# A Hierarchical Method for Stochastic Motion Planning in Uncertain Environments

Michael P. Vitus
University of California at Berkeley

Wei Zhang
Ohio State University

Claire J. Tomlin
University of California at Berkeley

***Abstract*— This paper considers the problem of stochastic motion planning in uncertain environments, and extends existing chance constrained optimal control solutions. Due to the imperfect knowledge of the system state caused by motion uncertainty, sensor noise and environment uncertainty, the system constraints cannot be guaranteed to be satisfied and consequently must be considered probabilistically. To account for the uncertainty, the constraints are formulated as convex constraints on a random variable, known as chance constraints, with the violation probability of all the constraints guaranteed to be below a threshold. Standard chance constrained stochastic motion planning methods do not incorporate environmental sensing which typically leads to overly-conservative solutions. To address this, a novel hierarchical framework is proposed that consists of two main steps: an expected shortest path problem on an uncertain graph and a chance constrained motion planning problem. The first successful, real-time experimental demonstration of chance constrained control with uncertain constraint parameters and variables is also presented for a quadrotor equipped with a Kinect sensor navigating through an uncertain, cluttered 3D environment.**

## I. Introduction

Robotic and autonomous systems are becoming increasingly prevalent in everyday life; there are robots which clean floors, telepresence robots that can buy and deliver breakfast, personal robots for cleaning a room, robotic surgeons, driver assistance systems for automatic parking and adaptive cruise control, and even fully autonomous cars. A critical challenge for planning in all of these systems is the presence of uncertainty which must be accounted for explicitly in order to maximize the success of the plans.

The uncertainty in the planning problem arises from three different sources: (i) motion uncertainty, (ii) sensing uncertainty and (iii) environment uncertainty. The presence of these uncertainties means that the exact system state is never truly known. Consequently, in order to maximize the probability of success, the planning problem must be performed in the space of probability distributions of the system, defined as the belief space. For a stochastic system, however, planning in the belief space is not enough to guarantee success because there is always a small probability that a large disturbance will be experienced. Therefore, a trade-off must be made between the conservativeness of the plan and the performance of the system.

The problem of motion planning with motion noise, sensing noise and environment uncertainty has been studied in the past. Some previous planners [1], [2] account for the motion uncertainty of the system but do not account for the partial observability of the system state or the sensing uncertainty. Others [3], [4] have included both the motion and sensing uncertainty when planning paths through the environment, but they simplify the problem by assuming the maximum likelihood observation is received for all future time-steps. This approximation results in an inaccurate representation of the probability distribution of the state which can lead to a violation of the system constraints. Bry and Roy [5] proposed the rapidly-exploring random belief trees algorithm to account for state dependent stochasticity in the dynamics and measurements when planning through known environments.

Planning under uncertainty can alternatively be handled by *chance constrained programming* introduced by Charnes and Cooper [6]. This formulation allows constraints with non-deterministic constraint parameters, named chance constraints, while only guaranteeing constraint satisfaction up to a specified limit. Blackmore et al. extended chance constrained stochastic motion planning methods in several ways. In [7], they handle non-Gaussian belief distributions by approximating them using a finite number of particles. This transforms the original stochastic control problem into a deterministic one that can be efficiently solved. This sampling approach, however, becomes intractable as the number of samples needed to fully represent the true belief state increases. The work by Blackmore et al. [8] approximated the chance constraints using Boole's inequality and used the idea of risk allocation introduced by [9] to distribute the risk of violating each chance constraint while still guaranteeing the specified level of safety. Vitus and Tomlin [10], [11] also extended the chance constrained methods to incorporate environment uncertainty, but they did not account for environmental sensing.

Another method of modeling the problem of stochastic motion planning through uncertain environments is by using Partially Observable Markov Decision Processes (POMDPs). A POMDP models a system's decision process in which the system's dynamics are not necessarily deterministic, but rather the outcome of its actions could be stochastic, and the system does not know the true underlying state. Instead, the system must take measurements of the underlying state and maintain a probability distribution over all possible states. In order to solve a POMDP, a policy must be calculated for the entire belief space that determines the best action for every possible future belief state that it may encounter. The main difficulty in solving a POMDP is its large computational complexity, and in general solving a POMDP exactly is intractable.

Several researchers have proposed efficient solution techniques for solving POMDPs. The work of Hsu et al. [12] reduced the computational complexity by exploiting the mixed observability of the problem by only maintaining a belief space over the partially observable states. The use of macro-

This work is supported in part by NASA (#NNA06CN22A) and ONR under MURI grant N00014-09-1-1051.

actions has also been used to decrease the computational complexity [13]. Instead of using the full action space, these methods use a sequence of primitive actions called macro-actions; this restricts the policy space and reduces the action branching factor leading to longer planning horizons. However these methods are still too complex to solve the problem of interest.

This work extends previous chance constrained programming formulations used to solve the stochastic motion planning problem in uncertain environments by allowing the environmental uncertainty to be affected by onboard sensors. Typically, standard chance constrained algorithms only account for motion noise, system state sensing noise and *a priori* environmental uncertainty, but they neglect to include any environmental sensing which leads to conservative solutions. A novel hierarchical framework, composed of two main steps, is proposed that incorporates future information to improve the solution over traditional methods. First, an expected shortest path problem on an uncertain graph is solved to determine how the robot should navigate the environment; this step incorporates external sensing and information gain into the problem through the probabilities of the uncertain graph. Second, a chance constrained motion planning problem is solved to execute the expected shortest path plan. As the robot executes the trajectory, the robot's sensors are used to update the model of the environment and replanning occurs as necessary. Finally, the algorithms are validated on the first successful, real-time experimental demonstration of chance constrained control with uncertain constraint parameters and variables. The experiment consists of a quadrotor equipped with a Kinect sensor navigating through an uncertain, cluttered 3D environment.

The paper proceeds as follows. Section II describes the stochastic problem, and the standard chance constrained motion planning formulation. A motivating example for the need for incorporating environmental sensing into the stochastic control solution is presented in Section III. In Section IV the proposed hierarchical method is developed. The method is experimentally demonstrated in real-time on a quadrotor vehicle navigating through a 3D environment in Section V.

## II. Problem Formulation

Consider the following linear stochastic system defined by,

$$x_{k+1} = Ax_k + Bu_k + w_k, \, \forall k \in [0, N-1], \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the system state, $w_k \in \mathbb{R}^n$ is the process noise and $N$ is the time horizon. The initial state, $x_0$, is assumed to be a Gaussian random variable with mean $\bar{x}_0$ and covariance $\Sigma_0$ i.e., $x_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_0)$. At each time step, a noisy measurement of the state is taken, defined by

$$y_k = Cx_k + v_k, \, \forall k \in [1, N], \tag{2}$$

where $y_k \in \mathbb{R}^p$ and $v_k \in \mathbb{R}^p$ are the measurement output and noise of the sensor at time $k$, respectively. The process and measurement noise have zero mean Gaussian distributions, $w_k \sim \mathcal{N}(0, \Sigma_w)$ and $v_k \sim \mathcal{N}(0, \Sigma_v)$. The process noise, measurement noise and initial state are assumed to be mutually independent. For notational convenience, the state and control inputs for all time-steps are concatenated to form, $\mathbb{X} = \begin{bmatrix} x_1^{\mathrm{T}} & \ldots & x_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and $\mathbb{U} = \begin{bmatrix} u_0^{\mathrm{T}} & \ldots & u_{N-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$.

The control inputs are required to be in a convex region denoted by $F_U$ and the system state is restricted to be in a feasible region denoted by $F_X$. To simplify the presentation of the material, the feasible region $F_X$ is assumed to be convex. Nonconvex regions can still be handled, however, either by (i) performing branch and bound on the set of conjunction and disjunction linear state constraints directly, or by (ii) decomposing the space into convex regions and using branch and bound to determine when to enter/exit each convex subregion. Given this assumption, the feasible region can be defined by a conjunction of $N_{F_X}$ linear inequality constraints,

$$F_X \triangleq \bigcap_{i=1}^{N_{F_X}} \left\{ \mathbb{X} : h_i^{\mathrm{T}} \mathbb{X} \leq b_i \right\} \tag{3}$$

where $h_i \in \mathbb{R}^{nN}$ and $b_i \in \mathbb{R}$. In this work, the environment is uncertain but the parameters of the probability distribution describing $h_i$ and $b_i$ are assumed to be known.

In this formulation, it is assumed that a linear feedback trajectory controller has been designed and it uses a Kalman filter to estimate the state. Given these choices, the distribution of the closed-loop system can be calculated *a priori* and is given by a Gaussian distribution: $\mathbb{X} \sim \mathcal{N}\left(\bar{\mathbb{X}}, \Sigma_{\mathbb{X}}\right)$.

The general belief space planning problem is posed as the optimization program (4). The optimization variable is the desired trajectory through the environment that the controller should follow. The objective function, $f(\cdot)$, is assumed to be a convex function in $\mathbb{X}$ and $\mathbb{U}$.

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\left[f(\mathbb{X}, \mathbb{U})\right] \\ \text{subject to} & \\ & x_{k+1} = Ax_k + Bu_k + w_k, \, \forall k \in [0, N{-}1] \\ & y_k = Cx_k + v_k, \, \forall k \in [1, N] \\ & \bar{\mathbb{U}} \in F_U \\ & \mathrm{P}(\mathbb{X} \notin F_X) \leq \delta \end{array} \tag{4}$$

The difficulty in solving the optimization program (4) is in evaluating the chance constraints: $\mathrm{P}(\mathbb{X} \notin F_X) \leq \delta$. In particular, by allowing $H$ and $b$ to be uncertain, the distribution of $H^{\mathrm{T}}\mathbb{X} - b$ becomes a sum of products of random variables. In general, the properties of the probability distribution for this constraint are not easy to calculate analytically, increasing the complexity of the problem. For an in-depth discussion on efficient methods to solve the previous chance constrained optimization program please refer to [10], [11], and the following sections assume that the optimization program (4) can be solved.

The current formulation does not take into account the ability to sense the environment which will lead to a conservative strategy for navigating through the uncertain environment. The following section motivates the need for incorporating sensing of the uncertain environment into the stochastic control solution.

## III. Motivating Example for Dual Control

The system has double integrator dynamics with a time-step of $\Delta t = 0.1$ seconds and a time-horizon of $N = 65$.

The noise parameters are $\Sigma_w = 0.0002I$ and $\Sigma_v = 0.001I$. The allowed probability of constraint violation is $\delta = 0.05$. The objective function for this problem is quadratic in the final state as well as the control inputs $f(\bar{\mathbb{X}}, \bar{\mathbb{U}}) = (x_N - x_{\text{ref}})^{\mathrm{T}} Q_{\text{obj}} (x_N - x_{\text{ref}}) + \bar{\mathbb{U}}^{\mathrm{T}} R_{\text{obj}} \bar{\mathbb{U}}$, with $Q_{\text{obj}} = 50I$, $R_{\text{obj}} = 0.001I$ and $x_{\text{ref}} = \begin{bmatrix} 2 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$.

The environment is modeled by a set of half plane constraints defined by a series of end points, which are assumed to be uncertain. The uncertainty is modeled by a truncated Gaussian represented as the orange ellipses in Figure 1.

The solution without incorporating constraint sensing of the optimization program (4) is shown in Figure 1. Here, the system must take the longer route around the top of the obstacle because the bottom route violates the chance constraints due to the large environmental uncertainty in the second corridor. This results in a path length of 2 times larger than if the system was able to take the bottom route.

For this example, the system is forced to take the longer path because environmental sensing is not taken into account. In this case, it might have been beneficial for the system to gain information by exploring the environment to improve the solution of its task. For instance, the shorter path might have been enabled through sensing the walls in the environment while executing its trajectory. The proposed hierarchical framework developed in the following section systematically trades off between exploring to gain information and accomplishing the task while ensuring the chance constraints are satisfied.

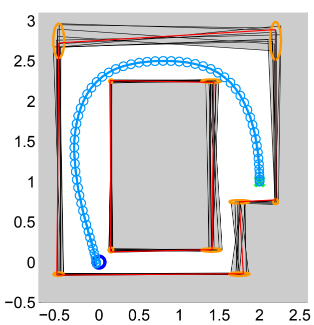



Fig. 1. The white area is the feasible region of the system state and the orange ellipses are the uncertainty of the environment. The blue, solid line is the solution when accounting for the uncertainty of the environment. The blue ellipses around the path indicate the uncertainty of the system. The start and goal location are marked by an 'o' and 'x', respectively.

## IV. Hierarchical Method

One way of incorporating environment sensing is by modeling the problem as a POMDP. The main difficulty in solving this POMDP is its large computational complexity, and in general solving a POMDP exactly is intractable. The two main causes of the large computational complexity is the curse of dimensionality and the long planning horizons. The previous state of the art methods [12], [13] for solving POMDPs do not result in a tractable solution method. Consequently, a new algorithm is needed to account for the environmental uncertainty and to reduce the large computational complexity.

In order to overcome the high computational complexity of solving the problem, it is broken down into two phases: (i) an expected shortest path problem on an uncertain graph and (ii) a chance constrained optimization problem.

This method reduces the complexity in two ways. The first step uses an abstraction of the dynamics and sensing to determine how the system should explore the uncertain environment. Instead of modeling the individual sensor measurements for the environment, only the outcome of the sensing is used. Specifically, the probability of the environment being passable after sensing is calculated. By using these two abstractions, this drastically reduces the size of the planning space. The second step reduces the computational complexity by solving the stochastic control problem around the highlevel plan from the first step. By decomposing the problem in this way, the size of the belief space is significantly reduced by planning only where it is best to observe the environment. This in turn greatly reduces the complexity. The hierarchical method is described in Algorithm 1.

The first step of the algorithm decomposes the environment into a set of nodes and edges as shown in Figure 2. Then the probability the link will be traversable after sensing is calculated. After the graph is built, the expected shortest path problem can be solved to provide a highlevel plan for the chance constrained stochastic motion planning algorithm. Then the robot will execute the plan and sense the environment to update the probability distribution of the environment. This procedure repeats until the robot completes its objective.

**Algorithm 1** Hierarchical Stochastic Motion Planning

```
1: while objective not complete do
2:     Decompose the environment into a graph
3:     Calculate the probabilities of the edges being
       traversable after sensing
4:     Solve the expected shortest path problem
5:     Solve the chance constrained stochastic motion plan-
       ning problem
6:     Execute the plan while sensing the environment
7: end while
```

### A. Environment Decomposition

The first step in using the hierarchical stochastic motion planning algorithm is to decompose the environment into a graph. There are several different methods that can be used and the design of the decomposition algorithm is out of the scope of this paper. One approach would be to build a graph based upon motion primitives. Care should be taken to provide a decomposition of the environment that is as dynamically feasible as possible to ensure that the robot can execute the highlevel plan. Once the graph has been constructed, the probability for each link being traversable *after* sensing needs to be calculated.

To simplify the discussion, the environmental uncertainty is assumed to only be in the parameter $b$. The probability of

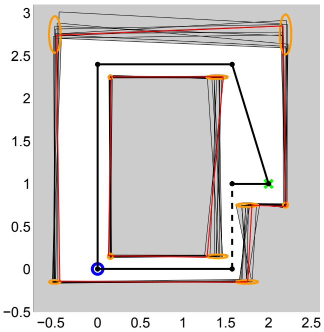



Fig. 2. A graph based decomposition of the environment. The dotted link has a large probability of failure before sensing but after sensing it has a large probability of being traversable.

collision before sensing for a single constraint (i.e. $H^{\mathrm{T}}\mathbb{X} \in \mathbb{R}$ and $b \in \mathbb{R}$) is calculated via

$$\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b\right) = \int\int \mathbf{1}\left(H^{\mathrm{T}}\mathbb{X} > b\right)\mathrm{P}(b)\mathrm{P}(\mathbb{X})dbd\mathbb{X}. \quad (5)$$

However, the statistic needed for each link in the expected shortest path problem is the probability that *after* sensing the link will be traversable. In order to calculate this, an allowed probability of collision ($\epsilon$) for the constraint needs to be allocated which will be discussed in the following section. Once the allowed probability of collision for the constraint has been determined, the probability the link will be traversable after sensing is calculated via

$$\mathrm{P}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|H, b\right) \leq \epsilon\right), \quad (6)$$

where $\epsilon$ is the allowed probability of constraint violation associated with the constraint. For the current derivation with uncertainty only in the parameter $b$, Eqn. (6) can be simplified further to

$$\mathrm{P}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon\right) = \int \mathbf{1}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon\right)\mathrm{P}(b)db. \quad (7)$$

If $b^*$ can be found such that $\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon$ for $b \geq b^*$ and $\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) > \epsilon$ for $b < b^*$, then the probability in Eqn. (7) can be simplified to

$$\mathrm{P}\left(\mathrm{P}\left(H^{\mathrm{T}}\mathbb{X} > b|b\right) \leq \epsilon\right) = \int_{b^*}^{\infty}\mathrm{P}(b)db. \quad (8)$$

For a given link, if there is more than one constraint, then a similar methodology can be used to determine an analytic formula for the probability. If both $H$ and $b$ are uncertain, the complexity of calculating the probability increases because it requires evaluating multivariate integrals, however, it can be efficiently calculated through Monte Carlo sampling.

Consider how this new formulation would effect the results in the previous example shown in Figure 1. The probability of failure for traversing the dotted link in Figure 2 *before* sensing is $0.15$ but the probability the link will be traversable *after* sensing is $0.64$ for an allowed probability of failure of $0.02$. The large probability of failure before sensing requires the robot to take the long route around the obstacle in Figure 1. However, the large probability that the link will be traversable *after* sensing suggests that it might be beneficial to first investigate whether or not the corridor is traversable.

The best strategy for navigating the environment can be calculated by solving the expected shortest path problem through the defined uncertain graph, however the strategy is highly impacted by the risk assigned to each edge. The following section discusses a method for allocating these risks through convex optimization.

### B. Risk Allocation for the Graph Decomposition

To calculate the probability of failure after sensing, the allowed risk for each link needs to be known. If the allowed risk is improperly assigned, then it may cause certain paths that should be explored to be overlooked by the expected shortest path algorithm. Therefore it is necessary to carefully allocate the risk in order to correctly trade off between exploration and execution in the expected shortest path calculation. The method presented below is based upon solving a convex optimization problem.

An example environment decomposition is given by the undirected graph in Figure 3(a). Here, $p_{ij}$ is the probability after sensing that the edge between node $i$ and $j$ will be passable and is defined as $p_{ij} = f_{ij}(\epsilon_{ij})$, with $f_{ij}(\epsilon_{ij})$ given by Eqn. (6). Let $e_{ij}$ be the edge between node $i$ and $j$. For any path through the graph, there is a set of edges $\{e^{(1)}, \ldots, e^{(n)}\}$ with associated probabilities $\{p^{(1)}, \ldots, p^{(n)}\}$ and probability functions $\{f^{(1)}(\epsilon^{(1)}), \ldots, f^{(n)}(\epsilon^{(n)})\}$.

For the systems considered the probability functions $f_{ij}$ are positive, nondecreasing functions of the allowed risk, and they are typically log-concave functions which will be exploited in formulating the problem. One example of the probability function is shown in Figure 3(b).

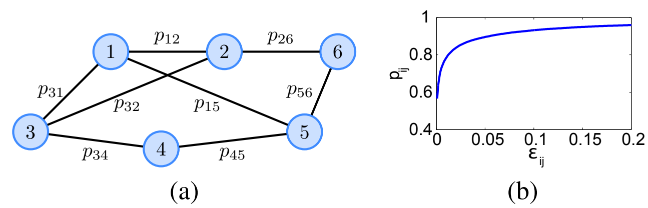



Fig. 3. (a) An example graph for the calculation of the expected shortest path. (b) An example of the probability of the link being traversable after sensing versus the allowed risk.

One method of assigning the risk for each edge is by maximizing the probability the path will be passable. The *a priori* probability that after sensing the path between any two nodes will be passable is the product of the probabilities for each edge that is traveled, i.e. $\prod_{m=1}^{n} p^{(m)}$. This assignment of risk can be obtained by solving the following optimization program.

$$\begin{array}{ll} \text{maximize} & \prod_{m=1}^{n} f^{(m)}(\epsilon^{(m)}) \\ \text{subject to} & \\ & \epsilon^{(m)} \geq 0, \quad m = 1, \ldots, n \\ & \sum_{m=1}^{n} \epsilon^{(m)} \leq \delta, \quad m = 1, \ldots, n \end{array} \quad (9)$$

The optimization program maximizes the probability of success subject to the constraint that the sum of the assigned risk over the path is smaller than the user allowed amount. Unfortunately, this program is not necessarily convex because the objective function is not guaranteed to be concave. To form a convex optimization program, a set of slack variables are introduced based upon the logarithm of the probability functions and the objective function is replaced by its logarithm. This results in the following equivalent convex optimization program.

$$\begin{array}{ll} \text{maximize} & \sum_{m=1}^{n} \alpha_m \\ \text{subject to} & \\ & \epsilon^{(m)} \geq 0, \quad m = 1, \ldots, n \\ & \sum_{m=1}^{n} \epsilon^{(m)} \leq \delta \\ & \alpha_m \leq \log\left(f^{(m)}(\epsilon^{(m)})\right), \quad m = 1, \ldots, n \end{array} \tag{10}$$

In this formulation, the optimization program will need to be solved for all paths without cycles between the start and goal location, and it may assign a different allowed risk for each link depending on the path. To obtain a risk assignment for all edges that is independent of the path, the problem formulation can be changed to a single optimization program that maximizes the probability of successfully traveling over *all* paths subject to the same constraints for all paths.

Using the risk allocation computed from the above formulation, the expected shortest path problem is solved next in order to determine the trade off between exploring to gain more information and completing the objective of the planning process.

### C. Expected Shortest Path

The expected shortest path problem on an uncertain graph has been extensively studied. The approach taken in this work is to convert the problem into a Markov Decision Process (MDP) as shown in [14]. There are many different solution methodologies to solve for the exact solution of a MDP, such as value or policy iteration. However, the formulation of the expected shortest path problem as a MDP has an exponential growth of states with respect to the number of uncertain edges in the graph. Since each uncertain edge can have three values associated with it, $\{0, 1, \mathrm{P}(e)\}$, the number of states in the graph could be as large as $|V(G)| \cdot 3^{|E(G)|}$. This exponential growth of the number of states will prohibit solving for the exact solution for large sized problems. Fortunately, there are many approximate MDP solvers [15] that can handle large size problems and provide near optimal solutions for the exploration phase of the algorithm.

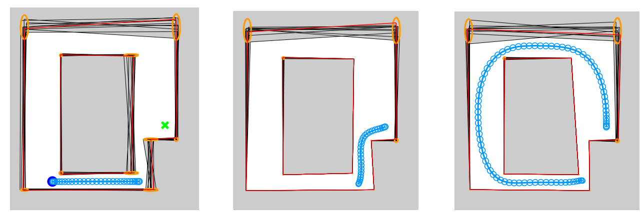
### D. Stochastic Motion Planning

Once the expected shortest path problem has been solved to determine the route the robot should take, the standard stochastic motion planning problem can be formulated and solved as shown in Section II. Since parts of the uncertain environment are too risky to pass through before sensing, the robot continuously senses the environment and updates the probability distribution of its parameters. If at any point it determines that the highlevel plan will violate the chance constraints, it will then re-iterate the algorithm solving for a new highlevel plan.

### E. Example

The same example as in Section III is solved through the proposed hierarchical stochastic motion planning algorithm and the solution is shown in Figure 4. After solving the expected shortest path problem, the best path for the robot to take is around the bottom of the obstacle, as shown in Figure 4(a), even though the probability of failure in the second corridor before sensing violates the constraints. As the robot executes the plan, it continuously senses the environment and updates the probability distribution of the environment and determines that for this environment the corridor is passable. Consequently, it proceeds directly to the intended goal location as shown in Figure 4(b). However, if after sensing the environment the robot determined that the corridor was indeed impassable then it would have backtracked and proceeded around the top of the obstacle as displayed in Figure 4(c).

As compared to the previous chance constrained programming solution shown in Figure 1, the proposed algorithm is able to take into account the environmental sensing and reduces the cost by $30\%$ on average.



(a) (b) (c)

Fig. 4. The solution from the hierarchical stochastic motion planning algorithm. (a) The best strategy is to attempt to take the shortest path to the goal location to sense the uncertain corridor. (b) After incorporating the measurements of the environment, the robot replans and is able to take the shortest path to the goal location. (c) For a different realization of the environment, after sensing the environment the probability of failure is too large for the shortest path and forces the robot to backtrack.

## V. Experimental Demonstration

The hierarchical stochastic motion planning algorithm was also evaluated on a quadrotor unmanned aerial vehicle. To the authors knowledge, this is the first successful, real-time experimental demonstration of chance constrained control through uncertain environments. All the algorithms were executed in real-time on a Intel Core i7 2.67 GHz computer without any human intervention. This experiment demonstrates the ability to use the chance constrained programming techniques for real-time applications. A video of the experiment can be found at `http://hybrid.eecs.berkeley.edu/~vitus/iros2012`

The vehicle has an onboard inertial measurement unit which provides three-axis attitude, attitude rate and acceleration measurements. An external Vicon positioning system is also used to provide measurements of the quadrotor's position with respect to a global coordinate frame. While the Vicon positioning system does provide very accurate measurements, the system is still greatly affected by disturbances created by the downwash of the rotors interacting with nearby objects. Therefore, the motion of the vehicle must be consider probabilistically. To sense the environment, the quadrotor was equipped with a Kinect sensor that provides a dense 3D point cloud measurement of the environment.

An initial map of the environment was constructed by manually flying the quadrotor around. The quadrotor is navigating through a cluttered 3D environment as displayed in Figure 5. In this example there are two sources of environment uncertainty: (i) the obstacle locations are not known exactly due to the sensing error of the Kinect sensor, (ii) the cabinet that is outlined with the orange dotted line in Figure 5 has a 0.3 probability of its door being open. If the door is open it would prevent the quadrotor from taking the shortest path to the goal location. The error from the Kinect sensor can be directly incorporated into the chance constrained stochastic motion planning algorithm by including it in the probability distribution of the uncertain environment.

The proposed hierarchical motion planning algorithm was solved in real-time and the results are shown in Figure 6. The solution from the expected shortest path problem instructed the quadrotor to first determine if the cabinet door was open then either proceed directly to the goal location or backtrack through the middle corridor. To solve the chance constrained stochastic motion planning problem the environment was decomposed into convex polytopes and the quadrotor's state was constrained to be within one of the polytopes. An initial path was determined through $A^*$ search as developed in [16]. Figure 6 shows the final map of the environment in which each point is color coded based upon the height off the floor. The start and goal location are shown as the green and red sphere, respectively. The planned trajectory is shown as the orange line and the executed trajectory of the quadrotor is the blue line. In this experimental trial the cabinet door was open which prevented the quadrotor from taking the shortest path to the goal location.


Fig. 5. An image of the stochastic environment. The start position is out of view on the bottom lefthand corner of the image and the goal location is marked by the red circle. The cabinet on the right of the image marked by the orange dotted rectangle is stochastic in the nature that the door could be open or closed which might prevent the robot from passing through that corridor.

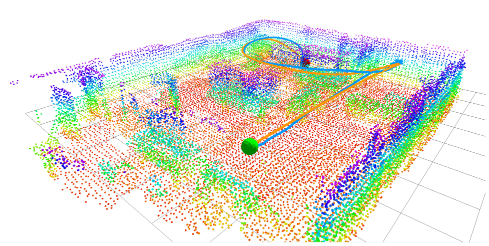


Fig. 6. The final point cloud representation of the environment. The start and goal locations are the green and red spheres, respectively. The points correspond to obstacle locations and are color coded based upon the height off the floor. The planned trajectory is shown as the orange line through the environment and the trajectory of the quadrotor is shown as the blue line.

## VI. Conclusion

In this work, a hierarchical stochastic motion planning algorithm was proposed to incorporate external sensing. The algorithm is composed of two steps: an expected shortest path problem is solved to determine the route the robot should take through the environment, and a chance constrained optimization problem is solved for the robot's trajectory. The algorithm was successfully demonstrated in real-time on a quadrotor vehicle navigating through a cluttered 3D environment.

## Acknowledgements

The authors would like to thank Patrick Bouffard for performing the system identification of the quadrotor testbed.

## References

[1] N. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proceedings IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007.
[2] R. Alterovitz, T. Simeon, and K. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
[3] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.
[4] R. Pepy and A. Lambert, "Safe path planning in an uncertain-configuration space using RRT," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
[5] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.
[6] A. Charnes and W. Cooper, "Deterministic equivalents for optimizing and satisfying under chance constraints," *Operations Research*, no. 11, pp. 18–39, 1963.
[7] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Keystone, Colorado, August 2006.
[8] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois, August 2009.
[9] A. Prékopa, "The use of discrete moment bounds in probabilistic constrained stochastic programming models," *Annals of Operations Research*, vol. 85, pp. 21–38, 1999.
[10] M. P. Vitus and C. J. Tomlin, "Belief space planning for linear, Gaussian systems in uncertain environments," in *18th World Congress of the International Federation of Automatic Control*, Milan, Italy, September 2011.
[11] ——, "A hybrid method for chance constrained control in uncertain environments," in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2012.
[12] Y. Du, D. Hsu, H. Kurniawati, W. Lee, S. Ong, and S. Png, "A POMDP approach to robot motion planning under uncertainty," in *International Conference on Automated Planning & Scheduling, Workshop on Solving Real-World POMDP Problems*, 2010.
[13] M. Hauskrecht, N. Meuleau, L. P. Kaelbling, T. Dean, and C. Boutilier, "Hierarchical solution of Markov decision processes using macro-actions," in *Proceedings of Uncertainty in Artificial Intelligence*, 1998, pp. 220–229.
[14] D. M. Blei and L. P. Kaelbling, "Shortest paths in a dynamic uncertain domain," in *Proceedings of the IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*, 1999.
[15] J. Barry, L. Kaelbling, , and T. Lozano-Perez, "Hierarchical solution of large Markov decision processes," Massachusetts Institute of Technology, Tech. Rep., 2010.
[16] N. Vandapel, J. Kuffner, and O. Amidi, "Planning 3-D path networks in unstructured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.