Non-Gaussian Belief Space Planning: Correctness and Complexity

Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, Russ Tedrake

***Abstract*— We consider the partially observable control problem where it is potentially necessary to perform complex information-gathering operations in order to localize state. One approach to solving these problems is to create plans in *belief-space*, the space of probability distributions over the underlying state of the system. The belief-space plan encodes a strategy for performing a task while gaining information as necessary. Unlike most approaches in the literature which rely upon representing belief state as a Gaussian distribution, we have recently proposed an approach to non-Gaussian belief space planning based on solving a non-linear optimization problem defined in terms of a set of state samples [1]. In this paper, we show that even though our approach makes optimistic assumptions about the content of future observations for planning purposes, all low-cost plans are guaranteed to gain information in a specific way under certain conditions. We show that eventually, the algorithm is guaranteed to localize the true state of the system and to reach a goal region with high probability. Although the computational complexity of the algorithm is dominated by the number of samples used to define the optimization problem, our convergence guarantee holds with as few as two samples. Moreover, we show empirically that it is unnecessary to use large numbers of samples in order to obtain good performance.**

I. Introduction

A fundamental objective of robotics is to develop systems that can function robustly in unstructured environments where the state of the world is only partially observed and measurements are noisy. For example, robust robot manipulation is well modeled as partially observable problem. It is common to model control problems such as these as partially observable Markov decision processes (POMDPs). However, in general, finding optimal solutions to POMDPs has been shown to be PSPACE complete [2]. Even many approximate approaches are computationally complex: the time complexity of standard point-based algorithms, such as HSVI and SARSOP, is exponential in the planning horizon [3], [4], [5].

A growing body of work is focused on planning in *belief space*, the space of probability distributions over the underlying state space. Most of this work assumes that belief state can be accurately described by a Gaussian distribution. For example, in prior work, we and others have explored approaches to planning in belief space based on assuming that belief state can always be described accurately as a Gaussian distribution [6], [7], [8]. Another recent class of approaches avoids the complexity of belief space planning by evaluating a large number of candidate trajectories in the underlying state space in terms of the information that is likely to be gained during execution and the chances of colliding with problem constraints [9], [10], [11]. In domains with relatively simple observation dynamics, such as navigation domains with range beacons, the Gaussian assumption can be reasonable and the approach can work well. However, when the observation function is complex, such as in robot manipulation domains [1], this assumption can easily become arbitrarily inaccurate. In these contexts, belief space planning methods that assume Gaussian distributions can make arbitrarily poor belief state estimates.

Recently, we proposed an approach to belief space planning with non-Gaussian belief state distributions that was applied to a robust robot grasping problem [1]. The approach creates belief space plans by solving a non-linear optimization problem that is defined in terms of a hypothesis state and a set of additional samples. This paper analyzes the correctness and computational complexity of the algorithm. We show that, under certain conditions, the algorithm terminates with probability one in a belief state where the probability that the system has achieved task objectives is greater than a user-specified threshold. Moreover, we provide an analysis of how quickly belief state changes as a function of the quality of the intermediate information-gathering plans. The computational complexity of the algorithm is dominated by the number of samples that are used to define the optimization problem. It turns out that our convergence guarantee holds with as few as two samples. Moreover, our experiments indicate that, for relatively simple problems at least, it is unnecessary to use large numbers of samples in order to obtain good plans.

II. Problem Statement

We are concerned with the class of control problems where it is desired to reach a specified goal state even though state may only be estimated based on partial or noisy observations. Consider a discrete-time system with continuous non-linear deterministic process dynamics, $x_{t+1} = f(x_t, u_t)$, where state, x, is a column vector in $\mathbb{R}^n$, and action, $u \in \mathbb{R}^l$. Although state is not directly observed, an observation, $z_t = h(x_t) + v_t$, is made at each time t, where $z \in \mathbb{R}^m$ is a column vector and v_t is zero-mean Gaussian noise with covariance Q.

Bayes filtering can be used to estimate state based on the previous actions taken and observations perceived. The estimate is a probability distribution over state represented by a probability density function (pdf), $\pi(x; b) : \mathbb{R}^n \to \mathbb{R}^+$ with parameter vector, $b \in \mathscr{B}$. The parameter vector is called the *belief state* and the parameter space, $\mathscr{B}$, is called the *belief-space*. For deterministic process dynamics, the Bayes filter

This work was performed in the Computer Science and Artificial Intelligence Laboratory at MIT. Robert Platt is now with the Computer Science and Engineering Department at SUNY Buffalo. Email: {rplatt,lpk,tlp,russt}@csail.mit.edu

update can be written:

$$\pi(f(x,u_t);b_{t+1}) = \frac{\pi(x;b_t)P(z_{t+1}|x,u_t)}{P(z_{t+1})}. \tag{1}$$

The Bayes update calculates a new belief state, b_{t+1}, given b_t, u_t, and z_{t+1}. It will sometimes be written, $b_{t+1} = G(b_t,u_t,z_{t+1})$. In general, it is impossible to implement Equation 1 exactly using a finite-dimensional parametrization of belief-space. However, a variety of approximations exist in practice [12].

Starting from an initial belief state, b_1, the control objective is to achieve a task objective with a specified minimum probability of success, $\omega \in [0,1)$. Specifically, we want to reach a belief state, b, such that

$$\Theta(b,r,x_g) = \int_{x\in B_n(r)} \pi(x+x_g;b) \geq \omega, \tag{2}$$

where $B_n(r) = \{x \in \mathbb{R}^n, x^T x \leq r^2\}$ denotes the r-ball in $\mathbb{R}^n$ for some $r > 0$, and ω denotes the minimum probability of success.

III. Algorithm

The algorithm that we proposed in [1] can be viewed as a receding horizon control approach that creates and executes nominal belief space plans. During execution, it tracks a belief distribution over underlying state based on actions and observations. If the true belief state diverges from the nominal trajectory, our algorithm re-plans and the process repeats.

A. *Creating plans*

The key to the approach is a mechanism for creating horizon-T belief-space plans that guarantee that new information is incorporated into the belief distribution on each planning cycle. Given a prior belief state, b_1, define a "hypothesis" state at the maximum of the pdf, $x^1 = \arg\max_{x\in\mathbb{R}^n} \pi(x;b_1)$. Then, sample $k-1$ states from the prior distribution, $x^i \sim \pi(x;b_1), i \in [2,k]$, such that the pdf at each sample is greater than a specified threshold, $\pi(x^i;b_1) \geq \varphi > 0$, and there are at least two unique states among the $k-1$. We search for a sequence of actions, $\mathbf{u}_{T-1} = (u_1,\ldots,u_{T-1})$, that result in as wide a margin as possible between the observations that would be expected if the system were in the hypothesis state and the observations that would be expected in any other sampled state. As a result, a good plan enables the system to "confirm" that the hypothesis state is in fact the true state or to "disprove" the hypothesis state. If the hypothesis state is disproved, then the algorithm selects a new hypothesis on the next re-planning cycle, ultimately causing the system to converge to the true state.

To be more specific, let $F_t(x,\mathbf{u}_{t-1})$ be the state reached at time t if the system begins in state x and takes actions $\mathbf{u}_{t-1}$. Recall that the expected observation upon arriving in state x_t is $h(x_t)$. Therefore, the expected sequence of observations is:

$$\mathbf{h}(x,\mathbf{u}_{t-1}) = \left(h(F_1(x,\mathbf{u}_1))^T,\ldots,h(F_{t-1}(x,\mathbf{u}_{t-1}))^T\right)^T.$$

We are interested in finding a sequence of actions that minimizes the probability of seeing the observation sequence expected in the sampled states when the system is actually in the hypothesis state. In other words, we want to find a sequence of actions, $\mathbf{u}_{T-1}$, that minimizes

$$\tilde{J}(x^1,\ldots,x^k,\mathbf{u}_{1:T-1}) = \sum_{i=2}^{k} N\left(\mathbf{h}(x^i,\mathbf{u}_{T-1})|\mathbf{h}(x^1,\mathbf{u}_{T-1}),\mathbb{Q}\right)$$

where $N(\cdot|\mu,\Sigma)$ denotes the Gaussian distribution with mean μ and covariance Σ and $\mathbb{Q} = diag(Q,\ldots,Q)$ is the block diagonal of measurement noise covariance matrices of the appropriate size. When this sum is small, Bayes filtering will more accurately be able to determine whether or not the true state is near the hypothesis in comparison to the other sampled states.

The above expression for observation distance is only defined with respect to the sampled points. However, we would like to "confirm" or "disprove" states in regions about the hypothesis and samples – not just the zero-measure points themselves. This can be incorporated to the first order by defining small Gaussian distributions in state space with unit covariance and taking an expectation:

$$\begin{aligned} &J(x^1,\ldots,x^k,\mathbf{u}_{1:T-1}) \\ &= \sum_{i=2}^{k} \mathbb{E}_{y^i\sim N(\cdot|x^i,I),y^1\sim N(\cdot|x^1,I)} N\left(\mathbf{h}(y^i,\mathbf{u}_{T-1})|\mathbf{h}(y^1,\mathbf{u}_{T-1}),\mathbb{Q}\right) \\ &= \sum_{i=2}^{k} N\left(\mathbf{h}(y^i,\mathbf{u}_{T-1})|\mathbf{h}(y^1,\mathbf{u}_{T-1}),\Gamma(x,\mathbf{u}_{T-1})\right), \end{aligned} \tag{3}$$

where

$$\begin{aligned} \Gamma(x,\mathbf{u}_{T-1}) = 2\mathbb{Q} &+ \mathbf{H}(x,\mathbf{u}_{T-1})\mathbf{H}(x,\mathbf{u}_{T-1})^T \\ &+ \mathbf{H}(x^1,\mathbf{u}_{T-1})\mathbf{H}(x^1,\mathbf{u}_{T-1})^T, \end{aligned}$$

$\mathbf{H}(x,\mathbf{u}_{1:t-1}) = \partial\mathbf{h}(x,\mathbf{u}_{1:t-1})/\partial x$ denotes the Jacobian matrix of $\mathbf{h}(x,\mathbf{u}_{1:t-1})$ at x. Rather than optimizing for $J(x^1,\ldots,x^k,\mathbf{u}_{1:T-1})$ (Equation 3) directly, we simplify the planning problem by dropping the normalization factor in the Gaussian and optimizing the exponential factor only. Let

$$\Phi(x^i,\mathbf{u}_{T-1}) = \|\mathbf{h}(x^i,\mathbf{u}_{T-1}) - \mathbf{h}(x^1,\mathbf{u}_{T-1})\|^2_{\Gamma(x^i,\mathbf{u}_{T-1})}.$$

The modified cost function is:

$$\bar{J}(x^1,\ldots,x^k,\mathbf{u}_{1:T-1}) = \frac{1}{k}\sum_{i=1}^{k} e^{-\Phi(x^i,\mathbf{u}_{T-1})}. \tag{4}$$

The optimization problem becomes:

Problem 1:

$$\begin{aligned} \text{Minimize} \quad & \bar{J}(x^1,\ldots,x^k,\mathbf{u}_{T-1}) + \alpha\mathbf{u}_{T-1}^T\mathbf{u}_{T-1} && (5) \\ \text{subject to} \quad & x^i_{t+1} = f(x^i_t,u_t), i \in [1,k] && (6) \\ & x^1_T = x_g, x^i_1 = x^i, i \in [1,k]. && (7) \end{aligned}$$

Equation 5 adds an additional quadratic cost on action that adds a small preference for short trajectories. The associated weighting parameter should be set to a small value ($\alpha \ll 1$).

Problem 1 can be solved using a number of planning techniques such as rapidly exploring random trees [13], differential dynamic programming [14], or sequential quadratic programming [15]. We use sequential quadratic programming to solve the direct transcription [15] of Problem 1. Although direct transcription is only guaranteed to find locally optimal solutions, we have found that it works well for many of the problems we have explored. The direct transcription solution will be denoted

$$\mathbf{u}_{T-1} = \textsc{DirTran}(x^1, \dots, x^k, x_g, T), \tag{8}$$

for samples, $x^1, \dots, x^k$, goal state constraint, x_g, and time horizon, T. Note that the dimensionality of Problem 1 is nk – linear in the dimensionality of the underlying state space with a constant equal to the number of samples. This compares favorably with the approaches in [6], [7], [8] that must solve planning problems in n^2-dimensional spaces (number of entries in the covariance matrix).

B. Re-planning

After creating a plan, our algorithm executes it while tracking the belief state using the user-supplied belief-state update, G. If the actual belief state diverges too far from a nominal trajectory derived from the plan, then execution stops and a new plan is created. The overall algorithm is outlined in Algorithm 1. The outer *while* loop iteratively creates and executes plans until the planning objective (Equation 2) is satisfied. Step 2 sets the hypothesis state to the maximum of the prior distribution. Step 3 samples $k-1$ additional states. Step 4 finds a nominal belief-space trajectory, $\bar{b}_{1:T}$ that optimizes Problem 1. Steps 6 through 12 execute the plan. Step 9 updates the belief state given the new action and observation using the user-specified Bayes filter implementation. Step 10 breaks plan execution when the actual belief state departs too far from the nominal trajectory, as measured by the KL divergence, $D_1\left[\pi(\cdot;b_{t+1}), \pi(\cdot;\bar{b}_{t+1})\right] > \theta$. The second condition, $\bar{J}(x^1, \dots, x^k, \mathbf{u}_{t-1}) < 1-\rho$, guarantees that the *while* loop does not terminate before a (partial) trajectory with cost $\bar{J} < 1-\rho$ executes. We show in the next section that the second condition guarantees that the algorithm makes "progress" on each iteration of the *while* loop.

```
Input : initial belief state, $b$, goal state, $x_g$, planning
        horizon, $T$, and belief-state update, $G$.
1  while $\Theta(b, r, x_g) < \omega$ do
2      $x^1 = \arg\max_{x \in \mathbb{R}^n} \pi(x; b)$;
3      $\forall i \in [2,k], x^i \sim \pi(x;b) : \pi(x^i;b) \geq \varphi$;
4      $\bar{b}_{1:T}, \mathbf{u}_{T-1} = \texttt{DirTran}(b, x^1, \dots, x^k, x_g, T)$;
5      $b_1 = b$;
6      for $t \leftarrow 1$ to $T-1$ do
7          execute action $u_t$, perceive observation $z_{t+1}$;
8          $b_{t+1} = G(b_t, u_t, z_{t+1})$;
9          if $D_1\left[\pi(x;b_{t+1}), \pi(x;\bar{b}_{t+1})\right] > \theta$ and
           $\bar{J}(\mathscr{G}, \mathbf{u}_{t-1}) < 1-\rho$ then
10             break
11         end
12     end
13     $b = b_{t+1}$;
14 end
```

Algorithm 1: Belief-space re-planning algorithm

IV. Analysis

We are interested in the correctness of Algorithm 1. Can we guarantee that Algorithm 1 eventually reaches a belief state where it is very likely that the system has achieved its goal? We show that if G is an exact implementation of Equation 1, and if DirTran in Algorithm 1 (step 4) always finds plans with a cost strictly less than one, then the algorithm terminates with probability one in a belief state where the probability that the system has achieved task objectives is greater than the user-specified threshold, ω. Moreover, we find a specific bound on how quickly the system localizes true state as a function of the quality of the intermediate plans.

We start by providing a lower bound on the expected probability of states in a neighborhood of the true state. On a particular iteration of the outer *while* loop in Algorithm 1, suppose that the system begins in belief state, b_1, while the true state is κ, and executes a sequence of actions, $\mathbf{u} = (u_1, \dots, u_{T-1})$ (subscript dropped for conciseness). During execution, the system perceives observations $\mathbf{z} = (z_2, \dots, z_T)$ and ultimately arrives in belief state b_T. The probability of a state, $y = F_T(x, \mathbf{u})$, estimated by recursively evaluating Equation 1 is:

$$\pi(y; b_T) = \pi(x; b_1) \frac{q_x(\mathbf{z}, \mathbf{u})}{p(\mathbf{z}, \mathbf{u})}, \tag{9}$$

where

$$q_x(\mathbf{z}, \mathbf{u}) = N(\mathbf{z} | \mathbf{h}(x, \mathbf{u}), \mathbb{Q}) \tag{10}$$

is the probability of the observations given that the system starts in state x and takes actions, $\mathbf{u}$, and

$$p(\mathbf{z}, \mathbf{u}) = \int_{x \in \mathbb{R}^n} \pi(x; b_1) N(\mathbf{z} | \mathbf{h}(x, \mathbf{u}), \mathbb{Q}) \tag{11}$$

is the marginal probability of the observations given $\mathbf{u}$. The following Lemma shows that $\pi(y; b_T)$ can be lower-bounded in terms of the proximity of x to the true state, κ.

Lemma 1: Suppose we are given an arbitrary sequence of actions, $\mathbf{u}$, and an arbitrary initial state, $x \in \mathbb{R}^n$. Then, the expected probability of $y = F_T(x, \mathbf{u})$ found by recursively evaluating the deterministic Bayes filter update (Equation 1) is

$$\mathbb{E}_{\mathbf{z}} \left\{ \frac{\pi(y; b_T)}{\pi(x; b_1)} \right\} \geq \exp\left(D_1(q_\kappa, p) - D_1(q_\kappa, q_x) \right),$$

where q_κ, q_x, and p are defined in Equations 10 and 11 and D_1 denotes the KL divergence between the arguments.

Proof: The log of the expected change in the probability of x is:

$$\begin{aligned}\log \mathbb{E}_{\mathbf{z}} \left\{ \frac{\pi(y;b_T)}{\pi(x;b_1)} \right\} &= \log \mathbb{E}_{\mathbf{z}} \left\{ \frac{q_x(\mathbf{z},\mathbf{u})}{p(\mathbf{z},\mathbf{u})} \right\} \\ &= \log \int_{\mathbf{z}\in\mathbb{R}^m} \frac{q_\kappa(\mathbf{z},\mathbf{u}) q_x(\mathbf{z},\mathbf{u})}{p(\mathbf{z},\mathbf{u})} \\ &\geq \int_{\mathbf{z}\in\mathbb{R}^m} q_\kappa(\mathbf{z}) \left(\log q_x(\mathbf{z},\mathbf{u}) - \log p(\mathbf{z},\mathbf{u})\right) \\ &= D_1(q_\kappa, p) - D_1(q_\kappa, q_x),\end{aligned}$$

where the third line was obtained using Jensen's inequality and the last line follows from algebra. Taking the exponential gives us the claim. ■

Lemma 1 expresses the bound in terms of the divergence, $D_1(q_\kappa, p)$, with respect to the true state, κ. However, since κ is unknown ahead of time, we must find a lower bound on the divergence $D_1(q_y, p)$ for arbitrary values of y. The following lemma establishes a bound on this quantity. We use the notation that $\|a\|_A = \sqrt{a^T A^{-1} a}$ denotes the Mahalanobis distance with respect to A.

Lemma 2: Given an arbitrary $\mathbf{u}$ and a distribution, π, suppose $\exists \Lambda_1, \Lambda_2 \subseteq \mathbb{R}^n$ such that $\forall x_1, x_2 \in \Lambda_1 \times \Lambda_2, \|\mathbf{h}(x_1,\mathbf{u}) - \mathbf{h}(x_2,\mathbf{u})\|^2_{\mathbb{Q}} \geq \zeta^2$ and $\int_{x\in\Lambda_1} \pi(x) \geq \gamma$, $\int_{x\in\Lambda_2} \pi(x) \geq \gamma$. Then

$$\min_{y\in\mathbb{R}^n} D_1(q_y, p) \geq 2\eta^2\gamma^2 \left(1 - e^{-\frac{1}{2}\zeta^2}\right)^2,$$

where $\eta = 1/\sqrt{(2\pi)^n |\mathbb{Q}|}$ is the Gaussian normalization constant.

Proof: By Pinsker's inequality, we know that KL divergence can be bounded by total variation: $D_1(q_y, p) \geq 2\sup_{\mathbf{z}} \left(q_y(\mathbf{z},\mathbf{u}) - p(\mathbf{z},\mathbf{u})\right)^2$. We lower bound the total variation by considering only those parts of the distributions where $\exists x \in \Lambda_1 \cup \Lambda_2$ such that $\mathbf{z} = \mathbf{h}(x,\mathbf{u})$:

$$\begin{aligned}D_1(q_y, p) \geq 2\sup_{\mathbf{z}} \Bigg(&\int_{x\in\Lambda_1} \pi(x) q_y(\mathbf{z}) - \pi(x) p(\mathbf{z}) \\ &+ \int_{x\in\Lambda_2} \pi(x) q_y(\mathbf{z}) - \pi(x) p(\mathbf{z}) \Bigg)^2.\end{aligned}$$

The right hand side above is minimized when $y \in \Lambda_1$ or $y \in \Lambda_2$. Without loss of generality, assume that $y \in \Lambda_1$:

$$\begin{aligned}\min_{y\in\mathbb{R}^n} D_1(q_y, p) &\geq 2\sup_{\mathbf{z}} \left(\int_{x\in\Lambda_2} \pi(x) q_y(\mathbf{z}) - \pi(x) p(\mathbf{z}) \right)^2 \\ &\geq 2\left(\gamma\eta(1 - e^{-\frac{1}{2}\zeta^2})\right)^2.\end{aligned}$$

■

As a result of Lemmas 1 and 2, we know that we can lower bound the expected increase in probability of a region about the true state by finding regions, Λ_1 and Λ_2, that satisfy the conditions of Lemma 2 for a given $\mathbf{u}$. The following lemma shows that these regions exist for any $\mathbf{u}$ with a cost (Equation 3) $J < 1$. For the proof of this Lemma, we refer to reader to [16].

Lemma 3: Suppose that $\mathbf{u}$ is a plan with cost $J = J(x^1, \ldots, x^k, \mathbf{u})$ defined over the samples, $x^i, i \in [1,k]$. If the maximum eigenvalue of the Hessian of h is λ, then $\exists i \in [1,k]$ such that $\forall r \in \mathbb{R}^+, \forall \delta_1, \delta_2 \in B_n(r)$:

$$\begin{aligned}\|\mathbf{h}(x^i + \delta_1, \mathbf{u}) - \mathbf{h}(x^1 + \delta_2, \mathbf{u})\|^2_{\Gamma(x^i,\mathbf{u})} \\ \geq \left[\sqrt{-\log J} - 2(r + cr^2)\right],\end{aligned}$$

where $c = \lambda \|\mathbf{1}\|_{\mathbb{Q}}/2$ and $B_n(r) = \{x \in \mathbb{R}^n; x^T x \leq r^2\}$.

We now state our main theoretical result regarding Algorithm 1 correctness. Two conditions must be met. First, the planner in step 4 of Algorithm 1 must always find low-cost plans successfully. Essentially, this guarantees that each plan will acquire useful information. Second, step 8 of Algorithm 1 must use an exact implementation of the Bayes filter. In practice, we expect that this second condition will rarely be met. However, our experiments indicate that good results can be obtained using practical filter implementations (Section V).

Theorem 1: Suppose we have:

1) a prior distribution, $\pi(x; b_1)$;
2) $k \geq 2$ samples, $x^1, \ldots, x^k$, from $\pi(x; b_1)$ such that $\exists r, \varphi \in \mathbb{R}^+$ where $\forall \delta B_n(r), i \in [1,k]$ we have $\pi(x^i + \delta; b_1) \geq \varphi$;
3) a trajectory, $\mathbf{u}_{T-1}$, with cost $J = \bar{J}(x^1, \ldots, x^k, \mathbf{u}_{T-1})$.

If an exact implementation of Bayesian filtering were to track state while executing $\mathbf{u}_{1:T-1}$ resulting in a distribution at time T of $\pi(x; b_T)$, then the probability of all states within a ball of radius r about the true state at time T is expected to increase by a factor of

$$\exp\left[2\eta^2\gamma^2 \left(1 - e^{-\frac{1}{2}\left(\sqrt{-\log J} - 2(r+cr^2)\right)^2}\right)\right] \quad (12)$$

relative to its value at time 1, where $\eta = 1/\sqrt{(2\pi)^n |\mathbb{Q}|}$ is the Gaussian normalization constant, $\gamma = \epsilon Vol_n(r)$, $Vol_n(r)$ is the volume of the r-ball in n dimensions, $c = \frac{1}{2}\lambda\|1\|_Q$, and λ is the maximum eigenvalue of the Hessian of $h(F_t(x, \mathbf{u}_{T-1}))$ over all t, x, and $\mathbf{u}_{T-1}$.

Proof: Lemma 3 gives us two samples, x^i and x^1 such that $\forall \delta_1, \delta_2 \in B_n(r)$,

$$\begin{aligned}\|\mathbf{h}(x^i + \delta_1, \mathbf{u}) - \mathbf{h}(x^1 + \delta_2, \mathbf{u})\|^2_{\Gamma(x^i,\mathbf{u})} \\ \geq \|\mathbf{h}(x^i + \delta_1, \mathbf{u}) - \mathbf{h}(x^1 + \delta_2, \mathbf{u})\|^2_Q \\ \geq \left[\sqrt{-\log J} - 2(r + cr^2)\right]^2.\end{aligned}$$

Lemma 2 gives us a lower bound of $D_1(q_y, p)$ by setting $\zeta = \sqrt{-\log J} - 2(r + cr^2)$. Lemma 1 gives us the conclusion of the Theorem by noting that $D_1(q_\kappa, q_x) = 0$ when $x = \kappa$. ■

The above theorem enables us to conclude that Algorithm 1 is guaranteed to terminate in the goal region of belief space.

Theorem 2: Suppose that Algorithm 1 executes with $k \geq 2$ samples. Suppose that DIRTRAN (step 4) always finds a plan with a maximum cost, $\epsilon < 1$: $\bar{J}(x^1, \ldots, x^k, \mathbf{u}_{1:T-1}) < \epsilon < 1$. Suppose that G (step 8) is implemented using an exact implementation of Bayes filtering. Then Algorithm 1

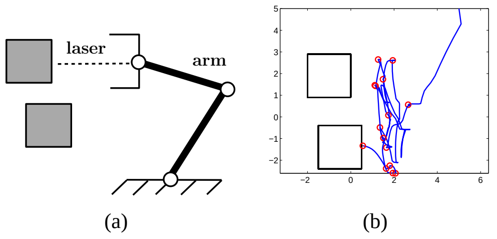

Fig. 1. (a) the experimental scenario. (b) a path found by Algorithm 1 with a nine-sample planner. It starts in the upper right and ends at a point directly in front of the right-most box. The red circles denote where re-planning occurred.

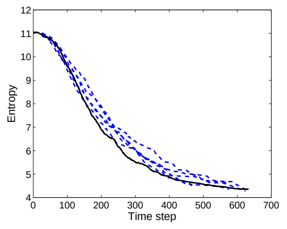

Fig. 2. Belief state entropy as a function of time step. The solid black line corresponds to the trajectory shown in (b). The dashed blue lines correspond to five additional nine-sample runs.

terminates with probability one in a belief state, b, where $\Theta(b,r,x_g) \geq \omega$.

Proof: Notice that if $\bar{J}(x^1,\ldots,x^k,\mathbf{u}_{1:T-1}) < \varepsilon$, then there exists some strictly positive radius, $r > 0$, such that the expression in Equation 12 strictly greater than a lower bound, $\varpi > 1$. Let κ_T denote the location of the true state at time T. Using the result of Theorem 1, we know that $\Theta(b,r,\kappa_T)$ grows arbitrarily close to one, and we know that Algorithm 1 must ultimately terminate for any $\omega < 1$. ■

At the end of Section III-A, we noted that the planning problem solved in step 4 of Algorithm 1 was linear in the dimensionality of the underlying space. Theorem 2 asserts that the algorithm is correct with as few as two samples. As a result, we know that the linear constant can be as small as two.

V. Experiments

From a practical perspective, the preceding analysis is useful because it tells us that if we execute the *while* loop in Algorithm 1 a sufficient number of times, we can expect to localize the state of the system with arbitrary accuracy (we can drive $\Theta(b,r,x_g)$ arbitrarily low). However, for this result to hold, we require the planner to find low cost paths each time it is called and for the tracking Bayes filter to be an exact realization of Equation 1 (the premise of Theorem 2). Since these conditions are difficult to meet in practice, an important question is how well the approach works for approximately accurate Bayes filter implementations and for planners that only succeed some of the time. Furthermore, we are interested in knowing how the performance of the algorithm changes with the number of samples used to parametrized the planner. Figure 1(a) illustrates the experimental scenario. A two-link robot arm moves a hand in the plane. A single range-finding laser is mounted at the center of the hand. The laser measures the range from the end-effector to whatever object it "sees". The hand and laser are constrained to remain horizontal. The position of the hand is assumed to be measured perfectly. There are two boxes of known size but unknown position to the left of the robot (four dimensions of unobserved state). The boxes are constrained to be aligned with the coordinate frame (they cannot rotate). The control input to the system is the planar velocity of the end-effector. The objective is for the robot to localize the two boxes using its laser and move the end-effector to a point directly in front of the right-most box (the box with the largest x-coordinate) so that it can grasp by extending and closing the gripper. On each time step, the algorithm specified the real-valued two-dimensional hand velocity and perceived the laser range measurement. If the laser missed both boxes, a zero measurement was perceived. The (scalar) measurements were corrupted by zero-mean Gaussian noise with 0.31 standard deviation.

Figure 1(b) illustrates the path of the hand (a point directly between the two jaws of the gripper) found by running our algorithm parametrized by nine samples. The state space was four dimensional and comprised of two box locations ranging between $[-1,1]$ on the x-axis and $[-2,2]$ on the y-axis. The hand starts in the upper right corner at $(5,5)$ and ends at a point directly in front of the lower right box. The blue line shows the path and the red circles identify the points along the path at which re-planning occurred (there are 14 re-plan events in this example). The tracking Bayes filter was implemented using a gridded histogram filter comprised of 62500 bins over the four-dimensional space (the position of each of the two boxes was denoted by a point in a 10×25 grid). At the start of planning, the prior histogram distribution was assumed to be uniform. The cost function optimized by the DirTran planner (Equation 5) was parametrized by $\alpha = 0.01$ and $\mathbf{V} = diag(0.5)$ (Equation 3). The planning horizon was $T = 50$. The algorithm did not terminate until the histogram Bayes filter was 90% confident that it had localized the right-most box to within ± 0.3 of its true location ($\omega = 0.9$ in step 1 of Algorithm 1). Figure 3(a)-(d) show snapshots of the histogram distribution at time steps 10, 100, 200, and 300. (This is actually a two-dimensional projection of the four dimensional distribution illustrating the distribution over the location of *one* box only.) Figure 3(e)-(h) show the nine samples used to parametrize the planning algorithm at the four snapshots. Initially, (in Figures 3 (a) and (e), the distribution is high-entropy and the samples are scattered through the space. As time increases, the distribution becomes more peaked and the sample sets become more focused. The solid black line in Figure 1(b) shows the entropy of the histogram distribution as a function of time step. As expected, entropy decreases significantly over the trajectory. For comparison, the five additional blue dotted lines in Figure 2 show entropy results from five additional identical experiments. Note the relatively small

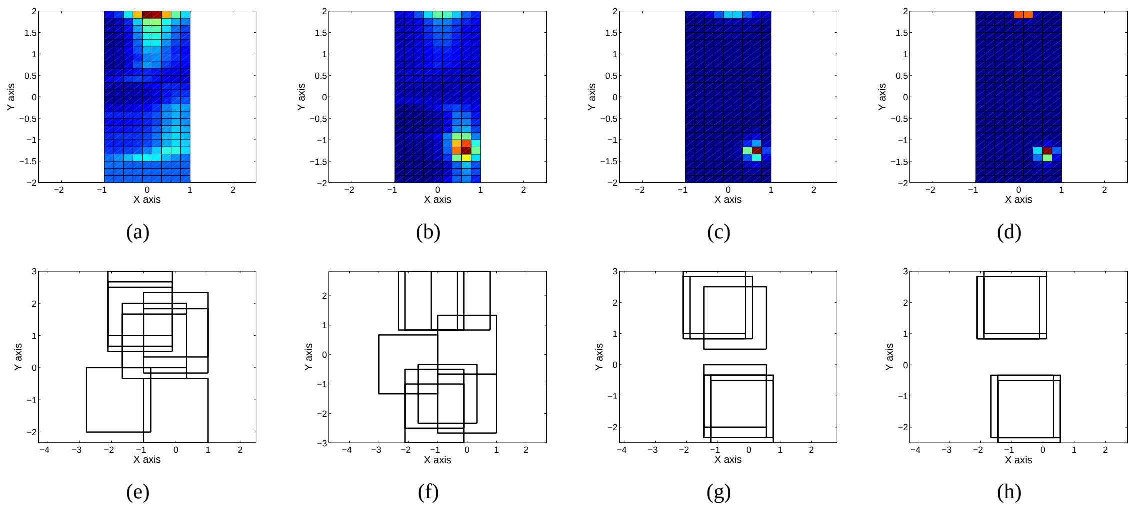

Fig. 3. Histogram probability distributions (a-d) and planner sample sets (e-h) at time steps 10, 100, 200, and 300 during the path shown in Figure 1(b).

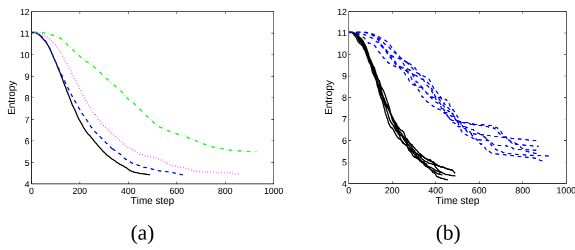

Fig. 4. (a) comparison of entropy averaged over six runs for four different planner sample set sizes (36 samples, solid black line; 9 samples, dashed blue line; 4 samples, dotted magenta line; 2 samples, dash-dot green line). (b) comparison of the six thirty-six-sample runs (solid black) with the six two-sample runs (dashed blue).

variance amongst trajectories. Even though the algorithm finds a very different trajectory on each of these runs, performance is similar. These results help answer two of the questions identified at the beginning of the section. First, Figure 3 suggests that in at least one case, the histogram filter was adequate to represent the belief state in the context of this algorithm even though it is a coarsely discretized approximation to the true distribution. The black line in Figure 2 suggests that DIRTRAN was an effective tool for planning in this scenario. The six additional runs illustrated in Figure 2 indicate that these results are typical.

The other question to be answered concerns the effect of the number of samples on algorithm performance. To find an answer, we have run the algorithm in the scenario described above for four contingencies where the planner was parametrized by two, four, nine, and thirty-six samples. Figure 4(a) compares the average (over six runs each) information-gathering performance for the four contingencies. Although increasing the number of samples improves algorithm performance, the gains diminish as the number of samples increases. Figure 4(b) compares the two-sample runs with the thirty-six-sample runs and demonstrates that the improvement is statistically significant. The comparison of Figure 4(c) with Figure 1(b) suggests that (in this experiment, at least) the trajectories produced by the high-sample planner are better than those produced by the low-sample planner because the high-sample planner does a better job covering the space in front of the boxes. These results show that it is valuable to expend computational effort planning an information-gathering trajectory, even in this simple example. The results also show that the performance of our algorithm smoothly degrades or improves with fewer or more samples used during planning. Even with the minimum of two samples, the algorithm is capable of making progress.

VI. CONCLUSIONS

Creating robots that can function robustly in unstructured environments has always been a central objective of robotics. In order to achieve this, it is necessary to develop algorithms capable of actively localizing the state of the world while also reaching task objectives. Recently, we proposed a belief space planning algorithm that is capable of planning in non-Gaussian belief spaces [1]. The non-Gaussian aspect of this algorithm is essential because in many robot problems it is not possible to track belief state accurately by projecting onto an assumed Gaussian density function (this is the case, for example, in many robot manipulation problems). However, since non-Gaussian belief space is potentially very high dimensional, it is important to know how effective the algorithm is and what its computational complexity is. This paper provides a novel sufficient condition for guaranteeing that the probability of the true state found by the Bayes filter increases (Lemma 1) and we show that this condition is met each time a low-cost plan executes. As a result, we can guarantee that the probability of the true state increases by a bounded amount on each re-planning iteration (Theorem 2). The algorithm is eventually guaranteed to converge to a goal region in belief space. We also characterize the expected computational complexity of the algorithm, which is dominated by the number of samples used to define the optimization problem. It turns out that our theoretical results hold with as few as two samples. In addition, we find that, for some problems, algorithm performance is nearly optimized using very few (between two and nine) samples.

ACKNOWLEDGMENTS

This work was supported in part by in part by the NSF under Grant No. 0712012, in part by ONR MURI under grant N00014-09-1-1051 and in part by AFOSR grant AOARD-104135

REFERENCES

[1] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, “Efficient planning in non-gaussian belief spaces and its application to robot grasping,” in *Int’l Symposium on Robotics Research*, 2011.

[2] C. Papadimitriou and J. Tsitsiklis, “The complexity of Markov decision processes,” *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[3] T. Smith and R. Simmons, “Point-based POMDP algorithms: Improved analysis and implementation,” in *Proc. Uncertainty in Artificial Intelligence*, 2005.

[4] H. Kurniawati, D. Hsu, and W. S. Lee, “SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2008.

[5] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, “Online planning algorithms for POMDPs,” *The Journal of Machine Learning Research*, vol. 32, pp. 663–704, 2008.

[6] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, “Belief space planning assuming maximum likelihood observations,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[7] T. Erez and W. Smart, “A scalable method for solving high-dimensional continuous POMDPs using local approximation,” in *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2010.

[8] S. Miller, A. Harris, and E. Chong, “Coordinated guidance of autonomous uavs via nominal belief-state optimization,” in *American Control Conference*, 2009, pp. 2811–2818.

[9] J. Van der Berg, P. Abbeel, and K. Goldberg, “LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[10] S. Prentice and N. Roy, “The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance,” in *12th International Symposium of Robotics Research*, 2008.

[11] N. Du Toit and J. Burdick, “Robotic motion planning in dynamic, cluttered, uncertain environments,” in *IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2010.

[12] A. Doucet, N. Freitas, and N. Gordon, Eds., *Sequential monte carlo methods in practice*. Springer, 2001.

[13] S. LaValle and J. Kuffner, “Randomized kinodynamic planning,” *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[14] D. Jacobson and D. Mayne, *Differential dynamic programming*. Elsevier, 1970.

[15] J. Betts, *Practical methods for optimal control using nonlinear programming*. Siam, 2001.

[16] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, “A hypothesis-based algorithm for planning and control in non-gaussian belief spaces,” Massachusetts Institute of Technology, Tech. Rep. CSAIL-TR-2011-039, 2011.