Stochastic Approaches to Mobility Prediction, Path Planning and Motion Control for Ground Vehicles in Uncertain Environments

by

Gaurav Kewlani

B. Tech., Mechanical Engineering (2007)
Indian Institute of Technology, Delhi

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

August 2009

Signature of Author .....
Department of Mechanical Engineering
August 14, 2009

Certified By .....
Dr. Karl Iagnemma
Principal Research Scientist
Department of Mechanical Engineering
Thesis Supervisor

Accepted By .....
Prof. David E. Hardt
Graduate Officer, Department of Mechanical Engineering

Stochastic Approaches to Mobility Prediction, Path Planning and Motion Control for Ground Vehicles in Uncertain Environments

by

Gaurav Kewlani

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

ABSTRACT

The ability of autonomous or semi-autonomous unmanned ground vehicles (UGVs) to rapidly and accurately predict terrain negotiability, generate efficient paths online and have effective motion control is a critical requirement for their safety and use in unstructured environments. Most techniques and algorithms for performing these functions, however, assume precise knowledge of vehicle and/or environmental (i.e. terrain) properties. In practical applications, significant uncertainties are associated with the estimation of the vehicle and/or terrain parameters, and these uncertainties must be considered while performing the above tasks. Here, computationally inexpensive methods based on the polynomial chaos approach are studied that consider imprecise knowledge of vehicle and/or terrain parameters while analyzing UGV dynamics and mobility, evaluating safe, traceable paths to be followed and controlling the vehicle motion. Conventional Monte Carlo methods, that are relatively more computationally expensive, are also briefly studied and used as a reference for evaluating the computational efficiency and accuracy of results from the polynomial chaos-based techniques.

KEY WORDS – vehicle mobility, path planning, motion control, uncertainty, Monte Carlo, polynomial chaos, response surface, roll-over, random tree, predictive control, UGV

Thesis Supervisor: Dr. Karl Iagnemma
Title: Principal Research Scientist, Department of Mechanical Engineering

ACKNOWLEDGEMENT

I would like to express my immense gratitude to Dr. Karl Iagnemma for his patient guidance, constant encouragement and supervision provided throughout the course of this project work.

I am also thankful to Mr. Edward H Linde whose generous contribution to the MIT Presidential Graduate Fellowship Program provided me with funding during the first year and to Army Research Office for sponsoring the research work. My deep appreciation also goes to the administration at the Department of Mechanical Engineering (particularly Leslie Regan and Joan Kravit) for their help in administrative matters as well as to the Department faculty, and also to all the staff who made the stay at MIT pleasant and hospitable.

I am also grateful to the members of the Robotic Mobility Group (notably Genya Ishigami, Steven Peters, Martin Udengaard and Sterling Anderson) for their very useful suggestions and to Natalie Illsley for the administrative support.

Finally I would like to acknowledge the invaluable help and support rendered to me by my family and friends, who always inspired me and kept my enthusiasm high.

Gaurav Kewlani

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation

Future army operations will employ autonomous or semi-autonomous unmanned ground vehicles (UGVs) in both cross-country and urban environments. Fundamental requirements for systems operating in such unstructured terrain include the capacity to quickly and accurately predict their ability to negotiate rugged regions and surmount obstacles, as well as to generate feasible trajectories online, and to have robust motion control schemes at their disposal to move the vehicle along desired paths. These capabilities are critical to the safe and efficient operation of vehicle navigation systems and hence to the successful deployment of UGVs that can operate effectively on challenging terrain with minimal human supervision.

With UGVs being employed for applications such as exploration, military reconnaissance and material transport on rugged terrain, autonomous navigation of such systems at high speeds has therefore received substantial research attention recently, as highlighted by the DARPA Grand Challenge competition. However, while significant work has been done to understand and predict the mobility of vehicles in natural terrain [1], [2], these efforts generally assume precise knowledge of vehicle parameters and wheel- (or track-) soil interaction properties, gathered from terrain measurement devices such as cone penetrometers. In the context of path generation, many techniques focus on generation of time- or distance-optimal paths while obeying some dynamic constraints [3]-[6], but often assume precise knowledge of the vehicle and/or terrain properties and do not analyze terrain mobility explicitly during the planning process. A similar deterministic approach is followed by most predictive motion control algorithms.

In summary, most previous methods for mobility prediction, path planning and predictive control rely on deterministic analyses that assume accurate knowledge of vehicle and/or terrain parameters. In field conditions, however, UGVs often only have access to sparse and uncertain soil parameter estimates. Moreover, significant uncertainties are often associated with estimates of vehicle parameters, due to effects such as loading, wear, fuel consumption,

etc. It is therefore critical to explicitly consider these uncertainties when deriving predictions of vehicle mobility, evaluating safe paths for the vehicle to follow and controlling vehicle motion along these routes.

1.2. Objective

As explained in Section 1.1, there has been little research that explicitly addresses the challenge of autonomously assessing the traversability of ground vehicles over a given terrain region. Further, most path planning approaches and motion control schemes do not explicitly consider parametric uncertainty during their analysis and predictions. It is essential that the effects of uncertainty be included in order to get more realistic predictions, especially for situations involving aggressive vehicle movement over uneven, uncertain terrain.

Many statistical techniques exist for evaluating processes that are subject to uncertainty and generating probabilistic results for various outputs. These include the conventional Monte Carlo approaches, as well as the relatively recent polynomial chaos-based techniques. The Monte Carlo approach involves sampling of random points within the domain of each uncertain parameter, followed by simulation of the model under consideration using this parameter set. The probability distribution for the output(s) and the corresponding statistical properties can then be obtained from the ensemble of simulation results. However, this technique is often time consuming and infeasible for most on-line planning and control operations for vehicle systems.

On the other hand, the relatively recent polynomial chaos-based techniques, although marginally compromising on the accuracy, are significantly more computationally efficient. These response approximation methods rely on constructing simplified models to capture the relationship between the stochastic inputs and outputs. While the original polynomial chaos framework could be applied only to normally distributed inputs, this was later extended to include inputs having other probability distributions, through the introduction of the generalized polynomial chaos method. Numerous extensions have been proposed to the underlying framework to enhance the applicability and accuracy of the technique, and these

include the stochastic response surface approach as well as multi-element generalized polynomial chaos method.

In this thesis, an attempt has been made to apply some of these approaches to the areas of vehicle dynamic analysis and navigation, and particularly to vehicle stability, path planning and motion control. A comparison has been performed between the various methods with regard to computation efficiency and accuracy.

1.3. Contributions

1.3.1. Mobility Analysis

There has been little research done in the past that explicitly deals with the challenge of autonomously assessing the traversability of a vehicle over a given terrain region or obstacle under uncertainty, as most techniques either do not explicitly analyze vehicle mobility on rough terrain or rely on a deterministic study that assumes precise knowledge of vehicle and/or terrain parameters.

In the present work, roll-over of the vehicle has been considered for studying its mobility in unstructured environments and a suitable metric has been adopted. Subsequently, a stochastic analysis using the polynomial chaos-based approach has been performed so as to incorporate the effects due to vehicle parameter uncertainty. Significant differences can be found between the deterministic and stochastic results, particularly when the variance in the predicted mean value is also considered in the stochastic analysis to indicate the vehicle roll-over tendency in realistic, uncertain scenarios. In other words, while the deterministic results may indicate safe traversal of the vehicle over a rugged terrain, especially for an aggressive maneuver, inclusion of parametric uncertainty can introduce a probability distribution for the expected output that provides an insight into the possibility of roll-over under uncertainty.

The approach has also been used to analyze the ability of the vehicle to traverse a terrain that has uncertainty in soil parameters. This scenario has been modelled by considering a wheeled ground vehicle traveling on flat, firm outdoor terrain (heavy clay), then attempting to navigate up an inclined region of highly deformable terrain (dry sand) and the mobility has

been defined as the probability that the vehicle will have a positive velocity after traversing the sandy incline.

In addition to mobility analysis, the method can also be applied to predict the expected mean value and the corresponding variance for each of the vehicle state variables such as slip angle and roll angle, as well as its path coordinates, in order to better approximate the vehicle dynamics in the presence of significant parametric uncertainty.

1.3.2. Path Planning

While evaluating a suitable path for the vehicle to follow through in an environment, important considerations include avoidance of obstacles and ensuring the stability of the vehicle along the proposed path. However, the presence of significant uncertainty in the vehicle and/or terrain parameters can lead to substantial deviation of the vehicle from the proposed path as it attempts to traverse the region. This may be attributed to the effect of uncertainty on the dynamics of the vehicle as it negotiates the terrain.

In other words, due to the uncertainty in the values of vehicle and/or terrain parameters, there exists a range of possible values for the position coordinates of the vehicle at any instant of time. As a result, any of the various possible paths may be tracked by the vehicle, which can even lead to collision with nearby obstacles while traversing certain trajectories. Further, there is a heightened probability of roll-over due to the variation of the dynamic state variables due to uncertainty induced effects. Incorporation of uncertainty within the planning framework has therefore been considered in the present work to deal with the above considerations, in order to generate safe and easily trackable paths.

1.3.3. Motion Control

Motion control constitutes another important feature that must be considered for successful operation of autonomous navigation systems. In the present work, the application of model predictive control (MPC) to UGV path tracking is studied due to its ability to systematically handle constraints and multi-variable systems. The technique uses a system model and optimization of constraints to determine control inputs that minimize a

performance objective and satisfy inequality constraints over a finite prediction horizon. While there have been attempts to make the framework more robust to uncertainty and include stochastic effects, there has been little research that explicitly considers uncertainty in vehicle and/or terrain parameters.

A feature of the predictive control approach is that it operates close to the constraint boundaries in order to obtain better performance than traditional approaches. However, the presence of uncertainty can lead to violation of these boundaries when the vehicle is actually traversing the region. Hence, inclusion of uncertainty is critical while determining the optimal control inputs, especially when the values of the state variables lie close to the constraint limits. In the present work, stochastic analysis is performed whenever such a case arises in order to obtain a better approximation to the expected values of the state variables. Further, control inputs are determined once the range of variation of the state variables from the corresponding mean values is incorporated in the analysis through a 'constraint tightening' approach. In other words, once the range of variation of the state variables is obtained through the stochastic analysis, the constraints are modified accordingly to incorporate the effects due to uncertainty. This has been found to yield more robust predictions.

1.4. Organisation of the Thesis

This thesis is organized as follows. In Chapter 2, conventional uncertainty analysis techniques are briefly introduced. This is followed by a description of the polynomial chaos-based approaches and their applicability to the study of robot dynamics and mobility analysis in Chapter 3. The application of these methods to predicting vehicle mobility is studied in Chapter 4, and to path planning is discussed in Chapter 5. The applicability of the approaches to a predictive control framework is also analyzed and simulation results for the integration of a polynomial chaos-based method into a model predictive control framework are shown in Chapter 6. Results from the conventional uncertainty analysis techniques are also compared to those from polynomial chaos-based methods and it can be seen that accurate, computationally efficient predictions can be achieved using the latter framework.

2. CONVENTIONAL UNCERTAINTY ANALYSIS TECHNIQUES

2.1. Overview

There exists a vast body of literature on techniques to estimate the probability distributions of processes that are subject to uncertainty. Such techniques can be applied to the domain of vehicle motion under uncertainty, by specifying the probability distribution of the uncertain vehicle and/or terrain parameters, defining a range for their probable values, and finally analyzing the performance of an analytical or numerical robot model over that parameter space, as in [7]. This analysis can be performed using a variety of techniques such as interval mathematics, fuzzy set theory and probabilistic methods such as the Monte Carlo approach among others [8]-[10].

The objective of interval analysis is to estimate the bounds on model outputs based on the bounds on the input parameters. In this approach, the uncertain parameters are assumed to be unknown but bounded (i.e. each parameter has lower and upper limits; knowledge of their probability distributions is not required) and described by an interval. Correspondingly, the outputs would also belong to respective intervals and be determined after performing suitable interval arithmetic [11]. Another approach for the study of uncertain systems and processes is the concept of fuzzy analysis, wherein the uncertain parameters are described by fuzzy numbers in contrast to random numbers (which are used in stochastic approaches).

Yet another traditional method for estimating the probability density function of a system's output response from known or estimated input distributions is the Monte Carlo method [12], [13]. This approach involves sampling values for each uncertain parameter from its uncertainty range, weighted by its probability of occurrence, followed by model simulation using this parameter set. This process is repeated many times to obtain the probability distribution of an output metric. Since parameter values are selected randomly (in case of traditional Monte Carlo methods), a large number of simulation runs is generally required to obtain reasonable results, leading to a (usually) high computational cost. Structured sampling techniques such as Latin hypercube sampling, importance sampling, and others can be used

to improve computational efficiency; however these gains may be modest for complex problems [14], [15].

2.2. Fuzzy Analysis

The modeling and simulation of uncertain systems within a fuzzy analysis framework can be performed through the numerical representation of the uncertain parameters as fuzzy numbers, followed by the use of the transformation method, as described in [9]. This is briefly discussed below.

In the first step, each fuzzy number $\boldsymbol{p_i}$ $(i = 1, 2, ..., n)$, is discretized into a number of intervals $\boldsymbol{X^{(j)}_i} = [a^{(j)}_i , b^{(j)}_i]$, assigned to the levels $\boldsymbol{\mu_j}$ $(j = 0, 1, ..., N)$, resulting from subdividing the possible range of membership into equally spaced units of $\boldsymbol{\Delta\mu} = 1/N$ (see Figure 2.1). Next, the input intervals $\boldsymbol{X^{(j)}_i}$ $(i = 1, 2, ..., n)$, $(j = 0, 1, ..., N)$, are transformed to arrays $\boldsymbol{X'^{(j)}_i}$ that are obtained from the upper and lower interval bounds after the application of a well-defined combinatorial scheme [9]. Each of these arrays represents a specific sample of possible parameter combinations and serves as an input parameter set to the problem to be evaluated. As a result of the evaluation of the model for the input arrays $\boldsymbol{X'^{(j)}_i}$, output arrays $\boldsymbol{Y'^{(j)}}$ are obtained which are then retransformed to the output intervals $\boldsymbol{Y^{(j)}} = [a^{(j)}, b^{(j)}]$ for each membership level $\boldsymbol{\mu_j}$ and finally recomposed to the fuzzy-valued output $\boldsymbol{q}$ of the system.

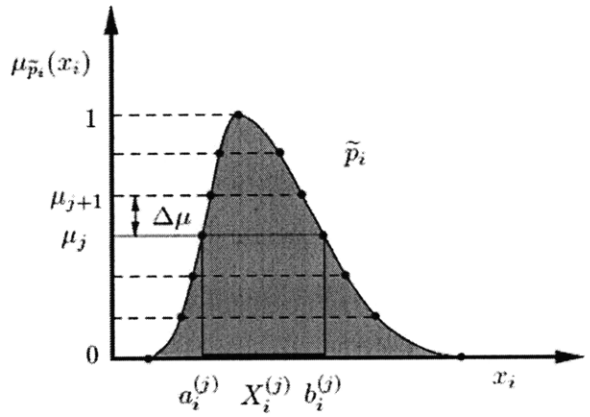

Figure 2.1. Decomposition of a fuzzy number $\boldsymbol{p_i}$ into intervals

2.3. The Monte Carlo Approach

With the advancements in computational technology, Monte Carlo techniques have found increasing application in numerous fields over the last several years. These methods typically involve the analysis of a (usually) large number of simulation runs of an analytical or numerical system model with various combinations of model parameters. In other words, the model parameters (known as "input parameters") are sampled from their respective probability distributions, which are assumed to be known (or can be estimated) a priori, and multiple simulation runs are conducted using each set of the input parameter values to obtain the corresponding outputs for each case. An estimate of the probability distribution of a user-defined output metric can then be estimated from this analysis.

In the "standard" Monte Carlo approach, random sampling of the input parameter distributions is performed. However, to ensure representation of the entire parameter range, a large number of simulations must be performed, often resulting in extensive computational costs. Over the years, various methods have been developed for efficient sampling from the input parameter probability distributions, including stratified, importance and Latin Hypercube sampling (among others) [16], [17]. Generally, these methods attempt to ensure that samples are generated from the entire range of the input parameter space while reducing computational costs, and are thus an improvement over the standard Monte Carlo method that is based on random sampling. In the present work, simulation results from the Monte Carlo approach are used as a baseline reference to validate the accuracy of results from the relatively recent polynomial chaos-based approaches (see Chapter 3).

2.3.1. Algorithmic Framework

The standard Monte Carlo approach considers functions of the form:

$$\mathbf{Y} = \mathbf{g}(\mathbf{X}) \tag{2.1}$$

where **g** represents the model under consideration, **X** is a vector of uncertain input variables and **Y** represents a vector of estimated outputs. A general procedure for the present analysis is as follows:

a) Construct a vector $\mathbf{X}$ consisting of n relevant vehicle and/or terrain parameters. To characterize the uncertainty in the elements of $\mathbf{X}$, define the range and probability distribution for each input parameter, based on corresponding engineering estimates. This defines the input parameter space.

While many forms of the input parameter distribution are possible, in this thesis, the parameter values are assumed to be either uniformly or normally distributed and to be uncorrelated (independent of one another), which is a reasonable assumption for most vehicle and/or terrain physical parameters.

b) Generate a sample value for each of the n input variables from the corresponding probability distribution. More specifically, a sample set:

$$\mathbf{X}_\mathbf{j} = \left[x_{j1}, x_{j2}, \ldots, x_{jn} \right] \tag{2.2}$$

is generated from the input parameter space. This set may be generated randomly or using structured sampling techniques such as stratified sampling, importance sampling or Latin Hypercube sampling.

c) Evaluate the output response from an analytical or numerical system model under consideration using the values from the input parameter set $\mathbf{X}_\mathbf{j}$ as model parameter values.

d) Repeat steps b) and c) to generate a distribution for the output metric. The number of simulations (N) is chosen to be large enough such that the output distribution converges to a stable value. The probability distribution of the output metric can then be determined, and various statistics such as its estimated expectation, μ, or variance, σ^2, be calculated as follows:

$$\mu = \frac{1}{N} \sum_{j=1}^{N} g(\mathbf{X}_\mathbf{j}) \tag{2.3}$$

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{N} (g(\mathbf{X}_\mathbf{j}) - \mu)^2 \tag{2.4}$$

As already mentioned in Chapter 1, for the analysis of vehicle motion over unstructured terrain, vehicle and/or terrain parameters are designated as uncertain input parameters. A

fundamental assumption of the proposed approach is that while these parameters may not be precisely known, engineering estimates of their distributions are available. This is a reasonable assumption for UGV physical parameter estimates, since the effects of loading, component wear, and parameter uncertainty can generally be bounded with reasonable accuracy. It is also a reasonable assumption for terrain parameter estimates, since many methods exist for coarsely classifying terrain from standard robotic sensors such as LIDAR and vision [18]-[20].

Figure 2.2 represents schematically the general Monte Carlo approach for uncertainty analysis.

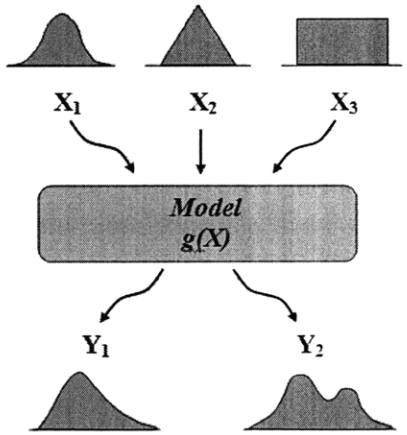

Fig. 2.2. Illustration of uncertainty analysis using the Monte Carlo method

2.3.2. Structured Sampling Techniques

As discussed in Section 2.1, using the standard Monte Carlo methods with random sampling can (typically) require extensive computational costs due to the large number of simulation runs required. Here some prominent structured sampling techniques are briefly discussed that can lead to an improvement in computational efficiency.

In the importance sampling method, sample points are generated to lie primarily in regions where the function (**g(X)**) is the strongest. Generating points evenly in the interval may sometimes give most of the points in the weak region of the function (depending on its probability distribution) and the contribution of these points to the total may be relatively small. As an example, consider the functional shape shown in Figure 2.3. It is quite obvious that most of the integral comes from the region of the peak. But if points are generated

evenly in the interval [*a, b*], most points won't be in the peak area, and their contribution to the total will be relatively small. The idea behind importance sampling is to transform **g(X)** into another, flatter function which is then analyzed using Monte Carlo method. Subsequently, there is a back-transformation to give the original output of interest. Though it can offer significant improvement over the standard Monte Carlo approach, importance sampling has the disadvantage that the function, or at least its overall shape, must be known. This is often not the case, as **g** may actually be a number returned by some other, complex simulation.

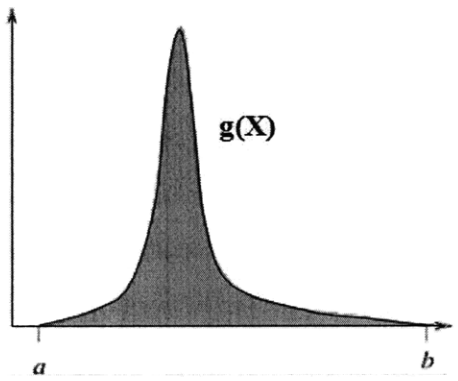

Figure 2.3. Functional form to illustrate use of importance sampling method

Stratified sampling, on the other hand, partitions the sample space into a number of strata, with each stratum having a specified probability of occurrence. Random samples are then drawn from each stratum. While this ensures dense coverage of the parameter space, it requires the definition of the strata and the calculation of their probabilities.

Latin hypercube sampling, however, can ensure dense coverage of the range of each input variable (say, *n* in number) while avoiding the difficulties associated with the above sampling techniques. It achieves this by exhaustively dividing each input parameter's range into disjoint intervals (say, *N* in number) of equal probability, and then randomly sampling a parameter value from each interval. This ensures representation from the entire range of each variable and is illustrated in Figure 2.4. The *N* values thus obtained for each of the *n* parameters are combined in a random manner with those from the other parameters until *N n*-

tuplets (N n-dimensional input vectors to be used for multiple simulation runs) are formed. More details can be found in [14].

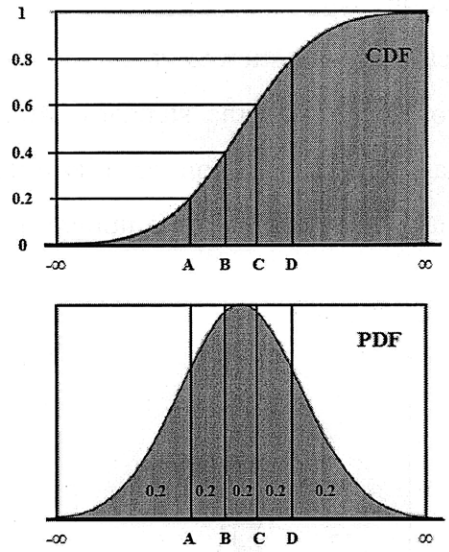

Figure 2.4. Illustration of sampling using the Latin hypercube method

3. POLYNOMIAL CHAOS-BASED UNCERTAINTY ANALYSIS TECHNIQUES

3.1. Overview

While Monte Carlo techniques have been in prominent use for quite some time, more recent approaches to stochastic simulation of processes subject to uncertainty include the polynomial chaos approach, which is based on Wiener's theory of homogeneous chaos. According to Wiener, a stochastic process can be represented through a spectral expansion using orthogonal polynomials as:

$$y = \sum_{j=0}^{\infty} y_j H_j(\xi) \tag{3.1}$$

where ξ is a vector of standard normal random variables (i.e. with zero mean and unit variance), H_j is the Hermite polynomial of order j, and y_j is the corresponding deterministic coefficient, to be calculated from a limited number of model simulations. The Hermite polynomial of degree q is given as:

$$H_q(\xi_1, \xi_2, ..., \xi_m) = (-1)^q e^{\frac{1}{2}\xi^T \xi} . \frac{\partial^q}{\partial \xi_1 ... \partial \xi_m} . e^{-\frac{1}{2}\xi^T \xi} \tag{3.2}$$

For this formulation, optimal convergence is obtained only for Gaussian stochastic processes. Recent extensions to the generalized polynomial chaos framework [21] have shown that optimal convergence can be achieved for more general stochastic phenomena. The fundamental principle behind this is that random processes of interest can be reasonably approximated using orthogonal polynomial basis functions of the Askey scheme (in terms of the corresponding random variable), and this allows treatment of a much broader range of stochastic problems.

Since the introduction of the spectral stochastic finite element method [22], polynomial chaos has been successfully applied to various structural and fluid mechanics problems. The primary advantage of the technique is the reduction in the number of model simulations relative to more conventional methods, such as Monte Carlo, thereby resulting in lower

computational cost. While being restricted to second order stochastic processes, the approach is still applicable to most physical phenomena as they typically have finite variance.

However, although the polynomial chaos technique has been widely used, it has been shown to perform inadequately for problems with discontinuities induced by random inputs, and for long-term integration. In [21], the method has been successfully applied to approximate the solution of a stochastic ODE while showing exponential convergence; however, it has been shown that those optimal results hold only for short times [23]. For long-term integration, the generalized polynomial chaos approximation to the analytical solution for a fixed polynomial degree is inaccurate, resulting in increased error levels. These problems can be overcome through implementation of the multi-element generalized polynomial chaos framework, which involves a decomposition of the random space, to yield more consistent results [23]. Another approach that has been developed to improve the robustness and performance of the polynomial chaos framework is the stochastic response surface method. These techniques are discussed in more detail in the subsequent sections.

3.2. The generalized polynomial chaos (gPC) method

As mentioned in Section 3.1, the generalized polynomial chaos method involves representing inputs and outputs of a system under consideration via series approximations using random variables, thereby resulting in a computationally efficient means for uncertainty propagation through complex models. In this approach, the same set of random variables that is used to represent input stochasticity is used for representation of the output(s). An equivalent reduced model for the output can thus be expressed in the form of a series expansion consisting of orthogonal polynomials (of the Askey scheme) in terms of the corresponding multi-dimensional random variable, as:

$$y = \sum_{j=0}^{\infty} y_j \Phi_j(\xi) \tag{3.3}$$

where y refers to an output metric, $\xi = [\xi_{i1} \ldots \xi_{im}]$ is the multi-dimensional random variable, $(\xi_{i1}, \xi_{i2}, \ldots)$ are i.i.d. (independent, identically distributed) random variables, $\Phi_j(\xi_{i1}, \xi_{i2}, \ldots, \xi_{im})$ is the generalized Askey-Wiener polynomial chaos of degree j, and y_j is the corresponding coefficient.

While for normal random variables Hermite polynomials are the basis functions, different orthogonal polynomial basis functions are used corresponding to the probability distributions of other non-normal variables [21]. This is shown in Table 3.1.

TABLE 3.1
POLYNOMIAL BASIS FUNCTIONS AND CORRESPONDING RANDOM VARIABLES

RANDOM VARIABLE	POLYNOMIAL FUNCTION
Gaussian	Hermite
Gamma	Laguerre
Beta	Jacobi
Uniform	Legendre

The series in (3.3) may be truncated to a finite number of terms and rewritten as:

$$y = \sum_{j=0}^{N_m} y_j \Phi_j(\xi) \tag{3.4}$$

where $N_m+1=(m+p)!/(m!p!)$, m is the number of random variables, and p is the maximum order of the polynomial basis.

The unknown coefficients in the expansion can be determined by projecting each state variable onto the polynomial chaos basis (i.e. using the Galerkin projection method) [24]. Another approach that is computationally more efficient is the collocation method [11], [25] wherein coefficient values are estimated from a limited number of model simulations [26]. This method imposes the requirement that the estimates of model outputs are exact at a set of selected collocation points, thus making the residual at those points equal to zero. The unknown coefficients are thus estimated by equating model outputs and the corresponding polynomial chaos expansion at this set of collocation points in the parameter space; the number of collocation points is equal to the number of unknown coefficients to be found. Thus, for each output metric, a set of linear equations are formed with the coefficients as the unknowns, which can be readily solved. If the governing equations are highly complex, the simplicity of the collocation-based framework results in a faster algorithm, particularly for high dimensional problems.

Though the accuracy of the gPC approach can be improved by increasing the polynomial order, it should be noted that as the number of inputs and the expansion order increase, the

number of unknown coefficients to be determined increases exponentially, thereby increasing the computational costs. The procedure describing the application of the generalized polynomial chaos is given below in more detail.

3.2.1. Algorithmic Framework

A summary of the gPC procedure, in the context of the present study, is presented here.

a) Represent uncertain input parameters in terms of random variables. Normally and uniformly distributed parameters will be considered in the present work. An uncertain vehicle and/or terrain parameter X_j can therefore be written as:

$$X_j = \mu_j + \sigma_j \xi \tag{3.5}$$

where μ_j is the mean, σ_j is a constant (and represents the standard deviation when X_j is normally distributed) and ξ is a random variable (i.e. the standard normal random variable *N(0,1)* when X_j is normally distributed).

b) Express the model output (y) under consideration in terms of the same set of random variables as:

$$y = \sum_{j=0}^{N_m} y_j \Phi_j(\xi) \tag{3.6}$$

c) Estimate the unknown coefficients of the approximating series expansion. This is accomplished by computing the model output at a set of collocation points, which results in a set of equations that can then be used to obtain the coefficient values. The Efficient Collocation Method (*ECM*) as proposed in [11] has been used in the present study.

d) Once the reduced order model is formulated, the mean and variance for orthonormal basis functions can be directly obtained [27] as:

$$\mu = y_0 \Phi_0(\xi) \tag{3.7}$$

$$\sigma^2 = \sum_{j=1}^{N_m} y_j^2 \tag{3.8}$$

As explained earlier in Section 3.1, the advantage of the polynomial chaos technique is that the number of model simulations is greatly reduced relative to more conventional methods such as the Monte Carlo method, thus improving computational efficiency. However, the approach is known to fail for long-term integration, losing optimal convergence behavior and developing high error levels [23]. This poor behavior can be somewhat mitigated by increasing the expansion order; however, this approach is undesirable for several reasons. First, in the general case, the gPC procedure becomes quite time consuming for high values of the polynomial order (since computational cost generally increases exponentially with increasing polynomial order). More importantly, increasing the maximal polynomial degree only postpones error growth, since for a fixed polynomial degree, error levels will become increasingly large over time. Hence, continuing to increase the integration time will require an ever increasing polynomial degree, which is not feasible in practice. This problem has been addressed in the multi-element generalized polynomial chaos technique [23], which solves the long-term integration issues faced in the gPC framework and can be used for arbitrary random variables as well. This will be discussed in the Section 3.3.

3.3. The multi-element generalized polynomial chaos (MEgPC) method

In [23], it has been shown that if the domain of random inputs is subdivided into multiple elements, the accuracy of stochastic solutions can be improved, especially for cases with discontinuities in stochastic solutions or for problems involving long-term integration. As a result, the integration error at each time step can be reduced and the domain of solutions' discontinuity can be approximated more accurately within a smaller decomposed domain. Further, a (relatively) lower order polynomial can be used in each random element since the local degree of perturbation has been scaled down, thereby enhancing the accuracy of solutions for long term integration. This is the fundamental principle underlying the MEgPC approach.

While the range of application of gPC is limited (since the polynomial order can not be increased arbitrarily high in practice), using MEgPC therefore allows this range to be extended. In this thesis, the standard MEgPC approach (as in [23]) has been employed while using uniformly distributed random input variables, and the Efficient Collocation Method has

been utilized (over the Galerkin projection approach), to result in a computationally more efficient algorithm. The general procedure is outlined below.

3.3.1. Algorithmic Framework

Let $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots \xi_m]$ denote an m dimensional random input vector, where ξ_i is an i.i.d. uniform random variable, $U[-1,1]$. Next, decompose the domain of the random input into N_r non-intersecting random elements. The domain of each element is contained within a hypercube, $[a^k_1, b^k_1) \times [a^k_2, b^k_2) \times \ldots \times [a^k_m, b^k_m)$, where a^k_i and b^k_i denote the lower and upper bounds of the local random variable ζ^k_i.

Then, define a local random vector within each element as $\boldsymbol{\zeta}^k = [\zeta^k_1, \zeta^k_2, \ldots \zeta^k_m]$, and subsequently map it to a new random vector in $[-1,1]^m$: $\boldsymbol{\xi}^k = g_k(\boldsymbol{\zeta}^k) = [\xi^k_1, \xi^k_2, \ldots \xi^k_m]$. This mapping is governed by the following relationship:

$$g_k(\boldsymbol{\zeta}^k): \zeta^k_i = \frac{b^k_i - a^k_i}{2}\xi^k_i + \frac{b^k_i + a^k_i}{2} \tag{3.9}$$

Consequently, the gPC framework can be used locally to solve a system of differential equations, with the random inputs as $\boldsymbol{\xi}^k$ instead of $\boldsymbol{\zeta}^k$, to take advantage of orthogonality and related efficiencies by employing Legendre Chaos. The global mean and variance can then be reconstructed once local approximations of the mean and the variance are obtained (see Equations 3.11-3.13).

Decomposition of the random space can be done *a priori* or adaptively. In the adaptive scenario, splitting of the random space occurs only when the local decay rate of the error of the gPC approximation η_k (see Equation 3.14) exceeds a threshold value. The basic steps are briefly discussed below.

Let the gPC expansion in random element k ($k = 1, 2, \ldots N_r$) be given as:

$$y_k = \sum_{j=0}^{N_m} y_{k,j}\Phi_j(\xi) \tag{3.10}$$

The approximated global mean and variance can then be written as:

$$\mu = \sum_{k=1}^{N_r} y_{k,0} V_k \tag{3.11}$$

where $V_k = \prod_{i=1}^{N_r} \frac{b_{k,i} - a_{k,i}}{2}$, and

$$\sigma^2 = \sum_{k=1}^{N_r} \left(\sigma_k^2 V_k + (y_{k,0} - \sigma)^2 V_k \right) \tag{3.12}$$

where the local variance estimated by polynomial chaos (using orthonormal basis functions) is obtained as:

$$\sigma_k^2 = \sum_{j=1}^{N_m} y_{k,j}^2 \tag{3.13}$$

To include adaptive decomposition of the random space, first define η_k as:

$$\eta_k = \frac{\sum_{j=N_{m-1}+1}^{N_m} y_{k,j}^2}{\sigma_k^2} \tag{3.14}$$

Then, split a random element if the following criterion is satisfied:

$$\eta_k^{\alpha} V_k \geq \theta_1, \quad 0 < \alpha < 1 \tag{3.15}$$

where θ_1 is a user-defined threshold parameter and α is a constant. Another parameter θ_2 can be used to choose the more sensitive random dimensions for decomposition, as in [23]. For this, first the sensitivity of each random dimension r_k is defined as:

$$r_{k,i} = \frac{y_{k,j(m)}^2}{\sum_{j=N_{m-1}+1}^{N_m} y_{k,j}^2} \quad , \; i = 1, \ldots, m \tag{3.16}$$

where $y_{k,j(m)}$ denotes the mode consisting only of random dimension ξ^k_i with polynomial order m. Only the random dimensions which satisfy the condition

$$r_i \geq \theta_2 . \max_{j=1,\ldots,m} r_j \quad , \; 0 < \theta_2 < 1 \quad , \; i = 1, \ldots, m \tag{3.17}$$

are split into two equal random elements in the next time step, while the other random dimensions stay unchanged. This reduces the total element number while gaining efficiency.

A critical numerical implementation involves assigning the initial condition after splitting the random dimension into multiple elements. This can be accomplished as follows:
First, represent the polynomial expansion of the current random field as:

$$\hat{y}(\hat{\xi}) = \sum_{j=0}^{N_m} \hat{y}_j \Phi_j(\hat{\xi}) \tag{3.18}$$

Once the random space is split, let the expansion in the next level be denoted as:

$$\tilde{y}(\tilde{\xi}) = \tilde{y}(g(\hat{\xi})) = \sum_{j=0}^{N_m} \tilde{y}_j \Phi_j(\tilde{\xi}) \tag{3.19}$$

To calculate the N_m+1 coefficients in this new representation, choose an equal number of uniformly spaced grid points in $[-1,1]^m$, and solve the following linear system:

$$\begin{bmatrix} \Phi_{00} & \Phi_{10} & \cdots & \Phi_{N_m 0} \\ \Phi_{01} & \Phi_{11} & \cdots & \Phi_{N_m 0} \\ . & . & \cdots & . \\ \Phi_{0N_m} & \Phi_{1N_m} & \cdots & \Phi_{N_m N_m} \end{bmatrix} \begin{bmatrix} \tilde{y}_0 \\ \tilde{y}_1 \\ . \\ \tilde{y}_{N_m} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{N_m} \hat{y}_i \Phi_i(g^{-1}(\tilde{\xi}_0)) \\ \sum_{i=0}^{N_m} \hat{y}_i \Phi_i(g^{-1}(\tilde{\xi}_1)) \\ . \\ \sum_{i=0}^{N_m} \hat{y}_i \Phi_i(g^{-1}(\tilde{\xi}_{N_m})) \end{bmatrix} \tag{3.20}$$

where $\Phi_{ij} = \Phi_i(\xi_j)$.

3.3.2. Study of Convergence

The convergence properties of the MEgPC approach are now studied, and its accuracy is compared to the gPC method. Consider a simple stochastic system: a first order linear ODE, described as:

$$\frac{dy}{dt} = -ky \quad \text{with } y_{t=0} = y_0 = 1 \tag{3.21}$$

Here, the decay rate coefficient k is considered to be a random variable, $k = \mu_k + \sigma_k \xi$, with a constant mean (μ_k=1) and σ_k=1, and ξ is a uniform random variable, $U[-1,1]$. While the deterministic solution $y(t)$ for the ODE above is $y_o e^{-\mu_k t}$, the mean of the stochastic solutions is given by:

$$\mu_{exact}(t) = y_o \int_S e^{-kt} f(k)dk = y_o \frac{e^{-(\mu_k+\sigma_k)t}(e^{2\sigma_k t}-1)}{2\sigma_k t} \tag{3.22}$$

To study the rate of convergence, define the error as an L_2 norm difference between the estimated result and the reference solution, normalized by the L_2 norm of the latter. This relative error measurement for the mean is expressed as:

$$\varepsilon_{mean}(t) = \frac{\| \mu(t) - \mu_{exact}(t) \|_2}{\| \mu_{exact}(t) \|_2} \tag{3.23}$$

where $\mu(t) = E[y(t)]$

Figure 3.1 shows the deterministic solution, the exact stochastic mean and the mean from the adaptive MEgPC method (P=3, θ_1=0.001, α=0.5) for the above first order linear system.

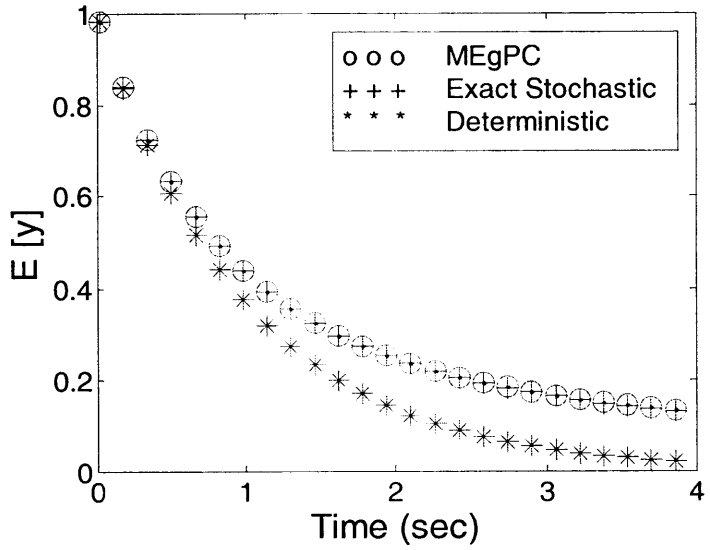

Fig. 3.1. Results for first order linear system

Next, the convergence behavior of MEgPC is analyzed, with varying values of the polynomial order P and number of elements N_r. For this analysis, the adaptive criterion is not applied, and instead the random space is decomposed according to the number of elements desired. The error in the mean is calculated at t = 4 s, and the results are similar to those obtained in [23]. In Figure 3.2, the exponential convergence of MEgPC for varying mesh size

(as represented by N_r) is shown. It can be observed that as the number of elements increases, not only does the error decrease, but the rate of convergence is higher as well.

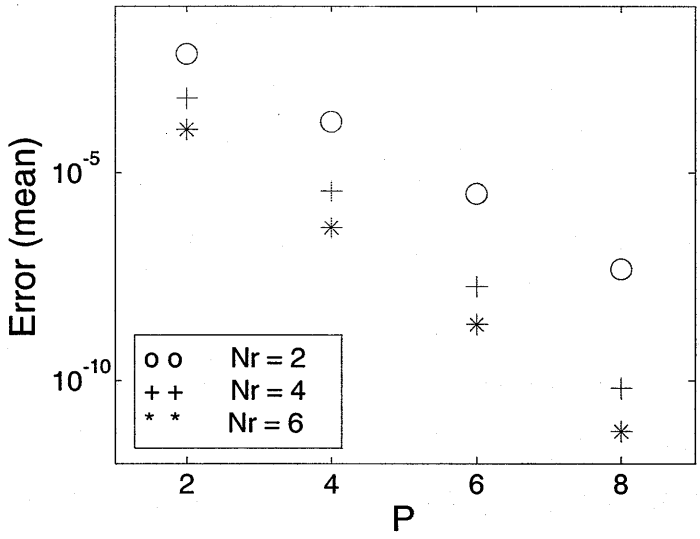

Fig. 3.2. Error convergence for MEgPC

In Figure 3.3, algebraic convergence of MEgPC in terms of the number of random elements N_r is shown. A sufficiently large algebraic index of convergence is seen, which indicates that random elements can influence the solution's accuracy dramatically.

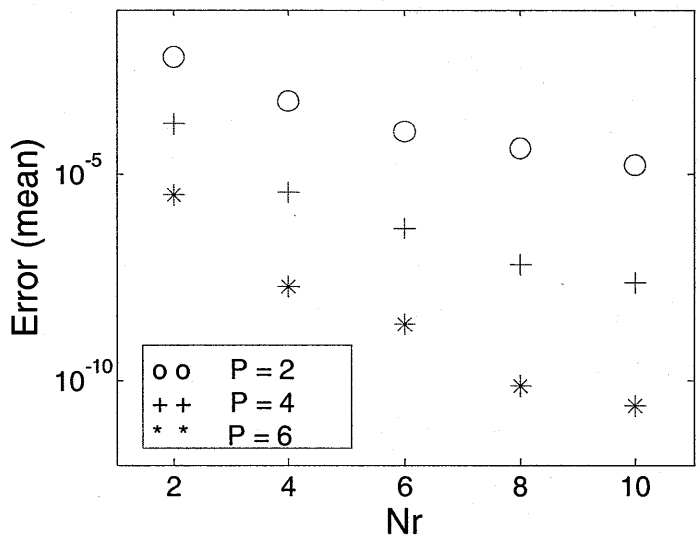

Fig. 3.3. Error convergence for MEgPC

Next, in Figure 3.4, the time evolution of the error for gPC and adaptive MEgPC approaches are compared at t=4s (with parameters P = 4, θ_1=0.001, α=0.5). It can be seen that when the error of gPC crosses the threshold limit, it triggers decomposition of the random space resulting in bounding of the error and a significant improvement in the accuracy.

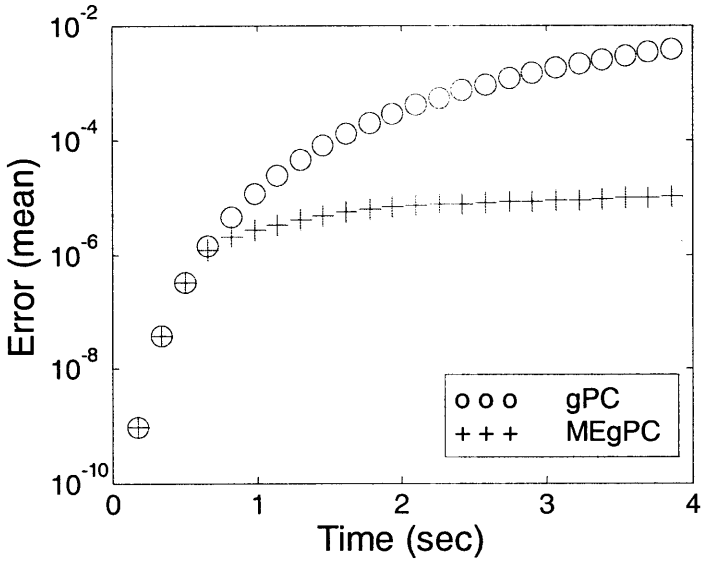

Fig. 3.4. Evolution of error for gPC and adaptive MEgPC

These simulation results clearly show that the MEgPC method represents a significant improvement over the gPC method in terms of accuracy of long-term predictions. In the next section, the technique known as the stochastic response surface method is described which addresses some of the stability issues associated with the collocation approach and proposes a regression-based framework to improve upon the robustness of the gPC method.

3.4. The stochastic response surface method (SRSM)

In recent years, researchers have applied the generalized polynomial chaos technique to various problems, including the dynamic simulation of a 7 DOF vehicle [24]. However, the collocation approach employed therein has been noted to be inherently unstable and exhibit convergence problems [28]. Moreover, different combinations of collocation points may lead to considerably different output estimates, or they may not correspond to high probability regions of the input parameter space [11]. In this context, the stochastic response surface method (SRSM) introduced in [11] provides a more robust alternative.

SRSM is similar to the generalized polynomial chaos method in the sense that it expresses random outputs in terms of a polynomial chaos expansion of generalized Askey-Wiener orthogonal polynomials; however, it uses an efficient collocation scheme with regression to determine the coefficients of the expansion. This polynomial form then allows straightforward determination of statistics such as the mean and variance, and of first and second order sensitivity information. The technique has been applied to fluid and structural mechanics problems and its computational advantages over the conventional Monte Carlo approaches have also been highlighted [11].

Collocation methods have been shown to be inherently unstable (especially with polynomial approximations of higher orders). Further, since the accuracy of these methods can typically depend on the selection of appropriate sample points (collocation points) as well, the regression-based response surface method provides a more robust means of estimating the coefficients of the reduced order functional approximation. The essence of this approach lies in using a more efficient sampling scheme (based on a heuristic technique) to obtain a set of collocation points larger in number than (typically, twice) the number of unknown coefficients, followed by computation of the model results at the selected points. This effectively moderates the influence of each individual collocation point. Calculation of the model output at these points therefore results in a system of equations, with the number of equations exceeding the number of unknown coefficients. This set of equations is then solved using the singular value decomposition technique, to obtain values for the deterministic coefficients in the spectral expansion.

Once the (statistically equivalent) reduced model is formulated, it can be used to facilitate analysis of the system under uncertainty. This procedure thus results in a reduction in the number of model simulations (and, therefore, a reduction in computational cost) required for estimation of output uncertainty, as compared to conventional probabilistic methods such as Monte Carlo methods. It also results in a more robust method than the generalized polynomial chaos framework [11].

3.4.1. Algorithmic Framework

Here a summary of the procedure employed in SRSM is presented, as applied to the present analysis.

a) Represent uncertain input parameters in terms of random variables. A vehicle and/or terrain parameter X_j can then be written (as in the gPC method) in the following way:

$$X_j = \mu_j + \sigma_j \xi \tag{3.24}$$

where μ_j is the mean, σ_j is a constant (and represents the standard deviation when Xj is normally distributed) and ξ is a random variable.

b) Express the model output under consideration in terms of the same set of random variables (in the form of a truncated series expansion) as:

$$y = a_0 + \sum_{i_1=1}^{m} a_{i_1} \Gamma_1(\xi_{i_1}) + \sum_{i_1=1}^{m} \sum_{i_2=1}^{i_1} a_{i_1 i_2} \Gamma_2(\xi_{i_1}, \xi_{i_2}) + \ldots \tag{3.25}$$

where y refers to an output metric, $\boldsymbol{\xi} = [\xi_{i1} \ldots \xi_{im}]$ is the multi-dimensional random variable, ξ_{i1}, $\xi_{i2}, \ldots$ are i.i.d. uniform random variables, $\Gamma_p(\xi_{i1}, \xi_{i2}, \ldots, \xi_{im})$ is the generalized Askey-Wiener polynomial chaos of degree p and a_{i1}, $a_{i1i2}, \ldots$ are the corresponding coefficients.

c) Estimate the unknown coefficients of the approximating series expansion. As explained earlier in Section 3.4, this is accomplished via a regression-based approach, by computing the model output at a set of collocation points [25], [26] selected using the heuristic technique proposed in [11]. Taking their number (N_c) to be nearly twice in number to the number of coefficients (N_m+1) has been shown to yield robust coefficient estimates [11], [25]. A system of linear equations with the number of equations exceeding the number of unknown coefficients is thus obtained after calculation of the model output at the sample points, as:

$$\begin{pmatrix} \Gamma_0(\xi_0) & \Gamma_1(\xi_0) & \ldots. & \Gamma_{N_m}(\xi_0) \\ \Gamma_0(\xi_1) & \Gamma_1(\xi_1) & \ldots. & \Gamma_{N_m}(\xi_1) \\ . & . & . & \\ . & . & . & \\ \Gamma_0(\xi_{N_c}) & \Gamma_1(\xi_{N_c}) & \ldots. & \Gamma_{N_m}(\xi_{N_c}) \end{pmatrix} \begin{pmatrix} y_0(t) \\ y_1(t) \\ . \\ . \\ y_{N_m}(t) \end{pmatrix} = \begin{pmatrix} y(t,\xi_0) \\ y(t,\xi_1) \\ . \\ . \\ y(t,\xi_{N_c}) \end{pmatrix} \tag{3.26}$$

This system can then be solved for the unknown coefficients using the singular value decomposition form of the linear least square method (See Appendix A1).

The reduced equivalent model can henceforth be used for analysis, which avoids the requirement of multiple runs of the (generally non-linear) model as in more conventional techniques, thereby resulting in reduced simulation time.

d) Estimate the statistics of the output metric, modeled as a stochastic response surface, using an efficient Monte Carlo method such as the Latin Hypercube Sampling Method [14], [15]. From the set of N samples (y_{s1}, y_{s2}, ... y_{sN}) thus generated, the mean and variance may be obtained as:

$$\mu = \frac{1}{N}\sum_{l=1}^{N} y_{sl} \tag{3.27}$$

$$\sigma^2 = \frac{1}{N}\sum_{l=1}^{N} (y_{sl} - \mu)^2 \tag{3.28}$$

The convergence of the approximation may also be determined through comparison with the results from a higher order approximation. This may be achieved by first using the next-higher-order polynomial chaos expansion, then repeating the process for the estimation of unknown coefficients. If the estimates of the probability density functions (*pdf*s) of output metrics are found to agree closely, the expansion is assumed to have converged, and the higher order approximation may be used to calculate the *pdf*s of the other output metrics. Further, it may be noted that as with gPC, the number of model simulations is greatly reduced relative to conventional methods, thus improving computational efficiency. Moreover, the accuracy of the computational model may also be increased by increasing the order of the polynomial chaos expansion.

3.5. Application Methodology

The generalized polynomial chaos method for uncertainty analysis can be extended to solve ODEs related to vehicle dynamics in a stochastic framework. This is discussed below. First the basic steps in applying polynomial chaos for a simple first order differential

equation are shown as an illustration. Then the method is applied to solve coupled differential equations for a quarter-car model.

3.5.1. Simple Stochastic ODE Model

Consider the following stochastic ordinary differential equation:

$$\frac{du(t)}{dt} + ku(t) = 0, \qquad u_{t=0} = 1 \tag{3.29}$$

For k as a random variable, $k = k(\xi)$, the solution $u(t)$ of the above equation will be a stochastic process $u(t,\xi)$. Here ξ is a standard normal random variable, $N(0,1)$. The equation can be rewritten as:

$$\frac{du(t,\xi)}{dt} + k(\xi)u(t,\xi) = 0, \qquad u_{t=0} = 1 \tag{3.30}$$

For the following analysis, the uncertain parameter k is considered to have a normal probability distribution, with a mean value of 1/2 and standard deviation of 1/8. Thus, k can be represented in terms of ξ as:

$$k(\xi) = \mu_k + \sigma_k \xi = \frac{1}{2} + \frac{1}{8}\xi \tag{3.31}$$

Next, $u(t,\xi)$ is represented in the form of a truncated series expansion consisting of Hermite polynomials of the random variable ξ, as:

$$u(t,\xi) = \sum_{j=0}^{N_0} a_j(t) H_j(\xi) \tag{3.32}$$

Propagating this through the ODE, equation (3.30) can be written as:

$$\sum_{j=0}^{N_0} \frac{da_j(t)}{dt} H_j(\xi) = -\sum_{j=0}^{N_0} a_j(t) k(\xi) H_j(\xi) \tag{3.33}$$

Choosing a set of Q collocation points, ξ^i with $0 \le i \le Q$, equality of (3.33) is enforced at these points:

$$\sum_{j=0}^{N_0} \frac{da_j(t)}{dt} H_j(\xi^i) = -\sum_{j=0}^{N_0} a_j(t) k(\xi^i) H_j(\xi^i) , \qquad 0 \le i \le Q \tag{3.34}$$

Also, define:

$$A_{j,i} = H_j(\xi^i) \tag{3.35}$$

Therefore equation (3.34) reduces to:

$$\sum_{j=0}^{N_0} \frac{da_j(t)}{dt} A_{j,i} = -\sum_{j=0}^{N_0} a_j(t) k(\xi^i) A_{j,i} , \qquad 0 \le i \le Q \tag{3.36}$$

This represents the $Q+1$ equations (for each ξ^i), with each equation having N_0+1 terms. Now combine equations (3.32) and (3.35) to get:

$$u_i(t,\xi^i) = \sum_{j=0}^{N_0} a_j(t) A_{j,i} , \qquad 0 \le i \le Q \tag{3.37}$$

where $u_i(t,\xi^i)$ reflects the result obtained after solving the differential equation in time, using the collocation point ξ^i.

As a result, the $Q+1$ equations for each ξ^i can be expressed as:

$$\frac{du_i(t,\xi^i)}{dt} = -\sum_{j=0}^{N_0} a_j(t) k(\xi^i) A_{j,i} , \qquad 0 \le i \le Q \tag{3.38}$$

Now $u_i(t,\xi^i)$ can be solved with respect to time using the initial conditions provided. Once the time evolution for each $u_i(t,\xi^i)$ is obtained, it can be used to determine the time evolution for the coefficients $a_j(t)$ as in Equation 3.26.

When implementing the generalized polynomial chaos method, the number of collocation points is chosen to be equal to the number of coefficients to be determined ($Q=N_0$) and the points are sampled using to the Efficient Collocation Method. The mean value at a time t is then given by:

$$\mu = a_0(t) H_0(\xi) \tag{3.39}$$

and the variance is obtained as:

$$\sigma^2(t) = \sum_{j=0}^{N_0}(a_j(t))^2\left\langle H_j^2\right\rangle - (a_0(t))^2 = \sum_{j=1}^{N_0}(a_j(t))^2\left\langle H_j^2\right\rangle \tag{3.40}$$

where $\langle\bullet\rangle$ represents the ensemble average. For orthonormal polynomials, $\left\langle H_j^2\right\rangle = 1$.

In the present analysis, following the SRSM implementation described in Section 3.4, Q is chosen such that the number of collocation points is more than (typically twice) the number of deterministic coefficients to be determined, and points are sampled according to the proposed heuristic technique. The resulting system of equations is then solved using singular value decomposition (See Appendix A1).

The mean value obtained after performing Monte Carlo runs on the reduced-order model can be compared to the deterministic solution of the equation, given by:

$$\mu_{exact} = e^{-\mu_k t} \tag{3.41}$$

as well as to the mean of the stochastic solutions, given by:

$$\mu_{exact}(t) = u_o \int_S e^{-kt} f(k)dk = y_o \frac{e^{-(\mu_k+\sigma_k)t}(e^{2\sigma_k t}-1)}{2\sigma_k t} \tag{3.42}$$

Figure 3.5 shows the stochastic solutions for the mean. The time evolution of the unit standard deviation plotted as error bars is shown in Figure 3.6.

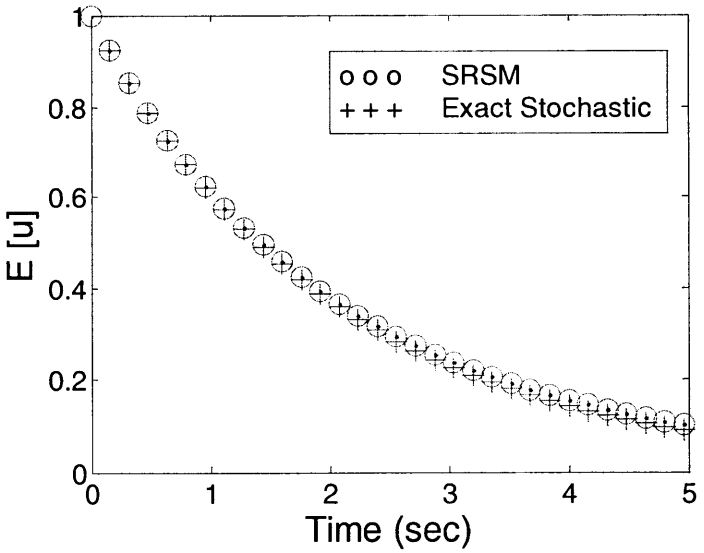

Fig. 3.5. Mean solution for the first order equation

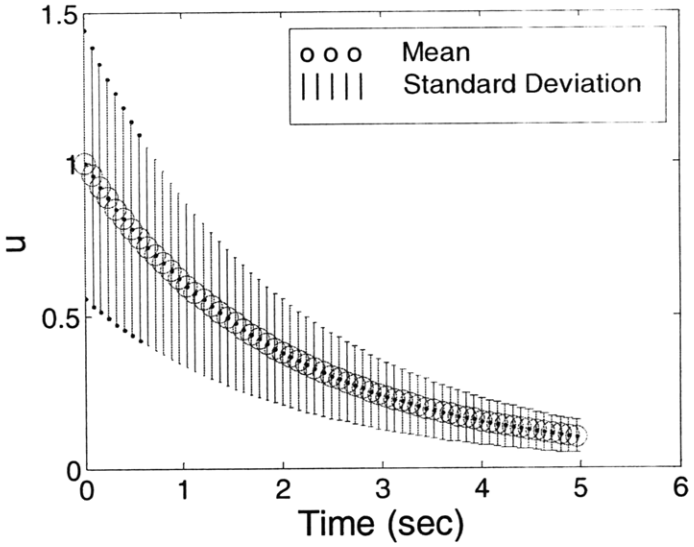

Fig. 3.6. Unit standard deviation from the mean plotted as error bars

3.5.2. Application to a Quarter-Car Model

A two degree of freedom quarter-car model of a vehicle suspension (see Figure 3.7) under uncertainty is now studied. The sprung mass, m_s, and unsprung mass, m_u, are connected by a nonlinear spring of stiffness k_s, and a linear damper with damping coefficient c. The input is applied through a forcing function $z(t)$, to m_u, through a linear spring k_u. This represents the interaction of the quarter car system with the terrain. The governing equations for the quarter car system are given as:

$$m_s \frac{d^2 x_1}{dt^2} = -k_s \left(x_1 - x_2\right)^3 - c\left(\dot{x}_1 - \dot{x}_2\right) \tag{3.43}$$

$$m_u \frac{d^2 x_2}{dt^2} = k_s \left(x_1 - x_2\right)^3 + c\left(\dot{x}_1 - \dot{x}_2\right) + k_u \left(z(t) - x_2\right) \tag{3.44}$$

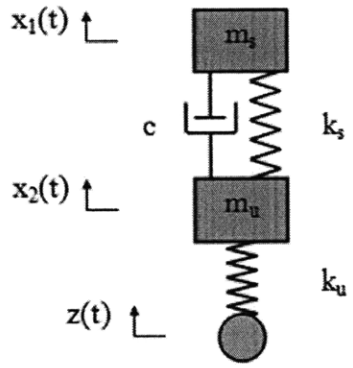

Fig. 3.7. Quarter-car model

Parametric uncertainty arises in the suspension stiffness. The two springs are considered to have uncertain spring constant values, uniformly distributed about a mean stiffness value. This can be represented as:

$$k_s = \mu_{k_s} + \xi_1 \sigma_{k_s} \tag{3.45}$$

$$k_u = \mu_{k_u} + \xi_2 \sigma_{k_u} \tag{3.46}$$

The sprung mass displacement is analyzed under parametric uncertainty and expressed as a spectral series expansion of Legendre polynomials of uniform random variables ξ_1 and ξ_2, in $U[-1,1]$. In general, the state can be expressed as:

$$\mathbf{X} = \left[x_1 \;\; x_2 \;\; \dot{x}_1 \;\; \dot{x}_2\right]^T \tag{3.47}$$

$$x_i(t,\xi) = \sum_{j=0}^{P} x_{i,j}(t)\Phi_j(\xi) \qquad \text{i=1,2} \tag{3.48}$$

$$\dot{x}_i(t,\xi) = \sum_{j=0}^{P} \dot{x}_{i,j}(t)\Phi_j(\xi) \qquad \text{i=1,2} \tag{3.49}$$

where $\xi = \left[\xi_1, \xi_2\right]$.

The parameter values used in this analysis are shown in Table 3.2.

TABLE 3.2
PARAMETERS IN QUARTER CAR MODEL

PARAMETER	μ	σ
k_s	400 N/m^3	40 N/m^3
k_u	2000 N/m	200 N/m
m_s	20 kg	-
m_u	40 kg	-
c	600 Ns/m	-

While the exact stochastic solutions may be easy to obtain for simple systems such as the one discussed above, they may be difficult to obtain for large and complex systems. For such scenarios, the exact solution can be replaced by a reference solution obtained from a standard Monte Carlo (SMC) analysis.

For a step input (with a step size of 0.2 m) - which models vehicle traversal over a bump or obstacle - it is observed that parametric uncertainty causes significant variation in the

resulting output of x_1, the sprung mass displacement (see Figure 3.8), thus indicating the importance of considering uncertainty during dynamic analysis. Similar results obtained for a sinusoidal input (with an amplitude of 0.1 m and time period of 1 s) are also shown in Figure 3.9.

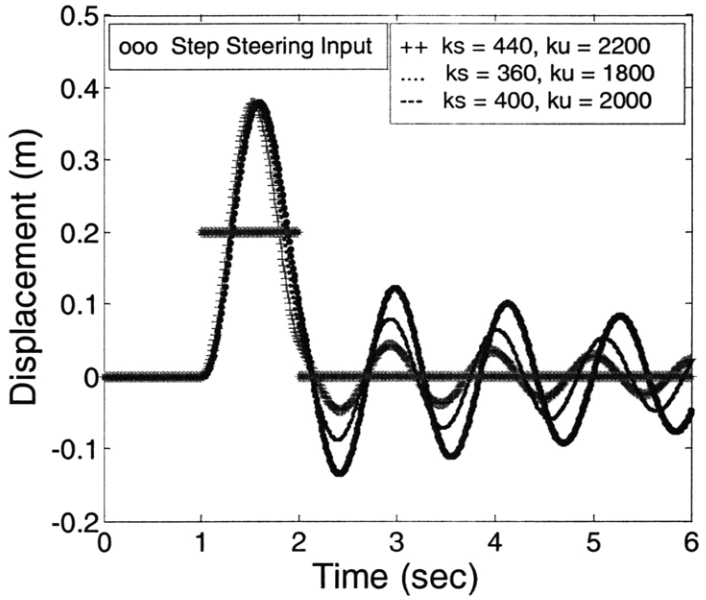

Fig. 3.8. x_1 (sprung mass displacement) for various stiffness and damping values

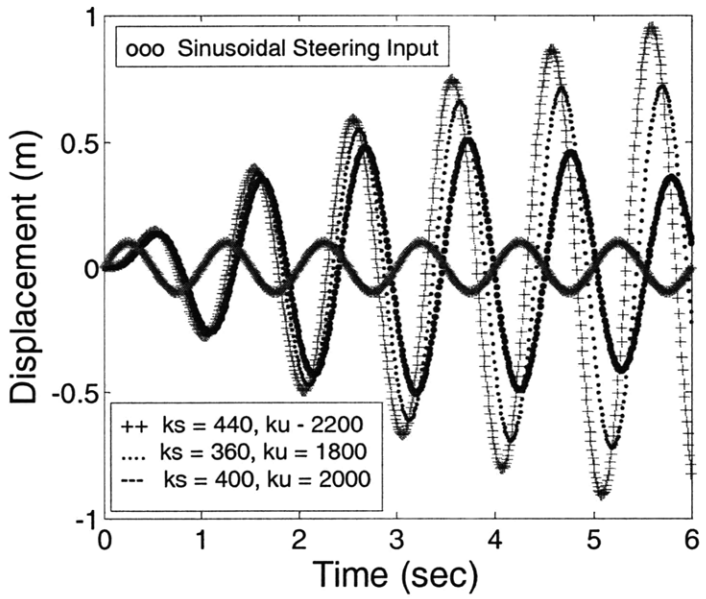

Fig. 3.9. x_1 (sprung mass displacement) for various stiffness and damping values

The time profiles obtained for the mean and the standard deviation of the displacement of x_1 (i.e. the sprung mass) are respectively shown in Figure 3.10 and Figure 3.11, for the gPC, MEgPC (P = 3, θ_1=0.001, α=0.5) and SMC methods. It is observed that while the results for the mean match closely, there is a substantial difference between the predicted variance from the two polynomial chaos-based techniques, with MEgPC yielding more precise results than gPC when compared to the baseline SMC analysis. This difference however, reduces with time due to the nature of the input considered. Computation times for SMC and MEgPC approaches with respect to the gPC method are shown in Table 3.3.

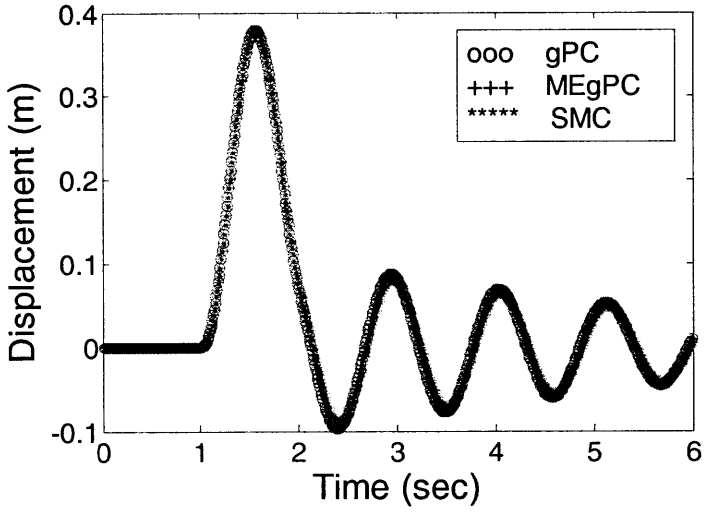

Fig. 3.10. Mean of x_1 (sprung mass displacement)

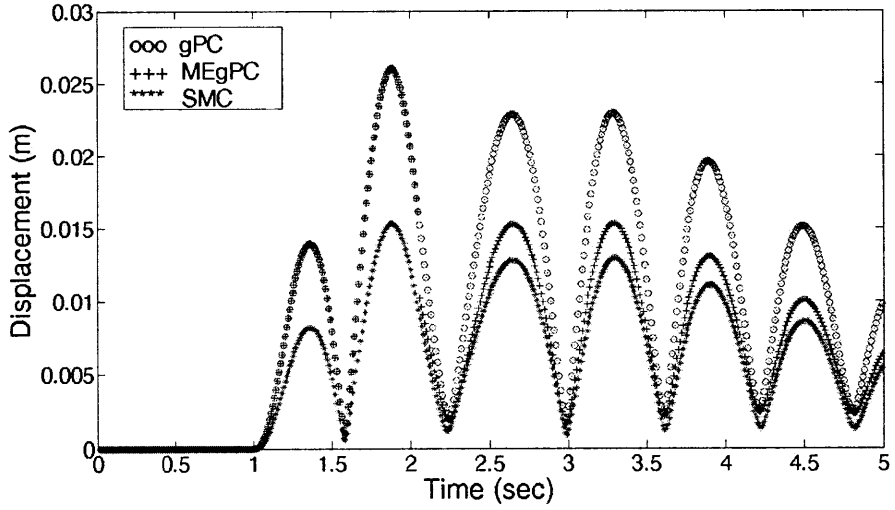

Fig. 3.11. Standard deviation of x_1 (sprung mass displacement)

TABLE 3.3
COMPUTATION TIME FOR THE VARIOUS APPROACHES

METHOD	RATIO OF SIMULATION TIME (s)
SMC (2000 runs)	191.96 s
MEgPC	1.585 s
gPC	1.00 s

Next, the system was analyzed for a sinusoidal terrain input. Though the mean results still agree closely, slight inconsistency is found in the variance predictions made using the gPC approach as expected (see Figures 3.12 and 3.13). This is, however, not the case with the MEgPC approach, which exhibits only a small bounded error over time. Further, for the sinusoidal input, this difference in the variance prediction does not decay as in the previous case. The computation times for SMC and MEgPC approaches with respect to the gPC method are shown in Table 3.4.

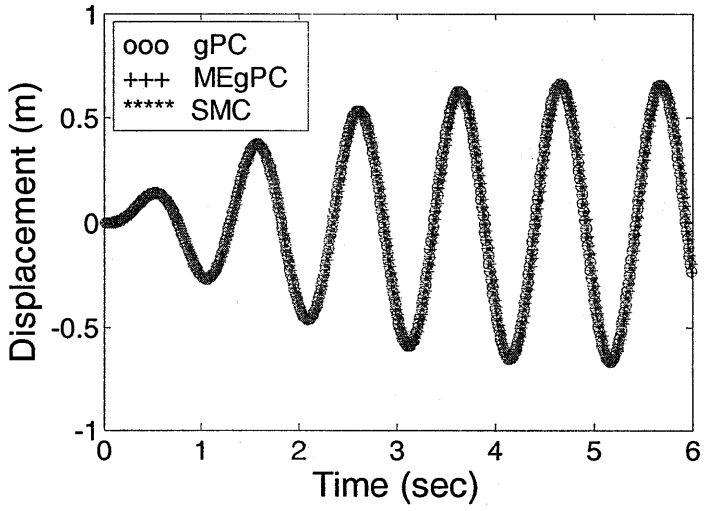

Fig. 3.12. Mean of x_1 (sprung mass displacement)

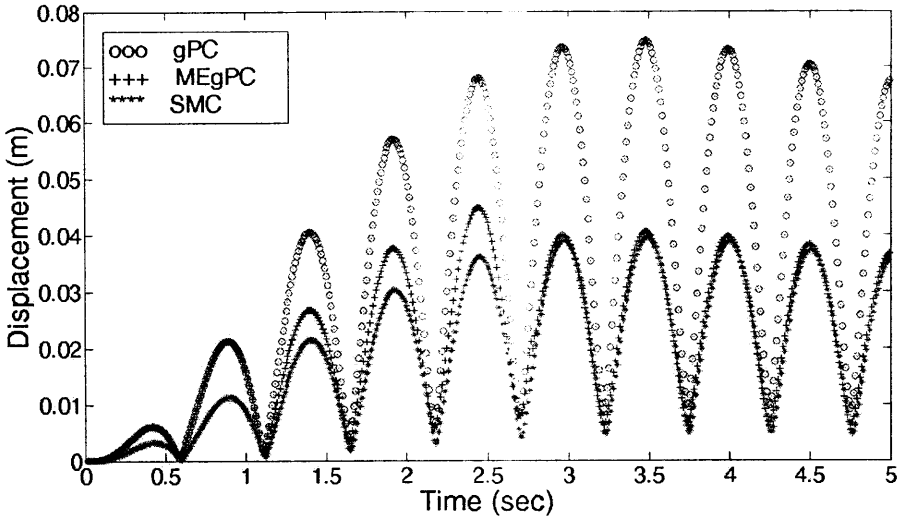

Fig. 3.13. Standard deviation of x_1 (sprung mass displacement)

TABLE 3.4
COMPUTATION TIME FOR THE VARIOUS APPROACHES

METHOD	RATIO OF SIMULATION TIME
SMC (2000 runs)	197.76 s
MEgPC	2.33 s
gPC	1.00 s

These results clearly show the applicability of polynomial chaos-based approaches and their advantage over the conventional Monte Carlo-based technique in terms of reduced computational costs. Applications of these approaches to vehicle mobility prediction, path planning and motion control will be discussed in subsequent chapters.

4. MOBILITY PREDICTION UNDER UNCERTAINTY

4.1. Overview

As noted in Section 1.1, a "mobility prediction" capability is often integrated into motion planning algorithms which allows the vehicle to evaluate the safety of its traversal over unstructured environments. This functional ability is therefore critical to the efficient operation of vehicle systems, and hence to the successful deployment of UGVs that can operate effectively on challenging terrain with minimal human supervision. However, there has been little research that explicitly addresses the challenge of autonomously assessing the traversability over a given terrain region or obstacle under uncertainty. While significant work has been done to understand and predict the mobility of vehicles in natural terrain [1], [2], these efforts assume accurate knowledge of vehicle parameters and wheel- (or track-) soil interaction properties (gathered from terrain measurement devices such as cone penetrometers). In field conditions, however, UGVs often only have access to sparse and uncertain parameter estimates drawn from "standard" robotic sensors such as LIDAR. Moreover, significant uncertainties are often associated with estimates of vehicle parameters, due to effects such as loading, wear, fuel consumption, etc. It is therefore imperative to consider these uncertainties when deriving predictions of vehicle mobility.

Various statistical methods for mobility prediction have been developed by the U.S. Army over the past 50 years, including the NATO Reference Mobility Model (NRMM), NRMM II, and others [1], [2], [29]. These are numerical algorithms for predicting cross-country vehicle movement at length scales of several meters to several kilometers. Further, they were developed for vehicles weighing 500 kg or more and are based on empirical results drawn from resource-intensive experimental testing. Thus, these techniques are generally inapplicable to small robotic vehicles, for which extensive empirical test data does not yet exist. Also, the mobility prediction problem considered in the present analysis is concerned with movement over particular vehicle-sized terrain regions and obstacles, rather than gross (i.e. km-scale) mobility characteristics.

Previous research in this area has focused on stochastic performance prediction of ground vehicles using classical Monte Carlo simulation methods [7]. Another recent approach relies on analysis of system performance over obstacle "primitives" such as single rocks and well-characterized rock fields; however it is unclear how these results can be generalized to complex terrain profiles [30]. In addition, the issue of mobility through terrain regions with non-geometric hazards (such as highly deformable or slippery regions) has not been addressed in this paradigm. Other related work has developed a stochastic analysis of terrain profiles and wheel-terrain interaction [24]. Although based on the stochastic analysis technique proposed here, it does not explicitly address the mobility prediction problem.

Most other research has attempted to designate terrain regions as "traversable" or "non-traversable" based solely on remotely-sensed terrain geometry. One such approach for outdoor robotic vehicles is described in [31]. An extension to the work, described in [32], attempts to characterize the nature of various outdoor obstacles; however this work focuses solely on identifying obstacles that are likely to be traversable despite their geometry (e.g. tall grass, which may possess an obstacle-like geometric profile but is often traversable due to its compliant nature). Another approach is presented in [33] to detect obstacles from color and LIDAR data. A terrain classification component is used to distinguish vegetation from the underlying terrain. This improves the estimate of the location of the load-bearing surface in thick vegetation; however it is not employed for mobility prediction.

In summary, most previous methods either do not explicitly analyze vehicle mobility on rough terrain or rely on deterministic analysis that assumes precise knowledge of vehicle and/or terrain parameters. The present work has attempted to address some of these concerns. In Section 4.2, the vehicle and wheel-soil interaction models that will be used in the present analysis are described. The application of the response surface-based uncertainty analysis technique to the domain of vehicle dynamic analysis and prediction of mobility characteristics is discussed in Section 4.3, and results are presented in Section 4.4. Long-term prediction results obtained using the MEgPC approach are also shown in Section 4.5.

4.2. Vehicle and Wheel-Terrain Interaction Models

4.2.1. Robot Dynamic Model

A three degree of freedom vehicle model (see Figure 4.1), is considered in this study that takes lateral acceleration, yaw motion and roll dynamics into account, as in [34]. The linearized equations for this model are given as:

$$mV(\dot{\beta}+\dot{\psi})-m_s h\ddot{\varphi}=\sum F_y=C_f(\delta-\frac{\dot{\psi}l_f}{V}-\beta)+C_r(\frac{\dot{\psi}l_r}{V}-\beta) \tag{4.1}$$

$$I_{zz}\ddot{\psi}=\sum M_z=C_f(\delta-\frac{\dot{\psi}l_f}{V}-\beta)l_f-C_r(\frac{\dot{\psi}l_r}{V}-\beta)l_r \tag{4.2}$$

$$(I_{xx}+m_s h^2)\ddot{\varphi}=\sum M_x=m_s gh\varphi+m_s hV(\dot{\beta}+\dot{\psi})+M_s \tag{4.3}$$

where $M_s=-(k_f+k_r)\varphi-(b_f+b_r)\dot{\varphi}$

A list of the parameter values used in the current study is provided in Table 4.1.

TABLE 4.1
VEHICLE MODEL PARAMETERS

Symbol	DESCRIPTION	MEAN VALUE	UNITS
δ	Front wheel steering angle	-	rad
β	Slip angle	-	rad
φ	Roll angle	-	rad
ψ	Yaw angle	-	rad
I_{xx}	Roll moment of inertia	834	kg m^2
I_{zz}	Yaw moment of inertia	2050	kg m^2
m	Total vehicle mass	2030	Kg
m_s	Sprung mass	1830	Kg
V	Longitudinal velocity	10	m/s
h	Height of center of gravity from roll axis	0.35	m
h_a	Height of roll axis from ground	0.21	m
y_w	Track width	1.56	m
C_f	Cornering stiffness of lumped front wheels	1440	N/rad
C_r	Cornering stiffness of lumped rear wheels	1280	N/rad
l_f	Distance of front axle from center of gravity	0.43	m
l_r	Distance of rear axle from center of gravity	0.33	m
k_f	Front roll stiffness	30000	Nm/rad
k_r	Rear roll stiffness	30000	Nm/rad
b_f	Front axle damping rate	3600	Nms/rad
b_r	Rear axle damping rate	3600	Nms/rad
g	Acceleration due to gravity	9.8	m/s^2

In addition to forces from tire compliance, lateral components of the contact forces on the vehicle can arise due to terrain unevenness. Given a terrain elevation map, modeled as a continuous, differentiable function of planar position *z(x,y)*, the terrain disturbance force T_i acting at each wheel can be written as [34]:

$$T_i = N_i \left((\partial z / \partial x_o)\hat{x}_o + (\partial z / \partial y_o)\hat{y}_o \right) \cdot \hat{y} \qquad i = 1...4 \tag{4.4}$$

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.

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

$$M_s = -k_f(\varphi - \varphi_f) - k_r(\varphi - \varphi_r) - b_f(\dot{\varphi} - \dot{\varphi}_f) - b_r(\dot{\varphi} - \dot{\varphi}_r) \tag{4.5}$$

where φ_f and φ_r are the front and rear terrain roll angles, with $\dot{\varphi}_f$ and $\dot{\varphi}_r$ as their corresponding rates.

To compute these 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)*, these quantities are calculated as [34]:

$$\varphi_i = (z_{i+1} - z_i) / y_w \tag{4.6}$$

$$\dot{\varphi}_i = (\dot{z}_{i+1} - \dot{z}_i) / y_w \tag{4.7}$$

where the rate of elevation change can be computed as:

$$\dot{z}_i = V\left((\partial z / \partial x)\cos(\psi + \beta) + (\partial z / \partial y)\sin(\psi + \beta) \right) \tag{4.8}$$

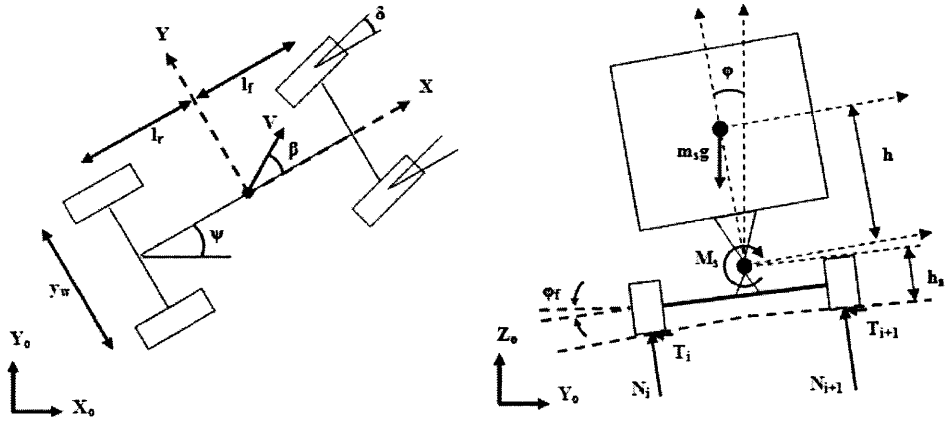

Fig. 4.1. Vehicle model for mobility analysis under uncertainty

Finally, the linearized equations for this model can be rewritten as:

$$\dot{\beta} = -\frac{GC}{mV}\beta + \left(-1 + \frac{KG}{mV^2}\right)\dot{\psi} + \frac{C_f G}{mV}\delta + \frac{m_s h M_s}{mVI_{xx}^o} + \frac{m_s^2 g h^2}{mVI_{xx}^o}\varphi + \frac{G}{mV}\sum T_i \tag{4.9}$$

$$\ddot{\varphi} = \frac{m_s gh}{I_{xx}^o}\varphi + \frac{M_s}{I_{xx}^o} - \frac{m_s Ch}{mI_{xx}^o}\beta + \frac{m_s Kh}{mVI_{xx}^o}\dot{\psi} + \frac{C_f m_s h}{mI_{xx}^o}\delta + \frac{m_s h}{mI_{xx}^o}\sum T_i \tag{4.10}$$

$$\ddot{\psi} = \frac{K}{I_{zz}}\beta - \frac{D}{VI_{zz}}\dot{\psi} + \frac{C_f l_f}{I_{zz}}\delta + \frac{1}{I_{zz}}\sum T_i l_i \tag{4.11}$$

where

$$C = C_f + C_r,\ K = C_r l_r - C_f l_f,\ D = C_f l_f^2 + C_r l_r^2,\ G = 1 + (m_s^2 h^2)/(mI_{xx}^o),\ I_{xx}^o = I_{xx} + m_s h^2\left(1 - m_s/m\right).$$

To measure vehicle stability, a roll-over coefficient is adopted from [34]. Using the principle of balance of moments and vertical forces, the roll-over metric for the linear model under consideration is given as:

$$R = \frac{2m_s}{mgy_w}(h_a + h)\left(v(\dot{\beta} + \dot{\psi}) - h\ddot{\varphi}\right) \tag{4.12}$$

where h_a is the height of the roll axis above the ground and y_w is the track width. The coefficient may further be expressed in terms of the state space variables from the equations of motion above. For this metric, $|R|>1$ indicates vehicle wheel liftoff and thus impending roll-over.

4.2.2. Bekker Wheel-Soil Interaction Model

A classical Bekker-type wheel terrain interaction model has been used to calculate the drawbar pull (i.e. net longitudinal wheel thrust) in the present study [35]-[37]. This model assumes quasi-static motion, and that the wheel is stiff relative to the terrain.

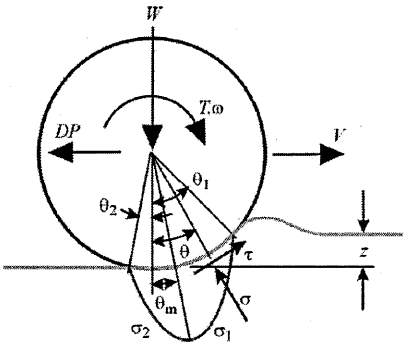

Fig. 4.2. Wheel-terrain interaction model for rigid wheel on deformable terrain

For the vehicle soil interaction model shown in Figure 4.2, the drawbar pull is given by:

$$DP = rb\left[\int_{\theta_1}^{\theta_2} \tau(\theta)\cos\theta \, d\theta - \int_{\theta_1}^{\theta_2} \sigma(\theta)\sin\theta \, d\theta\right] \tag{4.13}$$

where $\sigma(\theta)$ and $\tau(\theta)$ represent, respectively, the normal stress and the shear stress at the wheel-terrain interface (divided into two regions in Figure 4.2), and are given by:

$$\sigma_1(\theta) = \left(\frac{k_c}{b} + k_\varphi\right)\left(r(\cos\theta - \cos\theta_1)\right)^n \tag{4.14}$$

$$\sigma_2(\theta) = \left(\frac{k_c}{b} + k_\varphi\right)\left[r\left(\left(\cos\left(\theta_1 - \theta\frac{(\theta_1 - \theta_m)}{\theta_m}\right) - \cos\theta_1\right)\right)\right]^n \tag{4.15}$$

$$\tau(\theta) = (c + \sigma(\theta)\tan\varphi)\left(1 - e^{-\frac{r}{k}[\theta_1 - \theta - (1-i)(\sin\theta_1 - \sin\theta)]}\right) \tag{4.16}$$

The drawbar pull can hence be written as:

$$DP = rb\left(\int_0^{\theta_m} \tau_2(\theta)\cos\theta \, d\theta + \int_{\theta_m}^{\theta_1} \tau_1(\theta)\cos\theta \, d\theta - \int_0^{\theta_m} \sigma_2(\theta)\sin\theta \, d\theta - \int_{\theta_m}^{\theta_1} \sigma_1(\theta)\sin\theta \, d\theta\right) \tag{4.17}$$

The parameters employed in (4.14)-(4.17) are defined in Table 4.2.

TABLE 4.2
PARAMETERS INVOLVED IN DRAWBAR PULL CALCULATION

SYMBOL	DESCRIPTION
r	Wheel radius
b	Wheel width
θ_1	Angle corresponding to start of wheel-terrain contact
θ_2	Angle corresponding to loss of wheel-terrain contact
θ_m	Maximum stress angle
c	Cohesion
φ	Internal friction angle
i	Wheel slip
n	Sinkage exponent
k_c, k_φ	Pressure sinkage moduli
k	Shear deformation modulus

4.3. Mobility Analysis Scenarios

In this section simulation studies of the proposed method for mobility prediction under uncertainty are discussed. A brief description about the incorporation of vehicle and/or terrain parameter uncertainty, and of the mobility analysis scenarios, to study the

performance of the proposed polynomial chaos-based techniques, is also provided. In the present work, two scenarios are studied that involve terrain and vehicle parameter uncertainty. In the first scenario, a stochastic response surface will be generated for calculation of the mean drawbar pull (in lieu of using more computationally expensive Monte Carlo analysis-based on equations (4.14)-(4.17)). In the second scenario, a roll-over analysis will be performed using reduced order response surface expansions for the state variables associated with the vehicle's motion, for various steering maneuvers. These analyses are discussed below.

4.3.1. Scenario I

4.3.1.1. Inclusion of Uncertainty

A reduced stochastic model is developed for the drawbar pull considering c and φ as the uncertain parameters. The parameters are assumed to be normally distributed, though other possible probability distributions (such as uniform or beta distribution) can be considered as well. They are represented as:

$$c = \mu_c + \xi_1 \sigma_c \tag{4.18}$$

$$\varphi = \mu_\varphi + \xi_2 \sigma_\varphi \tag{4.19}$$

where μ_c and μ_φ represent the mean, and σ_c and σ_φ represent the standard deviation for c and φ respectively, and ξ_1 and ξ_2 are standard normal random variables. The drawbar pull is now expressed in the form of a second order polynomial chaos expansion as:

$$DP = a_0 + a_1\xi_1 + a_2\xi_2 + a_3(\xi_1^2 - 1) + a_4(\xi_2^2 - 1) + a_5\xi_1\xi_2 \tag{4.20}$$

The parameters c and φ are chosen since they exhibit significant influence on terrain thrust. The corresponding values for c and φ used in the present analysis can be found in Table 4.3.

TABLE 4.3
PROBABILITY DISTRIBUTION INFORMATION FOR UNCERTAIN TERRAIN PARAMETERS (c, φ)

PARAMETER	DISTRIBUTION FUNCTION	MEAN	STD. DEV.
c (Heavy Clay)	Gaussian	69 kPa	8.50 kPa
φ (Heavy Clay)	Gaussian	34 deg	2.10 deg
c (Dry Sand)	Gaussian	1.04 kPa	0.125 kPa
φ (Dry Sand)	Gaussian	28 deg	1.75 deg

4.3.1.2. Scenario Description

To demonstrate the application of the technique to mobility analysis, a simplified terrain traversal scenario is presented that considers a wheeled ground vehicle traveling on flat, firm outdoor terrain (here modeled as heavy clay), then attempting to navigate up an inclined region of highly deformable terrain (here modeled as dry sand). This is illustrated in Figure 4.3. It is assumed that significant uncertainty is associated with the terrain physical parameters c and φ. A reduced order model for the drawbar pull is then formulated (as shown in (4.20)) at each time interval to obtain the stochastic mean value for the drawbar pull to be used in the equation of motion for the vehicle.

A simple description of vehicle mobility in the proposed scenario is defined as the probability that, for a given initial velocity (u_o) at initial position (A) (see Figure 4.3), the vehicle will have a positive velocity at point (B), after traversing the sandy incline. This metric has been presented as a distribution of traversal probability versus initial velocity, which can be used to predict the velocities for which the vehicle will be able to traverse the deformable terrain region with a reasonably high probability.

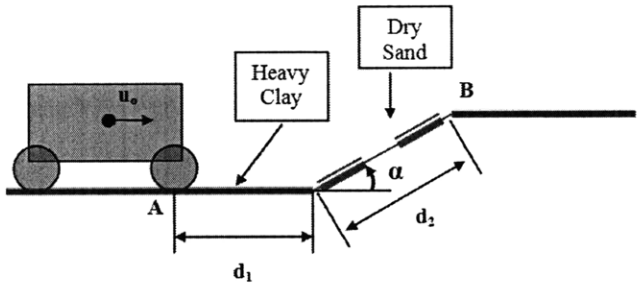

Fig. 4.3. Simplified scenario considered for mobility prediction under uncertainty

The governing equation of motion can be written as:

$$\ddot{x} = \frac{DP}{m} - g\sin\alpha \tag{4.21}$$

where m is the vehicle mass, g represents the acceleration due to gravity and α is the angle of the incline with respect to the horizontal. The parameter values used in this analysis are provided in Table 4.4.

TABLE 4.4
PARAMETERS INVOLVED IN DRAWBAR PULL CALCULATION

SYMBOL	UNITS	MEAN VALUE (HEAVY CLAY)	MEAN VALUE (DRY SAND)
R	m	0.1	0.1
B	m	0.1	0.1
M	kg	4	4
N	-	0.13	1.1
k_c	$N/m^{(n+1)}$	12.7	1
k_φ	$N/m^{(n+2)}$	1556	1528
K	m	0.025	0.025

The vehicle's mobility is then analyzed using a baseline "standard" Monte Carlo approach (SMC), the Latin Hypercube Sampling-based Monte Carlo method (LHSMC) and the SRSM technique.

4.3.2. Scenario II

4.3.2.1. Inclusion of Uncertainty

For the roll-over analysis scenario, the front and rear axle roll stiffness are considered to be normally distributed about their mean values, and are represented as:

$$k_f = \mu_{k_f} + \zeta_1 \sigma_{k_f} \tag{4.22}$$

$$k_r = \mu_{k_r} + \zeta_2 \sigma_{k_r} \tag{4.23}$$

where μ_{k_f} and μ_{k_r} represent the mean, and σ_{k_f} and σ_{k_r} represent the standard deviation for k_f and k_r respectively, and ζ_1 and ζ_2 are standard normal random variables. Then the output state variable X_i can be represented using Hermite polynomials as:

$$X_i(t,\zeta) = \sum_{j=0}^{P} X_{i,j}(t) H_j(\zeta) \tag{4.24}$$

where $\xi = [\zeta_1, \zeta_2]$.

The roll stiffness parameter values employed in the study are shown in Table 4.5.

TABLE 4.5
UNCERTAIN VEHICLE PARAMETERS IN ROLL-OVER ANALYSIS

PARAMETER	MEAN (Nm/rad)	STD. DEV. (Nm/rad)
k_f	30×10^3	4×10^3
k_r	30×10^3	4×10^3

4.3.2.2. Scenario Description

For the roll-over study, a spectral stochastic analysis [24], [38] is performed to obtain the time evolution of the roll-over coefficient, subject to various steering input functions (sinusoidal, ramp-like and a double lane change maneuver). The mean value as well as the variance in the roll-over coefficient is studied for each maneuver using Monte Carlo and polynomial chaos-based techniques. The vehicle dynamics are also studied using the polynomial chaos expansions for the various state variables from the governing equations of motion in (4.9)-(4.11).

4.4. Simulation Results

4.4.1. Mobility Prediction

First, results from the analysis of the mobility prediction scenario are presented here for inclination angle (α) equal to 6° and 15° (see Figure 4.4).

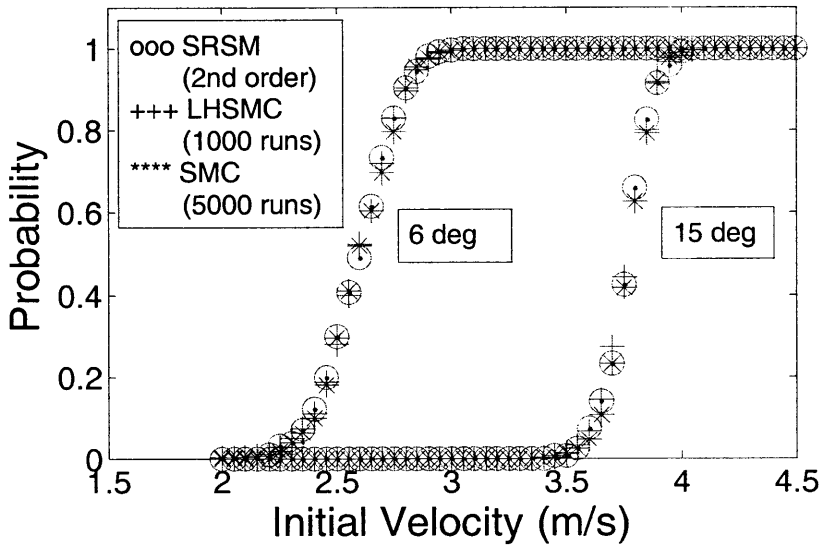

Fig. 4.4. Probability plots for mobility prediction scenario

The coefficient values obtained for the 2nd order expansion of drawbar pull (4.20) for the sandy slope (α = 15°) are:

a_0 = -1.4260, a_1 = 0.2981, a_2 = 0.5586, a_3 = 0.0000, a_4 = 0.0091, a_5 = 0.0000.

The results predict that increasing the robot's initial velocity increases the probability of safe slope traversal, as expected. Also, the minimum initial velocity required for successful traversal increases as the inclination increases. A clearly defined "transition region" can be observed, where the probability of safe traversal is a function of terrain parameter variance as well. This region effectively describes the "risk" of traversal at a certain critical velocity range.

The results from application of SRSM are compared to those obtained using SMC and LHSMC with respect to computational efficiency. To compare the methods, the ratios of the corresponding simulation time for the SMC (T_I) and LHSMC (T_{II}) approaches to the computation time using SRSM (T_o) were computed for the case when inclination angle is 10 degrees. These are given in Table 4.6.

TABLE 4.6
COMPUTATION TIME FOR MOBILITY PREDICTION ANALYSIS

METHOD	SIMULATION RUNS	RATIO OF SIMULATION TIME
SMC	5000	89.91
	20000	355.19
LHSMC	1000	18.39
	10000	179.21
SRSM (2^{nd} order)		1.00

Comparing the relative computation times, it can be inferred that the approach based on SRSM results in a significant computational reduction compared to the baseline approaches. There is however, not a significant difference between SMC and LHSMC because of the simplicity of the scenario considered.

4.4.2. Roll-Over Analysis

Results from analysis of the roll-over scenario described in Section 4.3.2 are now presented. Simulations for various vehicle maneuvers (i.e. a sinusoidal steering input with an amplitude of 0.1 radian and time period of 4 seconds, a ramp input with a slope of 0.4 degrees/second up to 4 seconds, and a double lane change steering input with an amplitude of 0.1 radian and the maneuver lasting 8 seconds) were conducted using the stochastic response surface method (SRSM), standard Monte Carlo approach (SMC) and Latin Hypercube

Sampling-based Monte Carlo method (LHSMC). The accuracy of the results from SRSM is compared to results from the application of Monte Carlo methods in Figures 4.5-4.7. Close agreement between the three methods can be observed.

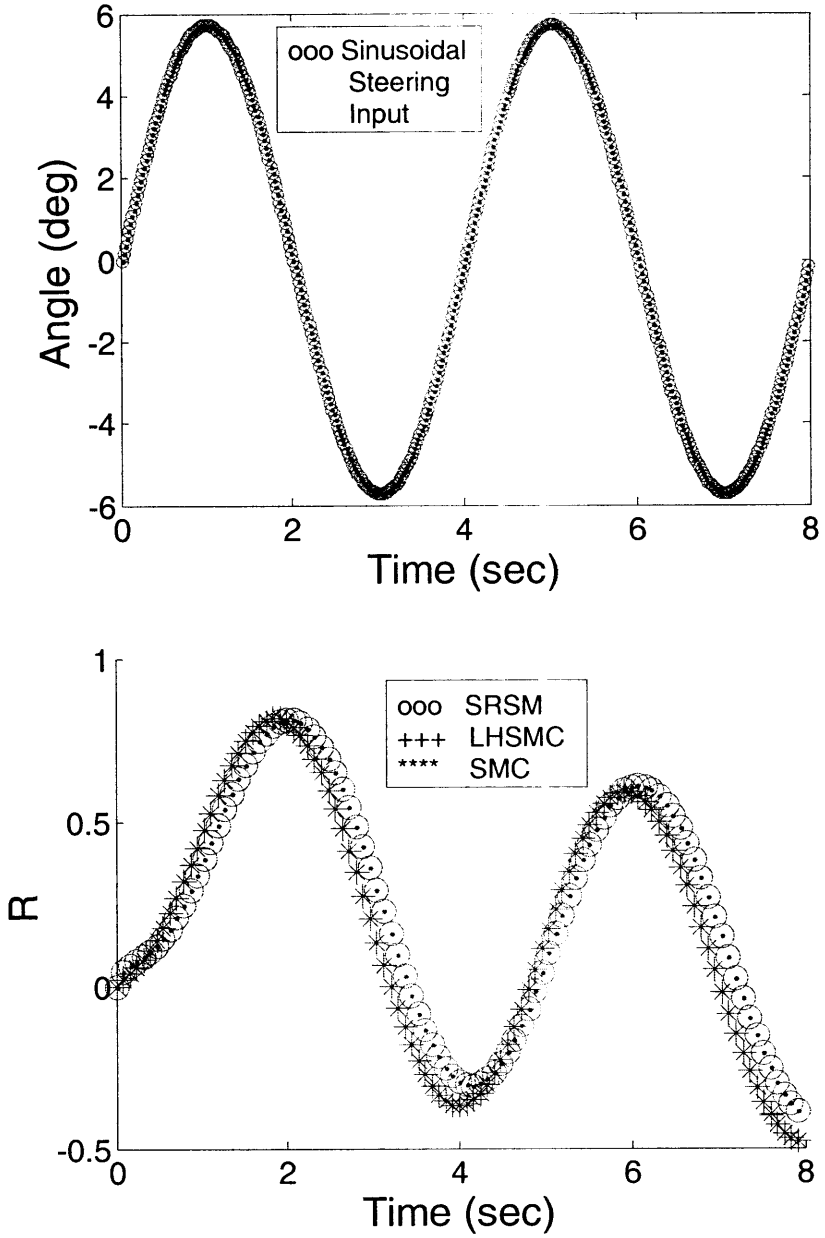

Fig. 4.5. Vehicle roll-over analysis for sinusoidal steering input

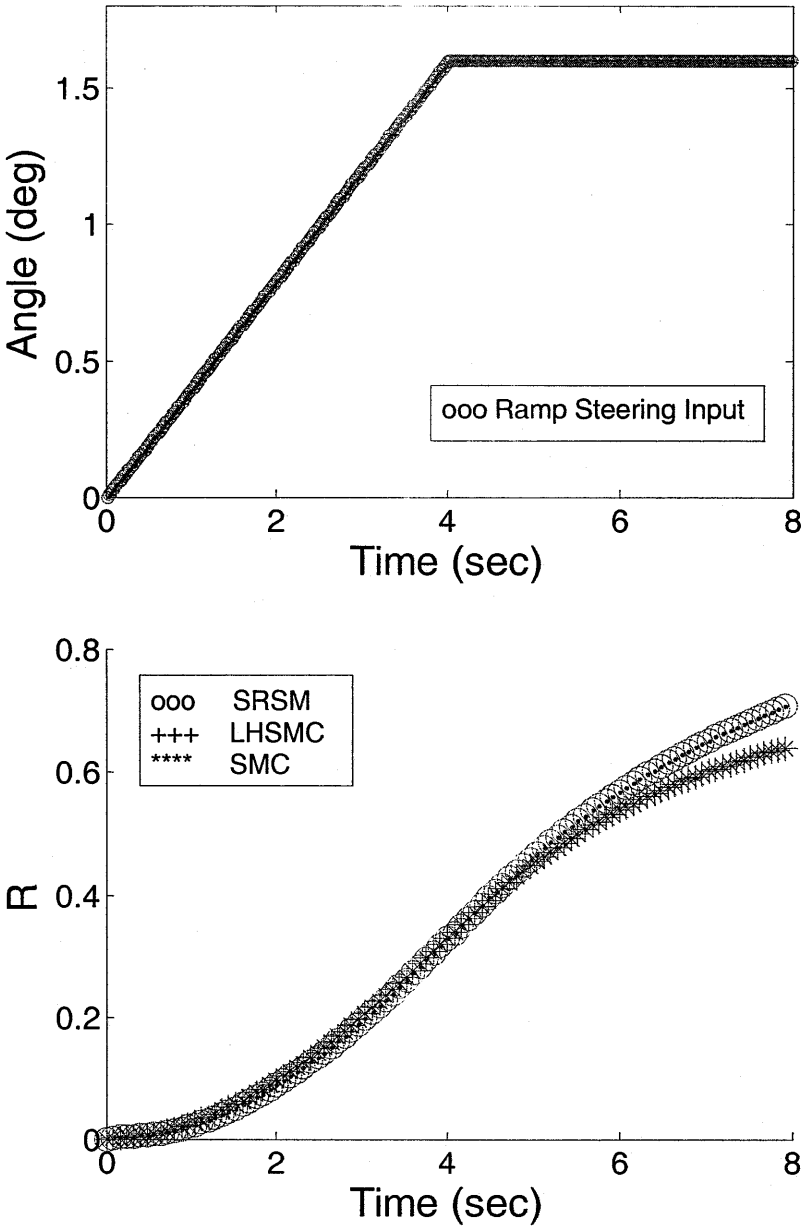

Fig. 4.6. Vehicle roll-over analysis for ramp-like steering input

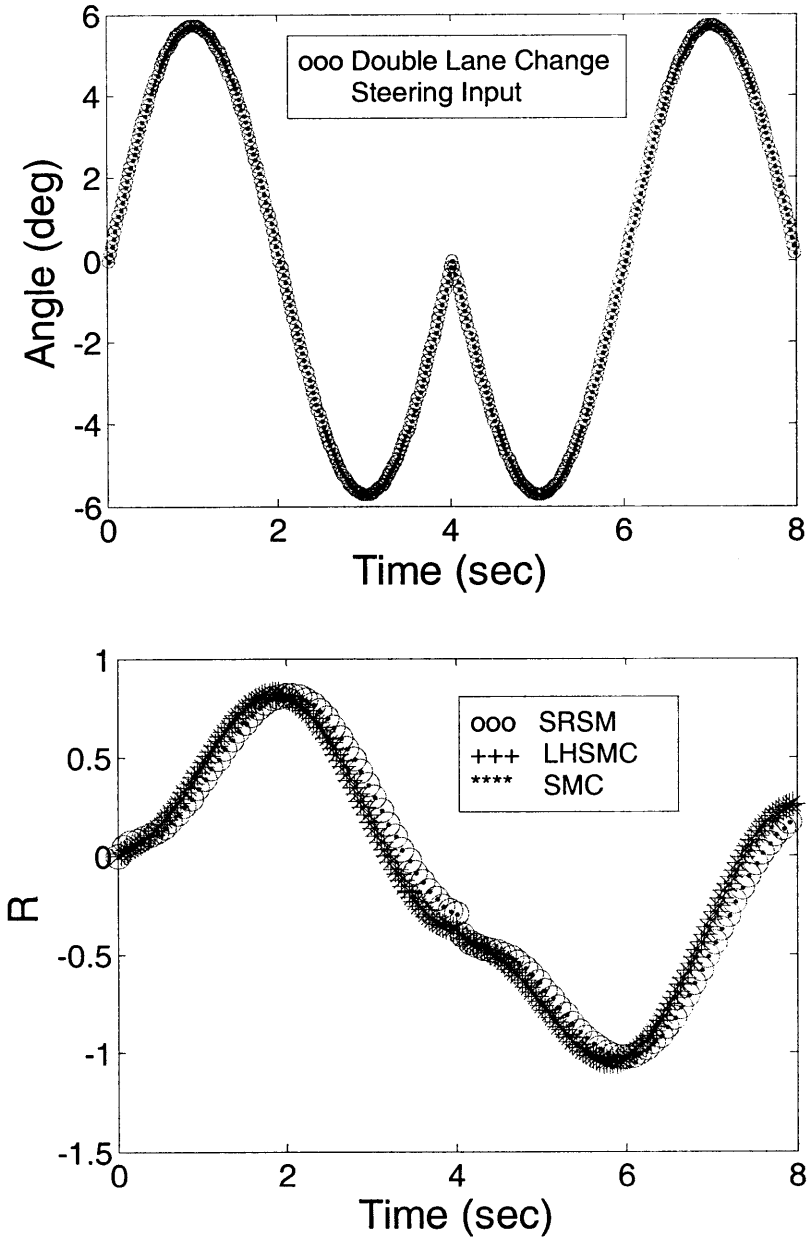

Fig. 4.7. Vehicle roll-over analysis results for double lane change steering input

Stochastic analysis also allows insight into the range of the variation of an output time series. In Figure 4.8, results are shown for the steering angle input and roll-over coefficient for a double lane change maneuver, here including uncertainty bounds on the 2σ variation. In this particular analysis, it can be observed that while the absolute value of the mean of the roll-over metric (corresponding closely to the result from a deterministic simulation) remains less

than one, the value exceeds one when prediction bounds from the stochastic analysis are also included, thus indicating a risk of vehicle roll-over when parameter uncertainty is explicitly considered.

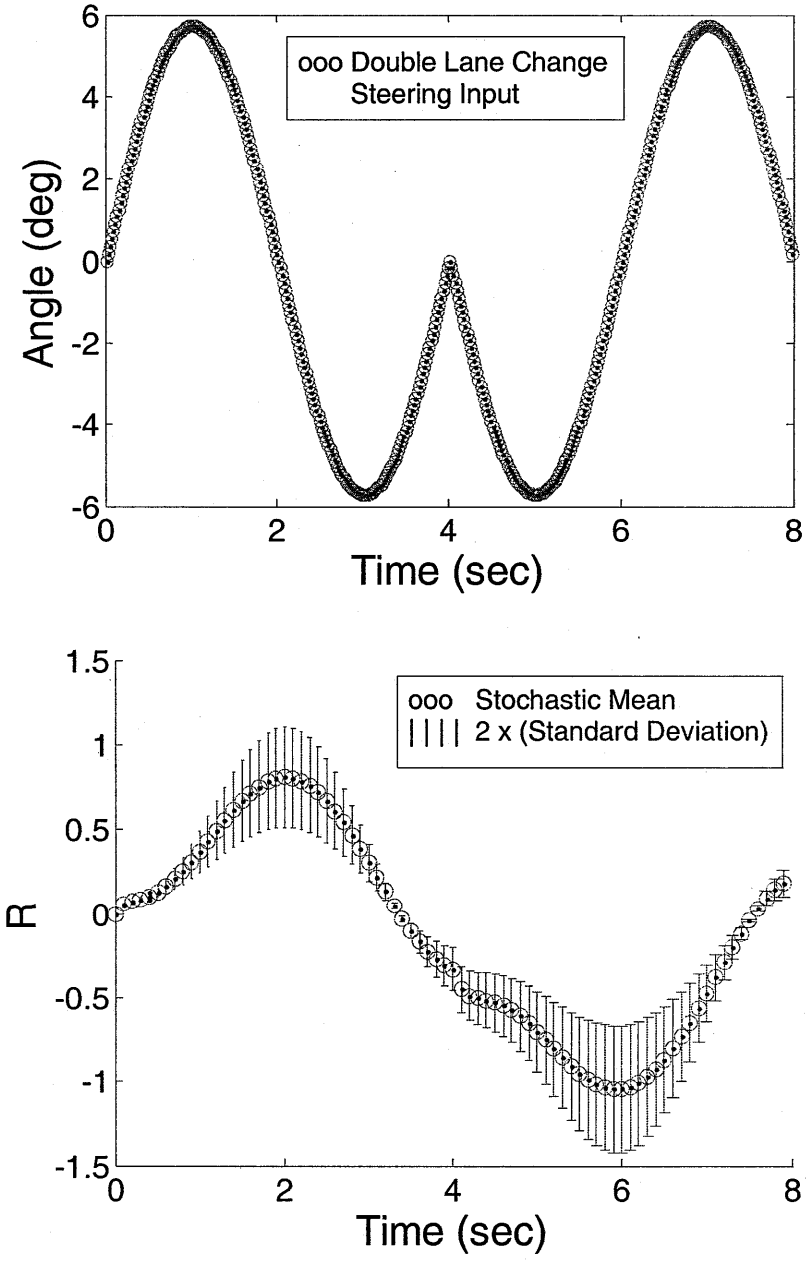

Fig. 4.8. Vehicle roll-over analysis using SRSM

Simulation times for the Monte Carlo approaches are compared to those for SRSM in Table 4.7. It can be observed that computation time for the response surface-based method is approximately two orders of magnitude lower than for the Monte Carlo-based methods.

TABLE 4.7
RATIO OF SIMULATION TIMES FOR VEHICLE ROLL-OVER ANALYSIS

STEERING INPUT	SMC (1000 RUNS)	LHSMC (400 RUNS)	SRSM (2^{ND} ORDER)
Sinusoidal	286.97	118.62	1.00
Ramp-Like	285.76	117.68	1.00
Double Lane Change	287.29	118.91	1.00

Results of simplified mobility prediction scenarios show that the proposed method represents a significant improvement over conventional Monte Carlo methods in terms of computational efficiency, while showing similar accuracy. It can therefore be used to robustly and efficiently predict the traversability of mobile robots in unstructured environments. However, for certain problems, such as those with discontinuities induced by random inputs, or for situations involving long-term integration, the technique may not give appropriate results. In such scenarios, the multi-element generalized polynomial chaos approach discussed in Chapter 3 can be utilized. This is discussed in Section 4.5.

4.5. Long-Term Predictions using MEgPC Approach

In this section, the MEgPC approach is applied to the domain of vehicle dynamic analysis and the results for long term integration are studied (as in Section 3.5.2). For this analysis, a double lane change steering maneuver is considered as an input, and the roll angle evolution under vehicle parameter uncertainty is studied using the gPC, MEgPC (P = 3, θ_1=0.05, α=0.5) and SMC approaches, for motion over uneven terrain. The terrain (see Figure 4.9) is represented using a combination of trigonometric functions as:

$$z(x,y) = A\sin\left(\frac{y}{2}\right) + B\sin\left(\frac{x}{2}\right) + C\cos\left(\frac{D}{2}\left(\sqrt{x^2+y^2}\right)\right) + E\cos\left(\frac{y}{2}\right) + F\sin\left(\frac{G}{2}\left(\sqrt{x^2+y^2}\right)\right) + H\cos\left(\frac{x}{2}\right) \quad (4.25)$$

where A,B,C,D,E,F,G,H are suitably chosen constants.

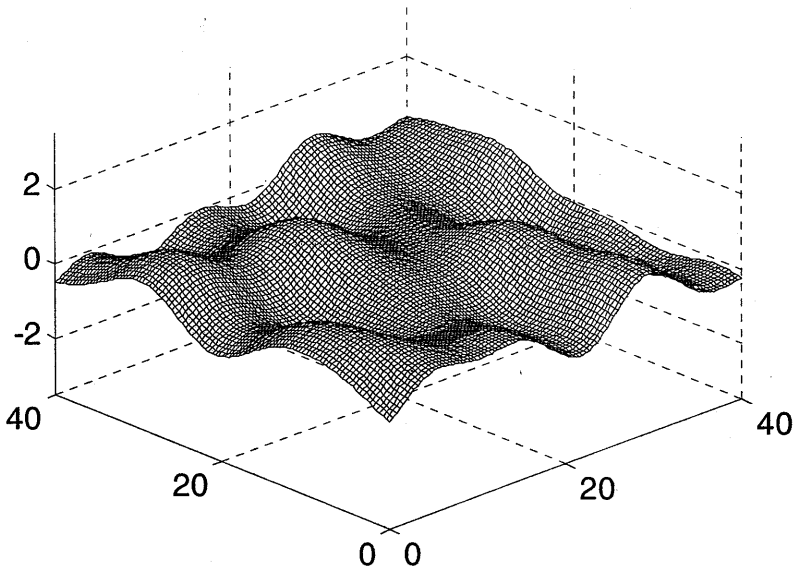

Fig. 4.9. Terrain map used in the analysis

Also, in the present analysis, roll stiffness parameters are considered to be uniformly distributed with the corresponding values shown in Table 4.8.

TABLE 4.8
UNCERTAIN VEHICLE PARAMETERS IN ROLL-OVER ANALYSIS

PARAMETER	μ (Nm/rad)	σ (Nm/rad)
k_f	30×10^3	4×10^3
k_r	30×10^3	4×10^3

A spectral stochastic analysis is performed to obtain the time evolution of the mean and standard deviation of the roll angle (see Figures 4.10-4.11). It can be seen that for the particular scenario, though the mean values match closely, the prediction from the gPC approach for the standard deviation differs substantially from the MEgPC and SMC results, even for relatively short times.

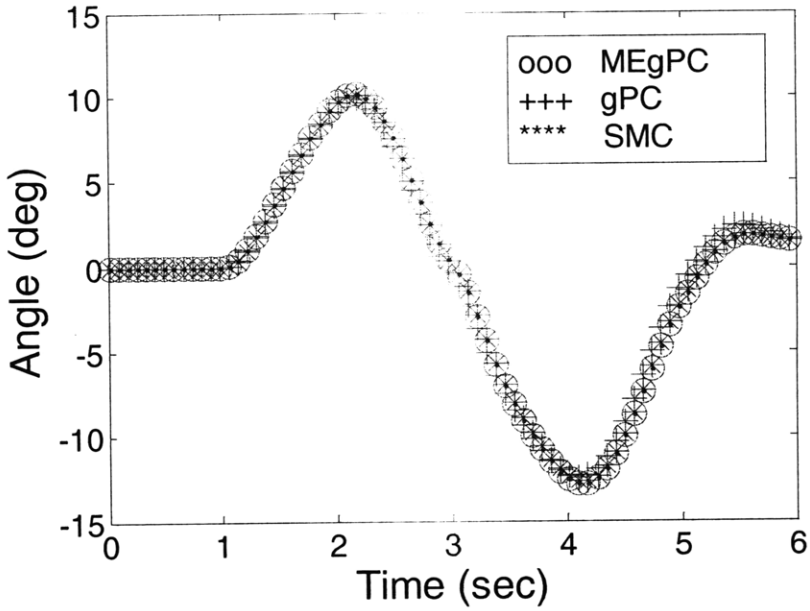

Fig. 4.10. Prediction of mean of roll angle

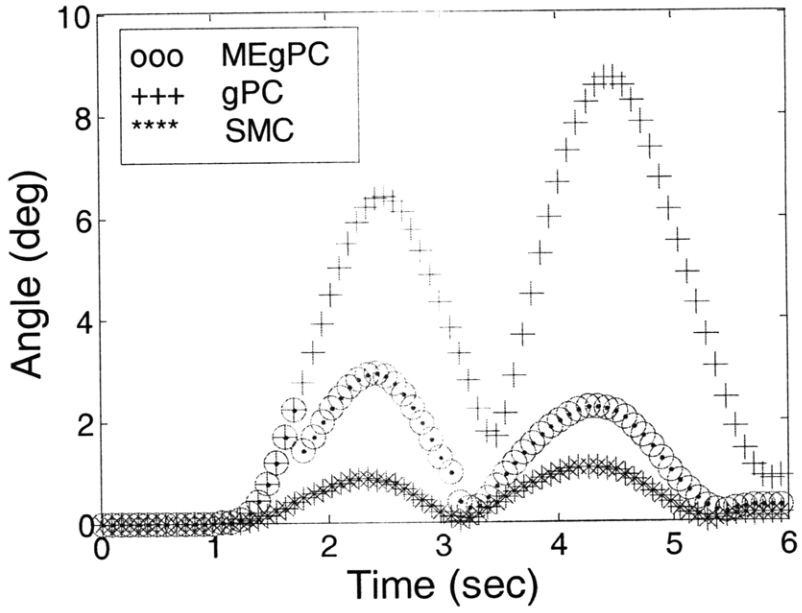

Fig. 4.11. Prediction of standard deviation of roll angle

Next, the time evolution of the standard deviation of R is studied for a sinusoidal input (see Figure 4.12). It can again be observed that there is significant difference in the predictions from the two polynomial chaos-based techniques.

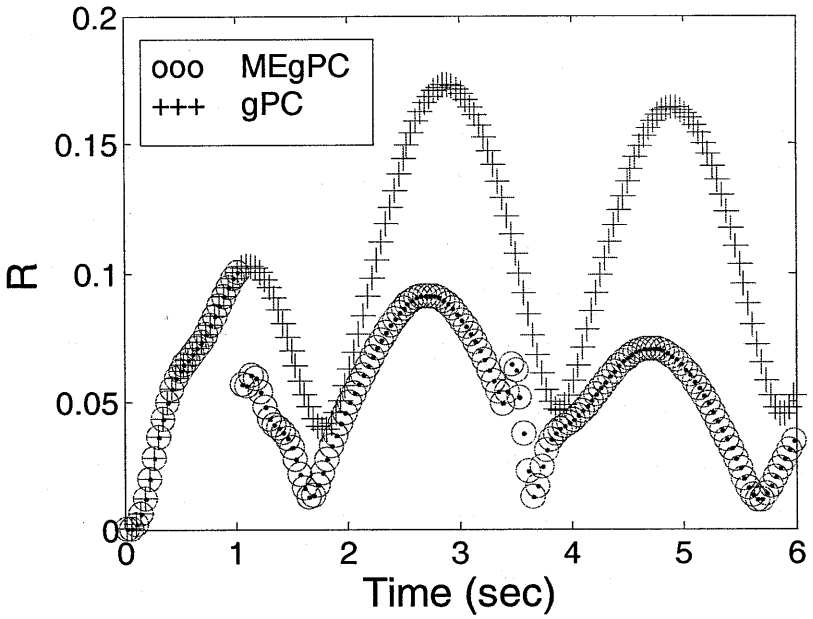

Fig. 4.12. Prediction of standard deviation of roll-over metric

Computation times for SMC and MEgPC approaches with respect to the gPC method, while predicting the roll-over coefficient R, can be seen in Table 4.9.

TABLE 4.9
COMPUTATION TIME FOR THE VARIOUS APPROACHES

METHOD	RATIO OF SIMULATION TIME
SMC (2000 runs)	197.49
MEgPC	5.37
gPC	1.00

Simulation results show that the method represents a significant improvement over the Monte Carlo technique in terms of computational cost, and over the gPC method in terms of accuracy of long-term predictions.

To summarize, the polynomial chaos-based approaches have been found to perform significantly better than conventional uncertainty analysis techniques for the mobility prediction and vehicle dynamics scenarios considered.

5. PATH PLANNING UNDER UNCERTAINTY

5.1. Overview

A fundamental requirement for autonomous ground vehicles moving on uneven, rugged terrain is the capacity to quickly and efficiently generate a feasible trajectory online that results in safe, rapid traversal while avoiding obstacles.

Substantial work in motion planning has been performed over the years, and the major techniques that have evolved include the A* and D* methods [3], potential field approaches [4], the probabilistic roadmap technique [5], and the rapidly-exploring random tree (RRT) algorithm [6]. These methods are aimed at determining suitable control inputs to move a robot from its initial position to its destination. Some methods attempt to accomplish this while obeying physics-based dynamic models and avoiding obstacles in the environment.

Recently, randomized approaches to kinodynamic motion planning [39] have proven to be an efficient tool for path generation, with RRTs proving to be a highly effective framework. In this technique, exploration and search are combined in a single method without substantial pre-computation that is often associated with a method such as the probabilistic roadmap. Further, the approach scales well for problems with high degrees of freedom and complex system dynamics, and its flexible framework simplifies the integration of uncertainty analysis techniques discussed in Chapter 3. RRTs will thus be the focus of the present analysis.

Since the introduction of RRTs, many extensions have been developed to the basic algorithm to improve its performance and better adapt to demands of specific systems [6]. 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 [40] while satisfying dynamic constraints. Another recent approach [41] explicitly models a robot's closed-loop controller in the planning methodology, thereby resulting in feasible paths. By construction, 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 vehicle and/or terrain parameters. As explained briefly in Chapter 1, most path planning techniques (including RRT-based approaches) rely on deterministic analysis that assumes precise knowledge of vehicle and terrain parameters. In field conditions, however, vehicles generally have access only to sparse and uncertain terrain parameter estimates, and vehicle parameters may be uncertain and time-varying. Failure to consider parameter uncertainty may lead to generation of unsafe trajectories and/or to the failure of the vehicle to track the generated paths, especially during high speed navigation in unstructured environments.

Recent work in this area uses a particle filter-based approach within the RRT framework, producing a distribution of vehicle states at each tree node [42] and uses a Monte Carlo-based simulation at each extension step to compute particles which are then clustered to form nodes in the tree [40]. Another study has focused on a modified RRT framework that includes a closed-loop prediction framework to reduce the effects of uncertainty on the generated paths [41]. However, for aggressive maneuvers or motion on highly deformable terrain, the presence of uncertainty may induce significant variation between the deterministic and stochastic performance prediction of the mobile robot. It is therefore critical to consider this uncertainty in the planning loop. The present work attempts to address these concerns through several extensions to the basic RRT algorithm to result in safe path generation over uncertain terrain.

5.2. RRTs – An Introduction

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 priori* known distribution (such that the tree expansion is biased towards the goal). Using biased sampling enhances the algorithmic performance; however, too strong a bias may adversely affect the random exploratory nature of the tree. Then, find the sampled point's nearest node in the current search tree based on an appropriate distance metric. Next, 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 tree.

A search tree is thus constructed which combines 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. This process is depicted in Figure 5.1 and the general steps (based on the standard RRT algorithm, described in pseudo-code in Table 5.1) are briefly outlined below. For more details, refer to [6], [39].

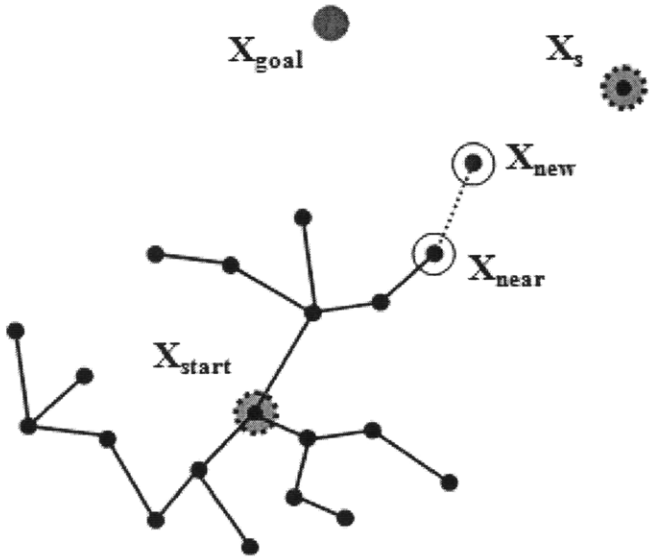

Fig. 5.1. Illustration of rapidly-exploring random tree expansion

5.2.1. Algorithmic Framework

a) Choose a target $\mathbf{X_s}$ ($\mathbf{X_s} = [x_{sample}, y_{sample}]^T$ in Cartesian space) from the domain using the function ***sample*** and determine the node in the tree $\mathbf{X_{near}}$ ($\mathbf{X_{near}} = [x_{near}, y_{near}]^T$) nearest to $\mathbf{X_s}$ using the ***nearest_node*** function. This calculation is performed using a suitable distance metric.

b) 'Grow' the nearest part of the tree towards the target $\mathbf{X_s}$ using the ***extend*** function, to reach the location $\mathbf{X_{new}}$. This node is added to the tree using the function ***add*** in case absence of collisions and if dynamic constraints are found to be satisfied using the function ***constraints***.

c) Terminate the algorithm when a node is selected that lies within some threshold distance D_o to the goal.

TABLE 5.1
BASIC RRT-BASED PLANNING ALGORITHM

```
01.  function create_tree(X_start, X_goal, E);
  [Get start location (X_start), goal location (X_goal) & environment (E).]
02.  T = initialize(X_start);
  [Initialize tree (T) using X_start.]
03.  while ~reached(X_goal, T);
  [Repeat steps below until X_goal is reached.]
04.    X_s = sample(E);
    [Choose sample node (X_s) in E.]
05.    X_near = nearest_node(X_s, T);
    [Search tree for nearest node (X_near) to X_s.]
06.    X_new = extend(X_near, X_s);
    [Move towards X_s from X_near to reach new location (X_new).]
07.    if ~constraints(X_new, T, E);
    [Check if constraints are satisfied.]
08.      T = add(X_new, T);
      [Add X_new to T if there is no collision.]
09.    end
10.  end
11.  return T;
```

A primary advantage of this framework is that it can be usually implemented for real-time, online planning, even for high degree-of-freedom dynamic models. Further, its flexibility allows trajectory-based checking of complex constraints as well as inclusion of mobility-based extensions that aid in the generation of safer paths for the vehicle to follow. Another feature is the ease with which integration of the proposed stochastic modeling approach for consideration of uncertainty can be performed within the RRT framework. These are discussed in the subsequent sections.

5.3. 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.

5.3.1. Distance Metric Calculation

Most approaches to RRT-based planning employ the Euclidean distance to calculate the distance from a node to the sample. However, many vehicles employ Ackermann (or Ackermann-like) steering, which restricts their path tracking capability to following smooth paths. Here, a distance metric similar to the Dubins path length [43] is employed for such vehicles. While Dubins curves are typically paths of the CCC/CSC sequence type (where C represents a circular arc and S refers to a straight line segment) between prescribed initial and terminal vehicle configurations, here paths of the CS/SC sequence type are considered, since the vehicle 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 vehicle heading and minimum turning radius, resulting in a more accurate estimate of the minimum path length the vehicle must travel to reach a sample from a given node (see Figure 5.2).

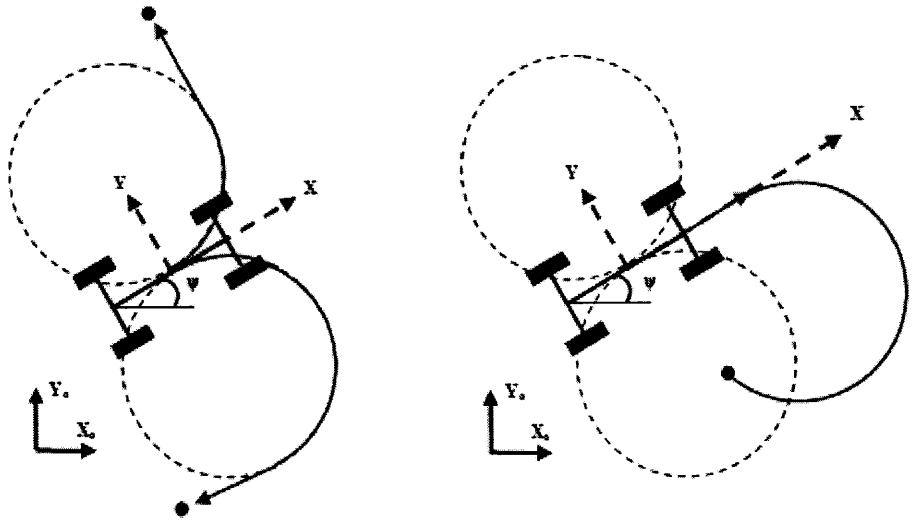

Fig. 5.2. Dubins-like paths CS (left) and SC (right) for nearest node calculations

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 vehicle at a node, the targeted sample point and the minimum turning radius of the vehicle ρ, the Dubins-like distance calculations are performed [44]. It should be noted that these calculations (based on the configuration in Figure 5.3) rely on a simple kinematic vehicle model, and thus serve as an approximation for high speed, dynamic systems.

For paths of type CS, the relations are obtained as:

$$D = \sqrt{x^2 + (y-\rho)^2} \tag{5.1}$$

$$L = \sqrt{D^2 - \rho^2} \tag{5.2}$$

$$\beta = \tan^{-1}(L/\rho) \tag{5.3}$$

$$\alpha = \tan^{-1}\{(y-\rho)/x\} \tag{5.4}$$

$$\theta = \pi/2 - (\beta - \alpha) \tag{5.5}$$

Then, $x_A = \rho \sin\theta$ and $y_A = \rho - \rho\cos\theta$

For paths of type SC, the relations can be given as:

$$\alpha = \tan^{-1}\{(y-\rho)/x\} \tag{5.6}$$

$$\varphi = \sin^{-1}\{(D\sin\alpha)/\rho\} \tag{5.7}$$

$$\gamma = \pi - (\varphi + \alpha) \tag{5.8}$$

$$L = (\rho/\sin\alpha)\sin\gamma = (D/\sin\varphi)\sin\gamma \tag{5.9}$$

Then, $x_A = L$, $y_A = 0$, and $\theta = (3/2)\pi - \varphi$

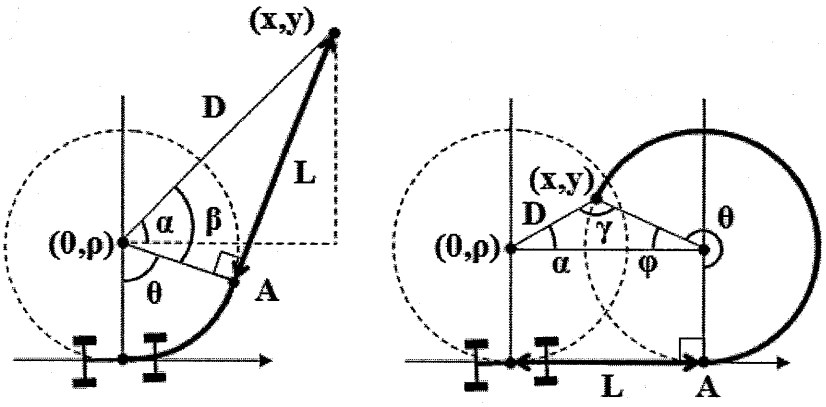

Fig. 5.3. Path length calculations for 2-D Dubins-like curves: CS (left) and SC (right)

5.3.2. 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 5.3.3). 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 roll-over metric R_o (see Section 4.2.2), averaged over the path segment (to give R_{avg_s}), 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.

5.3.3. 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 vehicle to roll-over. Here a roll-over metric (see Section 4.2.2) 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 vehicle will be generated. This node cost function is calculated as follows:

$$Q_k = \prod_{i=1}^{3} \left(C_{i,k} / \max(C_{i,j}) \right) \qquad j,k = 1...M \tag{5.10}$$

where

$$C_{1,k} = t_k \tag{5.11}$$

$$C_{2,k} = (R_{avg\_p,k} R_{max\_p,k})^h \tag{5.12}$$

$$C_{3,k} = d_k \tag{5.13}$$

Here t_k refers to the time to reach the k^{th} node from the vehicle'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 vehicle mobility, and depends on the particular application.

5.3.4. Pure Pursuit Controller

Closed-loop (rather than open-loop) model simulation is integrated in the proposed RRT framework, as in [41]. Here, a controller based on the pure pursuit algorithm [45] is employed due to its ease of implementation and widespread use. The closed-loop system is commanded to track a reference path input from the least cost node to the sample location.

The use of closed-loop control methodology (see Figure 5.4) 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 vehicle'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 thus enables generation of reasonably long and safe paths, as well as associated sequences of vehicle steering inputs.

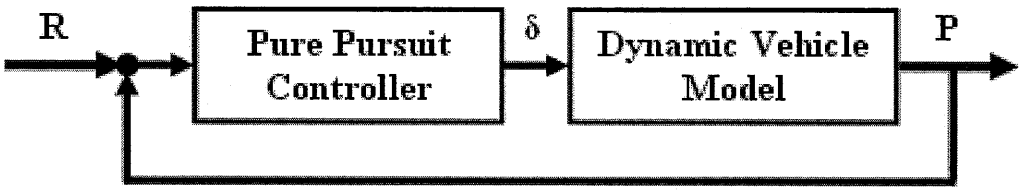

Fig. 5.4. Tracking of reference path input (R) by the controller after providing a suitable steering input (δ) to the vehicle, resulting in the traversed path (P)

The reference input to the closed-loop controller is the Dubins-like curve described in Section 5.3.1. Note, however, that only a section of the reference path might be tracked depending on the environment and the application scenario. An illustration of this approach is depicted in Fig. 5.5.

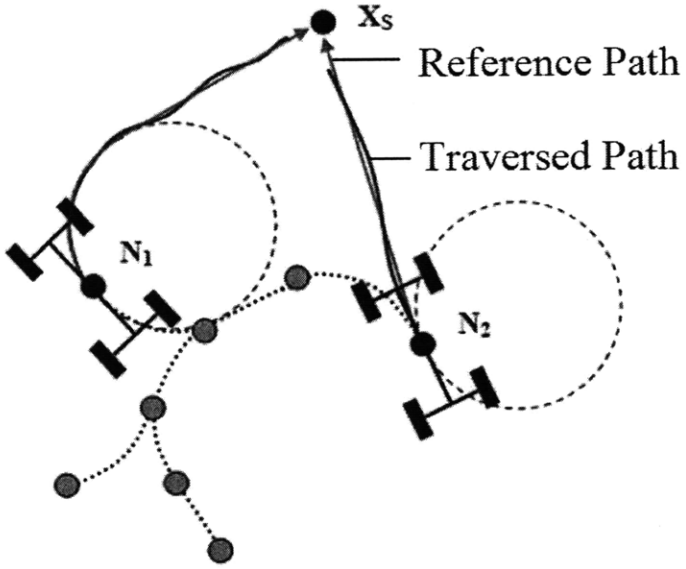

Fig. 5.5. Illustration of reference path tracking from two nearby nodes N_1 and N_2.

5.3.5. Intermediate Nodes

While long paths may be efficiently generated with the closed-loop control method, additional nodes are placed along the trajectory at short intervals. These are added to the tree if the mobility criterion (described in Section 5.3.2) 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 5.6).

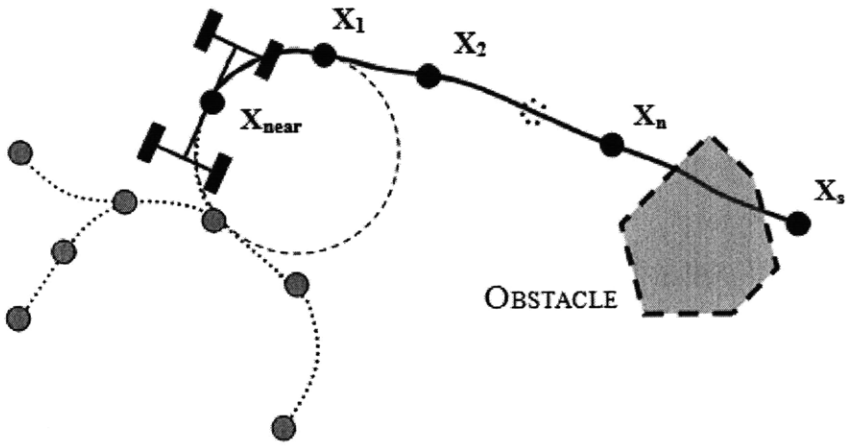

Fig 5.6. Placing of intermediate nodes along the traced path

The modified RRT algorithm with the above extensions is outlined in Table 5.2.

TABLE 5.2
MOBILITY-BASED RRT PLANNING ALGORITHM

```
01. function create_tree(X_start, X_goal, E);
    [Get start location (X_start), goal location (X_goal) & environment (E).]
02  T = initialize(X_start);
    [Initialize tree (T) using X_start.]
03. while ~reached(X_goal, T);
    [Repeat steps below until X_goal is reached.]
04.   X_s = sample_uniform(E);
      [Choose sample node (X_s) in E.]
05.   [X_near]= nearest_nodes(X_s, T);
      [Search tree for N nearest nodes [X_near] to X_s.]
06.   X_near= nearest_node([X_near]);
      [Choose nearest node (X_near) based on node costs.]
07.   [path] = create_path(X_near, X_s);
      [Create Dubins-like path to X_s from X_near.]
08.   [X_new] = extend_pure_pursuit([path]);
      [Move towards X_s from X_near to get nodes [X_new] along the path.]
09.   if ~constraints([X_new], T, E);
      [Check if constraints are satisfied.]
10.     T = add([X_new], T);
        [Add [X_new] to T if there is no collision.]
11.   end
12. end
13. return T;
```

5.4. Integration of SRSM with the RRT Framework

Parameter uncertainty, if not explicitly considered in the planning framework, can lead to uncertainty in vehicle mobility, stability, and path following characteristics. As depicted in Figure 5.7, for identical initial condition, various paths could be tracked by a closed-loop system depending on the values of uncertain vehicle and/or terrain parameters. Further, while traversing certain paths, the vehicle could collide with an obstacle, or may have a heightened possibility of roll-over (as measured with the averaged roll-over metric R_{avg_s}).

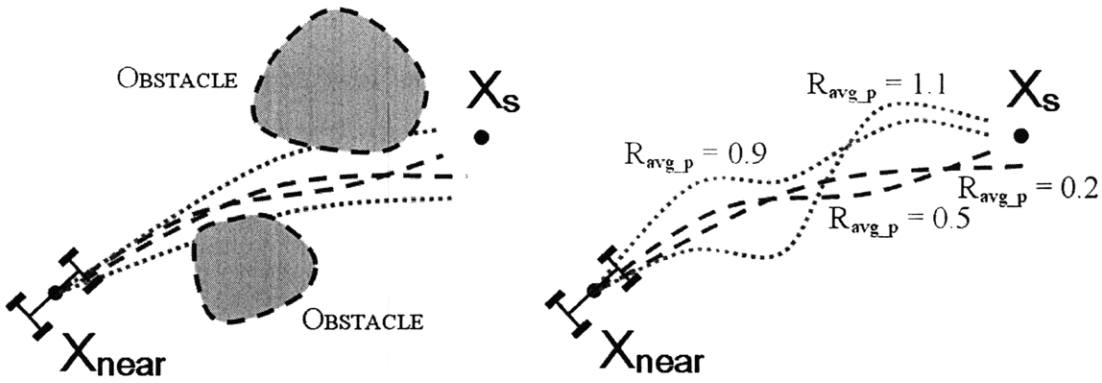

Fig. 5.7. Path and roll-over unpredictability under uncertainty

To explicitly consider uncertainty during planning, SRSM is integrated in the RRT framework. The general procedure is as follows:

Let m uncertain vehicle and/or terrain parameters, considered here to be normally distributed about their mean values, be represented using standard normal random variables ξ_k as:

$$P_k = \mu_{P_k} + \xi_k \sigma_{P_k} \quad , \quad k = 1...m \tag{5.14}$$

S state variables of interest, here including the vehicle's path coordinates, are then represented using Hermite polynomials of these standard normal random variables, as:

$$x_i(t,\xi) = \sum_{j=0}^{P} x_{i,j}(t) H_j(\xi) \quad , \quad i = 1...S \tag{5.15}$$

where $\xi = [\xi, \xi_2 \ldots \xi_m]$.

Spectral stochastic analysis [38] 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 vehicle's likely path of travel is obtained. By calculating confidence ellipses along the mean trajectory, paths that have a high probability of avoiding colliding with obstacles are chosen.

In addition to obtaining a more realistic approximation to the expected path traversed, the roll-over tendency of the vehicle along the trajectory is obtained. This can then be used in the mobility-based heuristic expansion in the RRT framework. These extensions are discussed in the following sections.

5.4.1. Confidence Ellipse Construction

The SRSM provides reduced order expansions for calculation of the vehicle's path coordinates, which are then utilized to obtain relevant statistics such as the mean and variance [11]. Based on these values, the mean path can be augmented with ellipses [46] that indicate confidence levels for the predicted position of the vehicle in the presence of uncertainty. These ellipses are then used to perform collision checks to avoid paths that are likely to collide with obstacles (see Figure 5.8). This is performed by placing the vehicle at suitable intervals along an ellipse and checking for collision with obstacles present in the environment. This approach represents an improvement over Monte Carlo methods by reducing the number of paths that must be generated to estimate the variation in path coordinates, thus reducing computational cost.

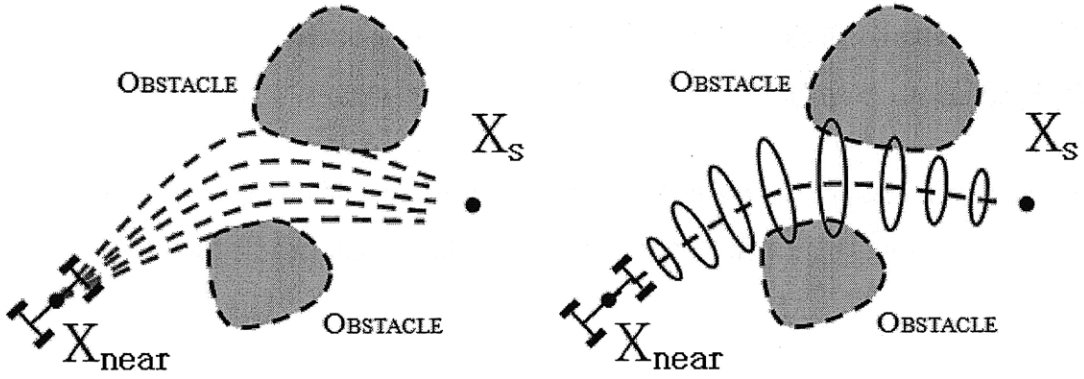

Fig. 5.8. Illustration of collision checking using confidence ellipses

Confidence ellipses centered at the mean path coordinates (see Figure 5.9) can be generated based on the following equation:

$$\frac{1}{1-r^2}\left[\frac{(\mu_x - x)^2}{s_x^2} - 2r\frac{(\mu_x - x)(\mu_y - y)}{s_x s_y} + \frac{(\mu_y - y)^2}{s_y^2}\right] = C^2 \tag{5.16}$$

where $C^2 = \left(\frac{N-1}{N}\right)\left((1-P)^{\frac{2}{2-N}} - 1\right), \mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$

μ_x and μ_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:

$$a_x = cs'_x,\ a_y = cs'_y \tag{5.17}$$

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, given as:

$$\beta = \frac{1}{2}\tan^{-1}\frac{2rs_x s_y}{s_x^2 - s_y^2} \tag{5.18}$$

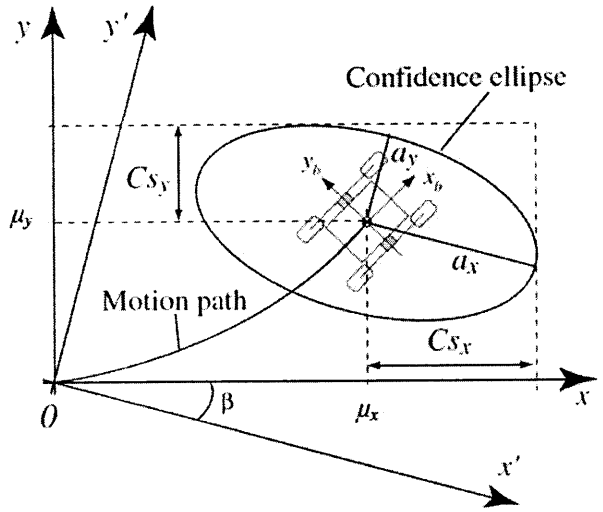

Fig. 5.9. Confidence ellipse construction

Information obtained from this analysis, such as the average variation in the vehicle 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 work.

5.4.2. Expansion Heuristic

As described in Section 5.3.3, a roll-over metric value R can be employed as a cost during tree expansion. This results in explicit consideration of vehicle mobility, albeit in a deterministic manner. To consider vehicle mobility in a stochastic manner, SRSM can be employed to yield an expected roll-over metric $E[R]$. This result can also be used during tree expansion described in Section 5.3.2. Once the expected value for the roll-over metric (E[R])

and its variance σ_R along a trajectory is obtained, R_{avg_s} can be replaced by $R'_{avg,s}$, where the latter is the path-averaged value of R_s, given as:

$$R_s = E[R] + f\sigma_R, \quad f \geq 0 \tag{5.19}$$

Thus, while extending towards a sample point from the least-cost node, $R'_{avg,s}$ is compared with the threshold R_o. Further, these stochastic values are utilized while assigning the node costs during the heuristically biased tree expansion.

5.4.3. Algorithmic Framework - Selective Implementation

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 vehicle roll-over. Here, the technique is employed when the following criterion is met:

$$|R_{avg\_s}| > R_I, \quad \text{where } R_I < R_o \tag{5.20}$$

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 roll-over risk.

Using the above extensions, an RRT algorithm that considers parameter uncertainty can be obtained that yields smooth and safe paths. The modified algorithm is outlined as a pseudo code in Table 5.3.

5.5. Simulation Studies

In this section simulation studies of the proposed method for path planning under uncertainty for the vehicle model seen in Section 4.2.1 are discussed. The scenarios chosen correspond to an obstacle laden field. First, the terrain environment is considered to be flat, and subsequently an uneven surface (represented using trigonometric functions) is considered. The use of SRSM within the framework allows incorporation of uncertainty effects (see Section 5.5.1) both while predicting mobility along the tree extensions, as well as while performing collision checks, as seen in Section 5.4. Suitable trajectory quality metrics have been defined (see Section 5.5.2) in order to compare the effectiveness of the modified

framework for the scenarios considered. The results shown in Section 5.6 indicate that the use of SRSM results in safer trajectories for the vehicle to follow.

TABLE 5.3
MODIFIED RRT-BASED PLANNING ALGORITHM

```
01. function create_tree(X_start, X_goal, E);
    [Get start location (X_start), goal location (X_goal) & environment (E).]
02  T = initialize(X_start);
    [Initialize tree (T) using X_start.]
03. while ~reached(X_goal, T);
    [Repeat steps below until X_goal is reached.]
04.   X_s = sample_uniform(E);
      [Choose sample node (X_s) in E.]
05.   [X_near]= nearest_nodes(X_s, T);
      [Search tree for N nearest nodes [X_near] to X_s.]
06.   X_near= nearest_node([X_near]);
      [Choose nearest node (X_near) based on node costs.]
07.   [path] = create_path(X_near, X_s);
      [Create Dubins-like path to X_s from X_near.]
08.   [X_new] = extend_pure_pursuit([path]);
      [Move towards X_s from X_near to get nodes [X_new] along the path.]
09.   If R_avg_s>R_I,, [X_new ,R'_avg_s]=SRSM([path]);
      [Call SRSM function if required, do collision-check using
       confidence ellipses.]
10.   if ~constraints([X_new],T,E);
      [Check if constraints are satisfied.]
11.     T = add([X_new],T);
        [Add [X_new] to T if there is no collision.]
12.   end
13. end
14. return T;
```

5.5.1. Inclusion of Uncertainty

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

$$k_f = \mu_{k_f} + \xi_1 \sigma_{k_f} \tag{5.21}$$

$$k_r = \mu_{k_r} + \xi_2 \sigma_{k_r} \tag{5.22}$$

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

$$X_i(t,\xi) = \sum_{j=0}^{P} X_{i,j}(t)\Phi_j(\xi) \tag{5.23}$$

where $\xi = [\xi_1, \xi_2]$.

The roll stiffness parameter values employed in the study are shown in Table 5.4.

TABLE 5.4
UNCERTAIN VEHICLE PARAMETERS

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

5.5.2. Description of Scenarios

Deterministic as well as stochastic analyses were performed for the environmental scenarios shown in Figures 5.10-5.11 to separately evaluate the improvements in the feasibility of path traversal 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 standard RRT algorithm was compared to a modified method that includes the mobility-based features described in Section 5.3. 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:

$$Q_{Ta} = \max\ (R_{avg\_s,i}) \tag{5.24}$$

where $R_{avg\_s,i}$ is the roll-over metric averaged along the path segment connecting the i^{th} node and its predecessor. Thus, Q_{Ta} refers to its maximum value among the nodes of the final path. The average roll-over coefficient along the final trajectory (R_{avg_p}) from the two approaches is also computed.

Uncertainty was then considered and the performance of a modified algorithm that included SRSM for each tree extension was compared to the non-SRSM case, in terms of the trajectory quality metric (Q_{Tb}), defined as:

$$Q_{Tb} = |R_{avg\_s}| \tag{5.25}$$

where R_{avg_s} is the path-average of the expected value of the roll-over metric along the trajectory under uncertainty. For the deterministic case, this metric was obtained by using a Monte Carlo (MC) analysis. Once the final path and associated steering inputs are obtained using the deterministic RRT algorithm, multiple simulations are performed corresponding to the sample parameter values from the respective uncertain distributions, while applying the steering inputs determined from the original analysis. Thus, the expected value of the roll-

over metric along the path is obtained which can then be utilized to get the required quality metric.

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 roll-over stability threshold R_I (from 5.20) 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.

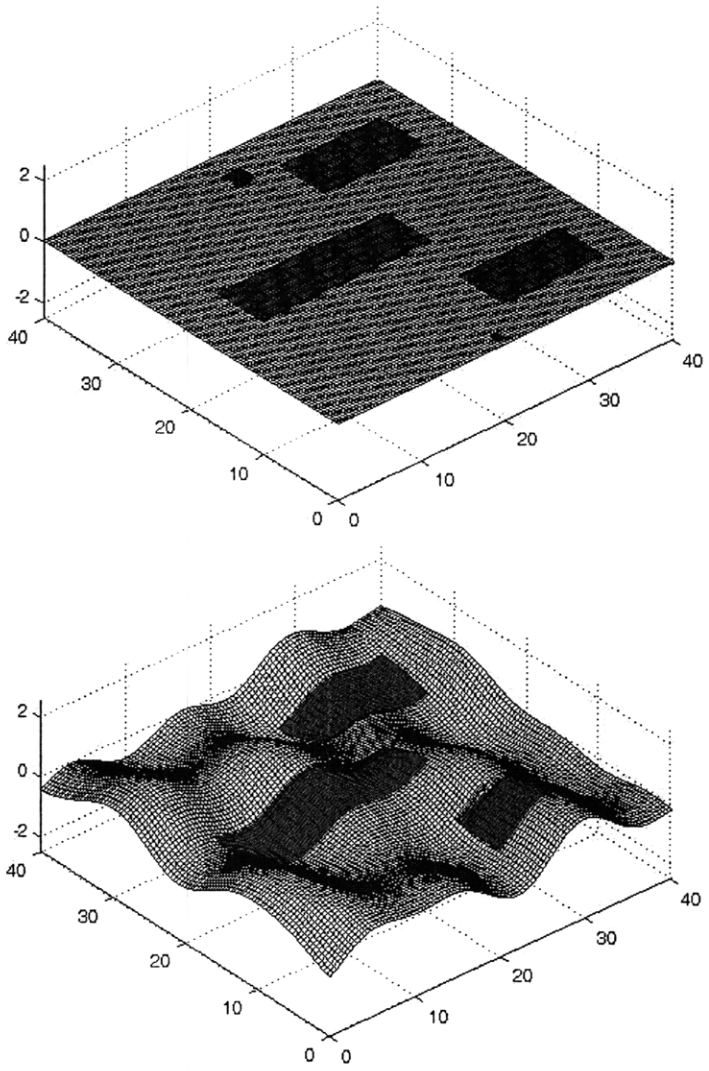

Fig. 5.10. Terrain environments considered in the analysis

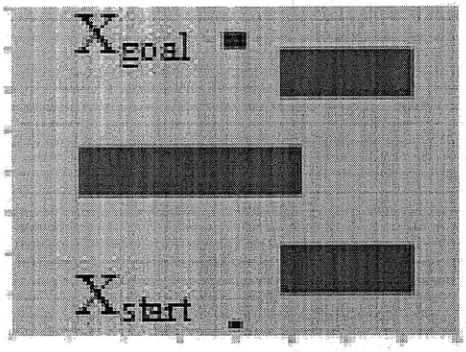

Fig. 5.11. Placement of obstacles for the scenarios (top view)

5.6. Path Planning Results

5.6.1. Deterministic Analysis

Plans were generated for the terrains of Figure 5.10. 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 5.5 and Table 5.6 respectively.

TABLE 5.5
TRAJECTORY QUALITY AND TRAVEL TIME
SCENARIO I

TECHNIQUE	h	R_O	Q_{Ta}	TRAVEL TIME, T_o (S)	R_{avg_p}
RRT (Basic)	-	-	0.457	18.26	0.372
Modified RRT (Non-SRSM)	1	0.4	0.380	17.31	0.285
	1	0.6	0.572	16.90	0.344
	1	0.8	0.769	16.51	0.470
	4	0.4	0.391	17.88	0.268
	4	0.6	0.584	17.14	0.323
	4	0.8	0.743	16.70	0.419

TABLE 5.6
TRAJECTORY QUALITY AND TRAVEL TIME
SCENARIO II

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 roll-over coefficient values. This is because the threshold value R_o limits the selection of tree extensions to those with absolute value of roll-over metric, averaged over the path segment, lower than R_o. 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 vehicle to overturn while negotiating the terrain is expected to be greater, especially at high speeds. Similarly, while T_o values may be lower, this comes at a cost to vehicle safety while negotiating the terrain. Further, reducing R_o may affect T_o depending on the nature of the terrain; however, increasing h is expected to cause a marginal increase in the travel time, when a larger sample of paths may be considered due to the additional emphasis on roll-over stability than on travel time.

The tree obtained from a typical simulation of the modified planning algorithm, is shown in Figure 5.12.

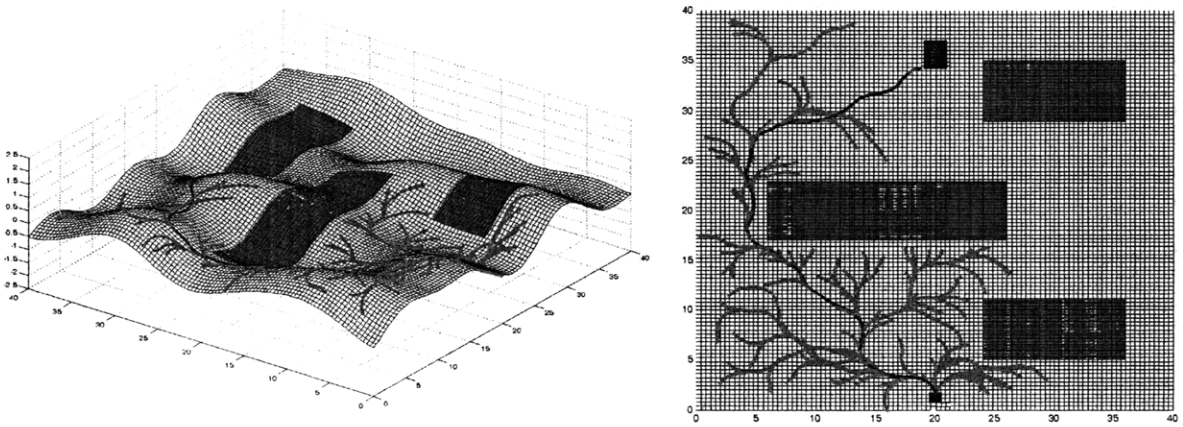

Fig. 5.12. Resulting tree and final path using the modified RRT algorithm (non-SRSM,h=1,R_o=0.6)

5.6.2. Stochastic Analysis

Further studies were conducted to consider uncertainty in vehicle parameters for varying values of R_o. As discussed in Section 5.5.2, the first case involved performing a

Monte Carlo analysis on the final path obtained through the deterministic RRT algorithm to get the expected mean value of the roll-over metric along the trajectory. The SRSM-based RRT approach was also used, while employing the response surface technique along each path segment (i.e. $R_I = 0$) to get the expected roll-over coefficient values along the tree segments, and subsequently the final path. Typical values obtained for the trajectory quality metric Q_{Tb} defined in (5.25) corresponding to the two scenarios are shown in Table 5.7.

TABLE 5.7
TRAJECTORY QUALITY FOR GENERATED PATHS

TECHNIQUE	R_O	*SCENARIO I* Q_{Tb}	*SCENARIO II* Q_{Tb}
Modified RRT (Non-SRSM, MC on final path)	0.5	0.312	0.378
	0.7	0.389	0.454
	0.9	0.497	0.584
Modified RRT (SRSM: $R_I = 0$)	0.5	0.301	0.363
	0.7	0.355	0.437
	0.9	0.456	0.560

For the first case, larger values of Q_{Tb} were observed, indicating that treatment of uncertainty is important to obtain accurate values for the expected roll-over metric along a path segment during tree expansion. Since the path is calculated through a deterministic analysis, certain segments that may be unsafe in actual, uncertain conditions can also be selected, resulting in a final path that may be infeasible for the vehicle to follow. In other words, 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 roll-over coefficient value may be significantly greater than R_o (or even 1, indicating failure). Consequently, a Monte Carlo analysis on the predicted trajectory results in a higher value for the quality metric Q_{Tb}. However, in the stochastic planning framework, paths segments that are likely to cause vehicle roll-over under uncertainty are disallowed, leading to trajectories that are safer for the vehicle to track.

The computational efficiency of SRSM was also compared to that of the Monte Carlo method in the planning framework. Typical results obtained for T/T_D are shown in Table 5.8. The computational efficiency for SRSM is significantly better than the Monte Carlo

approach, 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 5.8
TRAJECTORY QUALITY AND RELATIVE SIMULATION TIME

METHOD	R_o	*SCENARIO I*		*SCENARIO II*	
		T/T_D	Q_{Tb}	T/T_D	Q_{Tb}
Monte Carlo (400 runs) ($R_I = R_o - 0.1$)	0.5	280.2	0.309	288.6	0.369
	0.7	238.1	0.381	246.5	0.447
	0.9	120.4	0.465	127.1	0.575
SRSM (2nd order) ($R_I = R_o - 0.1$)	0.5	4.12	0.321	4.25	0.373
	0.7	4.01	0.395	4.17	0.449
	0.9	3.82	0.480	4.02	0.571

In summary, the framework for stochastic vehicle trajectory generation presented here explicitly considers vehicle mobility and parametric uncertainty. Simulation results for planning on uneven terrain show that the proposed method can generate safer paths compared to a basic RRT algorithm, and can be used to robustly and efficiently predicting safe, feasible trajectories for autonomous vehicles in unstructured, uncertain environments.

6. MOTION CONTROL UNDER UNCERTAINTY

6.1. Overview

Motion control on non-flat terrain is an important capability of mobile robots operating in outdoor environments and constitutes an important feature that must be considered for successful operation of autonomous navigation systems. However, most control schemes rely on a deterministic analysis, and do not explicitly consider parametric uncertainty while calculating the control inputs. In practical scenarios, however, the uncertainty in vehicle and/or terrain parameters can cause significant deviation from the predictions made according to a deterministic analysis. Consequently, most control algorithms can fail in controlling the motion of vehicles operating in such unstructured terrain. While stochastic control techniques have been developed recently to be applied to control of vehicle motion and dynamics, most of them either do not explicitly consider the uncertainty in the parameters or they tend to be computationally expensive and infeasible for application to online control of UGVs performing aggressive maneuvers on rugged terrain.

In the present work, model predictive control (MPC) has been used to achieve good path tracking performance due to its ability to systematically handle constraints and multi-variable systems. The technique uses a system model in a constrained optimization framework to determine control inputs that minimize a performance objective and satisfy constraints, including both dynamic equality constraints and physical inequality constraints.

A feature of model predictive control is that, due to its basis in optimal control, it operates close to the constraint boundaries. However, the presence of uncertainty can lead to violation of these boundaries, if the uncertainty is not explicitly considered in the control framework. Hence, inclusion of uncertainty is critical while computing the control inputs using MPC, especially when the state variables have values close to the constraint limits.

Extensive past research has been done in the area of predictive control and robustness to uncertainty (both parametric and exogenous), as well as in the domain of stochastic control. A traditional approach to considering uncertainty during model identification or

measurement is in the form of noise. Uncertainty-induced effects can be incorporated by defining the discrete-time systems as:

$$\mathbf{x_{k+1}} = a_k(\mathbf{x_k}, \mathbf{u_k}) + \mathbf{w_k} \tag{6.1}$$

where x_k and u_k denote the random vectors corresponding to the system state and applied control input, respectively, $a_k(.)$ is a linear or non-linear function, and w_k denotes additive white noise. While for linear systems, calculation of the probability density of the system state x_k at each time step can be performed using a Kalman filter, for non-linear systems it involves calculation of computationally demanding numerical integration. Recently, in [47], such systems have been approximated using a stochastic state prediction method wherein the transition density approximation is performed using hybrid Dirac and Gaussian densities.

Another formulation adopted in [48] solves closed-loop stochastic dynamic optimization problems, while explicitly considering stochastic properties of both exogenous and endogenous uncertainties. The non-linear predictive controller deals with model uncertainty and disturbances by replacing deterministic constraints in its formulation (represented as $y_{min}<y<y_{max}$), with chance constraints (i.e. constraints that hold true with a probability value greater than a given threshold) of the form:

$$\Pr\{y_{min}<y<y_{max}\} > \alpha \tag{6.2}$$

where α is a constant, $0 < \alpha < 1$. The main challenge here lies in the computation of the probability and its gradients. An inverse mapping approach that requires computationally expensive multivariate numerical integration is employed in [48].

Other approaches attempt to include uncertainty by performing optimization with respect to a single statistic, such as the expected value or the variance, thereby converting the stochastic problem to an optimal control deterministic optimization, by effectively removing the stochastic element of the problem. In [49], for example, optimal control of a linear, discrete time system subject to input constraints and stochastic disturbances is performed by employing a closed-loop optimization procedure. Although computationally demanding, the framework minimizes the expected value of the cost function (comprising of a performance measure and the expected value of the constraint violation cost, both expressed in terms of convex functions) and also takes disturbances into account.

In [50], explicit stochastic non-linear predictive control is introduced based on Gaussian process models. Non-parametric probabilistic black-box models directly provide uncertainty predictions, and once the probability distribution of the predicted states is obtained, the mean and variance can be incorporated within the constraint definitions. Other general stochastic techniques introduce a probabilistic formulation of the cost that includes probabilistic bounds of the predicted variable [51].

While there have been attempts to develop robust stochastic frameworks, most methods have not focused on the issue of computational efficiency. Also, most methods for ground vehicle control have not explicitly addressed uncertainty in vehicle and/or terrain parameters. The present work extends the response surface approach presented in Chapter 3 to a predictive control framework, in order to enhance the robustness of predictions while ensuring that computational costs are not excessive. An approach similar to the one adopted in [50] has been used to incorporate uncertainty effects in the control framework, while employing the response surface method to predict the state and to calculate moments to be utilized in modified constraint definitions. This is discussed in Section 6.3.

6.2. Linear Model Predictive Control (MPC) – An Introduction

Model Predictive Control has recently drawn attention from a variety of fields because of its ability to rigorously and systematically handle constraints [52]. Control inputs are obtained by repeatedly solving user defined minimization problems online as the system evolves over time. A key advantage is that these optimization-based controllers can operate close to constraint boundaries to obtain improved performance compared to most traditional approaches. As a result, however, disturbances can potentially drive systems into an infeasible region, and therefore it is important to carefully consider the effects of external disturbances, or mismatch between the model and actual process. A family of approaches, called robust MPC, exist that explicitly address this issue [53]. Stochastic MPC has also been the subject of research study for quite a few years [51].

Figure 6.1 illustrates the basic concept behind linear model predictive control. The goal of the controller is to make the actual output, **z**, as close to the reference, **r**, as possible. This is

done by calculating the optimal input, **u**, for each time-step. Feedback is often used to control the dynamic behavior of the system, and involves providing the system with the measured output, **y**, as feedback. Thus, information about the effect of the disturbance and/or uncertainty on the output is, in essence, provided back to the controller. This increases the robustness of the system by ensuring that the measured output is approximately similar to the reference value, **r**. When feed-forward is used, the controller analyzes potential future disturbances, **d**, and this allows the system to respond to these disturbances more quickly and robustly. Both feedback and feed-forward can be used simultaneously to improve the controller's robustness and reaction speed.

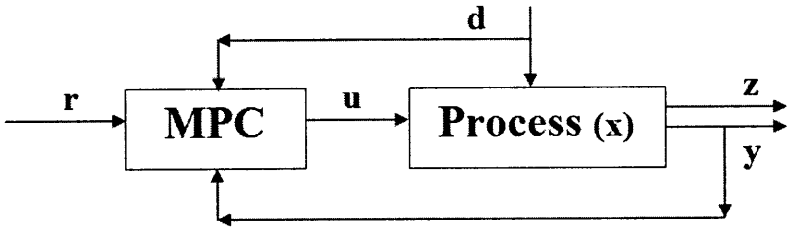

Fig. 6.1. Illustration of MPC framework

A brief formulation of the MPC framework is summarized in Section 6.2.1. More details can be found in [54], [55].

6.2.1. Algorithmic Framework

A system with linear time-invariant dynamics and discrete sampling time T_s is described by:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{E}\mathbf{d}_k \tag{6.3}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k \tag{6.4}$$

where **u**, **d**, **x** and **z** represent the inputs, disturbances, states and outputs of the system respectively.

Closed-loop system:

Performance objectives are specified through an objective function that is to be minimized, and inequality constraints on the inputs and outputs. For flexibility in specifying the objective function and constraints, the input $\mathbf{u}_k$ is defined as a cumulative sum of changes in input as:

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k \tag{6.5}$$

$$\mathbf{u}_{k+n} = \mathbf{u}_{k-1} + \sum_{j=0}^{n} \Delta\mathbf{u}_{k+j} \tag{6.6}$$

A prediction horizon is taken as N sampling intervals. The basic MPC problem is then defined that includes disturbance as well as constraints imposed on the input quantity, the input rate of change and on the output. In the deterministic analysis, it is assumed that for every time step, k, the constraint limits are the same. A quadratic objective function is then defined as:

$$\min J_k = \frac{1}{2}\sum_{k=0}^{N} \left\| \mathbf{z}_k - \mathbf{r}_k \right\|_{\mathbf{Q}_z}^2 + \frac{1}{2}\sum_{k=0}^{N-1} \left\| \Delta\mathbf{u}_k \right\|_{\mathbf{S}}^2 \tag{6.7}$$

s.t.

$$\begin{aligned}
&\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{E}\mathbf{d}_k, && k = 0,1,\ldots,N-1\\
&\mathbf{z}_k = \mathbf{C}\mathbf{x}_k, && k = 0,1,\ldots,N\\
&\mathbf{u}_{min} \le \mathbf{u}_k \le \mathbf{u}_{max}, && k = 0,1,\ldots,N-1\\
&\Delta\mathbf{u}_{min} \le \Delta\mathbf{u}_k \le \Delta\mathbf{u}_{max}, && k = 0,1,\ldots,N-1\\
&\mathbf{z}_{min} \le \mathbf{z}_k \le \mathbf{z}_{max}, && k = 1,2,\ldots,N
\end{aligned}$$

The first term in the objective function refers to the difference between the output, $\mathbf{z}_k$, and the reference, $\mathbf{r}_k$, while the second term (a regularization term) aims to reduce the difference between two consecutive steps in $\mathbf{u}$, which gives a smoother input. $\mathbf{Q}_z$ and $\mathbf{S}$ referred to in (6.7) represent the weighting matrices.

The objective function is then formulated as a QP problem. The vectors $\mathbf{z}$, $\mathbf{u}$, $\Delta\mathbf{u}$, $\mathbf{d}$ and $\mathbf{r}$ over the prediction horizon N are augmented into vectors $\mathbf{Z}$, $\mathbf{U}$, $\Delta\mathbf{U}$, $\mathbf{D}$ and $\mathbf{R}$ as:

$$\mathbf{Z}_{k+1} = \begin{bmatrix} \mathbf{z}_{k+1} \\ \mathbf{z}_{k+2} \\ \cdots \\ \mathbf{z}_{k+N} \end{bmatrix}, \mathbf{U}_k = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \cdots \\ \mathbf{u}_{k+N-1} \end{bmatrix}, \Delta\mathbf{U}_k = \begin{bmatrix} \Delta\mathbf{u}_k \\ \Delta\mathbf{u}_{k+1} \\ \cdots \\ \Delta\mathbf{u}_{k+N-1} \end{bmatrix}, \mathbf{D}_k = \begin{bmatrix} \mathbf{d}_k \\ \mathbf{d}_{k+1} \\ \cdots \\ \mathbf{d}_{k+N-1} \end{bmatrix}, \mathbf{R}_{k+1} = \begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{r}_{k+2} \\ \cdots \\ \mathbf{r}_{k+N} \end{bmatrix} \tag{6.8}$$

The remaining matrices that relate to the QP formulation are also determined as in [54]. Once the MPC problem is expressed as a quadratic program, it can be solved with conventional optimization routines.

However, the control problem can potentially be infeasible due to the presence of constraints. A common solution to this problem is to "soften" the constraints so that the boundaries may be violated when needed. This can be achieved by introducing a slack variable [54]. The soft output constraints aid in avoiding infeasibility, but can violate the physical limitations for the output. For further details of the MPC problem setup including soft constraints, refer to [54], [55].

The model predictive control framework therefore represents an effective control design methodology for handling both constraints and performance requirements. Moreover, the flexibility of the framework makes it convenient to integrate the proposed stochastic response surface approach in order to explicitly incorporate parametric uncertainty and generate robust predictions. This is discussed in Section 6.3.

6.3. Integration of SRSM with the MPC Framework

In MPC, a predictive control law is obtained by minimizing a receding horizon performance index that explicitly takes into account input and/or state constraints. However, it is well known that the action of a bounded disturbance can destabilize a predictive controller. Various algorithmic approaches have been developed to address this issue and yield more robust controllers. Similarly, uncertainty in the model and/or environment parameters can cause substantial deviations from the deterministic predictions and must to be considered in the analysis.

Parameter uncertainty, if not explicitly considered, can therefore lead to significant variation in vehicle mobility and stability, as well as path tracking characteristics. For example, while deterministic predictions may generate control inputs such that constraint limits are not violated (or marginally violated, due to the inclusion of soft constraints), the presence of uncertainty might cause constraint limits to be violated during actual system operation. Therefore, uncertainty should be considered in the MPC framework prior to determining the control inputs.

A simple approach to employing SRSM is to explicitly consider uncertainty pertaining to vehicle model and/or terrain parameters, and determining the stochastic mean and variance values for the state variables. In the presence of uncertainty, these serve as an appropriate means to quantify the variation in the state variables' values and can be utilized within the MPC framework to generate more robust predictions, while ensuring that computational burden on the controller during online implementation is not heavy. The general procedure is as follows:

Let m uncertain parameters, considered here to be normally distributed about their mean values, be represented using standard normal random variables ξ_q, as:

$$P_q = \mu_{P_q} + \xi_q \sigma_{P_q} \qquad q = 1...m \tag{6.9}$$

S state variables of interest (including the vehicle's path coordinates) are then represented using Hermite polynomials of these standard normal random variables, as:

$$x_i(t,\xi) = \sum_{j=0}^{P} x_{i,j}(t) H_j(\xi) \quad , \quad i = 1...S \tag{6.10}$$

where $\xi = [\xi, \xi_2 \ldots \xi_m]$.

Using these expansions, a spectral stochastic analysis is performed at each time step k, based on the control input determined by the controller in the previous step, to obtain the moment values of the state variables computed over the prediction horizon. The mean and variance values for a specific state variable can then be used to check whether any major constraint violation would occur in the presence of uncertainty. This condition may be formalized as:

$$\mathbf{z_{min}} \geq \boldsymbol{\mu}_{\mathbf{z_k}} - \lambda_1 \boldsymbol{\sigma}_{\mathbf{z_k}} \quad \textbf{or} \quad \boldsymbol{\mu}_{\mathbf{z_k}} + \lambda_1 \boldsymbol{\sigma}_{\mathbf{z_k}} \geq \mathbf{z_{max}} \tag{6.11}$$

where λ_1 is a user defined scalar, non-negative parameter.

In case this criterion is satisfied, the controller tightens the constraint based on the corresponding value of the standard deviation, by modifying them to accommodate the uncertainty induced variations in state variable values (e.g. z_{min} can be increased to $z_{\min} + \sigma_{z_k}$ while z_{max} can be reduced to $z_{\max} - \sigma_{z_k}$), and then recalculates the control input that would prevent a similar violation of the constraint. This in essence takes into account the

deviations that can arise due to uncertainty and leads to more robust predictions. Another feature of this approach is that a more accurate approximation to the expected state variable values of the vehicle, including those of the path coordinates, is obtained over the prediction horizon. The procedure is discussed in more detail in Section 6.3.1 and the modified algorithm is outlined as a flow chart in Figure 6.2.

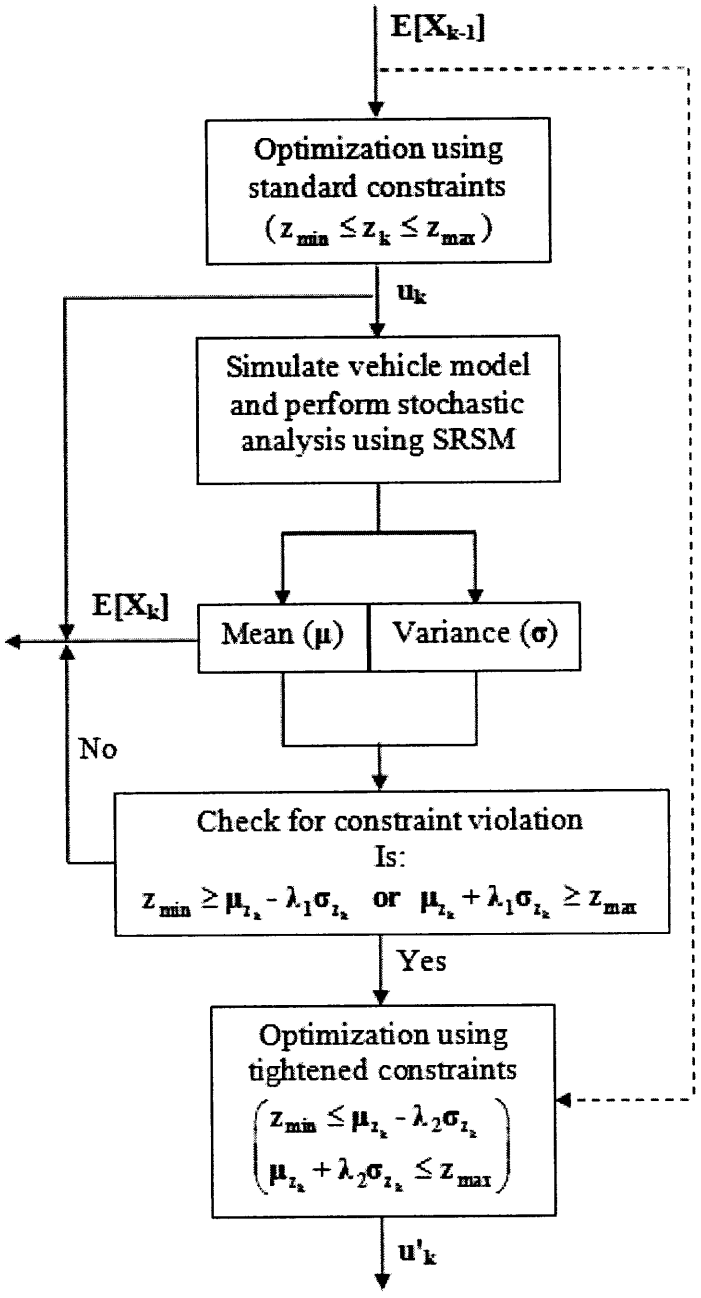

Fig. 6.2. Flow diagram showing structure of SRSM-MPC algorithm

6.3.1. Algorithmic Framework

The predictive controller uses an optimization algorithm to determine the control input at time step k while minimizing the objective function and satisfying the various constraints (as far as possible). However, it is possible that due to the presence of uncertainty, the resulting variance in the value of the output in the subsequent steps may lead to a violation of the constraint limits. Therefore, in order to incorporate these effects arising due to uncertainty, the control input obtained for time step k from a deterministic approach is used in the dynamic simulations and the variance values are obtained from a stochastic analysis. These are then used to evaluate the extent of the uncertainty induced effects on the values of the state variables. More specifically, uncertainty effects need to be considered if the condition in (6.11) is satisfied.

Hence, if a constraint is likely to be substantially violated, the control input needs to be determined again, while incorporating the variation in the predicted value for the corresponding state variable. This can be achieved by performing the optimization again, but modifying the constraints to include the variance, thereby obtaining a new control input value. The MPC problem can be rewritten as:

$$\min J_k = \frac{1}{2}\sum_{k=0}^{N}\left\|\boldsymbol{\mu}_{z_k} - \mathbf{r}_k\right\|_{Q_z}^2 + \frac{1}{2}\sum_{k=0}^{N-1}\left\|\Delta \mathbf{u}_k\right\|_S^2 \tag{6.12}$$

s.t.

$$\begin{aligned}
&\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{E}\mathbf{d}_k, && k = 0,1,\ldots,N-1\\
&\mathbf{z}_k = \mathbf{C}\mathbf{x}_k, && k = 0,1,\ldots,N\\
&\mathbf{u}_{min} \le \mathbf{u}_k \le \mathbf{u}_{max}, && k = 0,1,\ldots,N-1\\
&\Delta\mathbf{u}_{min} \le \Delta\mathbf{u}_k \le \Delta\mathbf{u}_{max}, && k = 0,1,\ldots,N-1\\
&\mathbf{z}_{min} \le \boldsymbol{\mu}_{z_k} - \lambda_2\boldsymbol{\sigma}_{z_k}, && k = 1,2,\ldots,N\\
&\boldsymbol{\mu}_{z_k} + \lambda_2\boldsymbol{\sigma}_{z_k} \le \mathbf{z}_{max}, && k = 1,2,\ldots,N
\end{aligned}$$

where the constraints are 'tightened' by an amount $\lambda_2\sigma_{z_k}$. Here, λ_2 is a scalar, non-negative parameter chosen based on the criticality of the operation and of constraint violation, and σ_{z_k} is the variance computed from the spectral stochastic analysis performed using the response surface expansions in (6.10). Using this approach therefore results in enhancing the

robustness of the predictions and more efficiently controlling vehicle motion along a trajectory. Further, note that the mean expected value μ_{z_k} obtained from the response surface-based stochastic analysis is used in the objective function (which may be done so for each time step, even if (6.11) is not satisfied).

6.4. Path Tracking Scenarios: SRSM-MPC

In this section, the problem of vehicle path tracking through an uncertain environment with sloped terrain is considered, for the model specified in Section 4.2.1. Details are provided about the incorporation of vehicle and/or terrain parameter uncertainty, and path tracking control is performed for a sinusoidal maneuver.

6.4.1. Inclusion of Uncertainty

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

$$k_f = \mu_{k_f} + \xi_1 \sigma_{k_f} \tag{6.13}$$

$$k_r = \mu_{k_r} + \xi_2 \sigma_{k_r} \tag{6.14}$$

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

$$X_i(t,\xi) = \sum_{j=0}^{P} X_{i,j}(t)\Phi_j(\xi) \tag{6.15}$$

where $\xi = [\xi_1, \xi_2]$.

The roll stiffness parameter values employed in the study are shown in Table 6.1.

TABLE 6.1
UNCERTAIN VEHICLE PARAMETERS

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

6.4.2. Description of Scenarios

Path tracking control is performed using the SRSM-MPC framework described in Section 6.3, for a sinusoidal reference path. It is assumed that the magnitude and rate of steering inputs is constrained as:

$$\delta_{\min} \le \delta_k \le \delta_{\max} \tag{6.16}$$

$$\dot{\delta}_{\min} T_s \le \Delta\delta_k \le \dot{\delta}_{\max} T_s \tag{6.17}$$

where δ_{min} and δ_{max} are the minimum and maximum values for the steering angle respectively, while $\dot{\delta}_{\min}$ and $\dot{\delta}_{\max}$ are correspondingly the minimum and maximum steering rate values.

Further, in order to ensure vehicle roll-over stability, constraints are also imposed on the body roll angle as:

$$\varphi_{\min} \le \varphi_k \le \varphi_{\max} \tag{6.18}$$

where φ_{min} and φ_{max} are the minimum and maximum acceptable roll angles, respectively. Other performance specifications (e.g. limits on sideslip angle) can easily be included in the framework, but have not been considered here. The values used in the present study are shown in Table 6.2.

TABLE 6.2
CONSTRAINT LIMITS

PARAMETER	VALUE (DEGREES)
δ_{min}	- 20
δ_{max}	20
$\Delta\delta_{min}$	- 3.75
$\Delta\delta_{max}$	3.75
φ_{min}	- 6
φ_{max}	6

Path tracking is performed for a sinusoidal maneuver, the reference path being tracked using the modified MPC framework (with soft constraints), while considering variation in the roll angle for the 'constraint tightening' approach. The results obtained are discussed in Section 6.5.

6.5. Path Tracking Results

Path tracking for a sinusoidal reference path is first studied, for a deterministic case. Figure 6.3 shows the path tracking performance of the closed-loop system, and the steering inputs are depicted in Figure 6.4. The time evolution of the slip angle and the roll angle are shown in Figures 6.5-6.6. It is observed that though there is a close resemblance between the reference and the tracked paths, the roll angle crosses the constraint boundaries at certain instants of time due to the presence of the soft constraints.

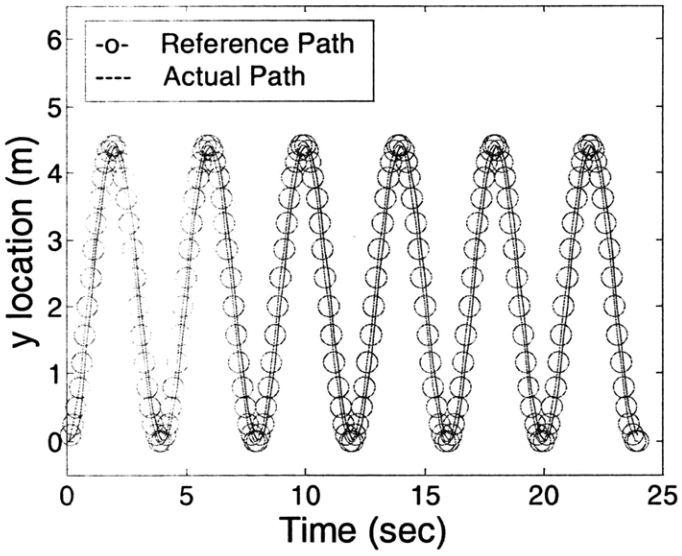

Fig. 6.3. Sinusoidal reference path tracking using MPC framework

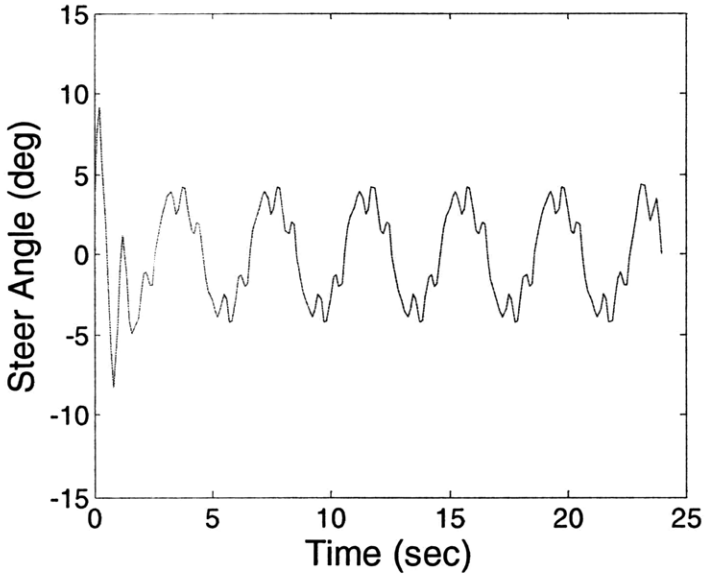

Fig. 6.4. Steering inputs for sinusoidal reference path tracking

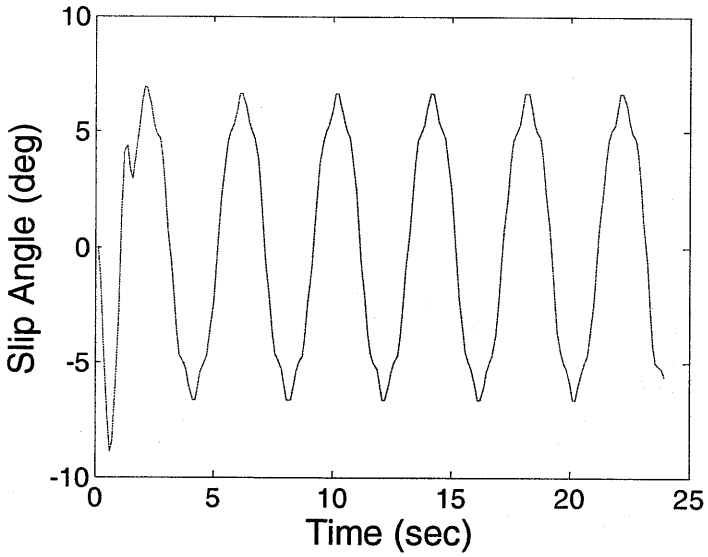

Fig. 6.5. Slip angle variation with time for sinusoidal reference path tracking

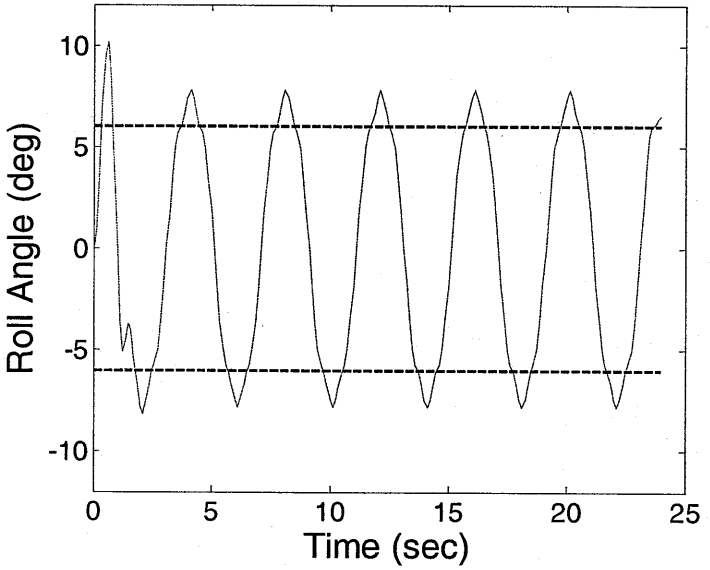

Fig. 6.6. Roll angle variation with time for sinusoidal reference path tracking

Uncertainty is now incorporated in the analysis and the evolution of the mean and variance values for the roll angle with time is analyzed, while employing the steering inputs calculated using the deterministic predictive control approach. For this, uncertainty in roll stiffness parameters is considered and Monte Carlo simulations are performed at each instant to observe the expected mean and variance values for the roll angle. The results shown in Figures 6.7-6.8 highlight the effects of inclusion of uncertainty within the MPC framework.

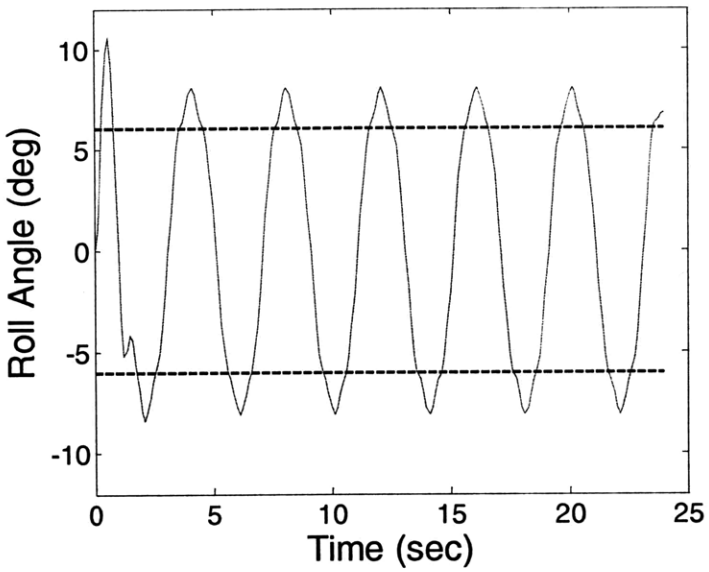

Fig. 6.7. Mean roll angle variation with time for sinusoidal reference path tracking

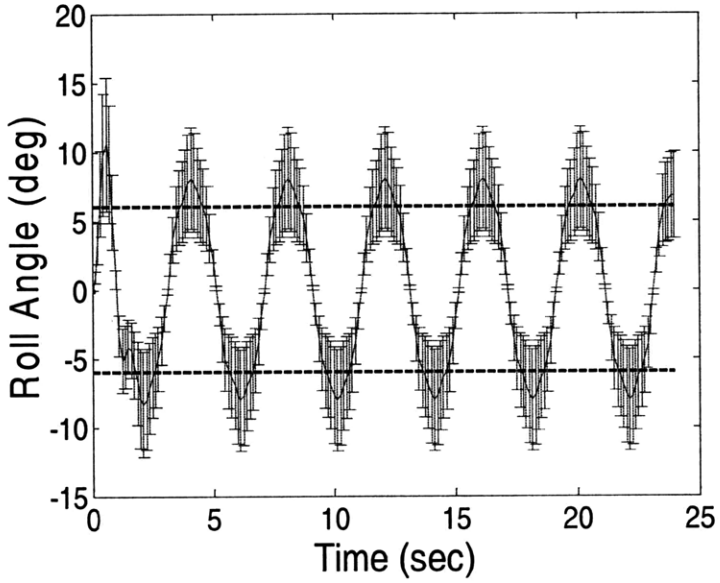

Fig. 6.8. Time evolution of roll angle variance for sinusoidal reference path tracking

It is observed that when variance in roll angle is also included in the results (as depicted by unit standard deviation error bars), the constraint boundaries may be significantly violated in realistic, uncertain situations thereby indicating the importance of including uncertainty in the analysis. It should also be noted that using Monte Carlo runs at each instant of time is inappropriate due to the heavy computational burden involved. Therefore, the SRSM-based MPC is considered for the path tracking problem. As mentioned in Section 6.3, the

constraints are reset at each step in case the condition in (6.9) is satisfied. This 'constraint tightening' approach for the scenario considered is shown in Figure 6.9 and the time evolution for the mean roll angle is depicted in Figure 6.10.

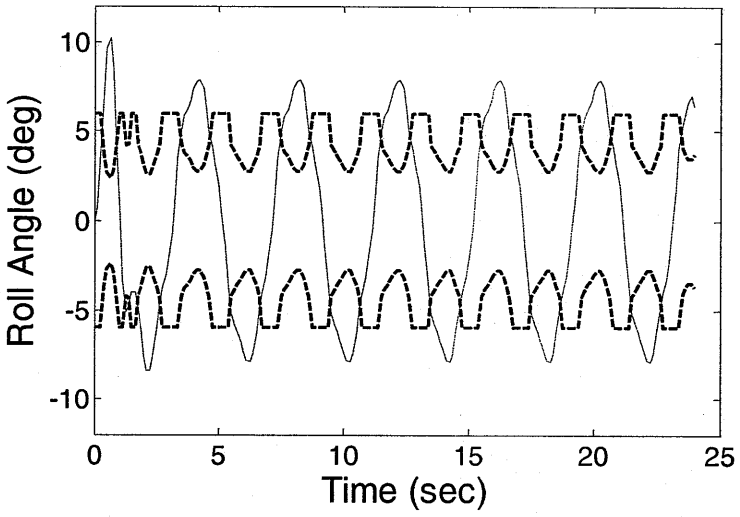

Fig. 6.9. Constraint tightening with time for the roll angle for sinusoidal reference path tracking

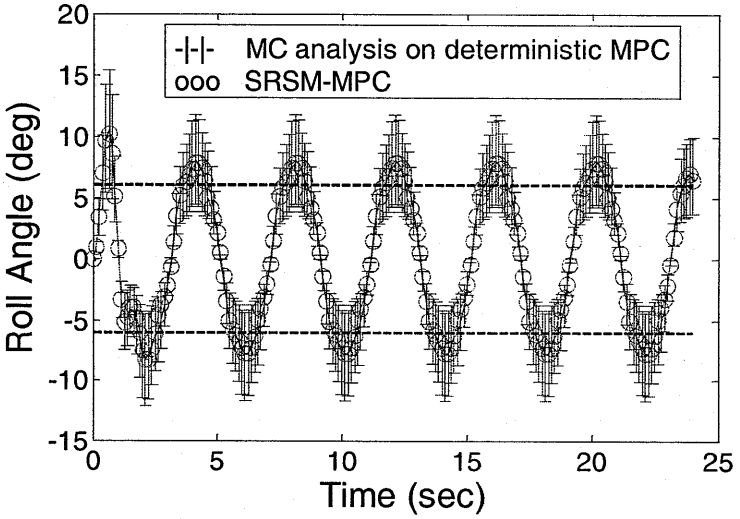

Fig. 6.10. Mean roll angle variation with time for sinusoidal reference path tracking

Results show that the controller tries to accommodate the uncertainty related effects by tightening the constraints, though at certain instants, it is becomes impossible to stay within the boundaries, and due to the presence of soft constraints, the roll angle violates the limits. This influences the path tracking behavior (see Figure 6.11) as well; however, the variance in the vehicle path coordinates is expected to be within the prescribed limits. This crossing of

the constraint boundaries due to the presence of soft constraints, which was also observed for the deterministic case in Figure 6.6, may be avoided by tightening the constraints further or by employing a higher penalty on the soft constraint violation. Further, while the focus here has been on vehicle stability and roll-over under uncertain circumstances, the path tracking aspect might as well be improved further by including a similar constraint tightening criteria for the path coordinates, and/or by choosing appropriate weighting matrices.

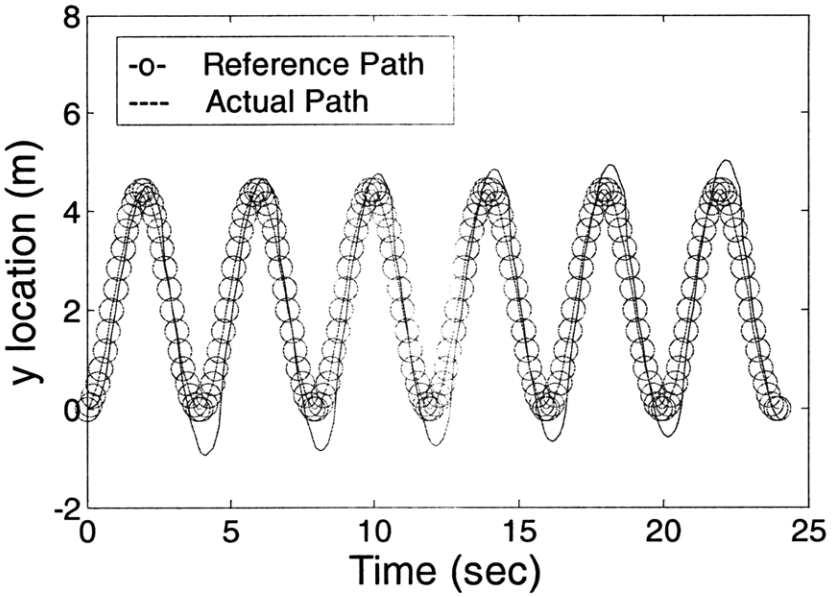

Fig. 6.11. Sinusoidal reference path tracking using SRSM-MPC framework

To summarize, the framework for stochastic vehicle motion control presented here explicitly considers parametric uncertainty, and simulation results for path tracking on uneven terrain show that the proposed method can follow through paths in unstructured, uncertain environments, while ensuring vehicle safety and avoiding roll-over, and can be used for robustly and efficiently controlling vehicle motion for autonomous vehicles on rugged terrain.

7. CONCLUSIONS

7.1. Conclusions

For unmanned ground vehicles to operate successfully in unstructured environments, the critical requirements include the ability to assess terrain traversability, compute feasible trajectories online and control vehicle motion. However, most algorithms employed to perform these functions do not explicitly consider vehicle and/or terrain parameter uncertainty during analysis. In practical scenarios, significant uncertainty is associated with these parameters and this must be included in order to obtain realistic and more robust predictions.

In this thesis, some of the prominent uncertainty analysis techniques have been applied to the areas of vehicle dynamic analysis and navigation and a comparison has been performed between the various approaches with regard to computational efficiency and accuracy. In Chapter 2, the conventional uncertainty analysis techniques, particularly the Monte Carlo methods were discussed. Although used frequently in various applications, these methods (typically) involve a large number of simulation runs and are generally associated with high computational costs. More recent approaches to stochastic simulation based on the polynomial chaos framework, described in Chapter 3, have been found to be relatively more computationally efficient. Since the framework involves representing the outputs of the system under consideration via series approximations, it requires less number of model simulations as compared to some of the conventional approaches, thereby resulting in a less computationally expensive means for uncertainty propagation through complex models.

The polynomial chaos-based techniques were applied to the area of mobility prediction, path planning and motion control in Chapters 4, 5 and 6 respectively, while explicitly considering uncertainty in vehicle and/or terrain parameters. Simulation results show that while the accuracy of results obtained is similar to the standard Monte Carlo methods, the computational costs are significantly less. The framework therefore can be effectively

applied to the areas of mobility analysis, path planning and motion control, enabling in the successful deployment of unmanned ground vehicles in unstructured, uneven terrain regions.

APPENDIX A1

Singular Value Decomposition (SVD)

Any real $m \times n$ matrix **A** can be decomposed uniquely as:

$$\mathbf{A} = \mathbf{R}\,\mathbf{D}\,\mathbf{S}^{\mathbf{T}} \tag{A.1}$$

where

R is $m \times n$ and orthogonal (its columns are eigenvectors of $\mathbf{A}\mathbf{A}^{\mathbf{T}}$)

S is $n \times n$ and orthogonal (its columns are eigenvectors of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$)

D is $n \times n$ diagonal (non-negative real values called singular values)

$$\mathbf{D} = diag(\sigma_1, \sigma_2, ..., \sigma_n) \text{ ordered so that } \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$$

Now consider the over-determined system of linear equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \text{ (where } \mathbf{A} \text{ is } m \times n \text{ with } m > n) \tag{A.2}$$

The least squares solution x with the smallest norm ||x|| is unique and it is given by:

$$\mathbf{A}^{\mathbf{T}}\mathbf{x} = \mathbf{A}^{\mathbf{T}}\mathbf{b} \text{ or } \mathbf{x} = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{b} = \mathbf{A}^{+}\mathbf{b} \tag{A.3}$$

If $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ is ill-conditioned or singular, SVD can be used to obtain a least squares solution as follows:

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{b} \approx \mathbf{S}\mathbf{D}_{\mathbf{0}}^{-1}\mathbf{R}^{\mathbf{T}}\mathbf{b} \tag{A.4}$$

where

$$\mathbf{D}_{\mathbf{0}}^{-1} = \begin{cases} 1/\sigma_i & \textbf{if } \sigma_i > t \\ \mathbf{0} & \textbf{otherwise} \end{cases}$$

where t is a small threshold value.

LIST OF REFERENCES

[1] Jurkat, M., Nuttall, C., and Haley, P., *The AMC '74 Mobility Model,* Technical Report 11921, US Army Tank Automotive Command, Warren, MI, 1975

[2] Ahlvin, R.B. and Haley, P.W., *NATO Reference Mobility Model Edition II, NRMM User's Guide,* Technical Report GL-92-19, U.S. Army WES, Vicksburg, MS, 1992.

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

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

[5] Kavraki L. E., Svestka P., Latombe J. C., and Overmars M. H., "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.

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

[7] Golda, D., Iagnemma, K., and Dubowsky, S., "Probabilistic Modeling and Analysis of High-Speed Rough-Terrain Mobile Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004

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

[9] Hanss, M., "The transformation method for the simulation and analysis of systems with uncertain parameters" *Fuzzy Sets and Systems*, v.130 n.3, p.277-289.

[10] Spanos, P.D., and Mignolet, M.P., "ARMA Monte Carlo Simulation in Probabilistic Structural Analysis," *The Shock and Vibration Digest*, 21, 3-14, 1989.

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

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

[13] Kalos, M. H., Whitlock, P. A., *Monte Carlo methods*. Vol. 1: Basics, Wiley-Interscience, New York, NY, 1986.

[14] McKay, M. D., "Latin hypercube sampling as a tool in uncertainty analysis of computer models" *Proceedings of the 24th conference on Winter simulation*, p.557-564, 1992, Arlington, Virginia, United States.

[15] Helton, J. C., Davis, F. J., "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems" *Reliability Engineering and System Safety* 81 (1), 23-69, 2003

[16] Helton, J. C., Johnson, J. D., Sallaberry, C. J., Storlie, C. B., "Survey of sampling-based methods for uncertainty and sensitivity analysis", *Reliability Engineering and Systems Safety*, 91, 1175-1209, 2006.

[17] McKay, M. D., Beckman, R. J., Conover, W. J., "A comparison of three methods for selecting values of of input variables in the analysis of output from a computer code", *Technometrics,* 21, 239-245, 1979.

[18] Vandapel, N., Huber, D.F., Kapuria, A., Hebert, M. (2004). "Natural Terrain Classification using 3-D Ladar Data", *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 5, 5117- 5122.

[19] Manduchi, R., Castano, A., Thalukder, A., and Matthies, L. (2005, May). "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, 18, 81-102.

[20] Halatci, I., Brooks, C., and Iagnemma, K., "Terrain Classification and Classifier Fusion for Planetary Exploration Rovers," *Proceedings of the IEEE Aerospace Conference*, 2007

[21] Xiu, D., Karniadakis, G. E., The Wiener--Askey Polynomial Chaos for Stochastic Differential Equations, *SIAM Journal on Scientific Computing*, v.24 n.2, p.619-644, 2002

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

[23] Wan, X., Karniadakis, G. E., An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, *Journal of Computational Physics*, v.209 n.2, p.617-642, 2005

[24] Li, L., and Sandu, C., "On the impact of cargo weight, vehicle parameters, and terrain characteristics on the prediction of traction for off-road vehicles," *Journal of Terramechanics*, in press

[25] Huang, S., Mahadevan, S., Rebba, R., "Collocation-based stochastic finite element analysis for random field problems," *Probabilistic Engineering Mechanics*, 22, 194-205, 2007.

[26] Isukapalli, S.S., Balakrishnan, S., Georgopoulos, P.G., "Computationally efficient uncertainty propagation and reduction using the stochastic response surface method", *IEEE Conference on Decision and Control*, p. 2237-2243, 2004.

[27] Blanchard, E., Sandu, A., Sandu, C., *Parameter Estimation for Mechanical Systems Using an Extended Kalman Filter,* Technical Report TR-08-18 (2008), Computer Science, Virginia Tech.

[28] Atkinson, K. E., *An Introduction to Numerical Analysis*, Second Edition, John Wiley & Sons, New York, 1988

[29] Rula, A, and Nuttall, C., *An Analysis of Ground Mobility Models (ANAMOB),* Technical Report M-71-4, US Army Engineer Waterways Experiment Station, Vicksburg, MS, 1971

[30] McBride, B, R.G. Longoria, and E. Krotkov, "Toward the Measurement and Prediction of the Off-Road Mobility of Small Ground Vehicles," *Proceedings of the 2003 Performance Metrics for Intelligent Systems Workshop (PerMIS'03),* NIST, Gaithersburg, MD, September 16-18, 2003.

[31] Talukder, A., Manduchi, R., Castaño, R., Owens, K., Matthies, L., Castano, A., Hogg, R., "Autonomous Terrain Characterization and Modeling for Dynamic Control of Unmanned Vehicles" *Proceedings of the IEEE Conference on Intelligent Robots and Systems*, 2002

[32] Talukder, A., Manduchi, R., Rankin, A., Matthies, L. "Fast and Reliable Obstacle Detection and Segmentation for Cross-country Navigation" *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2002

[33] Wellington, C., Courville, A., Stentz, A., "Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection" in *Proceedings of Robotics: Science and Systems,* June 2005.

[34] Peters, S., Iagnemma, K., "Mobile robot path tracking of aggressive maneuvers on sloped terrain," *Proceedings of IROS '08*, 2008.

[35] Bekker, G., *Theory of Land Locomotion*, University of Michigan Press, 1956.

[36] Bekker, G., *Introduction to Terrain-Vehicle Systems*, University of Michigan Press, 1969.

[37] Iagnemma, K., Shibly, H., Dubowsky, S., "On-Line Terrain Parameter Estimation for Planetary Rovers", *IEEE International Conference on Robotics and Automation*, 2002.

[38] Kewlani, G., Iagnemma, K., "Mobility prediction for unmanned ground vehicles in uncertain environments," in *Proc. of the SPIE Conference on Unmanned Systems Technology*, vol. 6962, pp. 69621G-1--69621G-12.

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

[40] Urmson C., and Simmons R., "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.

[41] Kuwata Y., Teo J., Karaman S., Fiore G., Frazzoli E., and How J., "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

[42] 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

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

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

[45] 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.

[46] 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.

[47] Weissel, F.; Huber, M.F.; Hanebeck, U.D., "Efficient Control of Nonlinear Noise-Corrupted Systems Using a Novel Model Predictive Control Framework," *American Control Conference, 2007. ACC '07* , vol., no., pp.3751-3756, 9-13 July 2007.

[48] Wendt M., Barz T., Wozny G., "Close-Loop Stochastic Dynamic Optimization Under Probabilistic Output Constraints", Lecture Notes in Control and Information Sciences, in Assessment and Future Directions of Nonlinear Model Predictive Control, vol 358, 2007.

[49] Batina I., Stoorvogel A. A., and Weiland S., "Optimal control of linear, stochastic systems with state and input constraints," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on* , vol.2, no., pp. 1564-1569 vol.2, 10-13 Dec. 2002

[50] Kocijan, J.; Murray-Smith, R.; Rasmussen, C.E.; Girard, A., "Gaussian process model based predictive control," *American Control Conference, 2004. Proceedings of the 2004* , vol.3, no., pp. 2214-2219 vol.3, 30 June-2 July 2004.

[51] Kouvaritakis B., Cannon M., and Tsachouridis V., "Recent developments in stochastic MPC and sustainable development", Annual Reviews in Control, Volume 28, Issue 1.

[52] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality", Automatica, vol. 36, no. 6, pp. 789-814, 2000.

[53] Bemporad A., and Morari M., "Robust Model Predictive Control: A Survey", Lecture Notes in Control and Information Sciences, in Robustness in Identification and Control, vol 245, 1999.

[54] Sorenson M., Kristiansen S., "Model Predictive Control for an artificial pancreas", B.Sc. Thesis, Technical University of Denmark, 2007.

[55] Maciejowski J. M., "Predictive Control with Constraints", Pearson Education, 2001.