frame is given by $\eta_\mathbf{i}$.

7.3 Ackermann Steering Controller

In order to actuate a rotation about the circle centered at $\mathbf{p_{ICR}}$ the horizontal plane wheel offsets from the body frame, the wheel x-axis orientation and the wheel radius; denoted $\eta_\mathbf{i}$, Φ_i and W_i respectively for wheel number $i = 1 \cdots 6$ must be known. The system calculates the steering angles ϕ_i and wheel speeds ω_i given by the respective equations (11) and (12) , where $\mathbf{p_{ICRx}}$, $\mathbf{p_{ICRy}}$, η_{ix} and η_{iy} are the x and y components of $\mathbf{p_{ICR}}$ and η_i respectively. These equations are evaluated for each wheel to achieve the ICR at a nominal tangential speed v_n; our implementation uses $v_n = 0.15$ m/s.

$$\phi_i = -atan2(\mathbf{p_{ICRx}} - \eta_\mathbf{ix},\ \mathbf{p_{ICRy}} - \eta_\mathbf{iy}) \quad (11)$$

$$\omega_i = \frac{||\eta_i - \mathbf{p_{ICR}}||\mathbf{v_n}}{||\mathbf{p_{ICR}}||\mathbf{W_i}} \quad (12)$$

7.4 Velocity Controller

Given the position $\mathbf{p_r}$ from the navigation system, the discrete derivative $\hat{\mathbf{p}}_\mathbf{r}$ is calculated using equation (13) where $\mathbf{p_{r,prev}}$

and ΔT are respectively the position measured during the previous timestep, and the time difference in seconds since the last timestep. The error between measured and nominal speed is given by ϵ_v and is the difference between the nominal speed v_n and the magnitude of the discrete derivative of the rover position $\dot{\mathbf{p}}_\mathbf{r}$. The wheel speed compensation factor Ω is given by (15) , where $\epsilon_{v,prev}$ is the value of ϵ_v at the last timestep, and K_p and K_i are the proportional and integral constants respectively; our implementation uses $K_p = 0.8$ and $K_i = 0.5$. The adjusted wheel speeds for each wheel i are then given by $\bar{\omega_i}$ (16) .

$$\hat{\dot{\mathbf{p}}}_\mathbf{r} = (\mathbf{p_r} - \mathbf{p_{r,prev}})/\mathbf{\Delta T} \tag{13}$$

$$\epsilon_v = v_n - ||\hat{\dot{\mathbf{p}}}_\mathbf{r}|| \tag{14}$$

$$\Omega = \epsilon_v(K_p + K_i\Delta t) - K_p\epsilon_{v,prev} \tag{15}$$

$$\bar{\omega_i} = \Omega\omega_i \tag{16}$$

7.5 Rover Mobility Constraints

In the final stage of the control system, the steering constraints of the rover are enforced. EMR-Mawson cannot actuate a steering angle $|\phi_i| > 90°$. If a steering angle is outside of the range $[-90, 90]°$ then it is rotated by $180°$ towards $0°$ and ω_i is negated. The effect of this is motion in the direction originally specified by ϕ_i, however, the wheel must rotate $180°$ every time ϕ_i crosses the $|\phi_i| = 90°$ boundary, causing unnecessary friction. It is noted that during Ackermann steering pure pursuit this generally does not occur unless paths involve turns much smaller than the EMR.

8 Experiments and Results

A series of both user-selected and planner-generated paths were followed using the path follower on EMR-Mawson. We define the error terms $\epsilon_\mathbf{p}$ and ϵ_θ and use the previous definition of ϵ_v in Section 7.4 to evaluate the path tracking performance. The positional deviation from the path is given by $\epsilon_\mathbf{p}$ which is the perpendicular distance between the rover position $\mathbf{p_r}$ and the closest point to the rover on the path, $\mathbf{p_c}$, which was calculated in Section 7.1. The heading error ϵ_ψ is the angle between the rover heading ψ_r and the heading of the path tangent at $\mathbf{p_c}$. We use the map Fig. 5 (b) (${}^{60}c_j^R$ evaluated over a grid) to qualify the terrain roughness of the paths. Each of these error terms are evaluated at each timestep during the execution of the path-follower. The results of these experiments are summarised in Table 1 .

A typical simulated science mission in the Mars Yard is shown in Fig. 10 : the user seeks out features of interest that could possibly be high-value science targets such as stromatolites, meteorites and a crater. The dashed lines are the designated paths for the user-generated (blue) and planner-generated (green) paths from the starting point to each subsequent features-of-interest: (a) stromatolite, (b) ancient stromatolite, (c) meteorite and (d) crater. The path passes through

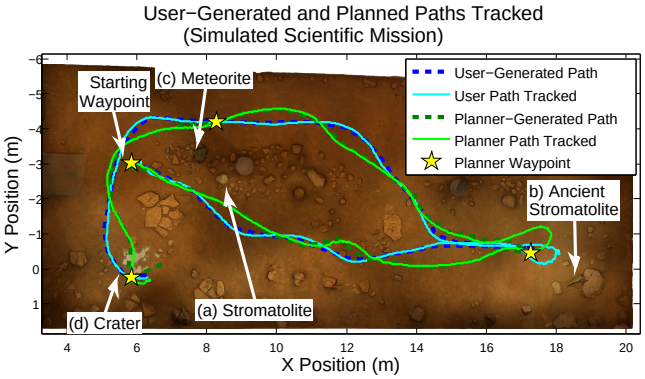


Figure 10: A typical simulated scientific mission in the Mars Yard.

areas of the Mars Yard with medium, then low terrain roughness. The solid lines are the paths actuated by the rover when following the respective paths using a pursuit distance of $d_p = 0.5m$. The measures of error of this path are evaluated and presented in Fig. 11 .

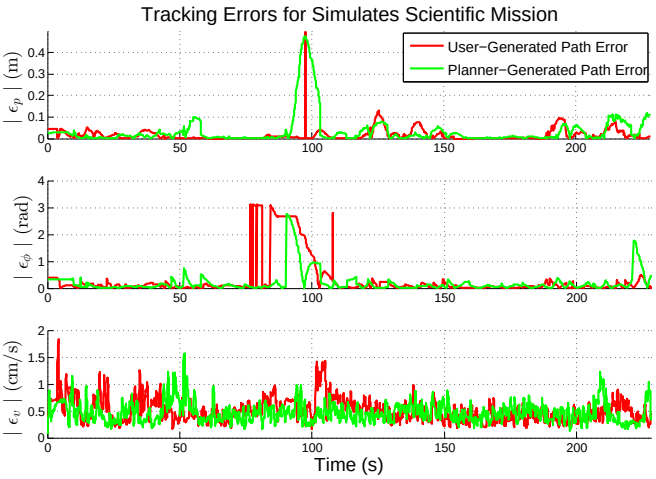


Figure 11: Position, heading and speed errors of the simulated scientific mission.

For the next experiment, EMR-Mawson follows a user-generated path across the roughest terrain in the Mars Yard using three different pursuit distances: $d_p = 30, 40$ and 50 cm. The path passes involves only low terrain roughness. The paths and tracking errors are shown in Fig. 12 and Fig. 13 respectively. The rover successfully traversed the paths with 50cm and 60cm pursuit distances smoothly and at a steady speed within 5% of the nominal 15cm/s.

The final experiment shows a single-waypoint planner-generated path across a flat section of the Mars Yard that contains few traversable obstacles and low terrain roughness. Similarly these paths are tracked using pursuit distances 30cm (green), 40cm (blue), 50cm (red) and are shown in Fig. 14 with tracking errors shown in Fig. 15 . It was observed that when actuating a planned path, the planner frequently

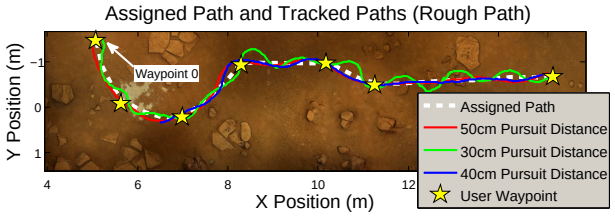


Figure 12: A user-generated path over rough terrain using a user-generated paths being tracked with three different pursuit distances: 30cm (green), 50cm (blue), 60cm (red).

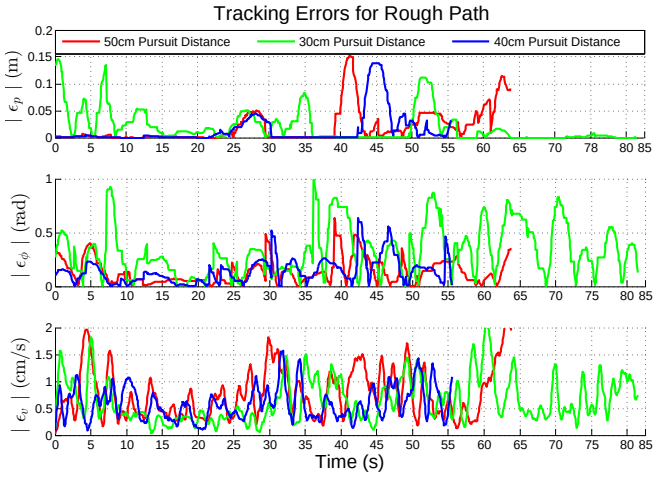


Figure 13: Position, heading and speed errors during tracking a rough path using pursuit distances: 30cm (green), 50cm (blue), 60cm (red).

would choose paths that involved overcoming traversible obstacles. The results show obstacle traversal in both Fig. 11 and Fig. 15 as small peaks across the position, heading and speed errors simultaneously.

In Fig. 11 a large peak is observed in position and heading errors at $t = 100s$; this corresponds to the "U-turn" that occurs around Fig. 11 (b) and is the result of sharply defined waypoints. To contrast, the single waypoint path defined in Fig. 14 shows much smaller positional and heading error. Both experiments shows spikes in heading error occasionally (e.g. Fig. 11 at $t = 75 \cdots 100$s) due to the non-continuous transition in heading between path edges.

As seen in Table 1 the flat and rough path experiments have local minima in ϵ_p, ϵ_v and ϵ_s simultaneously for the pursuit distances $d_p = 50$cm and $d_p = 40$cm respectively. We note this relationship between the quality of the terrain roughness and the best performing pursuit distance.

The rough path involved the most obstacle traversals, including three wheel-sized rocks between waypoints 4 and 5, pictured in Fig. 13 at approx $t = 30 \ldots 35$s on the $d_p = 40$cm (blue) path. Both the flat and rough paths show continuous oscillations in heading when $d_p = 30$cm, however it is noted that on rough terrain the magnitude of the

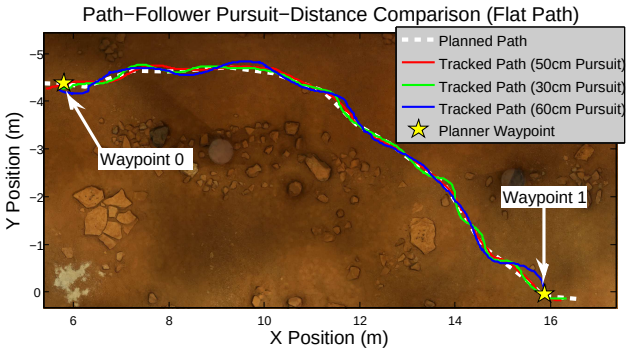


Figure 14: A comparison of planner-generated paths being followed using pursuit distances: 30cm (green), 50cm (blue), 60cm (red).

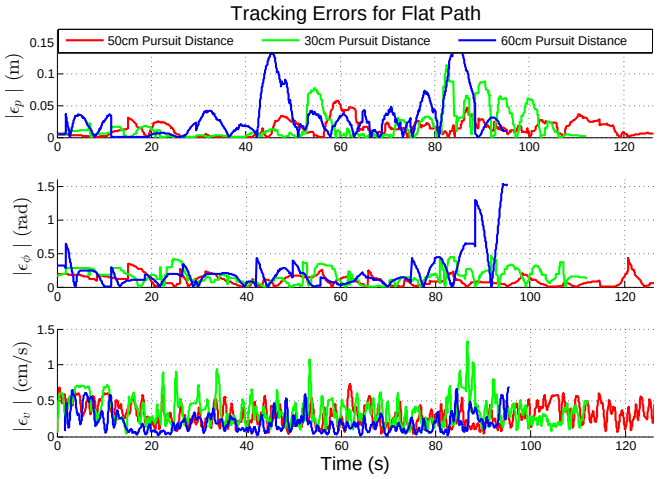


Figure 15: Position, heading and speed errors during tracking a flat path using pursuit distances: 30cm (green), 50cm (blue), 60cm (red).

oscillations is far greater.

In the experiments the velocity tracker performed with errors within 1.6% and 6.2% of the nominal speed, even on paths that involved traversing rocks and overcoming steep slopes.

9 Conclusion and Future Work

Autonomous capabilities including terrain based path planning and path-following have been added to EMR-Mawson to support space research and education in the Mars Yard. The processes described here take advantage of an *a priori* point cloud of the terrain and an *a priori* occupancy grid to support simulated missions on the Mars Yard for further research and education. The guidance system helps to minimise risk to the EMR by implementing terrain-based cost functions in the path-planning. The navigation system described provides fast, accurate vision-inertial pose updates of the EMR that can be used to control it. The effectiness of the control system has been empirically observed and it outperforms even

Table 1: Tracking Errors in Path Following Experiments.

Path Name	d_p (cm)	$\lvert\epsilon\rvert$ (cm)	$\lvert\epsilon_\theta\rvert$ (°)	$\lvert\epsilon_v\rvert$ (cm/s)	Duration (sec)
Mission	60	7.7	43.7	0.6556	230
*Misson	60	15.4	25.1	0.6114	235
*Flat	30	5.3	11.2	0.4011	112
*Flat	50	4.9	7.0	0.3322	128
*Flat	60	10.3	20.0	0.2346	93
Rough	30	10.1	22.9	0.9172	85
[1]Rough	40	5.9	10.5	0.7111	63
Rough	50	6.2	10.1	0.9275	56

*Indicates a planned path.
[1] Path ended early due to collision.

expert human operators in tracking complex paths over the roughest terrain in the Mars Yard.

In this paper, we restricted our attention to a simple kinematic model of the EMR and investigated if the combination of terrain roughness and pursuit distance has an effect on tracking performance. We conclude that for system tested, there is a relation between terrain roughness, pursuit distance and tracking accuracy. This work has enabled future work that will involve using learning algorithms to stochastically find the relationship between terrain roughness, pursuit distance and tracking accuracy to calculate the optimum look-ahead distance in the planning stage and apply that to a control law. Since the revision of this paper, a variant of RAS has also been validated on the MAMMOTH quadruped wheel-on-leg rover also at the Powerhouse Museum.

Acknowledgments

We would like to acknowledge the Australian Governments Broadband Enabled Education and Skills Services Programme, which funded The Mars Lab project.

References

[Bollobas, 1998] Bela Bollobas. Modern Graph Theory. *Springer Science & Business Media, ISBN-10:0387984887, ISBN-13:9780387984889*, 1998.

[Campion *et al.*, 1996] Guy Campion, Georges Bastin, and Brigitte D Andrca-novel. Structural Properties and Classication of Kinematic and Dynamic Models of Wheeled Mobile Robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62, 1996.

[Clarkson, 1997] Clarkson, Kenneth L. Nearest Neighbor Queries in Metric Spaces. *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pp. 609–617, 1997

[Golub and Kahan, 1965] Golub, G. and Kahan, W. Singular value decomposition and principal component analysis. *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, pp. 205-224, 1965

[Helmick *et al.*, 2009] Daniel M. Helmick, Anelia Angelova, Larry Matthies. Terrain Adaptive Navigation for planetary rovers. *Journal of Field Robotics*, 26(4):391–410, Feb.5, 2009.

[Hoffman and Krotkov, 1989] Hoffman, R. and Krotkov, E., Terrain Roughness Measurement from Elevation Maps. *Proc. SPIE Mobile Robots*, Vol. 4, pp 104–114, Philadelphia, USA, 1989.

[Lepetit *et al.*, 2013] Lepetit, Vincent and Moreno-Noguer, Francesc and Fua, Pascal Epnp: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, Vol. 81, No.2, pp. 155–166, 2009.

[Mandel, 1982] Mandel, John. Use of the Singular Value Decomposition in Regression Analysis. *The American Statistician*, Vol 36, pp. 15-24, 1982

[Morales *et al.*, 2013] Morales, J., Martinez, J., Martinez, M. and Mandow, A Pure-Pursuit Reactive Path Tracking for Nonholonomic Mobile Robots with a 2D Laser Scanner. *EURASIP Journal on Advances in Signal Processing*, Vol. 2009, Article ID 935237, January, 2009.

[Martin and Corke, 2011] Martin, S. and Corke, P. Path Planning Using Surface Shape and Ground Properties. *Proceedings of Australasian Conference on Robotics and Automation*, 7-9 Dec 2011,

[Potiris, Tompkins and Göktoğan, 2013] Potiris, S. , Tompkins, A. and Göktoğan, A-H. A Vision-Based Autonomous Docking System of the Experimental Mars Rover. *Australian Space Science Conference (ASSC-2013)*, pp.295–306, Sydney, Australia, Sep.30-Oct.2, 2013.

[Rekleitis *et al.*, 2008] Rekleitis, I and Bedwani, J.-L., Dupuis, E. and Allard, P. Path Planning for Planetary Exploration. *Canadian Conference on Computer and Robot Vision*, pp.61–68, Windsor, Ontario, 28-30 May, 2008.

[Rekleitis *et al.*, 2007] Rekleitis, I and Bedwani, J.-L. and Dupuis, E. Over-the-Horizon, Autonomous Navigation for Planetary Exploration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2248–2255, 2007

[Tahirovic *et al.*, 2013] Tahirovic, A and Magnani, G. and Kuwata, Y. Path Planning for Planetary Exploration. *IEEE International Conference on Mechatronics (ICM)*, pp.242–247, 2013.

[Tompkins, Potiris and Göktoğan, 2013] Tompkins, A., Potiris, S. and Göktoğan, A.H. Development of a Low-Cost Vision-Based Localisation System for the Experimental Mars Rover. *Australian Space Science Conference (ASSC-2013)*, pp.283–294, Sydney, Australia, Sep.30-Oct.2, 2013.