The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

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

The authors are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: {gkewlani | ishigami | kdi}@mit.edu). This work was performed under US Army Research Office support (contract number W912HZ-08-C-0062).

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:

$$y = a_0 + \sum_{i_1=1}^{n} a_{i_1} \Gamma_1(\xi_{i_1}) + \sum_{i_1=1}^{n} \sum_{i_2=1}^{i_1} a_{i_1 i_2} \Gamma_2(\xi_{i_1}, \xi_{i_2}) + ... \quad (1)$$

where y refers to an output metric, $\xi_{i1}, \xi_{i2},\ldots$ are i.i.d. uniform random variables, $\Gamma_q(\xi_{i1}, \xi_{i2}, ..., \xi_{iq})$ is the Hermite polynomial of degree q and $a_{i1}, a_{i1i2},\ldots$ are the corresponding coefficients. For notational simplicity, the series may be written as:

$$y = \sum_{j=0}^{N_q} y_j \Phi_j(\xi) \quad (2)$$

where the series is truncated to a finite number of terms and there exists a correspondence between $\Gamma_q(\xi_{i1}, \xi_{i2}, ..., \xi_{iq})$ and $\Phi(\xi)$, 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 ρ, 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 approximation 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, $x_A = \rho \sin\theta$ and $y_A = \rho - \rho\cos\theta$.

For paths of type SC, $\varphi = \sin^{-1}((D\sin\alpha)/\rho)$, $\gamma = \pi - (\varphi + \alpha)$ and $L = (\rho/\sin\alpha)\sin\gamma = (D/\sin\varphi)\sin\gamma$. Then, $x_A = L$, $y_A = 0$, and $\theta = (3/2)\pi - \varphi$.

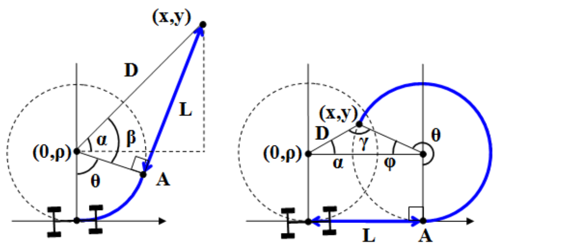

Fig. 1. Path length calculations for 2-D Dubins-like curves: CS (left) and SC (right)

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. $R_{avg\_s} < R_o$). 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:

$$Q_k = \prod_{i=1}^{3} \left(C_{i,k} / \max(C_{i,j}) \right) \qquad j,k = 1 \ldots M \tag{3}$$

where

$$C_{1,k} = t_k \tag{4}$$

$$C_{2,k} = (R_{avg\_p,k} R_{max\_p,k})^h \tag{5}$$

$$C_{3,k} = d_k \tag{6}$$

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.

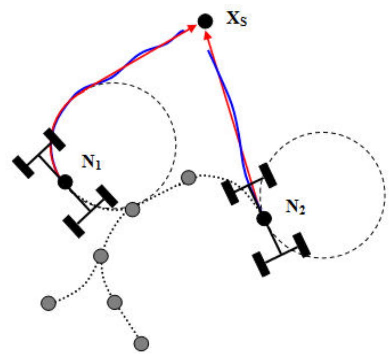

Fig. 2. Illustration of reference path tracking from two nearby nodes.

E. *Intermediate Nodes*

While long paths may be efficiently generated from the closed-loop control method, additional nodes are also placed along the trajectory at short intervals. These are added to the tree if the mobility criterion (described in IV.B) is satisfied for the path segment preceding the node under consideration.

This has been found to yield dense exploration and can save significant computational time in cases where there are collisions with obstacles, or if the mobility cost is exceeded for nodes at the end of long path segments (see Figure 3).

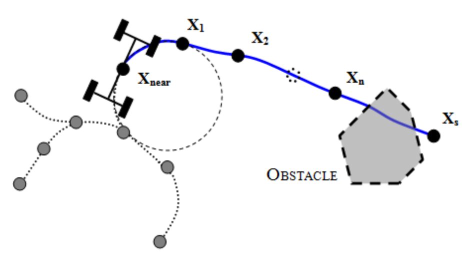

Fig 3. Placing of intermediate nodes along the traversed path.

V. Integration of SRSM With the RRT Framework

A. *Integration of SRSM with RRT framework*

Parameter uncertainty, if not explicitly considered in the planning framework, can lead to uncertainty in robot mobility and stability, and path following characteristics. As depicted in Fig. 4, for identical initial condition, various paths may be tracked by a closed-loop system depending on the values of 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_{avg_s}).

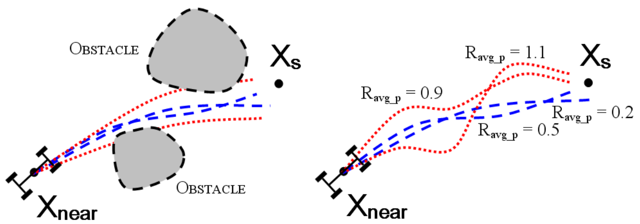

Fig. 4. Path and rollover unpredictability under uncertainty

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 ξ_m as:

$$P_m = \mu_{P_m} + \xi_m \sigma_{P_m} \qquad m = 1 \ldots N \tag{7}$$

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

$$x_i(t,\boldsymbol{\xi}) = \sum_{j=0}^{N_q} x_{i,j}(t)\Phi_j(\boldsymbol{\xi}) \qquad i = 1 \ldots S \tag{8}$$

where $\boldsymbol{\xi} = [\xi, \xi_2 \ldots \xi_N]$.

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.

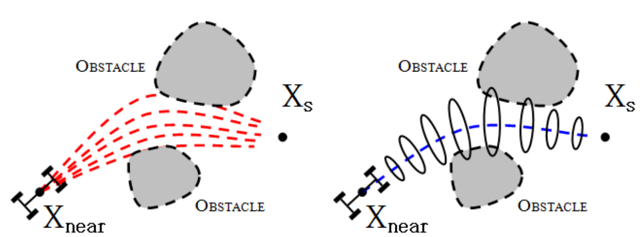

Fig. 5. Collision check using confidence ellipses

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

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

where $C^2 = \left(\frac{n-1}{n}\right)\left((1-P)^{\frac{2}{2-n}} - 1\right), \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

$\bar{x}$ 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:

$$a_x = cs'_x, \; a_y = cs'_y \tag{10}$$

where $s'_{x,y} = \left([s_x^2 + s_y^2 \pm \sqrt{(s_x^2 - s_y^2)^2 + 4r^2 s_x^2 s_y^2}]/2\right)^{1/2}$

The ellipse orientation is denoted by the inclination angle β:

$$\beta = \frac{1}{2}\tan^{-1}\frac{2rs_x s_y}{s_x^2 - s_y^2} \tag{11}$$

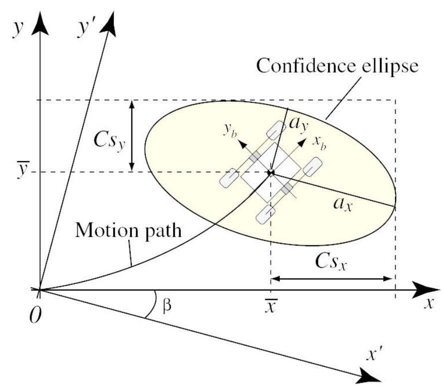

Fig. 6. Confidence Ellipse Construction

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 σ_R along a trajectory is obtained, R_{avg_s} of Equation (9) can be replaced by R'_{avg_s}, where the latter is the path-averaged value of R_s, given as:

$$R_s = E[R] + f\sigma_R, \quad f \geq 0 \tag{12}$$

Thus, while extending towards a sample point from the least-cost node, R'_{avg_s} is used to compare with the threshold R_o. 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:

$$|R_{avg\_s}| > R_I, \quad \text{where } R_I < R_o \tag{13}$$

Hence, if a path segment is likely to have an R_{avg_s} value close to the threshold R_o, 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
MODIFIED RRT-BASED PLANNING ALGORITHM

function *create_tree(**X***

function *create_tree*($\mathbf{X}_{start}$, $\mathbf{X}_{goal}$, $\mathbf{E}$);
[Get start location ($\mathbf{X}_{start}$), goal location ($\mathbf{X}_{goal}$) & environment ($\mathbf{E}$).]
$\mathbf{T}$ = *initialize*($\mathbf{X}_{start}$);
[Initialize tree ($\mathbf{T}$) using $\mathbf{X}_{start}$.]
while ~*reached*($\mathbf{X}_{goal}$, $\mathbf{T}$);
[Repeat steps below until $\mathbf{X}_{goal}$ is reached.]
    $\mathbf{X}_s$ = *sample_uniform*($\mathbf{E}$);
    [Choose sample node ($\mathbf{X}_s$) in $\mathbf{E}$.]
    [$\mathbf{X}_{near}$]= *nearest_nodes*($\mathbf{X}_s$, $\mathbf{T}$);
    [Search tree for N nearest nodes [$\mathbf{X}_{near}$] to $\mathbf{X}_s$.]
    $\mathbf{X}_{near}$= *nearest_node*([$\mathbf{X}_{near}$]);
    [Choose nearest node ($\mathbf{X}_{near}$) based on node costs.]
    [**path**] = *create_path*($\mathbf{X}_{near}$, $\mathbf{X}_s$);
    [Create Dubins-like path to $\mathbf{X}_s$ from $\mathbf{X}_{near}$.]
    [$\mathbf{X}_{new}$] = *extend_pure_pursuit*([**path**]);
    [Move towards $\mathbf{X}_s$ from $\mathbf{X}_{near}$ to get nodes [$\mathbf{X}_{new}$] along the path.]
    If $\mathbf{R}_{avg\_s} > \mathbf{R}_I$, [$\mathbf{X}_{new}$, $\mathbf{R'}_{avg\_s}$]=*SRSM*([**path**]);
    [Call SRSM function if required, do collision-check using confidence ellipses.]
    if ~*constraints*([$\mathbf{X}_{new}$], $\mathbf{T}$, $\mathbf{E}$);
    [Check if constraints are satisfied.]
        $\mathbf{T}$ = *add*([$\mathbf{X}_{new}$], $\mathbf{T}$);
        [Add [$\mathbf{X}_{new}$] to $\mathbf{T}$ if there is no collision.]
    end
end
return $\mathbf{T}$;

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 δ represents the front wheel steering angle. The linearized equations for this model are given as:

$$\dot{\beta} = -\frac{GC}{mV}\beta + \left(-1+\frac{KG}{mV^2}\right)\dot{\psi} + \frac{C_f G}{mV}\delta + \frac{m_s h M_s}{mVI^o_{xx}} + \frac{m_s^2 g h^2}{mVI^o_{xx}}\varphi + \frac{G}{mV}\sum T_i \tag{14}$$

$$\ddot{\varphi} = \frac{m_s g h}{I^o_{xx}}\varphi + \frac{M_s}{I^o_{xx}} - \frac{m_s C h}{mI^o_{xx}}\beta + \frac{m_s K h}{mVI^o_{xx}}\dot{\psi} + \frac{C_f m_s h}{mI^o_{xx}}\delta + \frac{m_s h}{mI^o_{xx}}\sum T_i \tag{15}$$

$$\ddot{\psi} = \frac{K}{I_{zz}}\beta - \frac{D}{VI_{zz}}\dot{\psi} + \frac{C_f l_f}{I_{zz}}\delta + \frac{1}{I_{zz}}\sum T_i l_i \tag{16}$$

where

$C = C_f + C_r$, $K = C_r l_r - C_f l_f$, $D = C_f l_f^2 + C_r l_r^2$, $G = 1 + \frac{m_s^2 h^2}{m I^o_{xx}}$,

and $I^o_{xx} = I_{xx} + m_s h^2 (1 - m_s / m)$. C_f and C_r are the cornering stiffness values of the lumped front and rear wheels, g is gravitational acceleration, and l_f and l_r 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_i acting at each wheel is:

$$T_i = N_i \left((\partial z / \partial x_o)\hat{x}_o + (\partial z / \partial y_o)\hat{y}_o \right) \cdot \hat{y}, \ \mathrm{i} = 1 \ldots 4 \tag{17}$$

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

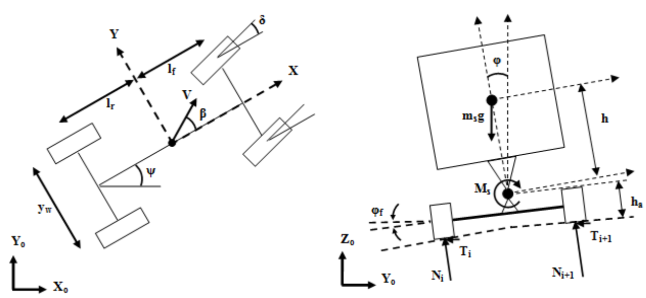

Fig. 7. Robot model for mobility analysis under uncertainty

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

$$M_s = -k_f(\varphi - \varphi_f) - k_r(\varphi - \varphi_r) - b_f(\dot{\varphi} - \dot{\varphi}_f) - b_r(\dot{\varphi} - \dot{\varphi}_r) \tag{18}$$

where k_f and k_r are the roll stiffness values, b_f and b_r are the damping rates of the front and rear axles, and φ_f and φ_r 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:

$$\varphi_i = (z_{i+1} - z_i) / y_w, \quad \dot{\varphi}_i = (\dot{z}_{i+1} - \dot{z}_i) / y_w \tag{19}$$

where the rate of elevation change can be computed as:

$$\dot{z}_i = V\left((\partial z / \partial x)\cos(\psi + \beta) + (\partial z / \partial y)\sin(\psi + \beta) \right) \tag{20}$$

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:

$$R = \frac{2m_s}{mgy_w}(h_a + h)\left(v(\dot{\beta} + \dot{\psi}) - h\ddot{\varphi} \right) \tag{21}$$

where h_a is the height of the roll axis above the ground and y_w 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:

$$k_f = \mu_{k_f} + \xi_1 \sigma_{k_f} \qquad k_r = \mu_{k_r} + \xi_2 \sigma_{k_r} \tag{22}$$

In the SRSM implementation, the output state variable X_i is represented as:

$$X_i(t, \boldsymbol{\xi}) = \sum_{j=0}^{N_q} X_{i,j}(t) \Phi_j(\boldsymbol{\xi}) \tag{23}$$

where $\boldsymbol{\xi} = [\xi_1, \xi_2]$.

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

TABLE 3
UNCERTAIN ROBOT PARAMETERS IN ROLLOVER ANALYSIS

PARAMETER	MEAN (Nm/rad)	STD. DEV. (Nm/rad)
k_f	60×10^3	15×10^3
k_r	60×10^3	15×10^3

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:

$$Q_{Ta} = \max\,(R_{avg\_s,i}) \tag{19}$$

where $R_{avg\_s,i}$ 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_{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:

$$Q_{Tb} = |R_{avg,s}| \tag{18}$$

where $R_{avg,s}$ 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.

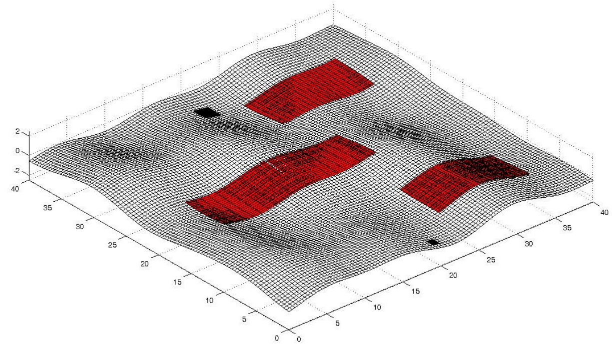

Fig. 8. Terrain environment considered in the analysis.

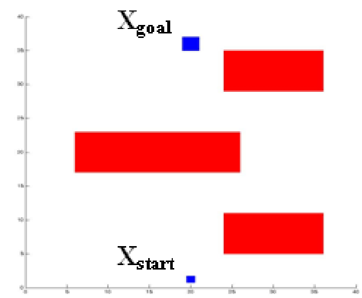

Fig. 9. Placement of obstacles for the scenario (top view).

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_o were considered and the typical values obtained for T_o, Q_{Ta} and R_{avg_p} for the two scenarios are shown in Table 4.

TABLE 4
TRAJECTORY QUALITY AND TRAVEL TIME

TECHNIQUE	h	R_O	Q_{Ta}	TRAVEL TIME, T_O (S)	R_{avg_p}
RRT (Basic)	-	-	0.692	19.02	0.485
Modified RRT (Non-SRSM)	1	0.4	0.397	18.01	0.331
	1	0.6	0.591	17.67	0.415
	1	0.8	0.776	17.38	0.531
	4	0.4	0.397	18.43	0.318
	4	0.6	0.588	17.98	0.403
	4	0.8	0.767	17.57	0.507

Paths generated by the proposed approach generally resulted in lower rollover coefficient values. This is because the threshold value R_o 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_o, therefore, results in paths with lower Q_{Ta} and R_{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_{avg_p} values.

While the R_{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.

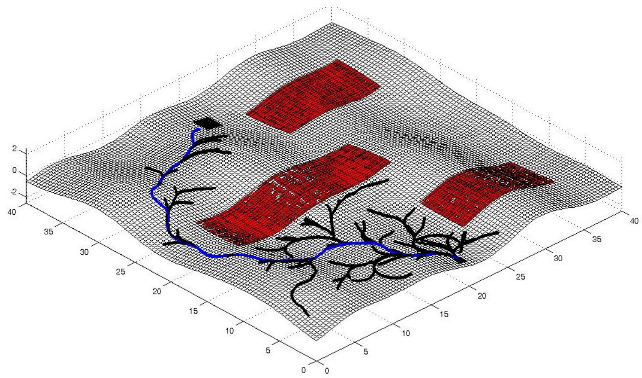

Fig. 10. Resulting tree and final path obtained using the modified RRT algorithm (non-SRSM). (h=1,R_o=0.6)

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 5
TRAJECTORY QUALITY FOR GENERATED PATHS

TECHNIQUE	R_O	Q_{Tb}
Modified RRT (Non-SRSM, MC on final path)	0.5	0.378
	0.7	0.454
	0.9	0.584
Modified RRT (SRSM: $R_I = 0$)	0.5	0.363
	0.7	0.437
	0.9	0.560

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_D 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 6
TRAJECTORY QUALITY AND RELATIVE SIMULATION TIME

METHOD	R_o	T/T_D	Q_{Tb}
Monte Carlo (400 runs) ($R_I = R_o - 0.1$)	0.5	288.6	0.369
	0.7	246.5	0.447
	0.9	127.1	0.575
SRSM (2^{nd} order) ($R_I = R_o - 0.1$)	0.5	4.25	0.373
	0.7	4.17	0.449
	0.9	4.02	0.571

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

[1] A. Stentz, "Optimal and efficient path planning for partially-known environments," *Proc. of IEEE International Conference on Robotics and Automation,* pp. 3310-3317, 1994.

[2] J. Barraquand, B. Langlois, and J.-C. Latombe; "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man and Cybernetics*, vol.22, no.2, pp.224-241, 1992.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol.12, no.4, pp.566-580, Aug 1996.

[4] P. Cheng, Z. Shen, and S. M. LaValle, "RRT-based trajectory design for autonomous automobiles and spacecraft," *Archives of Control Sciences*, 11(3-4):167--194, 2001.

[5] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM,* 40:1048–1066, 1993.

[6] C. Urmson, and R. Simmons., "Approaches for heuristically biasing RRT growth," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.2, pp. 1178-1183 vol.2, Oct. 2003.

[7] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. How, "Motion Planning in Complex Environments using Closed-loop Prediction," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI, Aug. 2008,* AIAA-2008-7166

[8] N. A. Melchior, and R. Simmons., "Particle RRT for Path Planning with Uncertainty," *IEEE International Conference on Robotics and Automation*, vol., no., pp.1617-1624, 10-14 April 2007

[9] J. C. Helton., "Uncertainty and Sensitivity Analysis Techniques for Use in Performance Assessment for Radioactive Waste Disposal", *Reliability Engineering and System Safety*, 42, 327-367, 1993.

[10] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, Inc., New York, NY, 1981.

[11] R. Ghanem, and P. D. Spanos., *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.

[12] S. S. Isukapalli, *Uncertainty analysis of transport-transformation models*, Ph.D. Thesis, Rutgers University, 1999.

[13] G. Kewlani, and K. Iagnemma, "Mobility prediction for unmanned ground vehicles in uncertain environments," *Proc. SPIE Conference on Unmanned Systems*, pp. 69621G1–69621G12, 2008.

[14] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.

[15] G. Cho, and J. Ryeu, "An Efficient Method to Find a Shortest Path for a Car-Like Robot," *Intelligent Control and Automation, 2006.*

[16] R. C. Coulter, "Implementation of the Pure Pursuit Path Tracking Algorithm," tech. report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, January, 1992.

[17] L. R. Paradowski, "Uncertainty Ellipses and Their Application to Interval Estimation of Emitter Position," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 33, no. 1 pp.126–133, 1997.