Stochastic mobility-based path planning in uncertain environments

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/IROS.2009.5354418">http://dx.doi.org/10.1109/IROS.2009.5354418</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Sat Dec 29 02:28:39 EST 2018</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/59445">http://hdl.handle.net/1721.1/59445</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.
Stochastic Mobility-based Path Planning in Uncertain Environments

Gaurav Kewlani, Genya Ishigami, Karl Iagnemma

Abstract — The ability of mobile robots to generate feasible trajectories online is an important requirement for their autonomous operation in unstructured environments. Many path generation techniques focus on generation of time- or distance-optimal paths while obeying dynamic constraints, and often assume precise knowledge of robot and/or environmental (i.e. terrain) properties. In uneven terrain, it is essential that the robot mobility over the terrain be explicitly considered in the planning process. Further, since significant uncertainty is often associated with robot and/or terrain parameter knowledge, this should also be accounted for in a path generation algorithm. Here, extensions to the rapidly exploring random tree (RRT) algorithm are presented that explicitly consider robot mobility and robot parameter uncertainty based on the stochastic response surface method (SRSM). Simulation results suggest that the proposed approach can be used for generating safe paths on uncertain, uneven terrain.

I. INTRODUCTION

AUTONOMOUS mobile robots are increasingly being used for operation on uneven, rugged terrain. A fundamental requirement for robots in such environments is the capacity to quickly generate a feasible trajectory that results in safe, rapid traversal while avoiding obstacles. This path planning capability is therefore critical to the safety and efficient operation of mobile robotic systems.

Substantial work has been performed in the field of motion planning over the years. Major techniques that have evolved include the A* and D* methods [1], potential field approaches [2], the probabilistic roadmap technique [3] and the rapidly-exploring random tree (RRT) algorithm [4]. These methods determine suitable control inputs to move a robot from its initial position to its destination while obeying physics-based dynamic models and avoiding obstacles in the environment. Recently, randomized approaches to kinodynamic motion planning [5] have proven to be a very efficient tool for the purpose of path generation, with RRTs proving to be a highly effective framework.

Since its introduction, many extensions to the basic RRT algorithm have been developed to improve its performance and better adapt to demands of specific systems [4]. However, little research has explicitly addressed the challenge of autonomously assessing a robot’s mobility over a given terrain region while planning a path. Consideration of robot mobility is important in field conditions, where terrain inclination, roughness, and/or mechanical properties can significantly impede robot motion. Such scenarios include planetary surface exploration, some search and rescue tasks, and many defense/security applications. Previous research has employed heuristically-biased expansion to generate efficient paths [6] while satisfying dynamic constraints. Another recent approach [7] explicitly models a robot’s closed-loop controller in the planning methodology, thereby resulting in trackable paths. However, these works do not explicitly address mobility aspects during the planning process.

Further, there has been little research that addresses the challenge of autonomously generating a path while explicitly considering uncertainty in the terrain and/or robot parameters. Most techniques rely on deterministic analysis that assumes precise knowledge of robot and terrain parameters. In field conditions, however, robots generally have access only to sparse and uncertain terrain parameter estimates, and robot parameters may be uncertain and time-varying (due to, for example, fuel consumption and mechanical wear). Failure to consider parameter uncertainty may therefore lead to failure of the robot to track generated paths, especially during high speed navigation in unstructured environments. Recently though, research work in this area has used a particle filter-based approach within the RRT framework, producing a distribution of robot states at each tree node [8], to capture uncertainty induced effects.

In summary, while many planning approaches have been developed that incorporate dynamic robot models and satisfy a variety of constraints (e.g. related to actuator physical limitations, kinematic constraints, etc.), very few explicitly consider robotic mobility in the planning process. Moreover, they typically employ a deterministic analysis and do not explicitly consider parameter uncertainty. This paper addresses these concerns through several extensions to the basic RRT algorithm that contribute to generation of a safe path over uncertain terrain.

This paper is organized as follows: Section 2 introduces the uncertainty analysis techniques employed in this work. The basic RRT algorithm is briefly presented in Section 3. This is followed in Section 4 by a description of various extensions to the RRT framework that consider robotic mobility. Section 5 discusses the integration of uncertainty analysis within the planning framework. The dynamic robot model employed for algorithm analysis is presented in Section 6 and simulation results are shown in Section 7. It is shown that safe trajectories can be generated for rapid traversal over unstructured terrain using the proposed framework.

II. UNCERTAINTY ANALYSIS TECHNIQUES

There exist numerous techniques to estimate the outputs for processes that are subject to uncertainty [9]. These may be applied to predict the ability of a robot to successfully traverse...
a given route during trajectory planning, while rigorously
considering parameter uncertainty. A traditional method for
estimating the probability density function of a system’s
output response while considering uncertainty is the Monte
Carlo method [10]. However, a large number of simulation
runs are generally required to obtain reasonable results, often
leading to high computational cost. Recently, efficient
approaches to uncertainty analysis have been introduced and
include the polynomial chaos approach [11] and the stochastic
response surface method (SRSM) [12]. The latter has been
employed in the present study and is described below.

A. Stochastic Response Surface Method (SRSM)

The stochastic response surface method involves
representing inputs and outputs of a system under
consideration via series approximations using standard
random variables, thereby resulting in a computationally
efficient means for uncertainty propagation through complex
numerical models. In this approach, the same set of random
variables that represents input stochasticity is used to express
the output(s). For normally distributed random inputs, an
equivalent reduced model for the output may be reproduced in
the form of a series expansion consisting of multi-dimensional
Hermite polynomials of normal random variables, as:

\[
y = a_0 + \sum_{i=1}^{\infty} a_i \Gamma_i(\xi_i) + \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} a_{ij} \Gamma_j(\xi_i, \xi_j) + \ldots
\]

(1)

where \(y\) refers to an output metric, \(\xi_i, \xi_{i2}, \ldots\) are i.i.d. uniform
random variables, \(\Gamma_i(\xi_i, \xi_{i2}, \ldots, \xi_{iq})\) is the Hermite polynomial
of degree \(q\) and \(a_{i1}, a_{i2}, \ldots\) are the corresponding coefficients.
For notational simplicity, the series may be written as:

\[
y = \sum_{j=0}^{N} \Phi_j(\xi)
\]

(2)

where the series is truncated to a finite number of terms and
there exists a correspondence between \(\Gamma_i(\xi_i, \xi_{i2}, \ldots, \xi_{iq})\) and
\(\Phi_j(\xi)\), and their corresponding coefficients.

The series expansion contains unknown coefficient values
that can be estimated from a limited number of model
simulations to generate an approximate reduced model. This
is achieved by choosing a set of suitable sample points
(collocation points) and generating model outputs at these
points. A regression-based approach is then utilized to obtain
the values for the unknown coefficients. Once the statistically
equivalent reduced model is formulated, it can be used to
facilitate analysis of the system under uncertainty, and obtain
relevant output statistics [12].

III. BASIC RRT ALGORITHM

The basic RRT planning algorithm can be briefly
summarized as follows: Given a robot in an initial
configuration in an environment, sample a point in space
(either randomly or according to a specified probability
distribution), then find its nearest node in the current search
tree based on an appropriate distance metric. Then,
forward-simulate a system model from the nearest node
towards the sampled point. If various constraints are satisfied,
a new location is reached and added to the search tree. A
search tree is thus constructed with a combination of random
exploration and (possibly) biased motion towards the goal,
while obeying various constraints. The algorithm terminates
when a node is selected that lies within some threshold
distance to the goal. For more details about the RRT
framework, refer to [4].

A primary advantage of this framework is that it can be
implemented for real-time, online planning, even for high
degree-of-freedom dynamic models. Further, its flexibility
allows trajectory-based checking of complex constraints and
integration of the proposed stochastic modeling approach.

IV. MOBILITY-BASED RRT EXTENSIONS

This section provides an overview of various extensions to
the basic RRT framework that aim to (implicitly or explicitly)
consider robot mobility, and thereby result in motion plans
that are safe and efficient, even over unstructured terrain.

A. Distance Metric Calculation

Most approaches to RRT-based planning employ the
Euclidean distance to calculate the distance from a node to the
sample point. However, many mobile robots employ
Ackermann (or Ackermann-like) steering, which restricts their
path following capability to following smooth paths. Here, a
distance metric similar to the Dubins path length [14] is
employed for such robots. While Dubins curves consider
paths of the CCC/CSC sequence type (where C represents a
circular arc and S refers to a straight line segment) to move
between prescribed initial and terminal robot configurations,
here paths of the CS/SC sequence type are considered, since
the robot orientation at the target point is not critical.

The proposed metric is more appropriate than a Euclidean
distance-based metric since it considers the initial robot
heading and minimum turning radius, resulting in a more
accurate estimate of the minimum path length a robot must
travel to reach a sample from a given node (see Figure 1).

To calculate this metric, the coordinates are first
transformed such that the node of interest (i.e. the potential
nearest node) lies at the origin. Then, based on the location
and orientation of the robot at a node, the targeted sample
point and the minimum turning radius of the robot \(\rho\), the
Dubins-like distance calculations are performed [15]. It
should be noted that these calculations rely on a simple
kinematic robot model, and thus serve as an approximate for
high speed, dynamic systems.

For paths of type CS, \(D = \sqrt{x^2 + (y - \rho)^2}\) and \(L = \sqrt{D^2 - \rho^2}\),
where \(\beta = \tan^{-1}(L/\rho)\), \(\alpha = \tan^{-1}((y - \rho)/x)\), and
\(\theta = \pi/2 - (\beta - \alpha)\). Then, \(x_d = \rho \sin \theta\) and \(y_d = \rho - \rho \cos \theta\).

For paths of type SC, \(\varphi = \sin^{-1}((D \sin \alpha)/\rho)\), \(\gamma = \pi - (\varphi + \alpha)\)
and \(L = (\rho/\sin \alpha) \sin \gamma = (D/\sin \alpha) \sin \gamma\). Then, \(x_d = L\), \(y_d = 0\),
and \(\theta = (3/2)\pi - \varphi\).
B. Use of Multiple Nearest Nodes

To enhance planning algorithm performance, \( M \) (here taken as 3) nearest nodes are calculated instead of just one during tree extension. These nodes are arranged in order of increasing cost (see Section IV.C). The least cost node is then chosen for expansion, provided the resulting trajectory towards the sample point has a reasonably high probability of safe traversal. This condition is satisfied when the rollover metric (see Section VI), averaged over the path segment, has an absolute value lower than a suitable threshold value (i.e., \( R_{k_{avg},s} < R_s \)). Keeping track of \( M \) nearest nodes prevents re-searching the entire tree in case the mobility-based criterion is not satisfied for the selected node. This improves the planner’s performance in rapidly finding a safe path.

C. Mobility-based Heuristic

Costs are assigned to nodes considering both temporal and mobility-based factors. While the former takes into account the time taken to reach a particular node, the latter considers the probability of successfully negotiating the terrain to do so.

This may be defined based on a metric related to the nearness of the robot to rollover. Here a rollover metric \( R \) is used to assign cost by computing it along the path leading to a node from the start location, thereby explicitly including mobility considerations in the planning process. By using this heuristic cost function, it is expected that paths that are safely traversable by the robot will be generated. This node cost function is calculated as follows:

\[
Q_k = \prod_{j=1}^{M} \left( C_{1,k} / \max(C_{i,j}) \right)
\]

where

\[
C_{1,k} = t_k
\]

\[
C_{2,k} = (R_{avg,p,k}R_{max,p,k})^h
\]

\[
C_{3,k} = d_k
\]

Here \( t_k \) refers to the time to reach the \( k^{th} \) node from the robot’s starting position, \( R_{avg,p,k} \) and \( R_{max,p,k} \) are, respectively, the average and maximum values of \( R \) along the entire path leading up to the node, \( d_k \) is the value of the distance metric to the sample point from the node, and \( h \) is a parameter to bias the search according to the relative importance of time and vehicular mobility, and depends on the particular application.

D. Pure Pursuit Controller

Closed-loop (rather than open-loop) model simulation is integrated in the proposed RRT framework, as in [7]. Here, a controller based on the pure pursuit algorithm [16] is employed to track a reference path input from the least cost node to the sample location. The use of a closed-loop control methodology has various advantages. First, upon integration with the RRT, the technique allows the planning framework to be applied to complex dynamic models by (potentially) transforming a high-dimensional search problem through the robot’s state space to a low-dimensional search through Cartesian space. Second, it yields trajectories that, by construction, are likely to be dynamically feasible. The technique also enables generation of reasonably long paths and associated sequences of robot steering inputs.

The reference input to the closed-loop controller is the same as the Dubins-like curve described in IV.A. Note, however, that only a section of the reference path might be tracked. An illustration of this approach is depicted in Fig. 2.
uncertain terrain and/or robot parameters. Further, while traversing certain paths, the robot may collide with an obstacle, or may have a heightened possibility of rollover (as determined through the averaged rollover metric $R_{av_g}$).

To explicitly consider uncertainty during planning, SRSM is here employed. The general procedure is as follows:

Let $N$ uncertain parameters, considered to be normally distributed about their mean values, be represented using standard normal random variables $\zeta_m$ as:

$$P = \mu_x + \bar{\zeta_m} \sigma_{\bar{\zeta_m}} \quad m = 1...N \quad (7)$$

$S$ state variables of interest are then represented using Hermite polynomials of these standard normal random variables as:

$$x_i(t, \zeta) = \sum_{j=0}^{N_i} x_{ij}(t) \Phi_j(\zeta) \quad i = 1...S \quad (8)$$

where $\zeta = [\zeta_1, \zeta_2, ... \zeta_N]$.

Spectral stochastic analysis [13] is then performed using the above expansions, resulting in the time evolution of the mean and variance values of the state variables during expansion of a given node. As a result, a description of the robot’s likely path of travel is obtained.

**B.1. Confidence Ellipse Construction**

SRSM provides reduced order expansions for calculation of the robot path coordinates, which are then utilized to obtain relevant statistics such as the mean and variance [12]. Based on these, the mean path can be augmented with ellipses [17] that indicate confidence levels for the predicted position of the robot in the presence of uncertainty. These are then used to perform collision checks to avoid paths that are likely to collide with obstacles (see Figure 5). The approach represents an improvement over Monte Carlo methods by reducing the number of paths that must be generated to estimate the path distributions.

Confidence ellipses centered at the mean path coordinate can be generated (see Figure 6) based on the following equation:

$$\frac{1}{1-R^2} \left[ \frac{(\bar{x} - x)^2}{s_x} - 2r \frac{(\bar{x} - x)(\bar{y} - y)}{s_x s_y} + \frac{(\bar{y} - y)^2}{s_y} \right] = C^2 \quad (9)$$

where $C^2 = \frac{n-1}{n} \left( \frac{2}{n} \right) \left( \frac{1}{P^2} \right) \cdot \frac{1}{\bar{\sigma}_x} \bar{\sigma}_y = \frac{1}{n} \sum_{i=1}^{n} y_i^2$.

$\bar{x}$ and $\bar{y}$ are the mean path coordinates, $s_x$ and $s_y$ are the sample standard deviations, $r$ is the sample correlation index, $n$ is the number of samples generated from the reduced model and $P$ is the confidence level of the predicted position, which may be chosen based on the criticality of the operation.

The principal semi-axes of the ellipse are given as:

$$a_i = cs_i \quad a_j = cs_j \quad (10)$$

where $s_{i,j} = \sqrt{(s_i^2 + s_j^2 + \sqrt{(s_i^2 - s_j^2)^2 + 4r^2 s_i^2 s_j^2})/2}$.

The ellipse orientation is denoted by the inclination angle $\beta$:

$$\beta = \frac{1}{2} \tan^{-1} \frac{2rs_{i,j}}{s_i^2 - s_j^2} \quad (11)$$

Information obtained from this analysis (such as the average variation in the robot position along the path, or the probability of collision with an obstacle) can also be used to alter the node costs in the RRT expansion heuristic. This has, however, not been considered in the present analysis.

**B.2. Expansion Heuristic**

As described in Section IV.C, a rollover metric value $R$ can be employed as a cost during tree expansion. This results in explicit consideration of robot mobility, albeit in a deterministic manner. To consider robot mobility in a stochastic manner, SRSM can be employed to yield an expected rollover metric $E[R]$. This result can also be used during tree expansion. Once the expected value for the rollover metric $E[R]$ and its variance $\sigma_R$ along a trajectory is obtained, $R_{av_g}$ of Equation (9) can be replaced by $R_{av_g}'$, where the latter is the path-averaged value of $R_n$, given as:

$$R_n = E[R] + f \sigma_R \quad f \geq 0 \quad (12)$$

Thus, while extending towards a sample point from the least-cost node, $R_{av_g}'$ is used to compare with the threshold $R_n$. Further, the stochastic values are utilized while assigning the node costs during the heuristically biased tree expansion.

**B.3. Selective Implementation of SRSM**

It can be inferred that the application of stochastic analysis along each path segment during tree growth can lead to increased computation times during planning. However, it may not be necessary to apply stochastic analysis for scenarios...
where the path segments are relatively smooth and flat. SRSM should be invoked only for tree expansions that may have a high likelihood of robot rollover. Here, the technique is employed when the following criterion is met:
\[ R_{\text{avg},a} > R_t, \quad \text{where} \ R_t < R^* \] (13)

Hence, if a path segment is likely to have an \( R_{\text{avg},a} \) value close to the threshold \( R^* \), SRSM is used to obtain a refined estimate of rollover risk.

Using the above extensions, an RRT algorithm that considers parameter uncertainty can be obtained, which yields smooth and safe paths. The modified algorithm is outlined in Table 2.

**TABLE 2**

<table>
<thead>
<tr>
<th>01.</th>
<th>function create_tree(Xsart,Xgoal,E);</th>
</tr>
</thead>
<tbody>
<tr>
<td>02.</td>
<td>( T = \text{initialize}(X_{\text{sart}}); )</td>
</tr>
<tr>
<td>03.</td>
<td>[Initialize tree (T) using Xsart.]</td>
</tr>
<tr>
<td>04.</td>
<td>( X_{\text{sart}} = \text{sample_uniform}(E); )</td>
</tr>
<tr>
<td>05.</td>
<td>[Choose sample node (Xsart) in E.]</td>
</tr>
<tr>
<td>06.</td>
<td>( [X_{\text{near}}] = \text{nearest_nodes}(X_{\text{sart}},T); )</td>
</tr>
<tr>
<td>07.</td>
<td>( [X_{\text{near}}] = \text{nearest_nodes}(E); )</td>
</tr>
<tr>
<td>08.</td>
<td>[Choose nearest node (Xnear) based on node costs.]</td>
</tr>
<tr>
<td>09.</td>
<td>( [X_{\text{near}}] = \text{extend_pure_pursuit}([\text{path}]); )</td>
</tr>
<tr>
<td>10.</td>
<td>( \text{SRR} = \text{SRSM}([\text{path}]); )</td>
</tr>
<tr>
<td>11.</td>
<td>( \text{Add [Xnear] to T if there is no collision.} )</td>
</tr>
<tr>
<td>12.</td>
<td>end</td>
</tr>
<tr>
<td>13.</td>
<td>end</td>
</tr>
<tr>
<td>14.</td>
<td>return T;</td>
</tr>
</tbody>
</table>

### VI. SIMULATED APPLICATION TO ROBOTIC SYSTEM

In this section the performance of the proposed mobility-based planning method is studied in simulation. Section VI.A describes the robot model, and Section VI.B describes the simulation scenario.

#### A.1. Dynamic Robot Model

Here a three degree of freedom robot model (see Figure 7) is considered that includes lateral acceleration, yaw and roll dynamics, as in [13]. The roll and yaw moments of inertia are represented by \( I_{xx} \) and \( I_{zz} \), respectively, \( m \) is the total robot mass, \( m_s \) is the sprung mass, \( V \) is the longitudinal velocity of the robot and \( \delta \) represents the front wheel steering angle. The linearized equations for this model are given as:

\[
\beta = \frac{GC}{mv} \beta + \frac{K_G}{mv} \beta + \frac{C_{bf}}{mv} \delta + \frac{m_h}{mv} \dot{\theta} + \frac{G}{mv} \sum T_i (14)
\]

\[
\dot{\psi} = \frac{mg_h}{r_s} \theta + \frac{m}{r_s} \theta + \frac{m_h}{r_s} \delta + \frac{m_h}{r_s} \sum T_i (15)
\]

\[
\dot{\psi} = \frac{K}{I_{xx}} \beta - \frac{D}{I_{xx}} \psi + \frac{C_{ff}}{I_{xx}} \delta + \frac{1}{I_{xx}} \sum T_i (16)
\]

where

\[
C = C_{ff} + C_{rr} \quad D = C_{ff} I_{xx} + C_{rr} I_{xx} \quad G = 1 + \frac{m_h^2 h^2}{m I_{xx}}
\]

and \( I_{xx} = m_h h^2 \left( 1 - m_s/m \right) \). \( C_{ff} \) and \( C_{rr} \) are the cornering stiffness values of the lumped front and rear wheels, \( g \) is gravitational acceleration, and \( I_{xx} \) and \( I_{zz} \) are the distances of the front and rear axles, respectively, from the center of gravity.

In addition to forces from tire compliance, lateral components of the contact forces on the robot can arise due to terrain unevenness. Given terrain elevation modeled as a continuous, differentiable function of planar position \( z(x,y) \), the terrain disturbance force \( T_r \) acting at each wheel is:

\[
T_r = N_i \left( (\partial z / \partial x_i) \dot{x}_o + (\partial z / \partial y_i) \dot{y}_o \right) \quad (17)
\]

where \( N_i \) is the normal contact force at wheel \( i \), \( \dot{x}_o \) and \( \dot{y}_o \) are unit vectors of the inertial reference frame, and \( \dot{y} \) is a unit vector lateral to the reference path.

![Fig. 7. Robot model for mobility analysis under uncertainty](image-url)

The suspension moment \( M_s \), including the body roll due to uneven terrain, is given as:

\[
M_s = -k_r \dot{\theta} - k_i \dot{\theta} - b_r \dot{\theta} - b_i \dot{\theta} (18)
\]

where \( k_r \) and \( k_i \) are the roll stiffness values, \( b_r \) and \( b_i \) are the damping rates of the front and rear axles, and \( \dot{\theta} \) and \( \dot{\theta} \) are the terrain roll angles.

To compute terrain roll angles and rates, it is assumed that the wheels always remain in contact with the terrain. Then, using knowledge of the position and velocity of each wheel and terrain elevation \( z(x,y) \), the disturbances are calculated as:

\[
\dot{\theta} = \left( \frac{z_{i+1} - z_i}{y_{i+1} - y_i} \right) \dot{y}_o \quad \dot{\theta} = \left( \frac{z_{i+1} - z_i}{y_{i+1} - y_i} \right) \dot{y}_o (19)
\]

where the rate of elevation change can be computed as:

\[
\dot{z} = V \left( (\partial z / \partial x) \cos(\psi + \theta) + (\partial z / \partial y) \sin(\psi + \theta) \right) (20)
\]

For measuring vehicular mobility, a rollover coefficient is defined, as in [13]. Using the principle of balance of moments and vertical forces, the rollover metric for the linear model under consideration is given as:

\[
R = \frac{2m}{mg_{gy}} \left( h_o + h \right) \left( \frac{1}{\psi} + h \frac{\psi}{\dot{\psi}} \right) (21)
\]

where \( h_o \) is the height of the roll axis above the ground and \( y_o \) is the track width. For this metric, \( R > 1 \) indicates robot wheel liftoff and thus impending rollover.
A.2. Inclusion of Uncertainty

In the present analysis, values of the front and rear axle roll stiffness values are considered to be normally distributed about their mean values, and are represented as:

\[ k_f = \mu_k + \xi_k \sigma_k \quad k_r = \mu_k + \xi_k \sigma_k \]

In the SRSM implementation, the output state variable \( X \) is represented as:

\[ X(t, \xi) = \sum_{j=0}^{N} X_{i,j}(t) \Phi_{j}(\xi) \]

where \( \xi = [\xi_f, \xi_r] \).

The roll stiffness parameter values employed in the study are shown in Table 3.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>MEAN (Nm/rad)</th>
<th>STD. DEV. (Nm/rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_f )</td>
<td>60×10^3</td>
<td>15×10^3</td>
</tr>
<tr>
<td>( k_r )</td>
<td>60×10^3</td>
<td>15×10^3</td>
</tr>
</tbody>
</table>

B. Description of Scenarios

Deterministic as well as stochastic analyses were performed for the environmental scenario in Fig. 8-9 to separately evaluate the improvements due to consideration of mobility-based features and stochastic analysis in the RRT framework. For the deterministic analysis, parameter uncertainty was neglected and the performance of a basic RRT algorithm was compared to the modified method that includes mobility-based features. Comparison metrics were calculated in terms of the travel time \( T_o \) and likelihood of safe traversal. To evaluate the latter, a trajectory quality metric (\( Q_{Ta} \)) is defined as:

\[ Q_{Ta} = \max \left( R_{\text{avg}_{xl}} \right) \]

where \( R_{\text{avg}_{xl}} \) is the averaged rollover metric along the path segment connecting the \( i^{th} \) node and its predecessor, and \( Q_{Ta} \) refers to its maximum value among the nodes of the final path. The averaged rollover coefficient along the final trajectory (\( R_{\text{avg}_{p}} \)) from the two approaches is also noted.

Uncertainty was then considered and the performance of the modified algorithm that included SRSM (without selective implementation) was compared to the non-SRSM case, in terms of the trajectory quality metric (\( Q_{Tb} \)), defined as:

\[ Q_{Tb} = |R_{\text{avg}_{l}}| \]

where \( R_{\text{avg}_{l}} \) is the path-average of the expected value of the rollover metric along the final trajectory, under uncertainty. For the deterministic case, this was obtained by using a Monte Carlo (MC) analysis for the final path, simulating over parameter value samples from the uncertain distributions, while applying the steering inputs determined from the original analysis.

The improvement in computational efficiency of SRSM over a Monte Carlo approach within the framework was also studied. Here, selective implementation was employed, where multiple simulations along a path segment were run only when the threshold \( R_i \) is crossed. To compare the two methods, the ratio of the corresponding simulation time (\( T \)) to the computation time for the deterministic run (\( T_D \)) was computed.

Fig. 8. Terrain environment considered in the analysis.

Fig. 9. Placement of obstacles for the scenario (top view).

VII. Simulation Results and Discussion

A. Deterministic Analysis

Plans were generated between random start/goal locations for the terrain of Figure 8. Various values for \( h \) and \( R_s \) were considered and the typical values obtained for \( T_o, Q_{Ta} \) and \( R_{\text{avg}_{p}} \) for the two scenarios are shown in Table 4.

<table>
<thead>
<tr>
<th>TECHNIQUE</th>
<th>( h )</th>
<th>( R_s )</th>
<th>( Q_{Ta} )</th>
<th>TRAVEL TIME, ( T_o ) (s)</th>
<th>( R_{\text{avg}_{p}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRT (Basic)</td>
<td>-</td>
<td>-</td>
<td>0.692</td>
<td>19.02</td>
<td>0.485</td>
</tr>
<tr>
<td>Modified RRT (Non-SRSM)</td>
<td>0.4</td>
<td>0.397</td>
<td>18.01</td>
<td>0.331</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.591</td>
<td>17.67</td>
<td>0.415</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.776</td>
<td>17.38</td>
<td>0.531</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.397</td>
<td>18.43</td>
<td>0.318</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.588</td>
<td>17.98</td>
<td>0.403</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.767</td>
<td>17.57</td>
<td>0.507</td>
<td></td>
</tr>
</tbody>
</table>

Paths generated by the proposed approach generally resulted in lower rollover coefficient values. This is because the threshold value \( R_s \) limits the selection of tree extensions to those with absolute value of rollover metric, averaged over the path segment, lower than its magnitude. Reducing \( R_s \) therefore, results in paths with lower \( Q_{Ta} \) and \( R_{\text{avg}_{p}} \) values. Similarly, increasing the value of the parameter \( h \) causes the expansion heuristic to select nodes on easily traversable paths, also leading to trajectories with marginally lower \( Q_{Ta} \) and \( R_{\text{avg}_{p}} \) values.

While the \( R_{\text{avg}_{p}} \) value may be lower for the basic RRT algorithm for certain scenarios, there is no control over the value of \( Q_{Ta} \) in the modified approach. Therefore, for the path obtained using basic RRT, the tendency for the robot to overturn while negotiating the terrain is expected to be greater.
especially at high speeds. Similarly, $T_o$ values may be lower as well; however this comes at a cost to robot safety while negotiating the terrain. The tree from a typical simulation of the modified planning algorithm is shown in Fig. 10.

![Fig. 10. Resulting tree and final path obtained using the modified RRT algorithm (non-SRSM). $\delta=1, R_o=0.6$](image)

### B. Stochastic Analysis under Uncertainty

Further studies were conducted with varying values of $R_o$. Typical values obtained for $Q_{Tb}$ are shown in Table 5.

<table>
<thead>
<tr>
<th>TECHNIQUE</th>
<th>$R_o$</th>
<th>$Q_{Tb}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified RRT (Non-SRSM, MC on final path)</td>
<td>0.5</td>
<td>0.378</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.454</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.584</td>
</tr>
<tr>
<td>Modified RRT (SRSM: $R_l = 0$)</td>
<td>0.5</td>
<td>0.363</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.437</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.560</td>
</tr>
</tbody>
</table>

For the non-SRSM case, larger values of $Q_{Tb}$ were observed, indicating that treatment of uncertainty is important to obtain accurate values for the expected rollover metric along a path segment during tree expansion. While the deterministic planning algorithm might assume that a path segment is safe for traversal using the threshold $R_o$, this assumption might be poor due to uncertainty that is present. In certain cases, the averaged rollover coefficient value may be significantly greater than $R_o$ (or even 1, indicating failure). These paths, however, are disallowed in the stochastic planning framework. In the present analysis, the quantitative difference in $Q_{Tb}$ is somewhat marginal, especially for low $R_o$ values, since moderate robot velocity was studied. However, the difference in $Q_{Tb}$ can be large for travel over highly uneven terrain and/or aggressive robot maneuvers.

The computational efficiency of SRSM was compared to that of the Monte Carlo method in the planning framework. Typical results obtained for $T/T_o$ are shown in Table 6. The computational efficiency for SRSM is significantly better than for Monte Carlo, particularly for low values of $R_o$, when the stochastic analysis is frequently invoked. The metric $Q_{Tb}$ was found to be similar for the two techniques, as expected.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>$R_o$</th>
<th>$T/T_o$</th>
<th>$Q_{Tb}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monte Carlo (400 runs)</td>
<td>0.5</td>
<td>288.6</td>
<td>0.369</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>246.5</td>
<td>0.447</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>127.1</td>
<td>0.575</td>
</tr>
<tr>
<td>SRSM (2nd order)</td>
<td>0.5</td>
<td>4.25</td>
<td>0.371</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>4.17</td>
<td>0.449</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>4.02</td>
<td>0.571</td>
</tr>
</tbody>
</table>

### VIII. CONCLUSION AND FUTURE WORK

This paper has presented a framework for stochastic robot path planning that explicitly considers robot mobility and parameter uncertainty. Simulation results for planning on uneven terrain have shown that the proposed method can generate safer paths compared to a basic RRT algorithm, and can be used for robustly and efficiently predicting safe paths for mobile robots in unstructured, uncertain environments.

### REFERENCES


