Sampling-Based Min-Max Uncertainty Path Planning

Brendan Englot Tixiao Shan Shaunak D. Bopardikar Alberto Speranzon

***Abstract*— We propose a new sampling-based path planning algorithm, the Min-Max Rapidly Exploring Random Tree (MM-RRT*), for robot path planning under localization uncertainty. The projected growth of error in a robot's state estimate is curbed by minimizing the *maximum* state estimate uncertainty encountered on a path. The algorithm builds and maintains a tree that is shared in state space and belief space, with a single belief per robot state. Due to the fact that many states will share the same maximum uncertainty, resulting from a shared parent node, the algorithm uses secondary objective functions to break ties among neighboring nodes with identical maximum uncertainty. The algorithm offers a compelling alternative to sampling-based algorithms with additive cost representations of uncertainty, which will penalize high-precision navigation routes that are longer in duration.**

I. INTRODUCTION

Sampling-based algorithms have been leveraged in recent years to solve, both heuristically and optimally, variants of robot path planning that pose challenging constraints and objectives alongside the standard requirement of collision avoidance with minimum distance, time, or energy. Sampling-based sensor coverage path planning has been solved heuristically for large-scale 3D coverage problems [1] and optimally for small-scale boundary coverage problems [2]. The projection of samples onto constraint manifolds has been used to find feasible solutions to complex manipulation tasks [3]. Rejection sampling has been used to solve optimal path planning problems with secondary cost criteria, such as maintaining maximum clearance from neighboring obstacles [4]. Maximally safe driving has been achieved using a cost function that quantifies the violations of a collection of safety rules [5]. Information-gathering tasks have been planned optimally under a limited energy or time budget via Sampling-Based Robotic Information Gathering (RIG) [6]. Approaches seeking feasible solutions have typically leveraged rapidly-exploring random trees (RRTs) [7] or probabilistic roadmaps (PRMs) [8], and approaches seeking optimal solutions have leveraged the asymptotic optimality properties of RRT*, rapidly-exploring random graphs (RRGs), or PRM* [9].

One of the most challenging variants of this type involves planning under uncertainty, which may seek to find feasible or asymptotically optimal plans in the presence of probabilistic actions, measurements, and/or environment maps. Sampling-based approaches to planning under uncertainty have typically assumed that belief spaces are Gaussian. The objective function itself may address the uncertainty associated with a localization process, as in the case of the belief roadmap (BRM) [10] and robust belief roadmap (RBRM) [11]. Standard minimum-distance or minimum-energy objective functions may also be used in combination with constraints related to localization uncertainty, as in the case of the rapidly-exploring random belief tree (RRBT) [12], bounded-uncertainty RRT* (BU-RRT*) [13], Box-RRT [14], and an expanded adaptation of the PRM [15]. Chance-constrained RRT* (CC-RRT*) has combined these concepts into a sampling-based algorithm that enforces constraints on a robot's collision probability while also penalizing collision risk in the objective [16]. Such problems can also be formulated as stochastic optimal control problems, solved by feedback motion planning, as with linear quadratic Gaussian motion planning (LQG-MP) [17] and the Feedback Controller-Based Information-State Roadmap (FIRM) [18], or by approximating optimal policies for Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs), as with the Stochastic Motion Roadmap (SMR) [19], incremental-MDP (iMDP) [20], and Adaptive Belief Tree (ABT) [21].

Despite these successes, path planning that addresses localization uncertainty in the objective function remains challenging and elusive in several ways. This approach is desirable in applications where the user does not know how to properly tune constraints governing uncertainty, and feasible solutions that curb estimation error are needed quickly. Prior work has sought to minimize the terminal, max, and sum of uncertainty over a path, though to date this has been achieved using feasible planning formulations only, or optimal formulations that curb uncertainty using constraints. Goal-state localization uncertainty has been minimized heuristically in [10] and [11], and min-max uncertainty has been addressed heuristically in the objective by [10] and in the constraints by [15]. Cost functions expressing a sum of uncertainties have been used to find locally optimal solutions in stochastic optimal control contexts [17], [18]. Constraints on collision probability have been enforced in [12], [13], and [16]. Related works that have achieved *asymptotically optimal* solutions, which approach global optimality in the limit, have done so by utilizing additive, cumulative representations of safety [5], information gain [6], and collision risk [16].

These asymptotically optimal approaches demonstrate compelling results for a variety of applications, but a *cumulative* representation of risk, safety, or uncertainty will inherently favor short paths over long ones. However, it may be desirable to follow a longer path if it offers superior

B. Englot and T. Shan are with the Department of Mechanical Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken NJ 07030 USA, {BEnglot, TShan3}@stevens.edu. S.D. Bopardikar is with United Technologies Research Center (UTRC), BopardSD@utrc.utc.com, and A. Speranzon was with UTRC at the time the work was performed. He is now with Honeywell Aerospace - Advanced Technology alberto.speranzon@honeywell.com.

management of uncertainty. A path of minimum mean uncertainty would not have this problem, but it would challenge our need for a cost metric (per the requirements on asymptotic optimality for sampling-based algorithms) that increases monotonically over the duration of a path. Most practical robot navigation problems will entail increases and decreases in uncertainty at various times. As a compromise, we propose taking a step in this direction by formulating a sampling-based algorithm for min-max planning under uncertainty.

Curbing the *maximum* uncertainty encountered over the duration of a path, as addressed previously in [10] and [15], does not guarantee a path of minimum mean uncertainty, but it no longer penalizes a highly safe path for its length. This approach also offers the possibility of a metric that is monotonically non-decreasing over the duration of a path. We propose a min-max variant of the RRT* algorithm we term the Min-Max Rapidly Exploring Random Tree, or MM-RRT*. In addition to utilizing a min-max uncertainty metric, which will be detailed in the sections to follow, the algorithm builds and maintains a tree that is shared in state space and belief space, with a singe belief per robot state. The algorithm also uses secondary objective functions to break ties among neighboring nodes with identical maximum uncertainty. Section II will define the problem of interest, and Section III will describe the proposed algorithm. Section IV will present an analysis of the algorithm, in which we describe its local optimality and reflect on the challenges of establishing global optimality. Section V will show computational results that demonstrate the algorithm's performance against an approach that minimizes a cumulative representation of uncertainty.

II. PROBLEM DEFINITION

A. Path Planning

Let $\mathcal{C}$ be a robot's configuration space. We will assume that a configuration $x \in \mathcal{C}$ describes the pose of the robot. $\mathcal{C}_{obst} \subset \mathcal{C}$ denotes the subset of configurations in $\mathcal{C}$ that are in collision with an obstacle, and $\mathcal{C}_{free} \subseteq (\mathcal{C} \setminus \mathcal{C}_{obst})$ denotes the configurations that are free of collision. We will assume that given an initial configuration $x_{init} \in \mathcal{C}_{free}$, the robot must reach a goal region $X_{goal} \subset \mathcal{C}_{free}$. Let a *path* be a continuous function $\sigma : [0, T] \to \mathcal{C}$ of finite length, traversed in finite time T, for which we assume a series of control inputs must be applied to the robot to achieve this path in $\mathcal{C}$. Let Σ be the set of all paths σ in a given configuration space. A path is collision-free if $\sigma \in \mathcal{C}_{free}$ for all arguments, and a collision-free path is *feasible* if $\sigma(0) = x_{init}$ and $\sigma(T) \in X_{goal}$. A *cost function* $c : \Sigma \to \mathbb{R}_+$ returns a strictly positive cost for all non-trivial collision-free paths.

B. State Estimation and Uncertainty

We consider a robot whose state evolves as a nonlinear, discrete-time dynamical system:

$$\mathrm{x}_{k+1} = \mathbf{f}(\mathrm{x}_k, \mathrm{w}_k) \tag{1}$$

$$\mathrm{y}_k = \mathbf{h}_k(\mathrm{x}_k, \mathrm{v}_k) \tag{2}$$

where $\mathrm{x}_k = \begin{bmatrix} \dot{x}_k & x_k \end{bmatrix}$ is the state of the system at time k, and describes the position and velocity of the robot in each degree of freedom. The robot's measurement at time k is y_k, and is supplied by a suite of sensors whose availability will vary as a function of the robot's location in the environment. At time k, the robot is influenced by independent zero-mean Gaussian process noise w_k, with covariance $\mathbf{Q}_k$, and sensor noise v_k, with covariance $\mathbf{R}_k$.

The robot's state may be estimated using an extended Kalman filter (EKF) [22], which is propagated as follows:

$$\mathbf{P}_{k+1}^{-1} = (\mathbf{F}_k \mathbf{P}_k \mathbf{F}_k' + \mathbf{Q}_k)^{-1} + \mathbf{H}_{k+1}' \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1} \tag{3}$$

$$\begin{aligned} \hat{\mathrm{x}}_{k+1} = \mathbf{P}_{k+1} \big(& (\mathbf{F}_k \mathbf{P}_k \mathbf{F}_k' + \mathbf{Q}_k)^{-1} \mathbf{F}_k \hat{\mathrm{x}}_k + \\ & \mathbf{H}_{k+1}' \mathbf{R}_{k+1}^{-1} (y_{k+1} - \mathbf{H}_{k+1} \hat{\mathrm{x}}_k) \big) \end{aligned} \tag{4}$$

where $\hat{\mathrm{x}}_k$ is the state estimate, $\mathbf{P}_k$ is the robot's estimation error covariance at time k, and $\mathbf{F}_k$ and $\mathbf{H}_k$ represent the Jacobians of $\mathbf{f}$ and $\mathbf{h_k}$ about $(\hat{\mathrm{x}}_k, 0)$ and $(f(\hat{\mathrm{x}}_k), 0)$, respectively. To penalize growth of the error covariance in the course of path planning, estimation error is ideally represented as a scalar cost metric. This has been achieved previously in the context of sampling-based planning algorithms using $tr(\mathbf{P})$, the error covariance trace [10], and $\overline{\lambda}(\mathbf{P})$, the maximum eigenvalue of $\mathbf{P}$ [11].

However, this latter work also proposes an upper bound on $\overline{\lambda}(\mathbf{P})$, $\ell \geq \overline{\lambda}(\mathbf{P})$, which, unlike the above metrics, admits optimal substructure when applied to the breadth-first search of a graph in belief space. A recursive update for ℓ results from the following inequality, which is derived from (3):

$$\overline{\lambda}(\mathbf{P}_{k+1}) \leq \frac{\overline{\lambda}(\mathbf{F}_k \mathbf{F}_k') \overline{\lambda}(\mathbf{P}_k) + \overline{\lambda}(\mathbf{Q}_k)}{(\underline{\lambda}(\mathbf{H}_k' \mathbf{R}_k^{-1} \mathbf{H}_k))(\overline{\lambda}(\mathbf{F}_k \mathbf{F}_k') \overline{\lambda}(\mathbf{P}_k) + \overline{\lambda}(\mathbf{Q}_k)) + 1} \tag{5}$$

where the operator $\underline{\lambda}(\ )$ represents the minimum eigenvalue of a positive definite matrix. Due to this inequality, the upper bound $\ell \geq \overline{\lambda}(\mathbf{P})$ may be propagated according to the following update rule, initialized using $\ell_0 = \overline{\lambda}(\mathbf{P}_0)$:

$$\ell_{k+1} = \frac{\overline{\lambda}(\mathbf{F}_k \mathbf{F}_k') \ell_k + \overline{\lambda}(\mathbf{Q}_k)}{(\underline{\lambda}(\mathbf{H}_k' \mathbf{R}_k^{-1} \mathbf{H}_k))(\overline{\lambda}(\mathbf{F}_k \mathbf{F}_k') \ell_k + \overline{\lambda}(\mathbf{Q}_k)) + 1}. \tag{6}$$

Using ℓ as an uncertainty metric offers a small improvement in computational efficiency over other metrics, since the propagation of uncertainty in (6) uses extreme eigenvalues [23], rather than the repeated computation of $\mathbf{P^{-1}}$ required in (3). We thus adopt ℓ as our scalar representation of uncertainty, although the proposed MM-RRT* algorithm may be used with any scalar uncertainty metric.

C. Robot Motion Assumptions

We assume a robot moves through $\mathcal{C}_{free}$ along paths obtained from a directed graph $G(V, E)$, with vertices V and edges E. Specifically, all graphs considered are *trees*. We make several assumptions regarding the consistency, stability, and synchrony of the estimation, motion, and measurement processes, adapted from [15]:

1) The filter produces a consistent estimate, and the corresponding error covariance is a measure of the precision of the filter.

2) There exists a low-level controller that ensures the robot follows the planned, nominal trajectory between any two vertices of the graph.
3) The measurements from all sensors are synchronized.

The first assumption implies that our motion and measurement models accurately represent the behavior of the robot. The second assumption implies that the robot is capable of recovering from process noise disturbances. The third assumption implies that measurements from all sensors will arrive in the order that their observations occur, such that a planning algorithm may predict their arrival. Our goal is to leverage our knowledge of the environment, motion and sensor models to prevent the growth of uncertainty to an extent that the above assumptions may be violated.

In general, a robot's state and uncertainty comprise a *belief*, (x, ℓ). A vertex of an arbitrary graph may represent many beliefs achieved over different motion and measurement histories. In a tree, however, there is only one path from the root x_{init} to each node, and if all paths are initialized from the root with the same uncertainty $\ell_{init} = \overline{\lambda}(\mathbf{P}_{init})$, there exists only one belief per node. Hereafter, when referring to a configuration x_k, we imply that x_k has an associated error covariance $\mathbf{P}_k$, an eigenvalue bound ℓ_k, a dynamical state x_k, and state estimate $\hat{\mathrm{x}}_k$ that are uniquely identifed by x_k in concert with the initial uncertainty at the root of the tree.

D. Cost Function

To aid in defining our cost function, we first define $G(\ell_0)|_{\sigma(0)}^{\sigma(T)}$, a function that represents the composition of evaluations of (6) over a path from $\sigma(0)$ to $\sigma(T)$, where ℓ_0 represents the uncertainty at the start of the path. Every discrete-time instant along the path represents a measurement update of the EKF. Our proposed cost function, $c_{max}(\sigma, \ell_0)$, then scores a path by evaluating G from the start of the path, where the intial uncertainty is ℓ_0, to every measurement update along the path, returning the maximum.

$$G(\ell_0)|_{\sigma(0)}^{\sigma(T)} := (\ell_T \circ \ell_{T-1} \circ ... \circ \ell_1)(\ell_0) \tag{7}$$

$$c_{max}(\sigma, \ell_0) := max\{G(\ell_0)|_{\sigma(0)}^{\sigma(T)}, G(\ell_0)|_{\sigma(0)}^{\sigma(T-1)}, ..., G(\ell_0)|_{\sigma(0)}^{\sigma(2)}, G(\ell_0)|_{\sigma(0)}^{\sigma(1)}\} \tag{8}$$

We will hereafter abbreviate $c_{max}(\sigma, \ell_0)$ using the notation $\overline{\ell}\,|_{\sigma(0)}^{\sigma(T)}$ to indicate the start and end states of the specific path evaluated, from which the maximum uncertainty $\overline{\ell}$ is returned. The algorithm we propose below builds a tree that minimizes $\overline{\ell}$ from root node x_{init}, initialized with uncertainty $\ell_{init} = \overline{\lambda}(\mathbf{P}_{init})$, to all destinations in $\mathcal{C}_{free}$.

In addition, we define a secondary cost and tertiary cost, which are integral cost functions, as follows:

$$c_{second}(\sigma) := \int_{\sigma(0)}^{\sigma(T)} Second(\sigma(s))ds \tag{9}$$

$$c_{third}(\sigma) := \int_{\sigma(0)}^{\sigma(T)} Third(\sigma(s))ds \tag{10}$$

where $Second(x)$ and $Third(x)$ are generalized functions of a single robot state. Each subsequent cost function in our hierarchy will be used for breaking ties that occur in the former. In the examples explored in our computational results below, all workspaces are comprised of two types of spatial regions: regions in which the robot's pose is not directly observable, in which the pose uncertainty grows under dead reckoning, and regions in which the robot's pose is directly observable (via GPS, the measurement of environmental features, or similar), where error growth is curbed. In these examples, $c_{second}(\sigma)$ returns the distance traveled by a path σ in all regions that lack observability of the robot pose, and $c_{third}(\sigma)$ returns the distance traveled by a path in all regions that offer such observability. We will alternately express $c_{second}(\sigma)$ using the notation $SI(\sigma(0), \sigma(T))$ and $c_{third}(\sigma)$ using the notation $TI(\sigma(0), \sigma(T))$ to indicate the limits of each integral.

III. ALGORITHM DESCRIPTION

Algorithm 1: MM-RRT*

Input: $V \leftarrow \{x_{init}\}; E \leftarrow \emptyset;$
1 **for** $i = 1, ..., n$ **do**
2 $x_{rand} \leftarrow Rand();\ x_{nearest} \leftarrow Nearest(V, x_{rand});$
3 $x_{new} \leftarrow Steer(x_{nearest}, x_{rand});$
4 **if** $ObsFree(x_{nearest}, x_{new})$ **then**
5 $X_{near} \leftarrow Near(V, E, x_{new}); V \leftarrow V \cup \{x_{new}\};$
6 $X_{min} \leftarrow \{x_{nearest}\};\ c_{min} \leftarrow MaxBnd(x_{nearest}, x_{new});$
for $x_{near} \in X_{near}$ **do**
7 **if** $ObsFree(x_{near}, x_{new})$ **then**
8 **if** $MaxBnd(x_{near}, x_{new}) < c_{min}$ **then**
9 $X_{min} \leftarrow \{x_{near}\};$
10 $c_{min} \leftarrow MaxBnd(x_{near}, x_{new});$
11 **else if** $MaxBnd(x_{near}, x_{new}) = c_{min}$ **then**
12 $X_{min} \leftarrow X_{min} \cup \{x_{near}\};$
13 $x_{min} \leftarrow Tiebreak(X_{min}, x_{new});$
14 $E \leftarrow E \cup \{(x_{min}, x_{new})\};$
15 **for** $x_{near} \in X_{near} \setminus \{x_{min}\}$ **do**
16 $replace \leftarrow false;$
17 **if** $ObsFree(x_{new}, x_{near})$ **then**
18 $c_{near} \leftarrow Cost(x_{near});$
19 $c_{new} \leftarrow MaxBnd(x_{new}, x_{near});$
20 **if** $c_{new} < c_{near}$ **then** $replace \leftarrow true;$;
21 **else if** $c_{new} = c_{near}$ **then**
22 $X_{pair} \leftarrow \{x_{parent}, x_{new}\};$
23 **if** $x_{new} = Tiebreak(X_{pair}, x_{near})$ **then**
24 $replace \leftarrow true;$
25 **if** $replace = true$ **then**
26 $E \leftarrow (E \setminus \{(x_{parent}, x_{near})\}) \cup \{(x_{new}, x_{near})\};$
$X_{children} \leftarrow Children(x_{near});$
27 $RecursivePropagate(x_{near}, X_{children});$
28 **return** $G = (V, E)$

MM-RRT* is outlined in Algorithm 1. The algorithm proceeds similarly to RRT*, beginning each iteration by drawing a random sample (line 3), steering toward the random sample from the nearest neighboring node in the existing tree (line 4), and subsequently searching for the

Algorithm 2: $Tiebreak(X_{tied}, x_{child})$

```
1  c_second ← +∞; c_third ← +∞;
2  for x_i ← X_tied do
3      if SI(x_init, x_i) + SI(x_i, x_child) < c_second then
4          c_second ← SI(x_init, x_i) + SI(x_i, x_child);
5          x_min ← x_i;
6      else if SI(x_init, x_i) + SI(x_i, x_child) = c_second then
7          if TI(x_init, x_i) + TI(x_i, x_child) < c_third then
8              c_third ← TI(x_init, x_i) + TI(x_i, x_child);
9              x_min ← x_i;
10 return x_min
```

best parent for the newly-generated candidate node by examining all neighbors within a ball radius that shrinks logarithmically according to $\gamma(log(card(V))/card(V))$, where $\gamma > 2(1+1/d)^{1/d}(\mu(\mathcal{C}_{free})/\mu^d_{ball})^{1/d}$ (lines 9-17). The volume of the d-dimensional unit ball in Euclidean space is indicated by μ^d_{ball}. The function $MaxBnd(x_{near}, x_{new})$ is used to evaluate the maximum uncertainty encountered while traveling to x_{new} from x_{init} on a path that includes x_{near}. To evaluate $MaxBnd(x_{near}, x_{new})$, we need only propagate (6) from x_{near} to x_{new}, and the maximum error covariance eigenvalue bound encountered, $\overline{\ell}\,|^{x_{new}}_{x_{near}}$, is compared to $\overline{\ell}\,|^{x_{near}}_{x_{init}}$, the stored maximum value associated with parent node x_{near}. The larger of the two values is returned by $MaxBnd(x_{near}, x_{new})$, and the resulting min-max after a survey of X_{near} is adopted as $\overline{\ell}\,|^{x_{new}}_{x_{init}}$, and stored as the cost associated with x_{new}. Because belief propagation is occuring on a tree, we will only have a single belief per node and there is no need to distinguish between geometric states x_i and their associated beliefs.

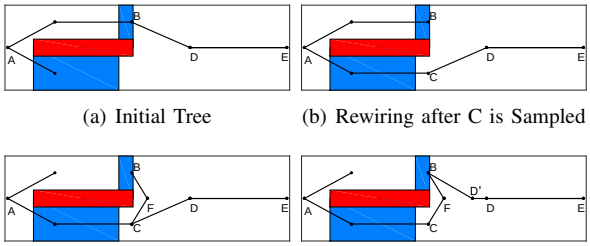
The min-max objective function will frequently result in ties among candidate nodes in X_{near}. This occurs when a maximum uncertainty value is derived from a common ancestor node and shared among multiple descendants. MM-RRT* implements tie-breaking using secondary and tertiary objective functions to impose an ordering scheme among nodes of identical maximum uncertainty. This is indicated by the use of the $Tiebreak()$ function in Algorithm 2. In the implementation of MM-RRT* discussed in this paper, c_{second}, the cumulative distance traveled in the regions where no GPS measurements are available, and c_{third}, the cumulative distance traveled in the regions where GPS measurements are available, are used as the secondary and tertiary objectives. Consequently, if two nodes, x_1 and x_2, are equally suitable parents of x_{new} by the min-max objective, the node offering the minimum c_{second} to x_{new} will be selected as the parent. If x_1 and x_2 offer the same primary *and* secondary cost to x_{new}, then the node offering the minimum c_{third} to x_{new} will be selected as the parent.

After a suitable parent is identified for x_{new}, both RRT* and MM-RRT* next attempt to improve the cost of the other nodes in X_{near} by considering the replacement of their parents by x_{new}, known as rewiring (lines 18-34). This is achieved by evaluating $MaxBnd(x_{new}, x_{near})$ and comparing the result with the existing value of $\overline{\ell}\,|^{x_{near}}_{x_{init}}$. If x_{new} offers a lower max uncertainty cost, it will replace the parent of x_{near}. If x_{new} ties with the existing parent, then the $Tiebrk()$ function is once again used to break the tie and choose a parent for x_{near}.

If the rewiring procedure succeeds in replacing the parent of x_{near}, the costs of all descendants of x_{near} must be updated. Unlike the standard RRT* algorithm, the min-max costs are not additive from node to node and the bound $\overline{\ell}$ must be propagated anew from x_{near} to all of its descendants. This is performed by the $RecursivePropagate()$ function, which recursively propagates the bound from the parent to all children until all branches of the tree that descend from x_{near} have been updated. The impact of this procedure on the algorithm's asymptotic optimality and computational complexity is discussed in the following section.

IV. PROPERTIES OF THE ALGORITHM

A. Monotonicity of the Min-Max Cost Metric



(a) Initial Tree (b) Rewiring after C is Sampled

(c) Rewiring after F is Sampled (d) Rewiring after D' is Sampled

Fig. 1. This example shows an instance in which a child node's (E) max uncertainty will first increase, then subsequently decrease, after a series of rewiring operations that are favorable for E's parent node, D. At top, path A-B-D-E will be rewired to A-C-D-E upon the sampling of node C, and at bottom, path A-C-D-E will be rewired to A-C-F-B-D'-D-E upon the sampling of nodes F, then D'. Blue regions contain GPS availability, with an obstacle depicted in red.

Unlike the standard RRT* algorithm, in MM-RRT*, it is possible for a rewiring operation, despite lowering the cost at a parent node, to cause the cost at one or more child nodes downstream from the parent to increase. A situation in which this can arise is illustrated in Figure 1, where an obstacle is depicted in red, regions of GPS availability are depicted in blue, and in all other regions, the robot, tasked with planning a path from A to E, must navigate using odometry. Upon the sampling of node C, a rewiring operation is performed, in which B, the original parent of node D, is replaced with C, a new parent offering lower $\overline{\ell}$, eliminating edge B-D. Despite the fact that a lower max uncertainty exists at node C, it also has a higher current uncertainty than node B. As a result, further downstream along the path from A to E, the current uncertainty at node C will eventually give rise to a new value of max uncertainty, which causes both the current uncertainty and max uncertainty at node E to increase. However, this increase in max uncertainty at E is eventually undone. When node F is added to the tree, it becomes the parent node of B through rewiring. Subsequently, node D' is added to the tree, and this triggers the rewiring of D, eliminates edge C-D, and

reduces both the current uncertainty and the max uncertainty at node E to the lowest levels yet.

We first address why this occurs, given that the min-max cost function appears to satisfy the two most commonly articulated properties for an admissible RRT* cost metric: *monotonicity*, as the cost along a path is monotonic non-decreasing, and *boundedness*, as the cost cannot instantaneously take on arbitrarily large values. However, a property that the min-max metric lacks is monotonicity with respect to the *initial value* of the cost metric at the beginning of a path. Consider the path from D to E. Although, at top of Figure 1, parent node C offers a lower min-max cost going into node D than its previous parent B, it results in higher min-max cost coming out at node E. This second type of monotonicity is trivial in the case of most cost metrics, and as such, it has not been articulated in prior analysis of RRT*. However, this property does not hold for MM-RRT* in examples where a path's uncertainty undergoes a reset, such as by entering a GPS zone like those of Figure 1. Rewiring a node's parent to improve the child node's uncertainty does not necessarily improve the uncertainty of all of that child's descendants.

Despite this, as $\mathcal{C}$ is populated with more nodes over the course of the MM-RRT* algorithm, these occasional instances of "bad rewiring" will be mitigated through subsequent rewiring of child nodes like E that are adversely affected by earlier events, as illustrated at bottom of Figure 1 after the new nodes F and D' are added to the graph. In our computational results below, we provide empirical support that the limiting behavior of the algorithm is for path costs to converge asymptotically, toward optimal values. However, we cannot claim that these asymptotes represent globally optimal solutions. Unlike the standard RRT* algorithm, a single instance of rewiring will not cause all affected nodes to improve in cost simultaneously; there will be fluctuations in cost with a net decrease that tends toward an asymptote. In future work, we hope to analyze the global optimality of tree-based algorithms that can undergo occasional setbacks en route to optimal solutions.

B. Computational Complexity

We now comment on the computational complexity of MM-RRT*. For a typical sampling-based path planning algorithm that does not propagate beliefs over the graph, the complexity of the $Near()$ operation and the number of calls made to $ObsFree()$ are typically of greatest concern, as these are expensive operations for large graphs that require many geometric primitives to represent the surrounding obstacles. In the case of RRT*, in a single iteration $\mathcal{O}(log(n))$ is the worst-case complexity of finding nearest neighbors, and $\mathcal{O}(log(n))$ total calls are made to $ObsFree()$. The time complexity of building a tree over n iterations is hence considered to be $\mathcal{O}(n\ log(n))$.

This complexity also holds for MM-RRT* with respect to these two operations. However, in the case of MM-RRT* we are also concerned about the number times Equation (6) will be propagated across an edge of the graph. This will happen $\mathcal{O}(n\ log(n))$ times if we count only the belief propagations that are used to evaluate candidate edges. These are the "expensive" belief propagation operations, occurring the first time an edge cost is computed. In addition, there are another worst-case $\mathcal{O}(n^2)$ propagations that will occur due to the calls made to $RecursivePropagate()$, in which new beliefs are propagated over pre-existing graph edges. These propagations may occur at substantially lower cost, as the min and max eigenvalue terms in (6) will have already been evaluated over an edge. Consequently, MM-RRT* requires $\mathcal{O}(n\ log(n))$ non-trivial belief propagation operations over new and candidate graph edges, which is the same worst-case number of candidate edge evaluations required by RRT*.

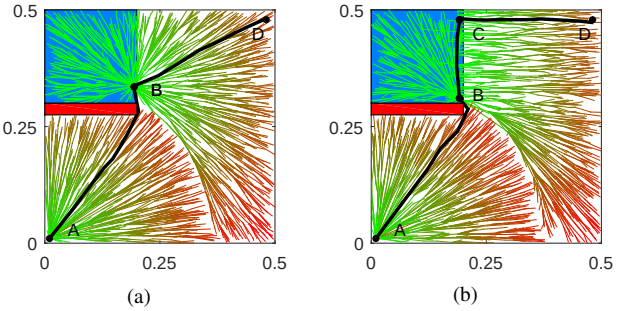
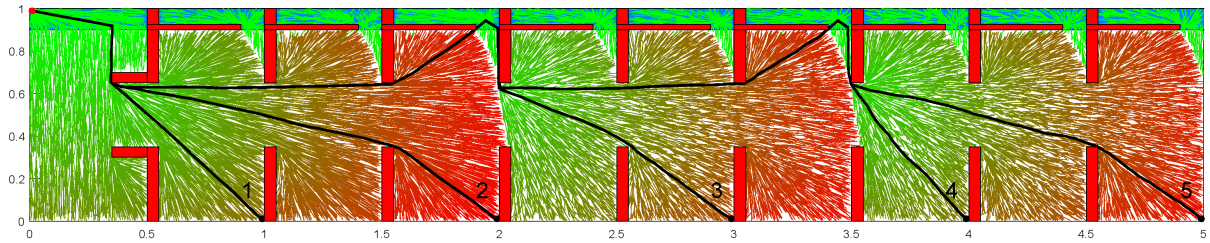


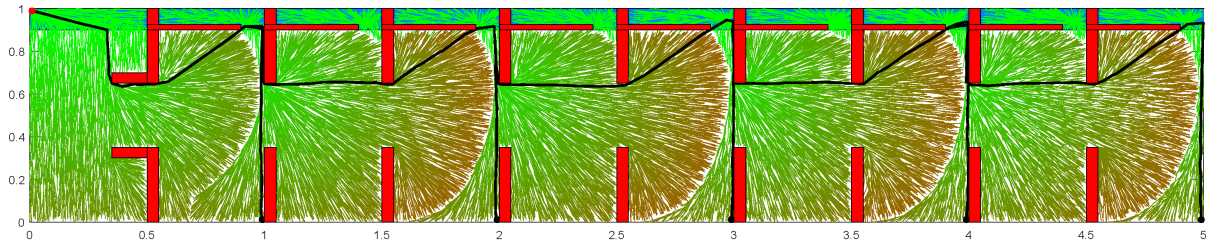
Fig. 2. When only one secondary cost function $Dist()$ is used to penalize distance traveled, the resulting tree and a path connecting start and goal nodes are shown in (a). When a secondary cost function $SI()$ and a tertiary cost function $TI()$ are used together, the resulting tree and respective path are shown in (b). This approach curbs the growth of uncertainty at D. The blue region indicates GPS availability, with an obstacle in red. Edges of the trees are plotted in color varying from green to red to indicate the current value of the eigenvalue bound ℓ (with higher ℓ in red) along each path.

C. On the Use of Hierarchical Cost Functions

Secondary and tertiary cost functions are used to curb additional growth of uncertainty when paths appear otherwise identical in primary cost. We can express the total accumulated distance along a path as $Dist(\sigma(0), \sigma(T)) = SI(\sigma(0), \sigma(T)) + TI(\sigma(0), \sigma(T))$ using these cost functions. The benefit of this approach is illustrated in Figure 2, in which there are four key tree nodes highlighted: A, the start node, B, the first node arriving at the GPS region, C, a node in the GPS zone which has the same y-axis value as node D, and D, the goal node. At node B, the path of interest enters the GPS region, and node B and its descendents will each have substantially lower localization uncertainty than the max uncertainty encountered a few steps earlier. As a result, ties exist among the children of node B, as they share the same primary cost. If only a secondary cost function $Dist()$ is applied here, the path to node D will not travel through node C, as a direct route from B to D will offer the shortest distance (shown in Figure 2(a)). However, we care more about the value of ℓ over the path than the path's length. When a tertiary cost function $TI()$ is introduced, the path, from B to C and finally to D, offers the shortest trajectory exposed to the regions that have no GPS access (shown in Figure 2(b)). In other words, we have $SI(B, C) + SI(C, D) < SI(B, D)$ as $SI(B, C)$ is equal to zero. All nodes in the GPS zone of Figure 2 share the same primary cost, as well as the same secondary cost from node

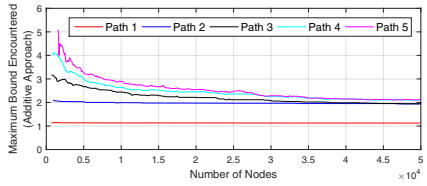


(a) Additive Approach

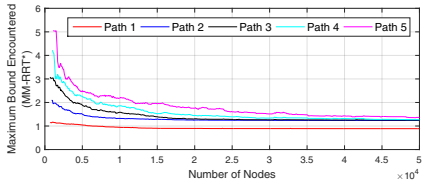


(b) MM-RRT*

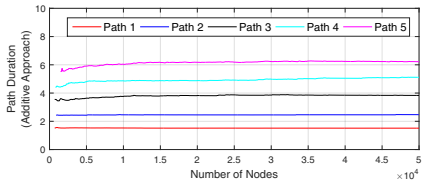
Fig. 3. Trees generated by two competing approaches for planning under uncertainty using the same sequence of random samples. Obstacles are indicated in red and GPS zones (at the upper boundary of the domain) are rendered in blue. Edges of the trees are plotted in color varying from green to red to indicate the current value of the eigenvalue bound ℓ (with higher ℓ in red) along each path.



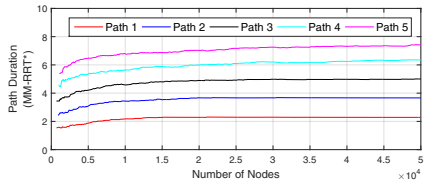
(a) Maximum Bound Encountered, Additive Approach



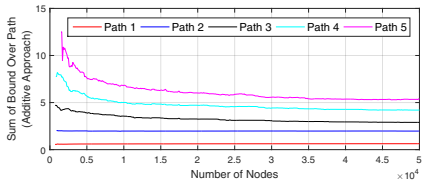
(b) Maximum Bound Encountered, MM-RRT*



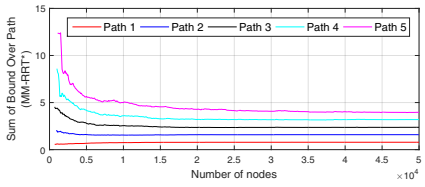
(c) Path Duration, Additive Approach



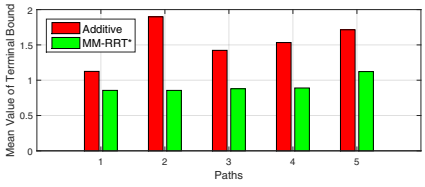
(d) Path Duration, MM-RRT*



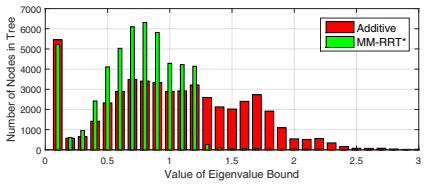
(e) Sum of Bound Over Path, Additive Approach



(f) Sum of Bound Over Path, MM-RRT*



(g) Mean Value of Terminal Bound



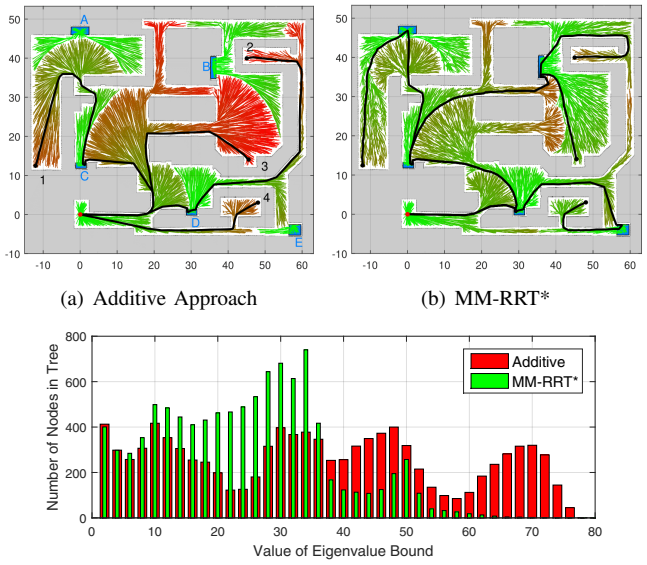
(h) Eigenvalue Bound at all Nodes in the Tree

Fig. 4. (a) - (f) show the evolution of the maximum eigenvalue bound encountered, path duration and the sum of the bound as a function of the number of nodes in the tree, over each of the 5 example paths from Figure 3, over 50 trials of our two competing approaches. (g) shows the mean terminal bound of each example path for the two approaches, and (h) shows a histogram of the bound's value at all nodes in the tree, averaged over 50 trials.

B. Therefore, ties exist between these nodes. $TI()$ can be used to break these ties, resulting in the selection of node C as a parent node.

V. COMPUTATIONAL RESULTS

We now describe a computational study performed to explore the effectiveness of planning under uncertainty using MM-RRT*. Two-dimensional robot workspaces are utilized to aid the visualization of the algorithm's performance, but it is also extensible to higher-dimensional systems. In our first example, we assume that a robot is capable of translation in two degrees of freedom, moves at constant speed, and is restricted to motion within the domain depicted in Figure 3. Throughout the domain the robot receives odometry measurements in both translational degrees of freedom. There are ten zones, illustrated in blue at the upper boundary of the domain, where the robot can receive GPS measurements. The robot's position estimate will drift and the error covariance terms associated with position will grow



(a) Additive Approach (b) MM-RRT*

(c) Eigenvalue Bound at all Nodes in the Tree

Fig. 5. A path planning example derived from a real-world building floorplan. Plots (a) and (b) show trees of 10,000 nodes each, generated by our two competing approaches using the same sequence of random samples. A histogram showing the value of the eigenvalue bound across all nodes in the tree, averaged over 150 trials, is shown in (c) for each approach.

TABLE I
MM-RRT* VS. ADDITIVE APPROACH (AA) AVERAGED OVER 150 TRIALS (BOUND IS GIVEN IN UNITS OF DISTANCE, SQUARED)

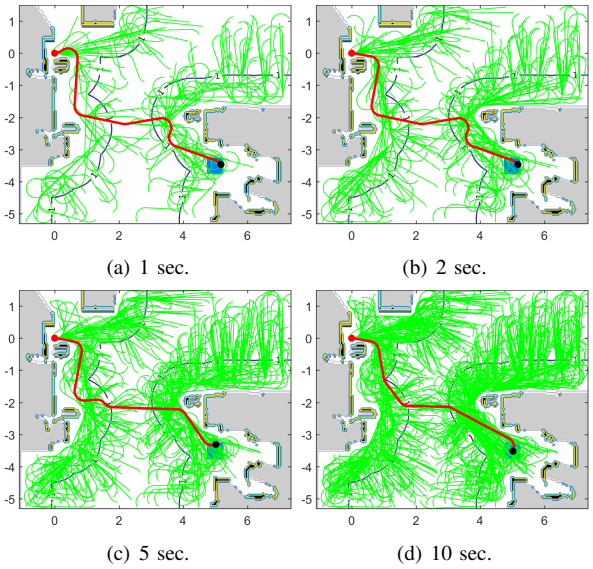
	Path	To 1	To 2	To 3	To 4
Max Bound Encountered	AA	52.837	69.947	67.907	50.386
	MM-RRT*	37.177	50.916	51.064	34.739
Sum of Bound	AA	2519	2926	2334	1288
	MM-RRT*	2313	3275	2515	1354
Terminal Bound	AA	52.837	69.947	67.907	50.386
	MM-RRT*	37.124	43.805	29.905	23.777
RunTime	AA	0.947s			
	MM-RRT*	1.042s			

unless it obtains GPS measurements to curb error growth. All path planning problems explored in this domain were rooted at $x_{init} = (0.01, 0.99)$. Trees were constructed using two different strategies and the results are compared for paths to five different goal regions. The goal regions are indicated by the terminal locations of the paths depicted in Figure 3(a); they are numbered from one to five. Each goal is a circular region of radius 0.02 units.

As a baseline for comparison, the first strategy explored was minimum-uncertainty path planning using an *additive* cost metric: the sum of the values of ℓ, summed at every filter update along a path. This metric, which penalizes the accumulation of uncertainty along a path, is intended to offer a basis for comparison for the MM-RRT* algorithm. A representative example of this approach is given in Figure 3(a). The second strategy explored is MM-RRT*, using $\bar{\ell}$ as the uncertainty cost metric, depicted in Figure 3(b). Fifty trials were performed of both the additive approach and MM-RRT* using ℓ as the basis for expressing uncertainty. In each trial, a tree of 50,000 nodes was constructed. Five quantities were compared between the two competing methods: (1) the maximum bound encountered over a path (Figure 4(a) and 4(b)), (2) path duration (Figure 4(c) and 4(d)), (3) the sum of the bound over each path (Figure 4(e) and 4(f)), (4) the mean value of each example path's terminal bound (Figure 4(g)), and (5) the value of the bound at all nodes in the tree (Figure 4(h)). All quantities shown in Figure 4 depict mean values averaged over 50 trials of each method. All computational experiments were performed using a single core of a laptop computer's Intel i7-4710MQ 2.5 GHz quad-core processor, equipped with 16GB RAM and the 64-bit Ubuntu 14.04 operating system. The RRT(*) C library provided by the authors of [9] was adapted to implement the algorithms considered in this paper.

In Figure 4(a) and 4(b), the maximum value of ℓ encountered over each of the five highlighted paths is shown as a function of the number of nodes comprising the tree, averaged over 50 trials. The plot begins where a feasible path has been returned from all 50 trials. The mean maximum bound obtained over these five example paths is reduced by 21.21%, 36.26%, 35.98%, 39.51% and 35.6% respectively by MM-RRT* in comparison to the additive approach. Figure 4(b) demonstrates that MM-RRT*'s paths exhibits asymptotic behavior, however, even when averaged over 50 trials, it is clear that paths undergo small fluctuations in cost as they approach their respective asymptotes. Path duration is also shown as a function of the number of nodes in Figures 4(c) and 4(d). Due to its priority on curbing uncertainty, the lengths of the paths offered by MM-RRT* are greater than those offered by the additive approach. The mean terminal bound of each path is depicted in Figure 4(g). MM-RRT* reduces the value of each path's terminal bound by 23.9%, 54.91%, 38.21%, 41.96% and 34.54% respectively in comparison to the additive approach. Figure 4(h), using a histogram, visualizes the values of the eigenvalue bound $\bar{\ell}$ for all nodes in the final tree, averaged over 50 trials. The min-max approach offers a narrower spread of uncertainty among the nodes of its tree; 98.99% of the nodes of MM-RRT* have an eigenvalue bound smaller than 1.2 units of distance, squared. For the competing additive approach, only 70.31% of the tree's nodes fall within this threshold. The "fat tail" in the histogram for the additive approach, which contains 29.69% of the tree's nodes, demonstrates that penalizing the sum of uncertainties will often favor short paths that do not curb uncertainty to the same extent as a min-max approach.

Further comparisons are made in a more complex scenario inspired by the layout of a real-world office environment. A robot translates in a 78.1m by 63.3m domain from $x_{init} = (0.0, 0.0)$ shown in Figure 5. The same color spectrum used in Figure 3 is applied here. There are five regions that have GPS availabilty, labeled A through E. Four goal regions used in the planning problem are numbered from 1 to 4. One hundred fifty trials are performed in this scenario, and each trial grows a tree comprised of 10,000 nodes. The same quantities compared in Figure 4 are also compared in this case, and the computational results of the two competing planning methods, over 150 trials, are shown in Table I. The paths of the additive approach only enter at most one GPS zone, while the paths produced by MM-RRT* enter at least



(a) 1 sec. (b) 2 sec. (c) 5 sec. (d) 10 sec.

Fig. 6. A real-time path planning example using an experimentally derived map from our lab, and a simulated ground robot with a three degree-of-freedom Dubins model. The robot can localize within a meter of the obstacles, but must rely on odometry elsewhere. The evolving solution and its tree is shown after 1, 2, 5, and 10 seconds of computation time.

two GPS zones each. 85.31% of the tree nodes of MM-RRT* have an eigenvalue bound smaller than $35m^2$. In the case of the additive approach, this number is only 51.33%.

A final comparison is presented in Figure 6, showing a three degree-of-freedom Dubins vehicle planning over an experimentally derived map of our lab at Stevens Institute, where it is capable of localizing only within a 1m visibility range of the surrounding obstacles. Elsewhere, it must rely on odometry for navigation. In this case, trees are grown that obey Dubins constraints for a 0.3m turning radius, and a solution gradually evolves in real-time that limits the duration in which the vehicle relies exclusively on odometry.

VI. CONCLUSION

We have proposed a new path planning algorithm, MM-RRT*, that curbs localization uncertainty by minimizing the maximum value of its uncertainty metric encountered during the traversal of a path. This approach offers an alternative to other sampling-based belief space planning algorithms that employ additive cost representations of uncertainty. It also makes efficient use of a tree structure, with only a single belief allowed per graph node. As demonstrated in our computational results, an additive approach may often provide a suitable solution that curbs the growth of localization error, but the inherent preference for shorter paths will occasionally eliminate families of paths that are capable of further uncertainty reduction and safer operation. This initial exploration of a min-max uncertainty cost function also leaves room for future work: the approach may be combined with uncertainty-based rejection sampling to reduce collision risk, and the prospect of establishing formal optimality guarantees will be further explored.

REFERENCES

[1] B. Englot and F.S. Hover, "Three-Dimensional Coverage Planning for an Underwater Inspection Robot," *The International Journal of Robotics Research*, vol. 32(9-10), pp. 1048-1073, 2013.

[2] G. Papadopoulos, H. Kurniawati, and N.M. Patrikalakis, "Asymptotically Optimal Inspection Planning using Systems with Differential Constraints," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4126-4133, 2013.

[3] M. Stilman, "Global Manipulation Planning in Robot Joint Space with Task Constraints," *IEEE Transactions on Robotics*, vol. 26(3), pp. 576-584, 2010.

[4] D. Devaurs, T. Simeon, and J. Cortes, "Efficient Sampling-Based Approaches to Optimal Path Planning in Complex Cost Spaces," *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2014.

[5] L.I. Reyes Castro, P. Chaudhari, J. Tumova, S. Karaman, E. Frazzoli, and D. Rus, "Incremental Sampling-Based Algorithm for Minimum-Violation Motion Planning," *Proceedings of the IEEE International Conference on Decision and Control*, pp. 3217-3224, 2013.

[6] G.A. Hollinger and G.S. Sukhatme, "Sampling-Based Robotic Information Gathering Algorithms ," *The International Journal of Robotics Research*, vol. 33(9), pp. 1271-1287, 2014.

[7] S.M. LaValle and J.J. Kuffner, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research*, vol. 20(5), pp. 378-400, 2001.

[8] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, vol. 12(4), pp. 566-580, 1996.

[9] S. Karaman and E. Frazzoli, "Sampling-Based Algorithms for Optimal Motion Planning," *The International Journal of Robotics Research*, vol. 30(7), pp. 846-894, 2011.

[10] S. Prentice and N. Roy, "The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance," *The International Journal of Robotics Research*, vol. 28(11-12), pp. 1448-1465, 2009.

[11] S.D. Bopardikar, B. Englot, A. Speranzon, and J. van den Berg "Robust Belief Space Planning Under Intermittent Sensing via a Maximum Eigenvalue-Based Bound," *The International Journal of Robotics Research*, DOI: 10.1177/0278364916653816, pp. 1-18, 2016.

[12] A. Bry and N. Roy, "Rapidly-Exploring Random Belief Trees for Motion Planning Under Uncertainty," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 723-730, 2011.

[13] B.D. Luders and J.P. How, "An Optimizing Sampling-Based Motion Planner with Guaranteed Robustness to Bounded Uncertainty," *Proceedings of the American Control Conference* , pp. 771-777, 2014.

[14] R. Pepy, M. Kieffer, and E. Walter, "Reliable Robust Path Planner," *Proceedings of the IEEE/RSJ Inernational Conference on Intelligent Robots and Systems*, pp. 1655-1660, 2008.

[15] S.D. Bopardikar, B. Englot, and A. Speranzon, "Multiobjective Path Planning: Localization Constraints and Collision Probability," *IEEE Transactions on Robotics*, vol. 31(3), pp. 562-577, 2015.

[16] B.D. Luders, S. Karaman, and J.P. How, "Robust Sampling-Based Motion Planning with Asymptotic Optimality Guarantees," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2013.

[17] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information," *The International Journal of Robotics Research*, vol. 30(7), pp. 895-913, 2011.

[18] A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-Based Feedback Motion Planning Under Motion Uncertainty and Imperfect Measurements," *The International Journal of Robotics Research*, vol. 33(2), pp. 268-304, 2013.

[19] R. Alterovitz, T. Simon, and K. Goldberg, "The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty," *Proceedings of Robotics: Science and Systems*, 2007.

[20] V. A. Huynh, S. Karaman, and E. Frazzoli, "An Incremental Sampling-Based Algorithm for Stochastic Optimal Control,". *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2865-2872, 2012.

[21] H. Kurniawati and V. Yadav, "An Online POMDP Solver for Uncertainty Planning in Dynamic Environment," *Proceedings of the International Symposium on Robotics Research*, 2013.

[22] A. Gelb, ed., *Applied Optimal Estimation*, Cambridge, MA: The MIT Press, 1974.

[23] Y. Okamoto and H. Maris, "A Novel Algorithm for Calculation of the Extreme Eigenvalues of Large Hermitian Matrices," *Computer Physics Communications*, vol. 76(2), pp. 191-202, 1993.