Learned Stochastic Mobility Prediction for Planning with Control Uncertainty on Unstructured Terrain

Thierry Peynot∗
School of Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, QLD 4001, Australia
t.peynot@qut.edu.au

Sin-Ting Lui
Australian Centre for Field Robotics
The University of Sydney
NSW 2006, Australia
a.lui@acfr.usyd.edu.au

Rowan McAllister∗
Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK
rtm26@cam.ac.uk

Robert Fitch
Australian Centre for Field Robotics
The University of Sydney
NSW 2006, Australia
rfitch@acfr.usyd.edu.au

Salah Sukkarieh
Australian Centre for Field Robotics
The University of Sydney
NSW 2006, Australia
salah@acfr.usyd.edu.au

Abstract

Motion planning for planetary rovers must consider control uncertainty in order to maintain the safety of the platform during navigation. Modelling such control uncertainty is difficult due to the complex interaction between the platform and its environment. In this paper, we propose a motion planning approach whereby the outcome of control actions is learned from experience and represented statistically using a Gaussian process regression model. This mobility prediction model is trained using sample executions of motion primitives on representative terrain, and predicts the future outcome of control actions on similar terrain. Using Gaussian process regression allows us to exploit its inherent measure of prediction uncertainty in planning. We integrate mobility prediction into a Markov decision process framework and use dynamic programming to construct a control policy for navigation to a goal region in a terrain map built using an on-board depth sensor. We consider both rigid terrain, consisting of uneven ground, small rocks, and non-traversable rocks, and also deformable terrain. We introduce two methods for training the mobility prediction model from either proprioceptive or exteroceptive observations, and report results from nearly 300 experimental trials using a planetary rover platform in a Mars-analogue environment. Our results validate the approach and demonstrate the value of planning under uncertainty for safe and reliable navigation.

∗Work done while the authors were at the Australian Centre for Field Robotics, The University of Sydney.
1 Introduction

Outdoor robots such as planetary rovers are designed for mobility in challenging environments, but are subject to significant control uncertainty due to complex interactions between the robot and the terrain (Schenker et al., 2003). Motion planning for planetary rovers must consider such control uncertainty, particularly in environments that expose the robot to the risk of serious mechanical damage, or in situations where robots operate in remote locations beyond the reach of human intervention. Loss of mobility can be catastrophic, yet these environments and situations represent a prime opportunity for mobile robotics to contribute to advancing scientific understanding through information gathering. In the case of planetary exploration, interesting science goals often lie within the most challenging terrain (Johnson, 2010). Our goal in this paper is to address the problem of motion planning with control uncertainty for the purpose of safe and reliable navigation for planetary rovers.

The goal of classical geometric motion planning is to minimise time or distance while avoiding obstacles (LaValle, 2006). The conceptual distinction between free space and obstacles for planetary rovers, however, is less clear. It is important to avoid obstacles, but it is also desirable to avoid free space where, due to control uncertainty, the robot has high likelihood of encountering an obstacle during execution. This situation cannot be modelled by simple distance thresholds surrounding obstacles because risk varies across free space and is not deterministic.

Motion planning under uncertainty fundamentally depends on the ability to predict the likely possible outcomes of an action in a given situation. Accurately predicting executed behaviour in response to a given control input is not straightforward in the case of planetary rovers due to complex terramechanics (Ishigami et al., 2010). For previously unobserved terrain, prior models of terrain properties may not be available.

Current work in online terrain estimation considers the learning problem where characterisation of local terrain is used to predict far terrain (Krebs et al., 2010). In this setting, metrics have been developed that relate terrain characteristics to rover performance, and these metrics can then be used to influence path planning. The focus of this work, however, is on hazard detection and avoidance (Brooks and Iagnemma, 2012). We are interested in terrain estimation that not only identifies hazards, but also provides a predictive model of control uncertainty in non-hazardous and semi-hazardous terrain.

In this paper, our approach is to learn the macroscopic behaviour of the rover directly. Instead of focusing on the complex low-level interaction between wheels and various soil types, we build a stochastic transition model of rover kinematics based on experience, and then use this stochastic transition model to generate a control policy that maximises safety and reliability under control uncertainty. Our intention is to capture
effects of terrain interaction explicitly at the whole-rover level. We consider rigid terrain as well as deformable terrain such as loose soil and movable rocks. The path planning objective is not to completely avoid areas with high control uncertainty such as deformable terrain, but to choose reliable paths that are consistently safe with respect to the expected rover behaviour.

We propose a navigation method that consists of two parts. First, we present a novel mobility prediction model based on Gaussian process (GP) regression that maps observed terrain features and rover control actions to resulting changes in rover pose. Because the terrain observations input to the GP model are represented continuously in feature space, our approach does not require the enumeration of terrain types as discrete classes. Second, we use this model in a Markov decision process framework and produce a control policy using dynamic programming. Inference in GP models provides a measure of prediction uncertainty that naturally instantiates the stochastic transition function used for planning. Execution of the policy results in reliable navigation because actions are chosen to maximise expected safety over time.

We implement two alternative methods for collecting experiential data to construct the GP model, one using proprioceptive measurements of rover state and the other using exteroceptive measurements, and report experimental data from nearly 300 trials with a rover platform operating in a Mars-analogue environment. The platform and environment are shown in Fig. 1. Our experimental results capture both rigid and deformable terrain cases, and show the benefits of our method in terms of navigation safety and reliability in comparison with planning with deterministic control. A main contribution of this research is to show that these benefits can be realised by using a nonparametric regression model that does not require extensive a priori understanding of the terramechanic properties of the environment. In this paper we extend a preliminary version of this work that considered only the case of learning from proprioception (McAllister et al., 2012); we consider the exteroceptive case and present further experimental results in more complex environments including deformable terrain.

The organisation of the paper is as follows. Section 2 discusses related work in motion planning and terrain estimation for planetary rovers. Sections 3 and 4 present our algorithmic contributions in mobility prediction and motion planning. The experimental system is described in Sec. 5, including a description of the experimental platform and Mars-analogue environment. Implementation, experiments, and results for the proprioceptive learning case are presented in Sec. 6, followed by the exteroceptive case in Sec. 7. Section 8 discusses the experimental results and concludes the paper.

2 Related Work

The problems of terrain traversability estimation and navigation for planetary rovers have been studied from several perspectives, motivated primarily by the need for robust navigation to support scientific information gathering on the surface of Mars (Johnson, 2010). Traversability analysis seeks to determine whether given terrain is traversable by the robot or not, or its degree of difficulty, for the purpose of motion planning. Traditional techniques evaluate the roughness of the terrain by compiling statistics about terrain geometry locally (Goldberg et al., 2002). The quality of traversability estimation can be improved by augmenting the traversability map with additional information such as tip-over stability (Iagnemma et al., 2001; Lacroix et al., 2002) and slip (Helmick et al., 2004; Shimoda et al., 2005). Slip is a measure of the lack of mobility or progress of a rover on the terrain; approaches that predict slip involve visual odometry (Helmick et al., 2004), stereo vision (Angelova et al., 2007), motion profile (Ishigami et al., 2007), current draw (Ojeda et al., 2006), wheel odometry (Ward and Iagnemma, 2008), wheel trace (Reina et al., 2008) and delayed-state filter for inertial navigation (Rogers-Marcovitz et al., 2012). A recent survey of traversability analysis is presented by Papadakis (Papadakis, 2013).

Physics-based mobility prediction in general is based on terramechanics, the study of soil-wheel interaction. A survey in the context of planetary rovers is presented in (Chhaniyara et al., 2012). Terramechanics provides detailed mobility models by considering specific terrain features, such as soil cohesion, density,
and angle of wheel/terrain contact (Iagnemma et al., 2001). Although these parameters can be estimated online (Iagnemma et al., 2004), estimation is difficult and can be highly uncertain, even on ideal homogeneous terrain. Recent work in statistical mobility prediction uses the stochastic response surface method (Ishigami et al., 2010; Kewlani and Iagnemma, 2008), where a Gaussian distribution is generated over predicted future states on homogeneous terrain, assuming noisy terramechanics parameters. However, navigation often involves heterogeneous and deformable terrain, and large parameter uncertainty makes it difficult to apply high-fidelity model-based methods directly in planning.

Recently introduced, near-to-far learning is an online approach to terrain classification and prediction where remotely sensed data are associated with proprioceptive metrics that can be used to predict rover mobility (Helmick et al., 2009; Krebs et al., 2010). The near-to-far approach allows the mobility model of the rover to be learned from experience, and terrain classification is not restricted to a predetermined set of classes. Krebs et al. (Krebs et al., 2010) relate softness, bumpiness, and visual features to a cost function for mobility prediction. Angelova et al. (Angelova et al., 2007) and Howard et al. (Howard et al., 2006) associate slip and wheel vibrations with classified visual data to augment the traversability map. Brooks and Iagnemma (Brooks and Iagnemma, 2012) propose a self-supervised learning framework for terrain classification. This work uses support vector machines (SVM) to learn discrete class labels from vibration features (measured in the rover’s suspension), and traction (measured by wheel torque and sinkage). These class labels are then associated with exteroceptive features (colour, visual texture, geometry-from-stereo) to predict the mechanical properties of far-away terrain. Although this work strongly connects observable terrain features with vehicle mobility, the discrete class labels do not include uncertainty estimates that could be exploited in planning. Other approaches to vibration-based classification include Gaussian mixture model representations (Weiss and Zell, 2008), and supervised learning (Halatci et al., 2008). Similarly, (Schwendner and Kirchner, 2010) suggested that the use of embodied data in addition to exteroceptive data could contribute to the prediction of a robot’s position, in the context of simultaneous localisation and mapping (SLAM). This was recently demonstrated in (Schwendner et al., 2014), where the uncertainties are well exploited within a hierarchical SLAM framework. In this paper, we focus on stochastic mobility prediction for planning, rather than localisation. Karumanchi et al. (Karumanchi et al., 2010) use Gaussian process regression to generate mobility prediction maps by learning permissible rover slip from past experiences. These mobility maps are then used in A* path planning. Our approach can be viewed as a type of near-to-far learning in that we associate locally sensed data to predict far-away mobility, but we explicitly model the stochastic response of the rover directly as opposed to indirectly through a traversability metric. Note that most of the aforementioned studies neglect possible terrain deformation during rover traversal, i.e. they assume that the terrain geometry as it is seen far away remains the same as that under the rover’s wheels.

Stochastic mobility prediction models are typically used in reactive control or control compensation, where predictions, e.g. risk of slip, help to minimise deviation from a reference path (Karumanchi and Iagnemma, 2012; Helmick et al., 2009). Techniques have also been proposed for learning such control compensation through experience in the context of self-modelling (Gloye et al., 2005; Bongard et al., 2006). Reactive techniques can compensate for control uncertainty, but require a reference path to be provided a priori.

Path planning for unmanned ground vehicles has been addressed using search algorithms such as A* (Karumanchi et al., 2010) and D* (Kelly et al., 2006) that find a path given a traversability map, assuming a deterministic mobility model. Model-based trajectory generation that can be used as part of a sampling-based planner is presented in (Howard and Kelly, 2007). Planning that considers the computational cost of terrain assessment was recently proposed (Stenning and Barfoot, 2012). (Gonzalez and Stentz, 2007) considered the problem of planning with uncertainty in position. In this paper, we focus on accounting for control uncertainty in planning.

A common approach for considering control uncertainty in motion planning is to express the uncertainty as a cost and then to plan a path that minimises this cost assuming deterministic control (Helmick et al., 2009). Another family of approaches plans a path using a sampling-based algorithm, and then evaluates the control uncertainty along the path selected (Bry and Roy, 2011; Ishigami et al., 2007). Various forms of control strategies (such as LQG) can be used to model potential deviations from a path and hence to select
a path with least risk in terms of platform safety (Patil et al., 2011; Berg et al., 2010).

For non-deterministic systems, Markov decision processes (MDPs) are commonly used to formulate problems in motion planning with uncertainty (LaValle, 2006; LaValle and Hutchinson, 1998). Control uncertainty is represented as a stochastic transition function, and a policy can be computed using dynamic programming (Alterovitz et al., 2007). The partially-observable Markov decision process (POMDP) is another common formulation (Kurniawati et al., 2011). However, these techniques are most often evaluated in simulation only and there is a critical need for validation using real robots. Recent experimental work that uses dynamic programming in path planning for planetary rovers is presented in (Plonski et al., 2013), although the objective is energy efficiency as opposed to navigation reliability.

In this work our model of stochastic actions is tied to observed terrain profiles that vary across the environment and is learned through experience. We furthermore consider risk at the level of motion primitives and construct a control policy that is executed directly. Our approach uses statistical regression in performing the inference, showing that meaningful improvements to motion planning are possible without a complex terramechanics model. However, these improvements are not restricted to the form of motion planning we present and likely may be realised by integrating our mobility prediction approach with other motion planning techniques.

3 Mobility Prediction

In this work, we consider a planetary rover interacting with unstructured, slightly deformable terrain which consists of both non-geometric (loose soil) and geometric (rocks, slopes) hazards. We propose to incorporate control execution uncertainty into motion planning by learning a stochastic mobility prediction model from experience. This model will capture the errors caused by small terrain deformations, wheel slippage and the actuators themselves. Given a cost map of the environment, our approach uses the mobility prediction as a stochastic state transition model to compute policies that minimise the cost over the entire state space rather than a single path. This section first presents how we build the mobility prediction model. The planning algorithm will then be described in Sec. 4.

The structure of this section is as follows. The stochastic mobility prediction model is first defined in Sec. 3.1. Section 3.2 then introduces the regression method used to generate the stochastic transition function by learning from experience. Section 3.3 describes the training process and Sec. 3.4 indicates how the learned mobility prediction model will be used in our planning framework.

3.1 Stochastic Mobility Prediction Model

Given an initial state of the rover, a mobility prediction model provides the potential subsequent states resulting from the execution of a given action. We define the state \( s \) of the rover using two dimensions \( x \) and \( y \) for position and one dimension \( \psi \) for orientation (yaw), i.e. \( s = \{x, y, \psi\} \). Consider \( A \) the set of actions \( a \) that the rover can execute. Given a vehicle state \( s \), a deterministic mobility prediction model provides the state \( s' \) resulting from the execution of action \( a \in A \). However, due to control uncertainty, in practice the resultant state is not deterministic (see Fig. 2). To encapsulate multiple possibilities of resultant states, we formulate the state transition function as the probability density function (PDF) of the relative transition between states, \( p(\Delta s|s, a) \), with \( \Delta s \equiv s' - s \):

\[
P(s'|s, a) = \int p(\Delta s|s, a)f(s + \Delta s, s') \, d\Delta s,
\]

where \( f(s_1, s_2) = 1 \) if the discretisation of \( s_1 \) corresponds to \( s_2 \) and \( f(s_1, s_2) = 0 \) otherwise.

On unstructured terrain, the subsequent states resulting from the execution of an action strongly depend on the nature and geometry of the terrain. For example, the same action executed on flat terrain and on rough
terrain will produce different outcomes. Thus, the transition model needs to take into account the geometry of the terrain traversed while executing action \( a \) from state \( s \). In our approach, this is represented by the *terrain profile* \( \lambda(s, a) \). Therefore, we have:

\[
p(\Delta s|s, a) = p(\Delta s|\lambda(s, a), a).
\]  

(2)

To directly capture the actual influence of the terrain geometry on the platform, in this paper \( \lambda(s, a) \) encodes the variations of vehicle attitude and configuration during the execution of action \( a \). This includes the variations of pitch (\( \phi \)), roll (\( \theta \)) and internal angles describing the configuration of the chassis.

We consider the \( N \) components \( \Delta s_i \) of \( \Delta s \), which can be defined using Cartesian or radial representations of the state space. In this paper, the components are the vehicle’s heading and distance travelled (see Fig. 2):

\[
\Delta s_1 = \Delta s_{\text{head}} = \tan^{-1}(\Delta y, \Delta x)
\]

(3)

\[
\Delta s_2 = \Delta s_{\text{dist}} = \sqrt{(\Delta x)^2 + (\Delta y)^2}.
\]

(4)

For holonomic vehicles, we also consider a third component, yaw:

\[
\Delta s_3 = \Delta s_{\text{yaw}} = \Delta \psi.
\]

(5)

Therefore, in our approach, \( \Delta s \) is defined by the tuple

\[
\Delta s \triangleq \{\Delta s_{\text{head}}, \Delta s_{\text{dist}}, \Delta s_{\text{yaw}}\}. 
\]

(6)

We propose to build the mobility prediction model by learning the relations between the outcomes of each action \( a \) and the terrain profiles \( \lambda(s, a) \), from experience. We first collect training data during multiple executions of each action \( a \) over a variety of terrain profiles, in a representative environment. We then build a model that can predict the distribution of control errors, expressed as deviations from the action’s expected outcome \( (\Delta s_a) \), i.e.:

\[
p(\Delta s_a|\lambda(s, a), a) = \overline{\Delta s_a},
\]

(7)

where \( \overline{\Delta s_a} \) is the mean value of \( \Delta s \) across all executions of action \( a \) in the training data. Using this formulation means that the training data for each action will have zero mean.

In practice, training can only provide a limited, sampled subset of the state space and of the possible terrain profiles. Therefore, we use a regression method to infer distributions of control errors that were not directly sampled in the training data. In this work, we use Gaussian process regression, which is a powerful technique to model sparse, spatially correlated data with uncertainty.
3.2 Gaussian Process Regression

Gaussian process regression is a non-parametric learning technique that estimates a single output from multiple inputs. Given a training set composed of $n$ inputs $X = \{x_j\}_{j=1}^n$ and the corresponding observed outputs, or targets $\{y_j\} = Y$, a GP provides a predictive distribution $f_*$ for any query inputs $x_*$. GPs assume noise to be additive, independent and Gaussian with zero mean (Rasmussen and Williams, 2006).

The covariance function we use to describe the spatial correlation between two input vectors $x$ and $x'$ is the standard squared exponential:

$$K(x, x') = \sigma_f^2 \exp \left( -\frac{1}{2} (x - x')^\top \Lambda^{-2} (x - x') \right) + \sigma_n^2 I,$$

where $\sigma_f^2$ is the variance of the noise-free input, $\Lambda$ is a length scale matrix of diagonal elements that describes the smoothness of the input data and $\sigma_n^2$ is the noise variance. In this work, these hyperparameters $\Theta = \{\sigma_f, \Lambda, \sigma_n\}$ are learned by maximising the log-marginal likelihood of the targets given the inputs and hyperparameters:

$$\log p(Y | X, \Theta) = -\frac{1}{2} Y^\top L^{-1} Y - \frac{1}{2} \log |L| - \frac{n}{2} \log 2\pi,$$

where $L = K + \sigma_n^2 I$. $f_*$ is then estimated as the Gaussian distribution:

$$p(f_* | X, Y, x_*) \sim \mathcal{N}(\mu_*, \Sigma_*),$$

with predictive mean

$$\mu_* = K(x_*, X)[K(X, X) + \sigma_n^2 I]^{-1} Y,$$

and variance

$$\Sigma_* = K(x_*, x_*) - K(x_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, x_*).$$

In this paper, we use GPs to build a continuous mobility prediction model. For each component $\Delta s_i$, $i \in [1, N]$ and each action $a \in A$, we define a separate GP (GP$_{i,a}$) to estimate the distribution:

$$p(\Delta s_{i,a} | \lambda(s, a), a) = \overline{s}_{i,a},$$

where $\Delta s_{i,a}$ is the $i^{th}$ component of the change of state $\Delta s$ resulting from executing action $a \in A$, and $\overline{s}_{i,a}$ is the mean value of $\Delta s_i$ across all executions of action $a$ in the training data. The predictive distribution of each GP can therefore be written as:

$$p(f_* | X, Y, x_*) = p(\Delta s_{i,a} | \lambda(s, a), a) - \overline{s}_{i,a} \sim \mathcal{N}(\mu_*, \Sigma_*).$$

In this method, we consider the uncertainty in each component $\Delta s_i$ by using the full distribution learned from $\Delta s_i$ and the expectation of the other components. Representing $\lambda(s, a)$ as $\lambda$, Eq. (2) is calculated as:

$$p(\Delta s | \lambda, a) = p(\{\Delta s_1, \ldots, \Delta s_i, \ldots, \Delta s_N\} | \lambda, a) \approx \{\mathbb{E}(\Delta s_1 | \lambda, a), \ldots, \mathbb{E}(\Delta s_i | \lambda, a), \ldots, \mathbb{E}(\Delta s_N | \lambda, a)\}. $$

Note that we model the uncertainty in the outcomes of each action separately, as well as for each component $\Delta s_i$. Therefore, in this paper, we learn a GP for each action $a$, and for each $\Delta s_i$, i.e. GP$_{i,a}$. In total, 18 GPs are calculated for six symmetrical actions (see Sec. 4.2) and three $\Delta s_i$ components.
3.3 Training

The training inputs $x$ represent the terrain profiles along the action execution, shifted to have zero mean, i.e., $x = \lambda(s,a)_{\text{train}} - \lambda(s,a)_{\text{train}}$, where $\lambda(s,a)_{\text{train}}$ is the mean of the terrain profile features in the training data. Let us name $\Phi(s)$ the vector of vehicle attitude and configuration angles when the rover is at position $s$. A full terrain profile can be represented by the set of all $\Phi$ for all discrete states during the execution of action $a$, i.e., $\{\Phi(s), ..., \Phi(s')\}$, where $s$ is the initial state, and $s'$ is the state reached at the end of the execution of action $a$. Depending on the resolution of the state space, it may not be practical to train our GPs using all of this information. Therefore, in our approach, $\lambda(s,a)$ are encoded more compactly using selected features to reduce the problem's dimensionality, and so that the over-fitting problem in learning can be mitigated. We determine the set of most informative features by performing a Principal Component Analysis (PCA) over a large variety of features capturing values and variations of the $\Phi$ sets.

The input training targets $y_{\text{train}}$ of each GP are the differences between the observed and average state transitions: $y_{i,a_{\text{train}}} = \Delta s_{i,a} - \Delta s_{i,a}$ for the component $\Delta s_i$ and action $a$, defined to have zero mean. The training of the GPs consists in learning from experience the correlations between terrain profiles and the control errors of each executed action, i.e., the correlation between $\lambda(s,a)$ and $y_{i,a}$. These correlations are represented by the matrix $K$.

In practice, to collect the required training data, the rover executes each action $a \in A$ multiple times over varying terrain profiles, on both rigid and deformable terrain, while recording: the action $a$, the difference between action outcome $s'$ and starting state $s$, and the attitude and configuration angles $\Phi(s)_{\text{train}}$ of the platform during the action execution. Then, features $\lambda(s,a)_{\text{train}}$ are computed from the $\Phi(s)_{\text{train}}$ sets, and the $\Delta s_i$ are calculated from $s$ and $s'$. In Sec. 6 and Sec. 7, we consider two ways to obtain the sets of $\Phi(s)_{\text{train}}$ used for training: from proprioception and from exteroception, respectively.

3.4 Learned Mobility Prediction Model

Once the GPs have been trained, the expected distributions of action outcomes $\Delta s_i - \Delta s_i$ can be computed for any terrain profile $\lambda(s,a)$, since the GPs provide continuous representations of these outcomes, with uncertainty. These distributions will be used to account for control uncertainty in planning. Figure 3 illustrates how our proposed approach will predict the heading and distance outcomes resulting from the execution of a given action $a$ from state $s$. First, we compute the expected attitude and configuration angles $\{\Phi(s), ..., \Phi(s')\}$ of the rover along the average executed path, of initial state $s$ and final state $s'$. Second, we calculate the corresponding terrain profile features $\lambda(s,a)$. Third, we query the estimated distributions of $\Delta s_i - \Delta s_i$ to the GP mobility prediction models built for action $a$, i.e. $GP_i,a$, with $\lambda(s,a)$ as an input. The GPs provide the expected mean $\mu_{s_i}$ and standard deviation $\sigma_{s_i}$ of each $\Delta s_i$ component. Using this process, the planner will be able to make a prediction of the outcomes of executing any action over any terrain profile, and account for the uncertainties in heading, distance, or yaw.

4 Planning

In this section we consider the integration of our learned mobility prediction model within a motion planning framework. We use the set of Gaussian process regression models that describe the behaviour of the robot to instantiate the transition function of a Markov decision process, and describe a set of motion primitives for a planetary rover that instantiate the action set of the MDP.
Figure 3: Predicting the uncertain outcomes resulting from the execution of a given action $a$, starting from position $s$ (top view), using the learned mobility prediction model. $\overline{s}'$ represents the average of all final positions, for all terrain profiles, when executing action $a$ in the training data ($\overline{s}' = s + \overline{\Delta s}$). The dots along the black lines show where the anticipated attitude and configuration angles $\{\Phi_*(s), \ldots, \Phi_*(\overline{s}')\}$ are evaluated. The mean of expected heading deviation, obtained when querying the GPs of action $a$, is shown by $\mu_{head}$ in (a), with the predicted distribution of outcomes illustrated in grey. Similarly, the mean of expected distance deviation, is shown by $\mu_{dist}$ in (b). In both cases, the corresponding predictive mean of final position is shown as $s'$ in grey.

4.1 MDP Formulation

One common approach to motion planning for planetary rovers is to first generate a path using a graph search algorithm such as A*, and then to pass this path to a controller for execution. If the controller is unable to follow the path, a replanning process is used to generate a new path. The planning objective must therefore favour areas of high controllability in order to minimise the need for replanning. In contrast, the MDP formulation is a sequential decision making process. A control policy is generated that maps states to actions, and the robot chooses and executes actions in an iterative manner until it reaches the goal state. Because the expected outcome of an action can be described probabilistically, as is the case in our mobility prediction model, control uncertainty is naturally considered in the planning process. The planner is thus free to choose actions that traverse areas of high control uncertainty so long as all possible outcomes lead to desirable states from which the robot is still able to reach the goal.

An MDP is defined by a 4-tuple $< S, A, T, R >$, where $S$ is a set of states, $A$ is a set of actions, $T$ is a transition function that maps a state-action pair to a set of resultant states, and $R$ is a reward function that specifies the cost of executing an action. The transition function $T$ can be stochastic and is typically represented as $P(s'|s,a)$, the probability of entering state $s'$ when executing action $a$ from initial state $s$. Our learned mobility prediction model provides such a function, defined earlier by Eq. (1). We consider the set of states $S$ to be a set of discrete positions and orientations of the robot within a given terrain map. An element of $S$ is defined as $s = \{x, y, \psi\}$. Reward function $R(s'|s,a)$ is interpreted as a cost function that encodes a measure of terrain difficulty. Actions $a \in A$ are motion primitives defined in terms of control inputs.

Our objective is to maximise the sum of rewards (or equivalently, minimise costs) accumulated over a sequence of actions. We compute a value function that maps states in $S$ to a scalar value that represents expected cost-to-goal. The value function is computed using dynamic programming (LaValle, 2006). The control policy greedily chooses the action with maximum expected value, where resultant states are predicted by the learned mobility model.

Because the learned mobility model comprises single-output GPs, we consider either distance uncertainty or heading uncertainty (but not both together) during planning. This is not a limitation of the MDP framework; correlated predictions could be included with no change to the planner.
4.2 Motion Primitives

We define actions as motion primitives specified in terms of control inputs. We consider two classes of motion primitives: crabbing and rotation. A crabbing action $crab(\beta)$ corresponds to executing a straight line translation in the $xy$-plane in a given direction $\beta$ and for a given duration, with no change in vehicle yaw $\psi$, and a constant linear velocity during the action execution. A rotation action $rotate(\Delta \psi)$ consists of a spin-on-the-spot motion, at constant angular velocity, that results in expected relative change in yaw $\Delta \psi$. These motion primitives treat the rover as a holonomic platform, but other motion primitives that respect nonholonomic constraints could be defined if desired.

Because the complexity of dynamic programming depends on the size of the action set, we limit the number of crabbing actions to eight directions at intervals of $\pi/4$. There are two rotation actions, $\pm \pi/4$. Figure 4 illustrates this set of motion primitives. Implementation details, including the linear and angular velocity values and durations used in our experiments, are described as part of the system description in Sec. 5.4.

5 Experimental System

This section specifies the implementation details of the approach proposed in this paper. In particular, after providing an overview of the system, we describe our test environment (Sec. 5.2), the experimental platform used to validate the method (Sec. 5.3), the implementation of motion primitives used in planning and control (Sec. 5.4), the terrain map (Sec. 5.5). Finally, we present how the cost and reward functions are computed (Sec. 5.6) and discuss planning and execution time (Sec. 5.7).

5.1 System Outline

Figure 5 gives an outline of our implementation of the proposed approach. First, the offline training allows for the generation of each covariance matrix $K$ that is needed for the GP regression. Online, a digital elevation map (DEM) is generated from point clouds acquired by a range sensor. The DEM is then converted into: a) a map of predicted rover configurations $\Phi_*(s)$, that the planner can query, and b) a cost map ($Cost(s)$).

In the second step, our planning algorithm (shown as the large grey box in Fig. 5) computes the optimal policy to reach the given goal, using dynamic programming, by evaluating state-action pairs $(s,a)$. For each $(s,a)$, the system generates the corresponding reward $R(s'|s,a)$ from the cost map and the stochastic transition function $P(s'|s,a)$ from the GPs. To obtain this probability $P$, the system first computes the terrain profiles $\Lambda_*(s,a)$ corresponding to the configurations $\Phi_*(s)$ that the algorithm needs to query. Then, it queries the appropriate GP for the predictive mean and covariance of state transition. Finally, once the optimal policy has been found, the rover greedily follows it to reach the goal.
5.2 Test Environment

Training and experiments were performed at the Mars Yard, a Mars-analogue terrain facility located inside the Powerhouse Museum in Sydney, Australia. This terrain is composed of solid and loose soil, gravel, slopes of different gradients and rocks of varying sizes and geometry. The slopes vary from $0^\circ$ (flat terrain) to $11.5^\circ$. Rocks heights vary from approximately $0.05m$ to $0.2m$ in radius (i.e. from 1 to 4 times the radius of the rover’s wheel, see below Sec. 5.3). The Mars Yard contains some rigid and some deformable terrain, both of which were considered in this study. The soil on rigid terrain (see experiments in Secs. 6.5.1 and 7.3.1) is mostly compact, and rocks do not shift during rover-terrain interaction, whereas the deformable terrain (see Secs. 6.5.2 and 7.3.2) consists of sandy loose-soil slopes and/or rocks that may shift due to rover-terrain interaction. Note that GP training was performed on different areas of the Mars Yard than the tests reported in Sec. 6.3 and Secs. 7.3.1-7.3.2.

5.3 Platform

The robot platform used in our experiments is the Mars rover prototype “Mawson”, shown in Fig. 6(a). Mawson is a six-wheeled holonomic rover with individual steering servo motors on each wheel and a Rocker-bogie chassis shown in Fig. 6(b). It is about $0.80m$ long and $0.63m$ wide, with a mast height of about $0.9m$ above the ground. The wheel radius is $0.25m$.

The rover is equipped with an RGB-D camera (Microsoft Kinect) mounted on a mast, tilted down $20^\circ$, which is used for terrain modelling. Note that our experiments were conducted on the Mars Yard, located indoors, allowing for the use of this sensor. Although the RGB-D camera is not a very reliable sensor for outdoor operation, another depth sensor such as stereo vision could be used alternatively without affecting the conclusions of this study. Three potentiometers provide the configuration of the chassis by measuring both bogie angles and the rocker differential ($\alpha_i$ in Fig. 6(b)). The rover is also equipped with an Intersense IS-1200 VisTracker device, composed of a camera and an inertial measurement unit (IMU). The system fuses camera observations of a constellation of fiducials in the environment and the IMU data to provide a 6-DOF localisation with $2cm$ average accuracy.
5.4 Implementation of Motion Primitives

We implemented the motion primitives described in Sec. 4.1 using constant-valued control inputs over a fixed time duration. For crabbing actions, linear velocity is $0.11 m/s$ and the time duration was calibrated on flat ground to effect an expected translation length of $0.3 m$. For rotation actions, angular velocity is $0.24 rad/s$ and the time duration was calibrated (also on flat ground) to effect the specified relative change in yaw (here, $\pm \pi/4$).

5.5 Rover Attitude and Configuration Prediction over DEM

DEM's are built by distributing the 3D point clouds obtained with the depth sensor onto a regular Cartesian grid. In our approach we use a state discretisation of $0.05 m \times 0.05 m \times \frac{\pi}{24} rad$, chosen to be smaller than the magnitude of the largest control errors (see Sec. 6.3). The resolution of the DEM grid was set at $0.05 m \times 0.05 m$ to match the state discretisation. Note that this resolution corresponds approximately to the radius of the rover’s wheel and is larger than the typical standard deviation of the rover’s localisation errors (see Sec. 5.3 above).

Given a position $s$ on the DEM, the rover attitude $\{\phi, \theta\}$ and chassis configuration $\{\alpha_1, \alpha_2, \alpha_3 = (\alpha_{3A} - \alpha_{3B})\}$ angles, i.e. $\Phi(s)$, are predicted using a kinematic model similar to (Tarokh and McDermott, 2005). Although the simplified model does not take into account the dynamics of the platform, this method provides a sufficient approximation for this implementation since the rover operates at low speeds. This prediction of rover attitude and configuration is then used in two ways (see Fig. 5). First, the $\{\Phi_s\}$ sets allow for the computation of the terrain profile features $\lambda_s(s,a)$ for each $(s,a)$ pair considered by the planner. These features are the queried inputs of the GP, used to generate the expected stochastic state transitions. Second, it allows us to represent the observed terrain as a cost map, as detailed below.

5.6 Implementation of Cost Map and Reward Function

As mentioned in Sec. 4.1, the reward function we use to compute policies over unstructured terrain is the inverse of a cost function, representing the cost associated to the terrain traversal, accumulated during the execution of an action. This cost is composed of a traversability cost, or terrain cost, and an action cost that is independent from the nature and difficulty of the terrain.
The traversability cost was chosen to be an image of the rover’s safety. It captures the changes of attitude and configuration of the rover, which are indicative of the difficulty to cross the terrain and of the risk to the stability of the platform. Therefore, the larger the absolute values of roll, pitch, and configuration angles of the chassis, the larger the cost. At any given position \( s = \{x, y, \psi\} \) in the DEM, we define the traversability cost as:

\[
cost_{\text{terrain}}(s) = (\text{cost}_{\phi\theta}(s) + \text{cost}_\alpha(s))^2,
\]

where

\[
\text{cost}_{\phi\theta}(s) = (\phi^2 + \theta^2),
\]

\[
\text{cost}_\alpha(s) = 0.5(\alpha_1^2 - \alpha_2^2).
\]

Note that the difference of internal chassis angles \( \alpha_3 \) is not accounted for in the calculation of the traversability cost because we observed experimentally that the differences between the two angles \( \alpha_3A \) and \( \alpha_3B \) (see Fig. 6(b)) were negligible on the terrains that we experimented on. This could be because the passive suspension of the rocker bogie frame dampens out \( \alpha_3 \).

Using this definition of terrain cost, a DEM can then be converted into a cost map, reflecting the difficulty of the terrain at each discretised position. The configuration of the robot at a given position on the elevation map depends on its orientation, therefore a 2D cost map needs to be generated for each discretised orientation, resulting in a 3-dimensional \((x, y, \psi)\) cost map.

The reward \( R(s'|s, a) \) accounts for the terrain cost accumulated during the execution of action \( a \). In addition, we want the planner to favour policies with a limited number of actions that need to be executed between the starting point and the goal. Therefore, the reward collected during the executing of action \( a \) is defined as the negative of: 1) the average cost of states that lie on a linear interpolation between the start state \( s \) and the resultant state \( s' \), and 2) a penalty \( \xi \) for executing an action. This leads to:

\[
R(s'|s, a) = -\xi - \frac{1}{M} \sum_{i=0}^{M} \text{cost}_{\text{terrain}}\left(s_x + \frac{i}{M}(s'_x - s_x), s_y + \frac{i}{M}(s'_y - s_y), s_\psi + \frac{i}{M}(s'_\psi - s_\psi)\right),
\]

where \( M \) is the sampling resolution of the path. In our implementation, we use \( M = 20 \) and a small penalty of \( \xi = 0.003 \).

5.7 Execution Time

The execution time of our approach is dominated by the pre-processing steps required to generate the policy. Given a policy, action selection is performed efficiently using a simple table look-up.

The computational burden of generating a policy depends on the size of workspace and the chosen resolution. In our implementation, using a standard desktop-class computer, each step represented in Fig. 5 requires several hours of computation time. However, optimising the pre-processing time is not our focus in the paper. Pre-processing time can be reduced if required through a more efficient implementation and more capable computing hardware.

6 Learning Mobility Prediction from Proprioception

This section discusses and evaluates the proposed approach with our first training strategy, in which the training data is collected from proprioception. Section 6.1 first specifies the training process involved. Section 6.2 defines the features \( \lambda(s, a) \) used to represent the rover attitude and configuration variations along an action execution. Section 6.3 then provides some illustrations of the training data obtained. In Sec. 6.5 we propose an experimental validation of the approach. Finally, the results are analysed and the limitations of the method are discussed in Sec. 6.6.
Figure 7: LfP training process outline. The GPs are trained using proprioceptive data provided by the IMU and the potentiometers.

Figure 8: An example of training data collected for GP training using the LfP method (top view). In training, the rover achieves multiple executions of action $a$ starting from position $s$, resulting in different executed paths, shown in grey. The actual path for the $k^{th}$ execution ended at position $s_k$. The average $\bar{s}$ of these final positions is shown in black. Along each grey line, the dots indicate where the measurements of attitude and configuration $\Phi(s)_\text{train}$ are obtained to populate each $\{\Phi(s)_\text{train}\}_{s \in S_a}$ set. Note that to train in a variety of terrain profiles, for each action the process is repeated using different initial positions $s$.

6.1 Training Data

In this first training strategy, Learning from Perception (LfP), the attitude and configuration data used for training the GPs are obtained from proprioception, i.e. from the onboard measurements of the IMU (for pitch and roll) and the potentiometers for the internal chassis angles $\alpha_i$ (see Fig. 7). During training, for each execution of an action we recorded: a) the set of attitude and configuration angles $\{\Phi(s)_\text{train}\}_{s \in S_a}$ observed by these proprioceptive sensors, where $S_a$ is the set of $M$ discretised positions of the rover during the $k^{th}$ execution of action $a$, and b) the outcome of the action execution: $s'_k$, given by the localisation system (see Fig. 8). For each action $a$, the average deviation $\Delta s_a$ needed in the GPs (see Sec. 3.1) was then obtained by computing the average of differences between each couple of initial and final positions $(s, s')$, i.e. $\Delta s_a = s'_a - s_a$.

To collect these training data, we performed more than 600 action executions over a large variety of terrain profiles. Approximately half of the actions were executed on rigid terrain (308), and the other half on deformable terrain (297). The following section describes how we determined the features $\lambda(s,a)$ to be computed from each $\{\Phi(s)_\text{train}\}_{s \in S_a}$ set.

6.2 Features describing the terrain profile: $\lambda(s,a)$

The $\lambda$ features are used: a) offline to train the GP ($\lambda_{\text{train}}$), and b) online to evaluate the mobility prediction ($\lambda_*$), see Fig. 7. They should capture the most informative part of the data contained in each $\{\Phi(s)\}_{s \in S_a,k}$.
set. To determine the best features to use, we performed a statistical analysis on a 70% of the data collected during training. The remaining 30% were used for cross-validation.

We first determined empirically that using only the attitude angles (pitch $\theta$ and roll $\phi$) was sufficient for our purpose. We considered the variations of values of $\theta$ and $\phi$ over the course of an action. To evaluate those variations, $\{(\theta_i, \phi_i)\}_{i \in S_k^a}$ were evenly discretised into $M = 20$ samples along $S_k^a$, i.e. using the same sampling as for the reward function evaluation (see Sec. 5.6). Thus, in the training data, to each execution $a$ of action $a$ corresponds a set of pitch and roll angles $\{(\theta_i, \phi_i)\}_{i \in \{1, M\}}$, respectively).

We considered 22 candidate features, including mean value of pitch and roll (representing the mean slope of the terrain profile), range of values (the overall change in slope), minimum and maximum values (the max/min in the slope profile), minimum and maximum continuous increase and decrease (the max/min continuous change in slope), minimum and maximum rates of change (the overall roughness of the terrain), minimum and maximum marginal increase and decrease (the incremental roughness of the terrain), and squared maximum increase and decrease of $\theta$ and $\phi$ values.

To determine the most informative features, we performed a Principal Component Analysis (PCA) over these 22 features. This analysis identified 6 features that together represent 90% of the data. Therefore, in this paper, $\lambda$ is a 6-dimensional vector:

$$\lambda \triangleq \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}. \quad (18)$$

where the $\lambda_i$ components are described below. $\lambda_1$ and $\lambda_2$ represent the range of $\phi_i$ and $\theta_i$ values, respectively:

$$\lambda_1 = \max_i(\phi_i) - \min_i(\phi_i) \quad (19)$$
$$\lambda_2 = \max_i(\theta_i) - \min_i(\theta_i). \quad (20)$$

$\lambda_3$ and $\lambda_4$ are the mean values of $\phi$ and $\theta$, respectively, which are indicative of the mean slope of the terrain:

$$\lambda_3 = \frac{\sum_{i=1}^{M} \phi_i}{M} \quad (21)$$
$$\lambda_4 = \frac{\sum_{i=1}^{M} \theta_i}{M}. \quad (22)$$

$\lambda_5$ and $\lambda_6$ represent the maximum rate of change in $\phi$ and $\theta$ during an action execution. This reflects the sharpness of the bumps in the terrain. This can be written as:

$$\lambda_5 = \max (\phi_j - \phi_i)/(j-i), \ \forall (i,j) \in [1, M]^2, i < j \quad (23)$$
$$\lambda_6 = \max (\theta_j - \theta_i)/(j-i), \ \forall (i,j) \in [1, M]^2, i < j. \quad (24)$$

6.3 Training Results

In this section, we illustrate the training data we collected and used to train our GPs. Figure 9(a-e) show the corresponding distributions of $\Delta s_{head}$ values for all terrain profiles experienced during training, i.e. the raw distributions $p(\Delta s_{i\cdot a}|a)$. Note that in practice, due to the left-right symmetry of the platform, we combined the training data obtained for symmetric actions, which means that in the final training data set only 6 of the 10 actions in $A$ have distinct distributions.

The executions of actions crab$(0\pi)$ and crab$(\pi)$, i.e. crabbing straight forwards and backwards, respectively, are the most consistent of the crabbing actions. This could be explained by the fact that the rocker-bogie chassis is designed to traverse obstacles efficiently when attacking them head on. Executing crabbing actions at $\pm \pi/2$, i.e. attacking obstacles on the side, leads to slightly less consistent outcomes. The most inconsistent executions among the crabbing actions are obtained for crab$(\pm\pi/4)$ and crab$(\pm3\pi/4)$. 


Figure 9: Training Data. (a-e) Distributions of $\Delta s_{\text{head}}$ for each crabbing action. (f) Distribution of $\Delta s_{\text{yaw}}$ for rotating actions.

Figure 9(f) shows the distribution of $\Delta s_{\text{yaw}}$ values for the rotating actions. We can see that the executions of these actions, shown as a single distribution because of the symmetry of the platform, are highly inconsistent. In most cases the rotation achieved was smaller than the target, and in extreme cases the rotation actually achieved was negligible. This happened when the wheels of the rover were right next to rocks that prevented the rover from achieving the rotation on the spot.

Figure 10 illustrates the distributions of $\Delta s_{\text{dist}}$ for all training profiles. We can observe that the inconsistencies in the actual distance travelled for each action are significant. Note that in some extreme cases, the recorded distance travelled was close to 0 m. This happened when a particularly challenging piece of terrain was located close to the initial position of the rover, and it was not able to overcome this situation in some of the action execution attempts.

Table 1 gives the means of the $\Delta s_i$ components for each action over all terrain profiles in the training data. It can be seen that the mean heading angle obtained for the multiplied executions of $\text{crab}(0\pi)$ and $\text{crab}(\pi)$ is close to the expected angles (0 and $\pi$), even across multiple terrain profiles. However, the mean outcomes of the other crabbing actions deviate further from their targets because of the difficulties encountered when traversing rough terrain profiles$^1$. The mean values of $\Delta s_{\text{dist}}$ show that for all crabbing actions, in average, the rover travels a shorter distance than intended, mostly because of rough and difficult terrain profiles. In particular, the average distance travelled is smaller for crabbing actions on the side ($\pm \pi/2$) and in diagonal ($\pm \pi/4$ and $\pm 3\pi/4$) than when the rover moves straight, whether forwards ($0\pi$) or backwards ($\pi$), since the platform performs better when travelling over rocks in the latter cases.

$^1$recall that the actions were calibrated on flat terrain
6.4 Learned Mobility Prediction Model

To illustrate how different the distributions of action outcomes can be for different terrain profiles, we computed the predictive distributions of \( \Delta s_{\text{dist}} \) for three specific terrain profiles using the training data and the GP for a particular action: \( a = \text{crab}(\pi/4) \). We chose three specific terrain profiles, i.e. three instances of \( \lambda \), in positions in the feature space where the density of training points was relatively high. Profile 1 represents relatively flat terrain: \( \lambda_1 = \{-0.13^\circ, -3.61^\circ, -0.13^\circ, -0.07^\circ, -0.06^\circ, -1.65^\circ\} \). For Profile 2, \( \lambda_2 = \{-1.15^\circ, -3.44^\circ, -0.57^\circ, -8.59^\circ, -2.86^\circ, -2.86^\circ\} \), which represents terrain that has a mean slope downhill. Profile 3 corresponds to terrain with a mean slope downhill towards the right. \( \lambda_3 = \{-1.10^\circ, 1.47^\circ, 6.45^\circ, 8.09^\circ, -0.86^\circ, 2.66^\circ\} \). \( \lambda_1 \), \( \lambda_2 \) and \( \lambda_3 \) represent terrain profiles of increasing difficulty for the rover. Table 2 shows the mean and standard deviation of the predicted distribution of \( \Delta s_{\text{dist}} \) for these \( \lambda \) values. We can see that, as the terrain becomes increasingly difficult, the mean of the estimated \( \Delta s_{\text{dist}} \) decreases, i.e. we predict the rover will travel shorter distances in average. Furthermore, the standard deviation increases, i.e. the prediction has more uncertainty since empirically the action outcomes were more spread.

<table>
<thead>
<tr>
<th>Action:</th>
<th>( \text{crab}(0\pi) )</th>
<th>( \text{crab}(\pm \pi/4) )</th>
<th>( \text{crab}(\pm \pi/2) )</th>
<th>( \text{crab}(\pm 3\pi/4) )</th>
<th>( \text{crab}(\pi) )</th>
<th>( \text{rotate}(\pm \pi/4) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta s_{\text{head}} ) (rad)</td>
<td>-0.0601</td>
<td>0.5248</td>
<td>1.1579</td>
<td>1.4525</td>
<td>3.1979</td>
<td>-</td>
</tr>
<tr>
<td>( \Delta s_{\text{dist}} ) (m)</td>
<td>0.2213</td>
<td>0.1670</td>
<td>0.1898</td>
<td>0.1861</td>
<td>0.2167</td>
<td>-</td>
</tr>
<tr>
<td>( \Delta s_{\text{yaw}} ) (rad)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.4057</td>
</tr>
<tr>
<td># samples</td>
<td>59</td>
<td>135</td>
<td>134</td>
<td>87</td>
<td>81</td>
<td>109</td>
</tr>
</tbody>
</table>

Figure 10: Training data. Distributions of \( \Delta s_{\text{dist}} \) for each action.
Table 2: Predicted mean and standard deviation of $p(\Delta s_{\text{dist}}|\lambda, a)$, for $a = \text{crab}(\pm \pi/4)$.

<table>
<thead>
<tr>
<th>Profile: $\lambda^* = \lambda^1 \ast \lambda^2 \ast \lambda^3$</th>
<th>$\lambda^1$</th>
<th>$\lambda^2$</th>
<th>$\lambda^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Mean $\mu_*$ (in m)</td>
<td>0.2635</td>
<td>0.2223</td>
<td>0.1637</td>
</tr>
<tr>
<td>Predicted Std. $\sigma_*$ (in m)</td>
<td>0.0038</td>
<td>0.0069</td>
<td>0.0102</td>
</tr>
</tbody>
</table>

6.5 Experimental Results

It is difficult to realistically simulate the actual errors generated by control action executions, especially in the case of possible terrain deformations. Therefore, it is necessary to experimentally validate the approach with a realistic rover on unstructured terrain. We performed a series of such tests with the experimental system described in Sec. 5.

We performed tests that vary across two dimensions: a) type of control uncertainty, and b) type of terrain. For control uncertainty, we considered three cases with heading and distance uncertainty treated independently as mentioned in Sec. 4.1. In the first case, uncertainty in distance (for crab actions) and in yaw (for rotation actions) was predicted according to the mobility prediction model, and heading was treated deterministically. Likewise in the second case, uncertainty in heading and yaw was predicted by the mobility model, and distance was treated deterministically. In the third case, heading, distance and yaw were all treated deterministically for comparison. For all deterministic predictions, the expected outcome was taken as the average action outcome in the training data, i.e. $\Delta s_a$. For the terrain type, we considered both rigid and deformable terrain.

Prior to testing, we placed the rover in the test environment and built a DEM from a single point cloud acquired by the depth sensor. From this DEM, the cost map and the $\lambda$ features used for mobility prediction were computed (see Fig. 5). For each test, the rover was placed at a known starting position within the DEM. We then generated and executed a policy with respect to a given goal region, also contained within the DEM. All computation was performed onboard the rover.

Multiple executions of each permutation of test conditions were performed. All tests with rigid terrain were performed with a fixed start and goal position within the test environment. All tests with deformable terrain were also performed with fixed start and goal positions, but in a different area of the test environment. Results are presented and discussed for both terrain types, representing over 100 trials in total.

6.5.1 Rigid Terrain

The first area used for the experimental validation was composed of rigid terrain only, i.e. the terrain never deformed significantly due to interaction with the rover. To illustrate this environment, Fig. 11 shows the cost map with example policies overlaid. It can be observed that when uncertainty was considered, the policy is smoother. During policy execution, the average distance travelled was 4.4m with no control uncertainty, 4.1m with distance uncertainty, and 4.2m with heading uncertainty.

Table 3 summarises results. Each run corresponds to one full trajectory executed by the rover. The policy for each experimental condition was fixed; the performance variation is due to control uncertainty during execution. The table details:

- the number of successful runs, i.e. when the rover was able to reach the goal, including runs in which the rover was temporarily stuck (i.e. at least one action execution resulted in no motion; this was observed by an operator),
- the number of failed runs, when the rover did not reach the goal. This happened, for example, when the rover was unable to run over a particular rock in practice. This is also expressed as percentage of the total number of runs, in parentheses.
Figure 11: Example of policies computed with the LfP method, without control uncertainty (a), and accounting for Heading & Yaw uncertainty (b). The policies shown are $\pi^*(x, y, 0)$, i.e. for yaw=0. The policies were computed using the full state resolution but are displayed using a coarser resolution of $0.2m \times 0.2m$ for clarity.

Table 3: Summary of all experimental runs using the LfP method on rigid terrain

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>Total runs</th>
<th>Successful runs (temporarily stuck)</th>
<th>Failed runs (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>24</td>
<td>17 (12)</td>
<td>7 (29%)</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>24</td>
<td>17 (8)</td>
<td>7 (29%)</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>21</td>
<td>17 (8)</td>
<td>4 (19%)</td>
</tr>
</tbody>
</table>

Although the number of failed runs is comparable for the methods without uncertainty and with uncertainty in distance, accounting for heading uncertainty significantly reduced this number. Note that the same test for each method was run until the number of successful runs for each method was the same (17), so that the statistics on cost and number of actions below could be comparable.

Table 4 summarises the statistics of actual total cost obtained for the trajectories executed by the rover during the aforementioned successful runs on rigid terrain. The table shows the mean and standard deviation of the total cost accumulated along the path executions, computed over all runs, as well as the cost reduction obtained by considering uncertainty compared with the method without control uncertainty. Figure 12 provides a decomposition of the total cost into its two components: the cost due to terrain only, and the number of actions executed by the rover to reach the goal. Note that the terrain cost was always calculated on the actual executed paths using the measurements from the IMU and the potentiometers onboard the rover.

Table 4: Cost statistics for LfP experimental results on rigid terrain

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>No. of runs</th>
<th>Cost mean</th>
<th>Cost std</th>
<th>Cost reduction</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>17</td>
<td>0.1653</td>
<td>0.0783</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>17</td>
<td>0.1336</td>
<td>0.0658</td>
<td>19.18%</td>
<td>0.0475</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>17</td>
<td>0.1138</td>
<td>0.0474</td>
<td>31.16%</td>
<td>0.0034</td>
</tr>
</tbody>
</table>
These experimental results on rigid terrain validate the proposed approach. When control uncertainty is considered in the planning, the rover performs substantially better than when following policies computed without considering uncertainty. The table shows that the terrain cost of the executed path is significantly reduced, by 19% in average when considering distance uncertainty and by 31% with heading uncertainty. The mean number of actions executed by the rover to reach the goal is also significantly reduced, showing that the execution was more efficient. In addition, the standard deviation of this number of actions was also clearly reduced, which indicates that the execution was more consistent. Therefore, considering uncertainty yielded improved results in terms of safety (terrain cost), and efficiency (number of actions). It can also be observed that considering heading uncertainty in planning provided better results than considering uncertainty in distance travelled.

To evaluate the statistical significance of our experimental results, we performed the significance test from (Moore and McCabe, 1993). Our “null hypothesis” $H_0$ is that the average cost obtained when considering uncertainty, $\mu$, is equal or superior to that obtained when no uncertainty is considered, $\mu_0$, i.e. $H_0 : \mu \geq \mu_0$. The significance test evaluates the evidence against this null hypothesis by computing the $P$-value. In this study the $P$-value is the probability, assuming $H_0$ is true, that the method accounting for uncertainty would provide an average cost $\bar{x}$ lower than, or equal to, that actually observed in our experiments. “The smaller the $P$-value, the stronger the evidence against $H_0$ provided by the data” (Moore and McCabe, 1993). Therefore, the $P$-value is the probability: $P = P(Z \leq z)$, where $Z$ is a standard normal random variable and $z$ is the standardised sample mean.

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}},$$  \hspace{1cm} (25)

where $\bar{x}$ is the sample mean ($\bar{x} = 0.1138$ when considering heading uncertainty), $\mu_0$ is the reference mean, obtained with no uncertainty ($\mu_0 = 0.1653$), $\sigma$ is the standard deviation of the reference ($\sigma = 0.0783$), and $n$ is the number of samples used to compute $\bar{x}$.

The $P$-values obtained are shown in the last column of Table 4. We find that there is very strong evidence that the mean cost obtained when considering heading uncertainty is in fact lower than when no uncertainty is considered (the data are statistically significant at level $\alpha = 0.34\%$). However, the evidence is not as strong for the method considering distance uncertainty, with a statistically significance level of $\alpha = 4.75\%$.

Figure 13 illustrates a subset of the multiple executed paths, shown over the cost map of the rigid terrain area used for these experiments. For clarity, we only show seven representative paths for each method. We
Table 5: Summary of all experimental runs of LfP method on deformable terrain

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>Total runs</th>
<th>Successful runs (temporarily stuck)</th>
<th>Failed runs (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>19</td>
<td>17 (7)</td>
<td>2 (10%)</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>20</td>
<td>17 (5)</td>
<td>3 (15%)</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>19</td>
<td>17 (6)</td>
<td>2 (10%)</td>
</tr>
</tbody>
</table>

Table 6: Cost statistics for LfP experimental results on deformable terrain

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>No. of runs</th>
<th>Cost mean</th>
<th>Cost std</th>
<th>Cost reduction</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>17</td>
<td>0.065</td>
<td>0.0321</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>17</td>
<td>0.0591</td>
<td>0.025</td>
<td>9.08%</td>
<td>0.2266</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>17</td>
<td>0.0621</td>
<td>0.0161</td>
<td>4.46%</td>
<td>0.3557</td>
</tr>
</tbody>
</table>

can observe that considering uncertainty led to smoother executed paths in average. Moreover, the executed trajectories appear to be spatially more consistent when considering control uncertainty, i.e. the variance of rover positions between trajectories was smaller. This may be because, when considering control uncertainty, the planner often preferred actions that have a smaller variance of $\Delta s$ in the training data, and in general, a smaller variance was obtained in areas with less terrain cost (i.e. less difficult terrain, or lower variations of rover attitude angles). When considering uncertainty, the rover steered clear of areas where the terrain cost is high. However, in the deterministic case the policy simply encodes least-cost paths and thus directed the rover to traverse dangerously close to areas of high cost.

6.5.2 Deformable Terrain

We also conducted similar experiments on deformable terrain. Since the terrain can change during each traversal of the rover, the environment was carefully reverted back to its original condition at the end of each run. The average traversal length during execution was 3.8m for the deterministic case, 2.8m with distance uncertainty, and 3.1m with heading uncertainty.

Table 5 summarises the experimental runs. We can observe that although considering control uncertainty using the LfP method led to a slight reduction in the number of temporarily stuck states, it did not reduce the number of failed runs. Table 6 shows the statistics of the experimental results obtained on deformable terrain, including the P-values of the significance test, obtained as described above. Figure 14 specifies statistics of the terrain cost and number of actions. Although considering control uncertainty also led to a slight reduction of terrain cost and number of executed actions in average, the improvement over the method with deterministic control is not statistically significant compared with the experiments conducted on rigid terrain.

6.6 Discussion

Although considering control uncertainty in the planning using the proposed LfP approach yielded improved results in average, the experimental results also showed that the impact was significantly smaller in the case of deformable terrain than in the case of rigid terrain. This indicates that whilst the LfP approach is efficient on rigid terrain, it has its limits on deformable terrain. This may be explained by inconsistencies between the observations of variations of platform configurations $\{\Phi\}$ along an action execution, represented by the $\lambda$ features, in the training phase and in the prediction phase, i.e. between $\lambda_{train}$ and $\lambda_{*}$.

The predictions need to be made before the rover actually traverses the terrain. Therefore, in our approach
Figure 13: Example of rover trajectories obtained by following the policies generated by the LfP method with: (a) no control uncertainty considered, (b) distance & yaw uncertainty, and (c) heading & yaw uncertainty. The trajectories are shown as coloured lines over the cost map. The colour bar illustrates the cost values found in the map. The dotted white box represents the goal region.
the $\lambda$, are obtained from exteroceptive data (interpreted by a configuration predictor, see Fig. 5). Conversely, in the LfP method the $\lambda_{\text{train}}$ are computed from proprioceptive data gathered by the rover while on the terrain (see Fig. 7). A necessary condition to have a perfect match between the $\lambda$ vectors obtained from exteroception and proprioception is that the terrain geometry observed by the exteroceptive sensors before rover traversal corresponds perfectly to the terrain during rover traversal. This condition can only be verified if the terrain is rigid, i.e. no terrain deformation occurs upon traversal.

Although it is common in the literature of planning for ground vehicles to assume that the terrain is rigid, this assumption is often invalid in the case of a rover evolving on rough unstructured terrain. This is particularly true in a Mars-analogue environment, where in some cases, the terrain is known to be highly deformable. Examples of such scenario include: small rocks that have a tendency to dislodge, and loose soil that shifts or compacts upon impact. Deformation is extremely hard to predict using a pre-defined model, therefore, it is preferable to learn from past experience.

The next section proposes an alternative method to train the GPs used for mobility prediction, which aims at mitigating the inconsistencies between the prediction and training. We will show that this makes the proposed planning approach efficient and reliable both in rigid and in deformable terrain.

# 7 Learning Mobility Prediction from Exteroception

This section discusses and evaluates the proposed approach with an alternative training strategy, Learning from Exteroception (LfE), where the training data is collected from exteroception. Section 7.1 first describes the training process involved in LfE. Section 7.2 specifies the features $\lambda(s, a)$. In Sec. 7.3 we propose an experimental validation of LfE on rigid terrain and deformable terrain, respectively. The performance of the LfE method will be compared to that of LfP in Sec. 8.

## 7.1 Training Data

Figure 15 illustrates the training strategy used in this method, which also involves executing each control action multiple times. However, unlike for the LfP method, the attitude and configuration data used for training the GPs were obtained from exteroception. We first built a DEM of the area where the multiple
executions of actions were performed by the rover. Then, the locations of the rover during the $k^{th}$ execution of action $a$, i.e. $\{s \in S^k_a\}$, were recorded from the localisation system, including the final action outcomes $s'^k_a$. The executed path was then discretised into $M = 20$ samples (as per the LfP method) and the corresponding set of attitude and configuration angles at these discrete locations, $\{\Phi(s)_{\text{train}} \}_{s \in S^k_a}$, was calculated based on the expected terrain profile obtained from exteroception, using the configuration predictor (see Sec. 5.5) on the DEM of the training area (see Fig. 16). We used the same configuration predictor as the one used to compute $\lambda_s(s,a)$ for planning (see Fig. 5).

Therefore, in this LfE method, the GPs are trained based on the terrain profile that was seen ahead of the rover prior to the action execution, whereas the LfP method was using the terrain profile as observed by the rover during the action execution. Note that for any execution of action $a$ starting from the same position $s$ the set $\{\Phi(s)_{\text{train}} \}_{s \in S^k_a}$ is the same, whereas the outcome $s'^k_a$ obtained can vary for each $k^{th}$ execution. The following sub-section describes how the features $\lambda(s,a)_{\text{train}}$ are obtained from each $\{\Phi(s)_{\text{train}} \}_{s \in S^k_a}$ set in this LfE method.

7.2 Features $\lambda(s,a)$

To determine the $\lambda$ features, we performed the same analysis with PCA as for the LfP method, with the new training data obtained from exteroception. We found that the most appropriate features to use were the same 6 features as defined in Sec. 6.2:

$$\lambda_{\text{LfE}} \triangleq \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}.$$ (26)

7.3 Experimental Results

Using the same experimental validation strategy as for the LfP method (in Sec. 6) we performed over 150 experimental runs. Once again we compare the performance of methods planning with stochastic control (using the LfE training data) with the reference method with deterministic control, over both rigid and deformable terrain. Note that the experiments reported in Secs. 7.3.1-7.3.2 were conducted on the same areas as for the LfP method, and with the same goal and initial position of the rover. Therefore, for a given category of terrain (rigid or deformable), the results obtained with the LfP and LfE methods can be compared.

7.3.1 Rigid Terrain

Figure 17 illustrates example policies generated by the LfE-based approach. Similar to Fig. 11, we can observe that when uncertainty is considered the policies appear to be smoother.

---

**Figure 15:** LfE training process outline. The GPs are trained using exteroceptive data interpreted by the configuration prediction.
Figure 16: An example of training data collected for GP training using the LfE method (top view). In training, the rover achieves multiple executions of action $a$ starting from position $s$, resulting in different executed paths, shown in grey. The actual path for the $k^{th}$ execution ended at position $s'_k$. The average $\bar{s}$ of these final positions is shown in black. The dots along the black line indicate where the attitude and configuration angles $\Phi(s)_{\text{train}}$ of the rover are evaluated from exteroception, using the configuration predictor on the DEM of the area.

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>Total runs</th>
<th>Successful runs (temporarily stuck)</th>
<th>Failed runs (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>22</td>
<td>17 (8)</td>
<td>5 (23%)</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>19</td>
<td>17 (5)</td>
<td>2 (10%)</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>19</td>
<td>17 (2)</td>
<td>2 (10%)</td>
</tr>
</tbody>
</table>

Table 7 summarises the experimental runs executed using the LfE-based method on rigid terrain. In particular it shows the number of successful runs (the robot reached the goal) and failed runs (where the rover was indefinitely stuck). The average traversal length during execution was 4.4$m$ for the deterministic case, 3.9$m$ with distance uncertainty, and 3.6$m$ with heading uncertainty. It can be noted that accounting for control uncertainty led to fewer failures than with deterministic control. Moreover, the number of runs with temporarily stuck states, within the successful runs, has decreased when incorporating uncertainty into the planning. The LfE-based method also failed fewer times overall than the LfP method in the same experimental conditions. Note that, as in Sec. 6, the same test for each method was run until the number of successful runs for each method was the same (17), so that the statistics on cost and number of actions below could be comparable.

Table 8 summarises the statistics of actual total cost obtained for the trajectories executed by the rover during the aforementioned successful runs on rigid terrain. The table shows that a substantial reduction of the overall cost was obtained by accounting for uncertainty: more than 25% when considering uncertainty in distance, and 41% when considering uncertainty in heading. This is to compare with the 19% and 31% reduction previously obtained with the LfP method (see Table 4). As indicated by the P-values, the cost reduction can be considered statistically significant, especially in the case of heading uncertainty.

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>No. of runs</th>
<th>Cost mean</th>
<th>Cost std</th>
<th>Cost reduction</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>17</td>
<td>0.1074</td>
<td>0.0560</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>17</td>
<td>0.0800</td>
<td>0.0413</td>
<td>25.51%</td>
<td>0.0217</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>17</td>
<td>0.0630</td>
<td>0.0249</td>
<td>41.34%</td>
<td>0.0005</td>
</tr>
</tbody>
</table>
Figure 17: Example of policies computed with the LfE method; without control uncertainty (a), and accounting for heading & yaw uncertainty (b). The policies shown are $\pi^*(x, y, 0)$, i.e. for yaw=0. The policies were computed using the full state resolution but are displayed using a coarser resolution of $0.2m \times 0.2m$ for clarity.

Table 9: Summary of all experimental runs of LfE method on deformable terrain

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>Total runs</th>
<th>Successful runs (temporarily stuck)</th>
<th>Failed runs (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>20</td>
<td>17 (6)</td>
<td>3 (15%)</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>22</td>
<td>17 (5)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>21</td>
<td>17 (4)</td>
<td>0 (0%)</td>
</tr>
</tbody>
</table>

Figure 18 specifies statistics of the terrain cost and number of actions. It shows a significant reduction in both terrain cost and number of actions obtained by accounting for uncertainty, compared with the method using deterministic control. The figure also shows that these costs accumulated along the executed paths were smaller with the LfE-based method than with the LfP-based method.

Figure 19 shows a subset of seven executed paths. By comparing Fig. 19 with Fig. 13, it can be seen that the executed paths for the LfE method did not traverse through the narrow pass situated at [6, -2] contrary to the LfP method. The cost associated with traversing through the narrow passageway may be low, however, if control execution precision is not as accurate then the executed path will have a much larger cost as the probability of traversing into an area of high cost (i.e. the rock slab) has increased. It should be noted that when uncertainty was considered, small rocks were not avoided. The rover traversed over rocks that it perceived as small and within its ability to overcome.

7.3.2 Deformable Terrain

The experimental runs executed on deformable terrain are summarised in Table 9. In these experiments, the LfE-based method was able to eliminate all failures (no failed runs) by accounting for control uncertainty, and therefore clearly outperformed the LfP-based method in terms of reliability.

The average length of the executed paths considering no uncertainty was 3m, considering distance uncertainty was 2.8m, and considering heading uncertainty was 3.2m. Table 10 summarises the statistics of actual total cost obtained for the trajectories executed by the rover during the successful runs on deformable terrain.
Table 10: Cost statistics for LfE experimental results on deformable terrain

<table>
<thead>
<tr>
<th>Uncertainty considered</th>
<th>No. of runs</th>
<th>Cost mean</th>
<th>Cost std</th>
<th>Cost reduction</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>17</td>
<td>0.0594</td>
<td>0.0147</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Distance &amp; Yaw</td>
<td>17</td>
<td>0.0521</td>
<td>0.0166</td>
<td>12.29%</td>
<td>0.0202</td>
</tr>
<tr>
<td>Heading &amp; Yaw</td>
<td>17</td>
<td>0.0498</td>
<td>0.0121</td>
<td>16.16%</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

The table shows that the LfE methods led to a clear reduction of the overall cost: more than 12% when considering uncertainty in distance, and 16% when considering uncertainty in heading. This is to compare with the reductions that were obtained with the LfP method (9% and 4%, respectively, see Table 6). Again the results can be considered statistically significant, especially when considering heading uncertainty.

Figure 20 specifies statistics of the terrain cost and number of actions. The reduction in terrain cost obtained by accounting for control uncertainty is clear, however, the number of actions executed is comparable to that obtained with deterministic control. This means that the method with stochastic control is clearly safer overall, and is comparably efficient.

7.3.3 Whole Mars Yard

Additional experiments were performed in a larger area, of dimensions $14m \times 6m$, on terrain composed of both rigid and deformable sections. In these experiments, we compare the results obtained by following policies computed using the LfE-based method accounting for heading and yaw uncertainty with those obtained using the reference method with deterministic control. Figure 21 shows the trajectories executed by the rover when starting from six different positions.

To quantify and compare the performances of the two methods, multiple runs were executed using a single pair of starting position and goal region, as previously. The results are summarised in Table 11. In these experiments, the number of failures reduced from 47% to 23% when heading uncertainty was considered, demonstrating increased reliability. Note that the rate of failures remains relatively high because the goal was purposely placed in a challenging area of the yard.

The trajectories obtained for the 10 successful runs for each method are shown in Fig. 22. The average length...
Figure 19: Example of rover trajectories obtained by following the policies generated by the LfE method (coloured lines) with: (a) no control uncertainty considered, (b) distance & yaw uncertainty, and (c) heading & yaw uncertainty. The dotted white box represents the goal region.
of these executed paths was 16.27m with no uncertainty, and 16.40m when considering uncertainty in heading and yaw. Along these paths, the rover experienced the following ranges of attitude and configuration angles: -15 to 15° roll, -15 to 10° pitch, and -17 to 17° internal chassis angles ($\alpha_1, \alpha_2$). One of the main differences between the paths of the two methods can be seen in the rocky area at $x \in [7, 9]$, where the rover often faced difficult terrain in practice when using the method without uncertainty, because of unexpected imperfect execution. In contrast, when control uncertainty was accounted for in the planning, the rover avoided this area since this imperfect execution was expected.

Table 12 summarises the statistics of actual total cost obtained for the trajectories executed by the rover during the successful runs on the whole Mars Yard. The table shows that considering heading uncertainty led to a reduction of the overall cost by 41% in average. This result is highly significant statistically, as shown by the very small P-value obtained. Note that the amplitude of the cost reduction is much stronger than that obtained in the smaller scale experiments conducted in deformable terrain (16% in Table 10). This suggests that the impact of the method is likely to be stronger as the goal is located further away from the starting point of the rover.

Figure 23 specifies statistics of the terrain cost and number of actions. The terrain cost and number of
actions executed were both clearly reduced by accounting for control uncertainty. Therefore, the method with stochastic control was both safer and more efficient.

Overall it is clear that the LfE method is more reliable and safer (reduced cost) when it accounts for control uncertainty. It is also more reliable and safer than the LfP method in both rigid and deformable terrain.

8 Discussion and Conclusion

In this paper we have presented a practical approach to path planning with control uncertainty on unstructured terrain. We used Gaussian processes to build continuous, stochastic mobility prediction models from empirical data. These models provided a stochastic transition function to an MDP-based planning algorithm, which used dynamic programming to build optimal policies for the rover to execute. We presented extensive empirical results that demonstrate this approach leads to more reliable and safer navigation.

Two different methods to train the stochastic mobility prediction were implemented and demonstrated. The first used proprioceptive data (LfP), and the second used exteroceptive data (LfE). To facilitate comparison of these two methods, we collect statistics that describe the overall success of each method, across all experimental trials reported in the paper, in Table 13.

Obtaining the training data for the LfP-based method was straightforward, as it simply required to record the attitude and configuration angles of the rover during multiple execution of its actions. The experimental results of LfP showed that accounting for control uncertainty in the planning led to enhanced reliability on rigid terrain, as illustrated by the reduction of failed runs (see Table 13). Furthermore, the executed paths were safer, as evidenced by the significant cost reduction for the executed paths (19% and 31% when considering distance and heading uncertainty, respectively, shown in Table 4). In addition to the increased reliability and safety, the reduction in the standard deviation of the number of executed actions demonstrated improved consistency in the path execution (shown in Fig. 12). However, no clear benefit of accounting for control uncertainty was observed when using the LfP-based method on deformable terrain. Therefore, accounting for control uncertainty within the LfP method was only beneficial if no terrain deformation was
Figure 22: Example of rover trajectories (coloured lines) obtained by following the policies generated by the LfE method with: (a) no control uncertainty considered, and (b) heading & yaw uncertainty. The dotted white box represents the goal region.
Figure 23: Average terrain cost (a) and number of actions (b) over the paths executed by the rover for experimental runs using the LfE method on the whole Mars Yard.

Table 13: Percentage of failed runs for the LfP and LfE methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Uncertainty considered</th>
<th>Failed runs, rigid terrain</th>
<th>Failed runs, deformable terrain</th>
<th>Failed runs, whole Mars Yard</th>
</tr>
</thead>
<tbody>
<tr>
<td>LfP</td>
<td>None</td>
<td>29%</td>
<td>10%</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Distance</td>
<td>29%</td>
<td>15%</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Heading</td>
<td>19%</td>
<td>10%</td>
<td>-</td>
</tr>
<tr>
<td>LfE</td>
<td>None</td>
<td>23%</td>
<td>15%</td>
<td>47%</td>
</tr>
<tr>
<td></td>
<td>Distance</td>
<td>10%</td>
<td>0%</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Heading</td>
<td>10%</td>
<td>0%</td>
<td>23%</td>
</tr>
</tbody>
</table>

experienced by the rover.

Training with the LfE-based method was more complex than for the LfP. This method required to build terrain maps and make predictions of attitude configuration angles on multiple positions along the expected path, rather than simply recording these angles from proprioceptive sensors. However, the LfE method clearly outperformed LfP. With the LfE-based method, accounting for control uncertainty in the planning strongly enhanced reliability, as the reduction in number of failed runs was greater than when using the LfP method\(^2\) (shown in Table 13). The executed paths were also safer, as shown by the cost reduction of 25% and 41% for executed paths considering distance and heading uncertainty, respectively (Table 8). In addition, the paths executed were more consistent, especially when considering heading uncertainty, as demonstrated by the reduced standard deviation in the number of executed actions to reach the goal (Fig. 18). On deformable terrain the LfE-based method also showed enhanced reliability and safety when accounting for uncertainty, as evidenced by the strong reduction of failed runs (Table 9) and a clear cost reduction for executed paths (12% and 16%, shown in Table 10). These results were even stronger for longer runs, with a cost reduction of 41% and a reduction of the number of failed runs by half obtained when considering heading uncertainty, as shown in Sec. 7.3.3. Therefore, accounting for control uncertainty within the LfE method proved beneficial in all the situations considered, including when the rover experienced terrain deformation. These results can be explained by better consistency between the predicted features $\lambda$, used in planning and the features $\lambda_{\text{train}}$ extracted from the training data. In conclusion, the results in this paper empirically validate the importance of coupling terrain profile features with control uncertainty.

Although accounting for control uncertainty in the planning allowed for a significant reduction of failed runs,\(^2\) this comparison is valid because the experimental tests have been run in the same conditions for both methods.
experimental runs, especially using the LfE method, some failures remained when the rover had to traverse challenging terrain. In most cases these failures were due to wheel slippage preventing a wheel from overcoming a rock. Slip is not captured directly in our cost function, but instead as control uncertainty in the stochastic transition model. Therefore, the rover is not necessarily avoiding such areas if it was capable of traversing comparable areas sufficiently often in the training phase. The rover may be aware that a particular action has some probability of falling short, but not that this may actually correspond to it being stuck for indefinite time. In future work, we may consider incorporating the risk of extreme slip in the cost function, and explicitly account for the risk of overall failures in the planning stage.

So far, we have considered individual dimensions of control outcomes independently, and shown that heading and distance uncertainty both contribute to safer navigation. In future work, we will consider the case where these predictions are correlated. This case is a natural extension of our learned mobility prediction model, but will require a more complex regression technique. One possibility for this purpose is multi-output GP (Boyle and Frean, 2005). We will also consider uncertainty in the cost map and the localisation, in addition to the control uncertainty. A specific study of the trade-off between training and performance will also be conducted. Finally, to facilitate the implementation of such approach in practice when the rover has to be deployed in terrain whose type is a priori largely unknown, we will consider online training strategies.

Acknowledgments

This work was partly supported by the Australian Government’s DIISR grant “Australian Space Research Program”, the Australian Centre for Field Robotics (ACFR) and the New South Wales State Government. This material is also based on research partially sponsored by the Air Force Research Laboratory, under agreement number FA2386-10-1-4153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References


