2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

# Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environment

Ruijie He, Sam Prentice and Nicholas Roy

***Abstract*— This paper describes a motion planning algorithm for a quadrotor helicopter flying autonomously without GPS. Without accurate global positioning, the vehicle's ability to localize itself varies across the environment, since different environmental features provide different degrees of localization. If the vehicle plans a path without regard to how well it can localize itself along that path, it runs the risk of becoming lost.**

**We use the Belief Roadmap (BRM) algorithm [1], an information-space extension of the Probabilistic Roadmap algorithm, to plan vehicle trajectories that incorporate sensing. We show that the original BRM can be extended to use the Unscented Kalman Filter (UKF), and describe a sampling algorithm that minimizes the number of samples required to find a good path. Finally, we demonstrate the BRM path-planning algorithm on the helicopter, navigating in an indoor environment with a laser range-finder.**

## I. INTRODUCTION

Unmanned air vehicles (UAVs) rely heavily on accurate knowledge of their position for decision-making and control. As a result, considerable investment has been made towards improving the availability of global positioning infrastructure, including utilizing satellite-based GPS system and developing algorithms to leverage existing RF signals such as WiFi. However, most indoor environments and many parts of the urban canyon remain without access to external positioning systems. Autonomous UAVs thus currently have limited ability to fly through these areas.

Vehicle localization using sonar ranging [2] or laser ranging [3] has been used extremely successfully in a number of applications and is now essentially a commodity technology, especially aboard ground robots. Unfortunately, the UAV community has not been able to leverage the ground vehicle successes for two reasons. First, some of the most successful demonstrations of long-term robot autonomy have used planar laser ranging based on the ubiquitous SICK laser range finder, which normally provides localization information for three dimensions without additional specialized hardware. While this is sufficient for ground vehicle localization, localization in six dimensions during flight requires considerably more data. Second, UAVs are severely constrained by weight and, consequently, power limitations. A vehicle small enough to fly indoors or through populated urban areas safely can carry very little in terms of sensor payload, leading to reduced range and field of view.

Nevertheless, most UAVs can carry *some* sensing capability for localization; they simply cannot carry sensors that enable themselves to localize *everywhere*. If the vehicle can use its sensor model to incorporate predicted measurements into its decision making, then the vehicle can plan trajectories that are robust to sensor limitations.

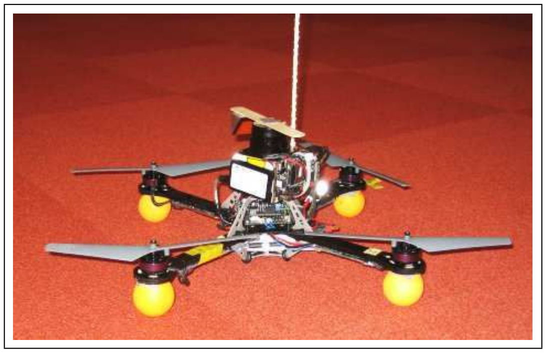


Fig. 1. Our quadrotor helicopter.

In this paper, we describe a planning algorithm for the quadrotor helicopter, shown in Figure 1, and built by Ascending Technologies [4]. We outfit this vehicle with a laser range-finder capable of estimating position, yaw angle and altitude information from environmental features within a $4m$ range in a $240°$ field of view. The limited range and field of view of the sensor lead to position estimates that vary in accuracy and confidence over the environment.

Our algorithm is based on the Belief Roadmap (BRM) algorithm [1], which is a generalization of the Probabilistic Roadmap (PRM) algorithm [5]. The BRM performs searches in the information space of the vehicle efficiently by using the sympletic form of the Extended Kalman Filter (EKF) to find the minimum expected cost path for the vehicle. We make two contributions in extending the BRM in this paper. First, we show how to generalize the BRM to use the Unscented Kalman Filter (UKF) [6] for position tracking, providing better approximation of the non-linearities of UAV motion and laser sensing. Second, we use the notion of a "Sensor Uncertainty Field" [7] and show how a model of sensor uncertainty can be used to generate a more efficient representation of the information space. Finally, we conclude the paper with a demonstration of the quadrotor helicopter using the BRM algorithm to navigate autonomously indoors.

## II. TRAJECTORY PLANNING

We first formulate the problem of motion planning for a UAV. We assume that the vehicle is holonomic and that we have full control authority, allowing us to ignore vehicle dynamics and treat the problem as a kinematic motion planning problem. $\mathcal{C}$ denotes the configuration space [8], the space of all vehicle poses, $\mathcal{C}_{free}$ is the set of all collision-free poses (based on the map $\mathcal{M}$ of obstacle positions) and $\mathcal{C}_{obst}$ is the set of poses resulting in collision with obstacles, so that $\mathcal{C} \equiv \mathcal{C}_{free} \cup \mathcal{C}_{obst}$. Given an initial vehicle state $s_0$ and a map of the environment, the planning problem is to

Ruijie He, Sam Prentice and Nicholas Roy are members of the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139. ruijie@mit.edu, prentice@mit.edu, nickroy@mit.edu

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

find a sequence of actions to move the vehicle from state $s_0$ to a goal state $s_g$ without collisions. Our UAV has 6 degrees of freedom ($x$, $y$, $z$, roll, pitch, yaw), so $\mathcal{C} = \mathcal{R}^6$, which is of moderately high dimension.

The Probabilistic Roadmap (PRM) is a common algorithm [5] for planning in high-dimensional problems, in which a discrete graph is used to approximate the connectivity of $\mathcal{C}_{free}$. The PRM builds the graph by sampling a set of states randomly from $\mathcal{C}$ (adding the start state $s_0$ and goal state $s_g$ to the sample set), and then evaluating each state for membership in $\mathcal{C}_{free}$; the assumption is that it is much cheaper to evaluate randomly sampled poses in higher dimensions than it is to build an explicit representation of $\mathcal{C}_{free}$. Samples that lie within $\mathcal{C}_{free}$ constitute the nodes of the PRM graph and edges are placed between nodes where a straight line path between nodes also lies entirely within $\mathcal{C}_{free}$. Given this graph, a feasible, collision-free path can be found using a standard graph search algorithm from the start node to the goal node. The path can be executed by using a simple controller to follow each edge to the goal.

However, the PRM and its variants are not yet well-suited to the problem of a GPS-denied UAV, in that executing a plan requires a controller that follows the straight-line edges joining graph points. If the UAV executing the plan does not have a good estimate of its state, it may not be able to determine when it has arrived at a graph node and should start to follow a new edge. Even more seriously, vehicle stability typically depends on accurate state estimation of higher order terms (such as velocity). Without enviromental feedback, IMU estimation can quickly drift causing catastrophic control failures.

## III. VEHICLE POSITION ESTIMATION

If the UAV does not have access to perfect state knowledge, such as through GPS, it can still localize itself by using sensors to measure environmental features and then registering those measurements against a pre-existing map. Bayesian filtering is one of the most robust methods of localization [2], in which a probability distribution $p(s_t|u_{1:t}, z_{1:t})$ is inferred over the (unknown) vehicle state $s_t$ at time $t$ following a series of noisy actions $u_{1:t}$ and measurements $z_{1:t}$. With some standard assumptions about the actions and observations, the posterior distribution (or belief) can be expressed as

$$p(s_t|u_{1:t}, z_{1:t}) = \frac{1}{Z} p(z_t|s_t) \int_S p(s_t|u_t, s_{t-1}) p(s_{t-1}) ds_{t-1}, \quad (1)$$

where $Z$ is a normalization factor. Equation (1), referred to as the Bayes' filter, provides an efficient recursion for updating the state distribution.

The Kalman filter is a form of Bayes filtering that assumes that all probability distributions are Gaussian, and that the transition and observation Gaussians are linearly parameterized by the state and control. The Extended Kalman filter (EKF) allows the same inference algorithm to operate with non-linear transition and observation functions by linearizing these functions around the current mean estimate. More formally, the next state $s_t$ and observation $z_t$ are given by the following functions,

$$s_t = g(s_{t-1}, u_t, w_t), \qquad w_t \sim N(0, W_t), \quad (2)$$

$$\text{and} \quad z_t = h(s_t, q_t), \qquad q_t \sim N(0, Q_t), \quad (3)$$

where $u_t$ is a control action, and $w_t$ and $q_t$ are random, unobservable noise variables. The EKF computes the state distribution at time $t$ in two steps: a process step based only on the control input $u_t$ leading to an estimate $p(\overline{s}_t) = N(\overline{\mu}_t, \overline{\Sigma}_t)$, and a measurement step to complete the estimate of $p(s_t)$. The process step follows as

$$\overline{\mu}_t = g(\mu_{t-1}, u_t), \qquad \overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t W_t V_t^T, \quad (4)$$

where $G_t$ is the Jacobian of $g$ with respect to $s$ and $V_t$ is the Jacobian of $g$ with respect to $w$. For convenience, we denote $R_t \triangleq V_t W_t V_t^T$. Similarly, the measurement step follows as:

$$\mu_t = \overline{\mu_t} + K_t(H_t \overline{\mu}_t - z_t), \qquad \Sigma_t = (I - K_t H_t)\overline{\Sigma}_t, \quad (5)$$

where $H_t$ is the Jacobian of $h$ with respect to $s$ and $K_t$ is known as the Kalman gain, given by

$$K_t = \overline{\Sigma}_t H_t^T \left( H_t \overline{\Sigma}_t H_t^T + Q_t \right)^{-1}. \quad (6)$$

An alternate form of the EKF represents the covariance by its inverse, the information matrix [9]. The information matrix updates can be written as

$$\overline{\Omega}_t = \overline{\Sigma}_t^{-1} = \left(G_t \Sigma_{t-1} G_t^T + R_t\right)^{-1} \quad (7)$$

$$\Omega_t = \overline{\Omega}_t + H_t^T Q_t^{-1} H_t. \quad (8)$$

For convenience, we denote $M_t \triangleq H_t^T Q_t^{-1} H_t$ such that $\Omega_t = \overline{\Omega}_t + M_t$. The distribution $p(s_t|u_{1:t}, z_{1:t})$ can be represented by the information vector $\xi_t$ and the information matrix $\Omega_t = \Sigma_t^{-1}$ and may be more efficient to compute in domains where the information matrix is sparse.

## IV. BELIEF SPACE PLANNING

Recall from section II that the PRM planning algorithm constructs a graph in the state space $\mathcal{C}_{free}$ of the vehicle. However, the vehicle does not know its actual state but only has access to the EKF state estimate $b = (\mu, \Sigma)$; by planning in the *belief space* (or information space), the vehicle can distinguish between state estimates where the norm of the covariance is small (i.e., the vehicle has high confidence in its mean state estimate) and state estimates where the norm of the covariance is large (i.e., the mean state estimate is uncertain). Ideally, beliefs with high uncertainty are to be avoided, and if encountered, conservative sensing action would be a reasonable response.

Conventional motion planners generally search for a collision-free path that minimizes the distance to the goal location. In belief space, every belief typically has some probability that the robot is at the goal state. A more appropriate objective function is therefore to maximize the probability of the goal state.

A naive approach to planning in belief space would therefore involve sampling beliefs directly from $(\mu, \Sigma)$, adding the initial belief $b_0$ to construct the graph nodes, placing edges between pairs of beliefs $(b^i, b^j)$ for which a controller exists that can take the vehicle from belief $b^i$ to $b^j$, and then carrying out graph search as before. Unfortunately, it has been shown [1] that the likelihood is zero of sampling any beliefs that are actually reachable from the initial belief $b_0$.

However, the EKF representation of the belief space carries an extremely useful property. Each belief $b_t$ is a

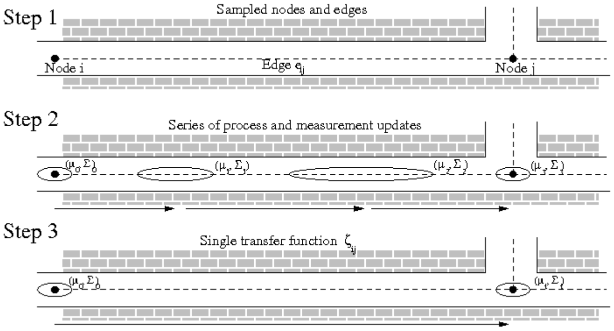


Fig. 2. The Belief Roadmap with one-step transfer functions calculated using the UKF. In step 1, the graph of mean poses is constructed, and mutually visible nodes are connected with edges. In step 2, the posterior covariance is calculated through a series of process and measurement updates. In step 3, the one-step covariance transfer function is calculated from the individual multi-step updates.

combination of $\mu$ and $\Sigma$. Under some mild assumptions of unbiased motion and sensor models, the reachability of any $\mu$ is a function of the vehicle kinematics and the environmental structure as in the PRM. For some $\mu$ that is reachable along a path from $\mu_0$, the corresponding reachable covariance can be predicted by propagating the initial covariance $\Sigma_0$ along the path using equations (4) and (5) and the motion and sensing models. Therefore, to construct a graph of the reachable belief space, the planner first samples a set of mean poses $\{\mu_i\}$ from $\mathcal{C}_{free}$ using the standard pose sampling of the PRM algorithm, and places an edge $e_{ij}$ between pairs $(\mu_i, \mu_j)$ if the straight line between poses is collision-free. Forward search is used to search for a path through the graph, but each step of the search computes the posterior covariance at each node instead of the standard cost-to-go.

### A. Belief Updating as a One-Step Operation

The most computationally demanding aspect of the graph-search algorithm described above is in propagating the initial covariance $\Sigma_0$ to each graph node. Covariance propagation requires multiple EKF updates along each edge $e_{ij}$, and while this operation is a constant multiplier of the asymptotic search complexity, it can still dominate the overall search time. Furthermore, these EKF updates are not a one-time cost; the search process will find multiple paths to node $i$. Each of these paths will lead to a different posterior covariance at node $i$, and each such covariance must then be propagated outwards from $i$ along edge $e_{ij}$ to reach node $j$, incurring the computational cost of propagating along the edge (a series of EKF updates) for each covariance. The BRM algorithm avoids this complexity by using an alternate representation of the covariance that allows multiple EKF updates to be compiled into single linear transfer function. By pre-computing the transfer function for each edge, the search complexity for belief space planning becomes comparable to configuration space planning.

It has been shown previously [1] that the covariance of a Kalman filter-based state estimator can be factored as $\Sigma = BC^{-1}$, where the combined process and measurement update for an EKF gives $B_t$ and $C_t$ as linear functions of $B_{t-1}$ and $C_{t-1}$.

$$\text{Given: } \Sigma_{t-1} = B_{t-1}C_{t-1}^{-1} \tag{9}$$

$$\Rightarrow \overline{\Sigma}_t = G_t B_{t-1} C_{t-1}^{-1} G_t^T + R_t \tag{10}$$

**Algorithm 1** The Belief Roadmap (BRM) algorithm.

```
Require: Start belief $(\mu_0, \Sigma_0)$, goal $\mu_{goal}$ and map $\mathcal{C}$
Sample poses $\{\mu_i\}$ from $\mathcal{C}_{free}$ to build belief graph node set $\{n_i\}$ such that $n_i = \{\mu = \mu_i, \Sigma = \emptyset\}$
Create edge set $\{e_{ij}\}$ between nodes $(n_i, n_j)$ if the straight-line path between $(n_i[\mu], n_j[\mu])$ is collision-free
Build one-step transfer functions $\{\zeta_{ij}\} \quad \forall \quad e_{ij} \in \{e_{ij}\}$
Augment node structure with best path $p{=}\emptyset$, such that $n_i{=}\{\mu, \Sigma, p\}$
Create search queue with initial position and covariance $Q \leftarrow n_0{=}\{\mu_0, \Sigma_0, \emptyset\}$
while $Q$ is not empty do
  Pop $n \leftarrow Q$
  if $n = n_{goal}$ then
    Continue
  end if
  for all $n'$ such that $\exists e_{n,n'}$ and not $n' \ni n[p]$ do
    Compute one-step update $\Psi' = \zeta_{n,n'} \cdot \Psi$, where $\Psi = \begin{bmatrix} n[\Sigma] \\ I \end{bmatrix}$
    $\Sigma' \leftarrow {\Psi'_{11}} \cdot {\Psi'_{21}}^{-1}$
    if $tr(\Sigma') < tr(n'[\Sigma])$ then
      $n' \leftarrow \{n'[\mu], \Sigma', n[p] \cup \{n'\}\}$
      Push $n' \rightarrow Q$
    end if
  end for
end while
return $n_{goal}[p]$
```

$$= (G_t B_{t-1})(G_t^{-T} C_{t-1})^{-1} + R_t \tag{11}$$

$$= \left(\overline{D}_t \overline{E}_t^{-1}\right)^{-1} \tag{12}$$

where $\overline{D}_t = G_t^{-T} C_{t-1}$ and $\overline{E}_t = G_t B_{t-1} + R_t(G_t^{-T} C_{t-1})$ and equation (12) follows from a matrix inversion lemma. The covariance update in the information form can similarly be factored as

$$\Sigma_t = (\overline{\Sigma}_t^{-1} + H_t^T Q_t^{-1} H_t^T)^{-1} \tag{13}$$

$$= (\overline{D}_t \overline{E}_t^{-1} + M_t)^{-1} \tag{14}$$

Using the same matrix inversion lemma,

$$= \overline{E}_t(\overline{D}_t + M_t \overline{E}_t)^{-1} \tag{15}$$

$$\Rightarrow \Sigma_t = B_t C_t^{-1}, \tag{16}$$

where $B_t = \overline{E}_t = G_t B_{t-1} + R_t(G_t^{-T} C_{t-1})$ and $C_t = (\overline{D}_t + M_t \overline{E}_t) = G_t^{-T} C_{t-1} + M_t G_t B_{t-1} + R_t(G_t^{-T} C_{t-1})$. In both cases, $B_t$ and $C_t$ are linear functions of $B_{t-1}$ and $C_{t-1}$. Collecting terms, we can write the complete update step linearly, such that

$$\Psi_t = \begin{bmatrix} B \\ C \end{bmatrix}_t = \begin{bmatrix} 0 & I \\ I & M \end{bmatrix}_t \begin{bmatrix} 0 & G^{-T} \\ G & RG^{-T} \end{bmatrix}_t \begin{bmatrix} B \\ C \end{bmatrix}_{t-1}, \tag{17}$$

where $\Psi_t$ is the stacked block matrix $\left[\begin{smallmatrix} B \\ C \end{smallmatrix}\right]_t$ consisting of the covariance factors and $\zeta_t = \left[\begin{smallmatrix} W & X \\ Y & Z \end{smallmatrix}\right]_t$ is the one-step transfer function for the covariance factors for $G_t$, $H_t$, $R_t$ and $M_t$.

Notice that all of the elements in $\zeta$ are directly controllable, except for $M_t$, which is related to the measurement $z_t$ but is not a function of the measurement itself. $M_t$ represents the total amount of information that the measurement provides at time $t$ and depends on the measurement noise model $Q$ (which is usually constant) and the measurement Jacobian $H_t$. The accuracy of the EKF approximation assumes that the measurement function is locally linear, which is exactly the approximation that the Jacobian is locally constant. As a result, whenever the EKF assumptions hold, then we can assume that $M_t$ is constant and known *a priori*. This allows us to determine $\zeta_t$ for any point along a trajectory; furthermore, the linearity of the update allows us to combine multiple $\zeta_t$ matrices into a single, one-step update for the covariance along the entire length of a trajectory. Therefore, for each edge $e_{ij}$ in the BRM graph, we can pre-compute each $\zeta_t$ along the edge from the relevant Jacobians and then multiply the set of $\zeta_t$'s into a single transfer function $\zeta_{ij}$ that will propagate an initial (factored) covariance along the length of the edge in a single matrix multiply. Figure 2 shows this process of constructing the transfer function for each edge. Table 1 describes the complete Belief Roadmap algorithm. Step 2 of the algorithm contains the pre-processing phase where each edge is labeled with the transfer function $\zeta_{ij}$ that allows each covariance to be propagated in a single step.

## V. THE UNSCENTED KALMAN FILTER

The critical step of the BRM algorithm is the construction of the transfer function, which depends on terms $R_t$ and $M_t$, the projections of the process and measurement noise terms into the state space. $R_t$ and $M_t$ also represent the information lost due to motion, and the information gained due to measurements. When using the Extended Kalman filter to perform state estimation, these terms are trivial to compute. However, the EKF is not always a feasible form of Bayesian filtering, especially when linearizing the control or measurement functions leads to a poor approximation. A particularly relevant application where EKF state estimation fares poorly is localization in discrete or grid-based maps. Grid map representations contain a strong independence assumption between the grid cells, which causes measurements of neighboring grid cells to appear uncorrelated. When computing the Jacobian of a measurement with respect to a grid cell, however, the gradient of the measurement is strongly correlated with the neighboring cells. As a result, EKF localization requires high-level features such as walls and corners to be extracted for use in both computing the innovation of the measurements and computing the Jacobians. This example is one of a number of problems that can occur with a standard EKF implementation.

In order to address the limitations of linearization, alternate forms of the Bayes filter have been developed. One recent extension is the Unscented Kalman filter (UKF) [6], which uses a set of $2n+1$ deterministic samples, known as "sigma points" from an assumed Gaussian density to represent the probability density of a space of dimensionality $n$. These samples are generated according to:

$$\mathcal{X}_t^0 = \mu_{t-1}, \tag{18}$$

$$\mathcal{X}_t^i = \mu_{t-1} + \left(\sqrt{(n+\lambda)\Sigma_t}\right)^i, \quad i = 1, \ldots, n \tag{19}$$

$$\mathcal{X}_t^i = \mu_{t-1} - \left(\sqrt{(n+\lambda)\Sigma_t}\right)^i, \quad i = n{+}1, \ldots, 2n \tag{20}$$

where $\left(\sqrt{(n+\lambda)\Sigma_t}\right)^i$ is the $i$th column of the root of the matrix. Each sigma point $\mathcal{X}^i$ has an associated weight $w_m^i$ used when computing the mean, and $w_c^i$ is the weight used when computing the covariance, such that $\sum_{i=0}^{2n} w_c^i = 1$, $\sum_{i=0}^{2n} w_m^i = 1$. The weights and the $\lambda$ parameters model the width of the covariance; the mechanism for choosing these parameters can be found in [6]. The samples are propagated according to the non-linear process model such that

$$\overline{\mathcal{X}}_t^i = g(\mathcal{X}_t^i, u, 0), \tag{21}$$

generating the process mean and covariance

$$\overline{\mu}_t = \sum_{i=0}^{2n} w_m^i \overline{\mathcal{X}}_t^i \tag{22}$$

$$\overline{\Sigma}_t = \sum_{i=0}^{2n} w_c^i (\overline{\mathcal{X}}_t^i - \overline{\mu}_t)(\overline{\mathcal{X}}_t^i - \overline{\mu}_t) + R_t. \tag{23}$$

The sigma points are used to create sigma points in the measurement space, which are then transformed to generate the posterior mean and covariance $(\mu_t, \Sigma_t)$, such that

$$\overline{\mathcal{Z}}_t^i = h(\overline{\mathcal{X}}_t^i, 0) \qquad \overline{\mu}_t^z = \sum_{i=0}^{2n} w_m^i \overline{\mathcal{Z}}_t^i \tag{24}$$

$$S_t = \left(\sum_{i=0}^{2n} w_m^i (\overline{\mathcal{Z}}_t^i - \overline{\mu}_t^z)(\overline{\mathcal{Z}}_t^i - \overline{\mu}_t^z)\right) + Q_t \tag{25}$$

$$K_t = \left(\sum_{i=0}^{2n} w_c^i (\overline{\mathcal{X}}_t^i - \overline{\mu}_t)(\overline{\mathcal{Z}}_t^i - \overline{\mu}_t^z)\right) S_t^{-1} \tag{26}$$

$$\mu_t = \overline{\mu}_t + K_t(z_t - \overline{\mu}_t^z) \tag{27}$$

$$\Sigma_t = \overline{\Sigma}_t - K_t S_t K_t. \tag{28}$$

The advantage to the UKF formulation is that the process and measurement functions are not projected into the state space by a linearization; instead, the Unscented Transform computes the moments of the process and measurement distributions directly in the state space itself. As a result, the UKF eliminates the need for linearization and captures the distribution accurately up to the second order, rather than just the first order fidelity of the EKF.

Unfortunately, although the UKF provides a mechanism of efficiently tracking the posterior distribution as a Gaussian while avoiding linearization of the measurement model, the UKF no longer calculates the $M_t$ matrix which is a critical piece of the individual transfer functions $\zeta_t$. However, we can still recover $M_t$ from the UKF update directly by working in the information form and noticing that $M_t$ is the information gain due to measurement $z_t$. We can therefore combine equation (8) and equation (28),

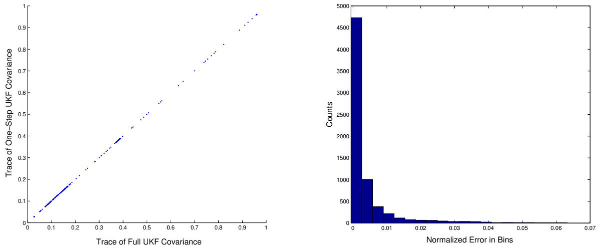
$$\Omega_t = \overline{\Omega}_t + M_t \tag{29}$$

$$\Rightarrow M_t = \Omega_t - \overline{\Omega}_t \tag{30}$$

$$= \Sigma_t^{-1} - \overline{\Sigma}_t^{-1} \tag{31}$$

$$= (\overline{\Sigma}_t - K_t S_t K_t)^{-1} - \overline{\Sigma}_t^{-1} \tag{32}$$

In order to calculate the $M$ matrix for a series of points along a trajectory, we can generate a prior covariance and compute the posterior covariance as in equation (28). Happily, the UKF covariance update does not depend on the actual measurement received, exactly like the EKF covariance update.



(a) Comparison of covariance predictions

(b) Distribution of error using constant prior approximation

Fig. 3. (a) Comparison of trace of covariance from full UKF filtering and trace of covariance from one-step transfer function using UKF $M$ matrix. (b) Distribution of ratio of error induced by computing the $M$ matrix for the one-step transfer function using a constant prior.

The UKF is still a projection of the measurement noise into the state space, but is a more accurate projection than an explicit linearization of the measurement model. By representing the belief update process with the one-step transfer function, we are approximating the non-linear UKF update. Figure 3(a) depicts the difference between covariances computed using the full UKF update and covariances computed using the one-step transfer function for a range of motions and randomized initial conditions. As expected, the one-step transfer function using the $M$ matrix calculated in equation (32) is an approximation to the UKF model but the induced error is low; the traces of the covariances are closely matched.

The UKF calculation of the information gain $M_t$ does, however, depend on the specific prior matrix $\overline{\Sigma}_t$. As a result, different choices of prior for equation (32) may result in different one-step transfer functions. Figure 3(b) shows a distribution of the ratio of the error of the one-step covariance to the full UKF covariance, where 7000 trials were performed using 100 different priors and a range of initial conditions and trajectories were used to calculate the $M$ matrix. The error induced in the one-step transfer function for using a constant $M$ is less than 2% with a significance of $p = 0.955$, indicating low sensitivity to the choice of prior over a range of operating conditions.

## VI. SAMPLING IN BELIEF SPACE

As the number of samples and the density of the graph grows, the BRM planning process will find increasingly low-covariances paths. However, as the density of the graph grows, the cost of searching the graph will also grow; searching the graph will have complexity $\mathcal{O}(b^d)$ for $b$ edges per node and path of length $d$ edges. We can reduce this complexity by minimizing the size of the graph, sampling nodes that reflect the useful part of the information space.

The optimal sampling strategy would generate samples that lie only on the optimal path to the goal; this would of course require knowing the optimal path beforehand. However, some samples are more likely to be useful than others: vehicle poses that generate measurements with high information value are much more likely to lie on the optimal path than vehicle poses that generate measurements with little information. If poses are initially sampled from $\mathcal{C}$ uniformly, but are retained according to the expected information gain from sensing at each point, the graph will still converge to one that maintains the connectivity of the free space but the graph nodes will be placed to generate sensor measurements that maximize the localization accuracy of the vehicle. We call this sampling strategy *sensor uncertainty* sampling, after the "Sensor Uncertainty Field" (SUF) defined by Takeda and Latombe [7]. The sensor uncertainty field is a mapping from location $x$ to expected information gain, $x \rightarrow \mathcal{I}(x)$, where locations with high information gain correspond to locations that generate sensor measurements that we expect to maximize the localization accuracy of the vehicle. Explicitly building this field is computationally expensive in practice; by sampling from this field in building the BRM graph, we gain the benefits of focusing the search on the states that lead to high information gain without the cost of explicitly building the sensor uncertainty field.

Information gain is calculated from the difference in entropy of the prior and posterior distributions,

$$\mathcal{I}(x) = H(p(x)) - H(p(x|z)) \tag{33}$$

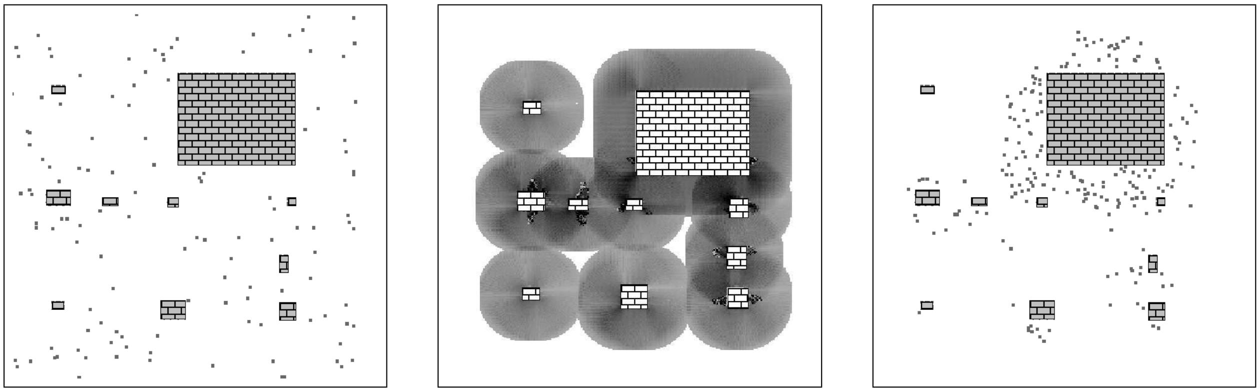
where entropy is $H(p(x)) = -\int p(x) \log p(x)$. Since our analysis (figure 3b) suggested that the measure of information gain was statistically insensitive to the choice of prior, we used a constant prior $p(x) = \Sigma_0$ such that $H(p(x)) = C$ while evaluating sensor uncertainty, and Bayes' rule to compute $p(x|z) = p(z|x) \cdot p(x)$, such that

$$\mathcal{I}(x) = C - H(p(x|z)) \tag{34}$$

where $z = \text{argmax}_z\, p(z|x)$ and $p(x|z)$ is calculated according to the UKF. For each sample, we simulate the sensor measurement and find the probability of observing the sensor measurement at each of the sigma points. The lower the probability of observation at the neighboring sigma points, the smaller the entropy of the posterior distribution, and therefore the greater the information gain. We normalize the posterior entropies so that $\mathcal{I}(x)$ lies in the range $[0, 1]$, allowing us to treat the information gain of $x$ as a probability that the sample is accepted or rejected.

Figure 4(a) shows a bird's-eye view of an example environment with limited structure and no GPS. The brick structures in figures 4(a) and (c) are the parking garage pillars and stairwell (top right). In figure 4(a), sample poses are drawn uniformly. Figure 4(b) shows the sensor uncertainty field [7] where equation (33) is evaluated at each location $(x, y)$ for fixed height and attitude. (The lack of smoothness between obstacles is an artifact of the rendering process and angular discretization.) The pixel intensity corresponds to the information gain, where darker pixels have more information. This field is shown only to illustrate the concept; computing the field for realistic domains is impractical. Finally, figure 4(c) shows samples drawn according to the sensor uncertainty. Note that the sample density is lowest far from the environmental structure where sensing provides the least amount of information.

Figure 5 shows the advantage of sampling according to the sensor uncertainty. The graph constructed using sensor uncertainty sampling consistently found a trajectory resulting in a covariance with trace 1.48 using 100 samples, whereas the uniform sampling method required 1000 samples to achieve a covariance of size 3.43. By sampling uniformly, the standard BRM requires a large and dense graph to achieve good



(a) Samples drawn uniformly (b) Sample map of sensor uncertainty field (c) Samples from sensor uncertainty field

Fig. 4. Bird's-eye view of unstructured, GPS-denied environment. The brick structures are pillars in the underground garage. (a) Distribution of samples drawn uniformly. (b) The sample map with the sensor uncertainty field. The intensity (darkness) of each pixel corresponds to the information gain available by sensing there. (c) Distribution of samples drawn according to the sensor uncertainty field.

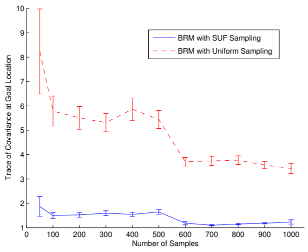


Fig. 5. Comparison of uniform vs. sensor uncertainty sampling strategies. The sensor uncertainty sampler finds accurate trajectories with considerably fewer samples than the uniform sampler.

localization accuracy. Table I shows a comparison of graph construction and planning times. The conventional PRM is clearly the fastest algorithm in both graph construction speed and path search, but as expected the localization performance is poor. The BRM with sensor uncertainty sampling requires additional time during the graph creation phase, but this time can be amortized across multiple queries, and results in measurably better paths.

| | Trace Goal Covariance | Graph Build Time (s) | Path Search Time (s) |
|---|---|---|---|
| PRM | 16.046 | 0.036 | 0.001 |
| BRM, Uniform Sampling | 4.223 | 18.920 | 0.039 |
| BRM, Sensor Uncertainty Sampling | 1.094 | 25.589 | 0.032 |

TABLE I

PERFORMANCE AND TIME COSTS OF DIFFERENT PLANNERS.

## VII. INDOOR NAVIGATION RESULTS

The BRM algorithm and sensor-maximizing sampling strategy were tested using the quadrotor helicopter, shown in Figure 1. Equipped with auto-stabilization rate gyros and accelerometers, the helicopter has on-board attitude control and thus acts as a stable sensor platform. The on-board environmental sensor is a Hokuyo URG laser sensor – a planar laser rangefinder that provides a 240° field-of-view at 10 Hz, up to an effective range of 3m. The laser is mounted in the X-Y plane of the helicopter, and we modified the laser to optically redirect 20° of its field-of-view to provide a small set of range measurements in the (downward) $z$ direction. In a single scan, the vehicle is therefore able to estimate its position, yaw orientation and altitude with respect to environmental features. Figure 6(c) shows an example scan. In practice, the measurement of the ground plane is relatively noisy, although sufficient for closed-loop altitude control.

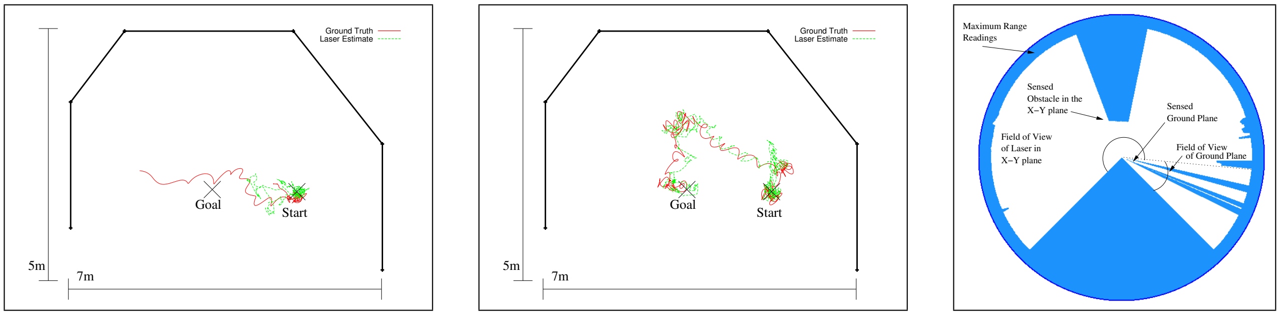
The helicopter is required to plan a path from the starting position to the end goal, shown in Figures 6(a-b), and must localize itself using the laser while executing the trajectory. We first plan a path for the helicopter using each method. We then attempt to fly the the helicopter autonomously through the environment using the planned trajectories, and determine if the helicopter is able to successfully reach the end goal by maintaining accurate localization. We compare the laser-localized state estimate against ground truth measured by a motion capture system.

Figure 6(a) shows an example trajectory generated by the traditional PRM planner, which finds a direct path from start to goal. Because this plan ignores the helicopter's need for sensor information to localize itself, the helicopter gets lost while in flight incurring a position estimation error of at least $3.6m$, falsely believing that it is still in the center of the environment when it has already flown to the left.

On the other hand, an example BRM trajectory using sensor uncertainty sampling enables the helicopter to stay well-localized incurring a position estimation error of $.17m$, as shown in Figure 6(b). The helicopter achieves this by detouring from the shortest path toward areas of high sensor information, successfully reaching its desired goal with high certainty. This demonstrates that the BRM trajectory leads to measurably more accurate performance.

## VIII. RELATED WORK

Modern approaches to planning with incomplete state information are typically based on the partially observable Markov decision process (POMDP) model or as a graph search through belief space [10]. While the POMDP provides a general framework for belief space planning, the complexity of the solution grows exponentially in the length of the policy and the number of potential observations. Numerous approximation algorithms exist to mitigate the problem of scalability [11], [12], but these techniques still face computational issues in addressing large problems. Alternatively, the Augmented MDP uses the concept of information gain by the sensor at each possible pose in freespace [13] to compute a dense policy. The Augmented MDP approach is strongly



(a) PRM trajectory, uniform sampling

(b) BRM trajectory, SU sampling

(c) Laser Perception Model

Fig. 6. (a) and (b) show an example indoor environment for comparing the performance of the BRM algorithm. The green line shows the helicopter's position estimate from the laser sensor measurements, which it uses for localization and control. The red line shows the true path of the helicopter. (a) Localization performance of helicopter executing PRM trajectory. (b) Localization performance of helicopter executing trajectory planned by the BRM using sensor uncertainty sampling. (c) The perception model of the onboard laser range-finder, including the field of view of the X-Y plane and the field of view of the ground plane.

related to the ideas in this paper, but does not scale well to more than two dimensions.

The extended Kalman filter and unscented Kalman filter have been used extensively. Ko et al. [14] use the iMote2 technology and the UKF for state estimation in aerial vehicles, and Valenti et al. [15] were the first to demonstrate reliable navigation and position estimation on quadrotor helicopters. The sympletic form (and related Hamiltonian form) of the covariance update has been reported before, most recently by Mourikis et al. [16]. Finally, laser range finding on-board helicopters is not a novel technology [17], [18], although we believe we are the first to demonstrate reliable autonomous localization and motion planning on an indoor helicopter using laser range finding.

## IX. Conclusion

In this paper, we have addressed the problem of a helicopter localizing and navigating in GPS-denied environments. The helicopter uses laser range data and an existing map to localize, but the laser has a limited field of view, causing the helicopter to lose track of its own position in certain configurations and in some parts of the environment. We showed how the Belief Roadmap algorithm [1] can be used to plan trajectories through the environment that incorporate a predictive model of sensing, allowing the planner to minimize the positional error of the helicopter at the goal using efficient graph search. The original BRM algorithm assumed an extended Kalman filter model for position estimation, and we showed how this algorithm can be extended to use the unscented Kalman filter. Furthermore, we showed that by choosing an appropriate sampling algorithm, the BRM can find better trajectories with fewer samples than using uniform sampling strategies.

## X. Acknowledgements

Ruijie He was supported by the Republic of Singapore Armed Forces, and Sam Prentice and Nicholas Roy were supported by Draper Laboratories under URPP "Robust Distributed Sensor Networks" and by the National Science Foundation Division of Computer and Network Systems under grant # 0707601 and the Division of Information and Intelligent Systems under grant # 0546467. This project was supported by the Office of the Dean, School of Engineering and the MIT Air Vehicle Research Center (MAVRC). Jan Stumpf and Daniel Gurdan provided the quadrotor helicopter and the support of Ascending Technologies. Col. Peter Young, Jonathan How, Spencer Ahrens and Brett Bethke provided additional support in the development of the vehicle. Dirk Hähnel, Karl Koscher, Jonathan Lester and Adam Rea provided considerable assistance with the iMote2 package including the MSB and LSB boards. Finally, Intel Labs Seattle and University of Washington donated the iMote hardware. The authors wish to thank this large group of people in support of this project.

## References

[1] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in linear pomdps by factoring the covariance," in *Proc. ISRR*, 2007.

[2] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[3] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[4] D. Gurdan, J. Stumpf, M. Achtelik, K. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 khz," in *Proc. ICRA*, 2007.

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

[6] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. ACC*, 1995.

[7] H. Takeda and J. Latombe, "Sensory uncertainty field for mobile robot navigation," *Proc. ICRA*, 1992.

[8] T. Lozano-Perez., "Spatial planning: A configuration space approach," *IEEE Trans. on Computers*, vol. C-32, no. 2, 1983.

[9] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[10] B. Bonet and H. Geffner, "Planning with incomplete information as heuristic search in belief space," *Proc. AIPS*, pp. 52–61, 2000.

[11] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," *Proc. IJCAI*, 2003.

[12] T. Smith and R. Simmons, "Heuristic search value iteration for POMDPs," *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 520–527, 2004.

[13] N. Roy and S. Thrun, "Coastal navigation with mobile robots," in *Advances in Neural Processing Systems 12*, vol. 12, 1999.

[14] J. Ko, D. Klein, D. Fox, and D. Hähnel, "GP-UKF: unscented kalman filters with gaussian process prediction and observation models," in *Proc. (IROS)*, 2007.

[15] M. Valenti, B. Bethke, G. Fiore, J. P. How, and E. Feron, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," in *Proc. AIAA GN&C.*, 2006.

[16] A. Mourikis and S. Roumeliotis, "Optimal sensor scheduling for resource constrained localization of mobile robot formations," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 917–931, October 2006.

[17] S. Thrun, M. Diel, and D. Hähnel, "Scan alignment and 3d surface modeling with a helicopter platform," in *Proc. FSR*, 2003.

[18] L. O. Mejias, S. Saripalli, P. Cervera, and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal of Field Robotics*, vol. 23, no. 3, pp. 185–199, 2006.