

# Convex receding horizon control in non-Gaussian belief space

Robert Platt Jr.

**Abstract** One of the main challenges in solving partially observable control problems is planning in high-dimensional belief spaces. Essentially, it is necessary to plan in the parameter space of all relevant *probability distributions* over the state space. The literature has explored different planning technologies including trajectory optimization [8, 6] and roadmap methods [12, 4]. Unfortunately, these methods are hard to use in a receding horizon control context. Trajectory optimization is not guaranteed to find globally optimal solutions and roadmap methods can have long planning times. This paper identifies a non-trivial instance of the belief space planning problem that is convex and can therefore be solved quickly and optimally even for high dimensional problems. We prove that the resulting control strategy will ultimately reach a goal region in belief space under mild assumptions. Since the space of convex belief space planning problem is somewhat limited, we extend the approach using mixed integer programming. We propose to solve the integer part of the problem in advance so that only convex problems need be solved during receding horizon control.

## 1 Introduction

The problem of controlling partially observable systems is extremely important in general and is particularly important in robotics. In partially observable systems, noisy and incomplete measurements make it hard to estimate state accurately. For example, it can be very challenging for a manipulation robot to estimate the position of objects to grasp accurately using laser, visual, or tactile data. Similarly, it can be difficult for an autonomous air or land vehicle to localize itself based on noisy range bearings to features in the environment. Since state is never known exactly in these problems, the objective of control should be to maximize the *probability* that the

---

Robert Platt Jr., SUNY at Buffalo, Buffalo, NY 14206, e-mail: robplatt@buffalo.edu

robot achieves its goals. These problems are frequently solved using planning and control in *belief space*, the space of probability distributions over an underlying state space. The solutions frequently involve strategies for actively gathering information in order to better localize state. This is in contrast to fully observable problems where, since state is fully observed, the only objective is to reach a goal state.

One of the challenges in controlling a partially observable system in belief space is that since belief space can be very high-dimensional, it can be difficult to compute a policy. In fact, finding optimal policies for Markov partially observable problems has been shown to be PSPACE-complete [7]. An alternative is to compute and execute plans instead and re-plan when necessary. However, in order for this to be effective, it is necessary to have the ability to re-plan quickly. This is because belief space dynamics can be very stochastic; a single observation can “convince” the system that it needs to adjust its belief state significantly (*i.e.* sequential Bayesian filtering can adjust its belief state estimate significantly based on only one or two observations). When this happens, the old plan becomes irrelevant and a new plan must be found. Unfortunately, most current belief space planning approaches are not fast enough to use in the inner loop of a receding horizon controller [8, 6, 12]. Moreover, most current approaches to belief space planning are based on assuming that belief space is Gaussian – that is, that belief state is always well-represented by a Gaussian [12, 14, 4, 11]. Unfortunately, this is not the case for many common robotics problems occurring in robot navigation or manipulation where multi-modal distributions are more common. In these situations, the Gaussian assumption can result in plans that are arbitrarily bad.

This paper proposes formulating the belief space planning problem as a convex program that is appropriate for receding horizon control. This is important for two reasons. First, since convex programs can be solved quickly and accurately in general, the method is fast enough for receding horizon control. Second, solutions to convex programs are guaranteed to be globally optimal. This is important because it enables us to guarantee that receding horizon control converges in the limit to a goal region in belief space as long as a path to the goal region exists. In this paper, we constrain ourselves to considering only linear systems with state-dependent observation noise. Although we are assuming a linear system, it should be noted that this problem setting is fundamentally different from the linear quadratic Gaussian (LQG) setting [2] because observation noise is state dependent. This one difference means that the certainty equivalence principle [2] does not apply. To our knowledge, this is the first belief space control strategy for systems with state dependent noise for which a such a convergence guarantee can be made. After introducing the basic convex formulation, this paper introduces an approach to using mixed integer programming to solve non-convex problems comprised of a small number of convex “parts”. Although it can take significantly longer to solve a mixed integer program than a convex program, this does not affect our on-line performance since the mixed integer program may be solved ahead of time and the integer variables fixed. We simulate our algorithm in the context of standard mobile robot localization and navigation problems where features in the environment make observations more or less noisy in different parts of the environment.

## 2 Problem Statement

We consider a class of Markov partially observable control problems where state,  $x \in \mathbb{R}^n$ , is not observed directly. Instead, on each time step, the system takes an action,  $u \in \mathbb{R}^l$ , and perceives an observation,  $z \in \mathbb{R}^m$ . The process dynamics are linear-Gaussian of the form,

$$x_{t+1} = Ax_t + Bu_t + v_t, \quad (1)$$

where  $A$  and  $B$  are arbitrary, and  $v_t$  is zero-mean Gaussian noise with covariance  $V$ . The observation dynamics are of the form,

$$z_t = x_t + q_t(x_t), \quad (2)$$

where  $q_t(x_t)$  is zero-mean Gaussian noise with state-dependent covariance,  $Q(x_t)$ . Notice that the presence of non-uniform observation noise invalidates the certainty equivalence principle [2] and makes this problem fundamentally different than the standard linear quadratic Gaussian (LQG) problem. In order to solve this problem using the convex formulation proposed in this paper, we require  $Q(x)^{-1}$  to be piecewise matrix convex in  $x$  (matrix convexity implies that  $a'Q(x)^{-1}a$  is a convex function for any constant vector,  $a$ ). In addition, we require all feasible trajectories to adhere to a chance constraint that bounds the probability that the system collides with an obstacle on a given time step. Let  $O_1, \dots, O_q \subset \mathbb{R}^n$  be a set of  $q$  polyhedral regions of state space that describe the obstacles. The probability that the system is in collision with an obstacle at time  $t$  is constrained to be less than  $\theta$ :

$$P(x_t \in \cup_{n=1}^q O_n) \leq \theta. \quad (3)$$

The objective of control is to reach a radius around a goal state,  $x_g$ , with high probability. Let  $b_t(x) = P(x|u_{1:t-1}z_{1:t})$  denote *belief state*, a probability distribution over system state that represents the state of knowledge of the system and incorporates all prior control actions,  $u_{1:t-1}$ , and observations,  $z_{1:t}$ . Our objective is to reach a belief state,  $b$ , such that

$$\Theta(b, r, x_g) = \int_{\delta \in B_n(r)} b(\delta + x_g) \geq \omega, \quad (4)$$

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  $\omega$  denotes the minimum probability of reaching the goal region.

### 3 Belief Space Planning

#### 3.1 Non-Gaussian belief space planning

We build upon a framework for non-Gaussian belief space control introduced in our prior work [8]. We use a receding horizon control strategy where planning and tracking occurs using different representations of belief state. Plans are created using a particular version of Bayesian filtering known as sequential importance sampling (SIS) [1]. However, belief state tracking occurs using a more accurate implementation of Bayesian filtering,  $G$ , that can be arbitrary. For concreteness, suppose that  $G$  is a histogram filter or a standard particle filter (SIR) [1]. At time  $t$ , the belief state,  $b_t(x)$ , is approximated by a set of  $k$  support points  $x_t^{1:k} = x_t^1, \dots, x_t^k$  and the corresponding un-normalized weights,  $w_t^{1:k}$ :

$$b_t(x) \approx \sum_{i=1}^k w_t^i \delta(x - x_t^i), \quad (5)$$

where  $\delta(x)$  denotes the Dirac delta function of  $x$ . As the number of samples,  $k$ , goes to infinity, this approximation becomes exact. When the system takes an action and perceives a new observation, SIS can be used to calculate the new belief state in two steps: the process update and the measurement update. Given a new action,  $u_t$ , the process update samples the support points at the next time step from the process dynamics,  $x_{t+1}^i \sim N(Ax_t^i + Bu_t, V)$ , where  $N(\mu, \Sigma)$  denotes the normal distribution with mean,  $\mu$ , and covariance matrix,  $\Sigma$ . Given a new observation,  $z_{t+1}$ , the measurement update adjusts the weight of each point according to the observation dynamics:  $w_{t+1}^i = w_t^i N(z_{t+1} - x_{t+1}^i, Q(x_{t+1}^i))$ , where we have used the fact that  $\mathbb{E}[z_{t+1}] = x_{t+1}$  (Equation 2). In this paper, we will track un-normalized weights and write the above measurement update as:

$$w_{t+1}^i = w_t^i \exp\left(-\|z_{t+1} - x_{t+1}^i\|_{Q(x_{t+1}^i)}^2\right), \quad (6)$$

where  $\|x\|_Q^2 = x'Q^{-1}x$  denotes the L2 norm of  $x$  weighted by  $Q^{-1}$ .

In order to plan in belief space, it is necessary to make assumptions regarding the content of future process noise and future observations. Following our prior work [8], we assume, for planning purposes only, that future observations will be generated as if the system were actually in the currently most likely state. The observation at time  $\tau > t$  is predicted to be  $z_\tau = x_\tau^1$  (recall the unit observation dynamics of Equation 2), where  $x_\tau^1$  denotes the state at time  $\tau > t$  corresponding to the most likely state at time  $t$ ,  $x_t^1 = \arg \max_{x \in \mathbb{R}^n} b_t(x)$ , after executing some sequence of actions. Later in the paper, we show that this assumption enables us to prove that the receding horizon control algorithm converges in the limit to a goal region in belief space. Algorithm 1 illustrates the complete replanning algorithm. Following our prior work in [8], we track belief state using  $G$ , an accurate implementation of sequential Bayesian filtering that is different than that used during planning.  $G$

**Input** : initial belief state,  $b_1$ , goal state,  $x_g$ , planning horizon,  $T$ , and belief-state update,  $G$ .

```

1  $t \leftarrow 1$ ;
2 while  $\Theta(b_t, r, x_g) < \omega$  do
3    $x_t^1 \leftarrow \arg \max_{x \in \mathbb{R}^n} b_t(x)$ ;
4    $\forall i \in [2, k], x_t^i \sim b_t(x) : b_t(x^i) \geq \varphi$ ;
5    $J_{1:T-1}, \mathbf{u}_{T-1} = \text{ConvexPlan}(b_t, x_t^1, \dots, x_t^k, x_g, T)$ ;
6   for  $\tau \leftarrow 1$  to  $T-1$  do
7     execute action  $u_\tau$ , perceive observation  $z_{\tau+1}$ ;
8      $b_{\tau+1} \leftarrow G(b_{\tau+1}, u_\tau, z_{\tau+1})$ ;
9     if  $J_\tau < \rho$  then
10      break
11   end
12 end
13  $t \leftarrow t + \tau$ ;
14 end
```

**Algorithm 1:** Receding horizon belief space control algorithm.

computes the next belief state,  $b_{t+1}$ , given that action  $u_t$  is taken from belief state,  $b_t$  and observation  $z_{t+1}$  is made:  $b_{t+1} = G(b_t, u_t, z_{t+1})$ . Steps 3 and 4 select the support points. Step 5 computes a belief space plan as described in the next section. The plan is represented by the action sequence,  $\mathbf{u}_{T-1} = (u_1, \dots, u_{T-1})$ , and the sequence of partial plan costs,  $J_{1:T-1} = (J_1, \dots, J_{T-1})$ . Steps 6 through 12 execute the plan. Step 9 breaks execution if at any point the partial plan cost drops below a threshold,  $\rho$ . The outer *while* loop executes until the belief space goal is satisfied. Later in this section, we give conditions under which this is guaranteed to occur.

### 3.2 Convex Formulation

The question remains how to solve the planning problem. We identify a class belief space trajectory planning problems that can be expressed as convex programs and solved using fast, globally optimal methods. Upon first approaching the problem, one might identify the following problem variables:  $x_\tau^i$ ,  $w_\tau^i$ , and  $u_\tau$ , for all  $\tau \in [t, t+T]$  and  $i \in [1, k]$ . However, notice that the weight update in Equation 6 is a non-convex equality constraint. Since we are interested in identifying a convex version of the problem, we express the problem in terms of the log-weights,  $y_\tau^i = \log(w_\tau^i)$  rather than the weights themselves. Equation 6 becomes:

$$y_{\tau+1}^i = y_\tau^i - \|x_{\tau+1}^1 - x_{\tau+1}^i\|_{Q(x_{\tau+1}^1)}^2. \quad (7)$$

The second term above appears to be bi-linear in the variables. However, because we have assumed linear process dynamics, we have:

$$x_{t+\tau}^i - x_{t+\tau}^j = A^\tau (x_t^1 - x_t^i),$$

where  $x_t^i, i \in [1, k]$  are the support points for the prior distribution at time  $t$  and therefore constant. As a result, the second term of Equation 7 is convex when  $Q(x)^{-1}$  is matrix concave in  $x$ . However, in order to express the problem in convex form, all convex constraints must be inequality constraints, not equality constraints. Therefore, we relax the constraint to become:

$$y_{\tau+1}^i \geq y_{\tau}^i - \|A^{\tau}(x_t^1 - x_t^i)\|_{Q(x_{\tau+1}^i)}^2. \quad (8)$$

The objective is to reach a belief state at time  $t + T$  such that Equation 4 is satisfied. This is accomplished by minimizing the average log weight,

$$J(t + T) = \frac{1}{k} \sum_{i=1}^k y_{t+T}^i. \quad (9)$$

The problem becomes:

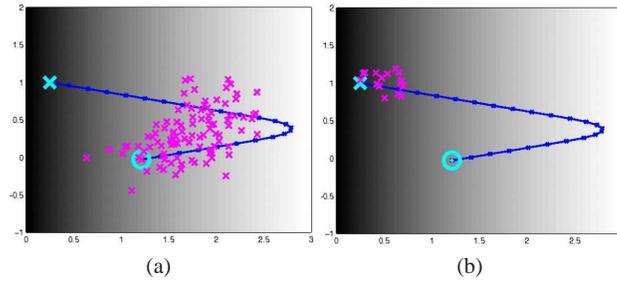
**Problem 1.**

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{k} \sum_{i=1}^k y_{t+T}^i + \alpha \sum_{\tau=1}^{T-1} u_{t+\tau}^T u_{t+\tau} \\ \text{subject to} \quad & x_{\tau+1}^i = Ax_{\tau}^i + Bu_{\tau}, i \in [1, k] \\ & y_{\tau+1}^i \geq y_{\tau}^i - \|A^{\tau}(x_t^1 - x_t^i)\|_{Q(x_{\tau+1}^i)}^2 \\ & x_t^i = x^i, w_t^i = 1, i \in [1, k] \\ & x_{t+T}^1 = x_{goal}. \end{aligned} \quad (10)$$

$\alpha$  is intended to be a small value that adds a small preference for shortest-path trajectories. It turns out that because the  $\frac{1}{k} \sum_{i=1}^k y_T^i$  term is part of the objective and the quadratic action cost term is never affected by  $y_T^i$ , that the relaxed inequality constraint on the log-weights (Equation 8) is always active (i.e. it is tight). As a result, there is effectively no relaxation. All solutions found to Problem 1 will satisfy the equality constraint in Equation 6. Problem 1 is a convex program when  $Q(x)^{-1}$  is matrix concave. When Algorithm 1 executes the convex plan in step 5, we mean that it solves an instance of Problem 1.

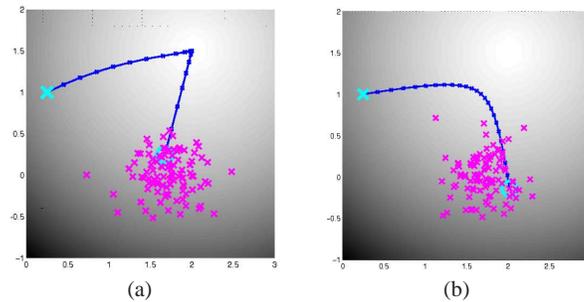
### 3.3 Examples: Single Beacon Domains

For example, Figure 1 illustrates the solution to Problem 1 for the a variant on the “light-dark” domain introduced in [11]. The cost function in this problem (Equation 10) uses  $\alpha = 0.4$ . In this problem, the process dynamics have unit transition matrices,  $A = I$  and  $B = I$  in Equation 1, with Gaussian process noise with a variance of 0.01. The observation dynamics are defined in Equation 6 with the following



**Fig. 1** Solution to the light-dark problem found using convex optimization. The magenta “x”s denote the support points in the initial (a) and final (b) distributions that had above average weight. The distribution is represented by 1000 support points. Entropy of the distribution is clearly much lower in (b) than it is in (a).

hyperbolic state dependent noise,  $Q(x) = 1/(1 + 2x)$ ,  $x \geq 0$ . The observation noise is illustrated by the shading in Figure 1. Noise is relatively high on the left side of state space and relatively small on the right. The system starts in an initial belief state described by a mixture of two Gaussians with means  $(1.75, 0)$  and  $(2, 0.5)$  and equal variances of  $0.0625$ . The distribution is represented by a set of 1000 support points. The solid blue line in Figure 1 denotes the planned trajectory of the hypothesis sample starting at the cyan circle and ending in the cyan “x” mark. For the hyperbolic observation noise given above, Problem 1 becomes a quadratic program. We solved it (parametrized by 1000 support points) using the IBM ILOG CPLEX Optimizer version 12.3 in 0.08 seconds on a dual-core 2.53GHz Intel machine with 4G of RAM. The magenta “x” marks show the locations of the importance samples with weights greater than or equal to the average. Initially (Figure 1(a)), the sampled distribution has a high entropy. However, entropy drops significantly so that by the end of the trajectory (Figure 1(b)), entropy is much lower.



**Fig. 2** Single-beacon problem solved for two different cost functions. (a) shows a solution found using a cost function (Equation 10) with  $\alpha = 1$ . (b) shows the solution found with a cost parameter  $\alpha = 5$ . The magenta “x”s denote the support points in the initial distribution.

The single-beacon domain is another example of a convex belief space planning problem. In this problem, the covariance of observation noise is a function of the L2 distance from a single point in state space (*i.e.* the beacon). Figure 2 illustrates the solution to this problem. The beacon is located at  $x_{beacon} = (2, 1.5)'$ . The covariance of the observation noise varies as a hyperbolic function of the L2 distance from the beacon center:  $Q(x) = 1/(6 - \|x - x_{beacon}\|_2)$ ,  $\|x - x_{beacon}\|_2 < 6$ . For this noise function, Problem 1 can be expressed as a second order cone program where the constraint on the domain of  $x$  is interpreted as a second order cone constraint on the positions of all the support points. We solved this program using CPLEX in less than one second for a distribution with 100 support points.

### 3.4 Analysis of algorithm convergence

For systems with deterministic process dynamics, it is possible to show that Algorithm 1 reaches a goal region of belief space in the limit as long as a feasible horizon- $T$  path with cost  $J \leq \rho$  exists from any starting configuration,  $x_1 \in \mathbb{R}^n$ , and as long as the implementation of Bayesian filtering,  $G$ , used in the algorithm is sufficiently accurate. Let  $F_{\mathbf{u}}(x) = (x_1, \dots, x_T)$  denote the sequence of  $T$  states reached after taking actions,  $\mathbf{u} = (u_1, \dots, u_{T-1})$ , starting from state  $x_1 = x$ . Let

$$q_{x,\mathbf{u}}(\mathbf{z}) = N(\mathbf{z}|F_{\mathbf{u}}(x), \mathbb{Q}(x)) \quad (11)$$

be the probability of making the observations,  $\mathbf{z} = (z_1, \dots, z_T)$ , given that the system starts in state  $x$  and takes actions,  $\mathbf{u}$ , where  $\mathbb{Q}_{\mathbf{u}}(x) = \text{diag}(Q(x_1), \dots, Q(x_T))$ ,  $(x_1, \dots, x_T) = F_{\mathbf{u}}(x)$  denotes the block diagonal matrix comprised of the covariance matrices of observation noise on each time step. Let

$$p_{b,\mathbf{u}}(\mathbf{z}) = \int_{x \in \mathbb{R}^n} b(x) q_{x,\mathbf{u}}(\mathbf{z}) \quad (12)$$

be the probability of the observations given  $\mathbf{u}$  marginalized over  $b$ . Our analysis is based on the following two lemmas that are adapted from [10] and stated here without proof.

**Lemma 1.** *Suppose we are given an arbitrary sequence of  $T - 1$  actions,  $\mathbf{u}$ , an arbitrary initial state,  $x_1 \in \mathbb{R}^n$ , and an arbitrary initial distribution,  $b_1$ . Then, the expected probability of  $x_T$ , where  $(x_1, \dots, x_T) = F_{\mathbf{u}}(x)$ , found by recursively evaluating the deterministic Bayes filter update  $T - 1$  times is*

$$\mathbb{E}_{\mathbf{z}} \left\{ \frac{b_T(x_T)}{b_1(x_1)} \right\} \geq \exp(D_1(q_{\kappa,\mathbf{u}}, p_{b_1,\mathbf{u}}) - D_1(q_{\kappa,\mathbf{u}}, q_{x_1,\mathbf{u}})),$$

where  $\kappa$  denotes the true state at time  $t$  and  $D_1$  denotes the KL divergence between the arguments.

It is possible to bound the KL divergence between the two distributions,  $q_{\kappa, \mathbf{u}}$  and  $p_{b_1, \mathbf{u}}$  using the following lemma.

**Lemma 2.** *Suppose we are given a partially observable system with state dependent noise that satisfies Equation 2, an arbitrary sequence of  $T$  actions,  $\mathbf{u}$ , and a distribution,  $b_1$ . Suppose  $\exists \Lambda_1, \Lambda_2 \subseteq \mathbb{R}^n$  such that  $\forall x_1, x_2 \in \Lambda_1 \times \Lambda_2, \|F_{\mathbf{u}}(x_1) - F_{\mathbf{u}}(x_2)\|_{\mathbb{Q}_{\mathbf{u}}(x_2)}^2 \geq \varepsilon^2$  and  $\int_{x \in \Lambda_1} b_1(x) \geq \gamma, \int_{x \in \Lambda_2} b_1(x) \geq \gamma$ . Then*

$$\min_{y \in \mathbb{R}^n} D_1(q_{y, \mathbf{u}}, p_{b_1, \mathbf{u}}) \geq 2\eta^2 \gamma^2 \left(1 - e^{-\frac{1}{2}\varepsilon^2}\right)^2,$$

where  $\eta = 1/\sqrt{(2\pi)^n |\mathbb{Q}_{\mathbf{u}}(x_2)|}$  is the Gaussian normalization constant.

In order to apply Lemma 2, it is necessary to identify regions,  $\Lambda_1, \Lambda_2 \subseteq \mathbb{R}^n$  and a  $\varepsilon$  such that  $\forall x_1, x_2 \in \Lambda_1 \times \Lambda_2, \|F_{\mathbf{u}}(x_1) - F_{\mathbf{u}}(x_2)\|_{\mathbb{Q}_{\mathbf{u}}(x_2)}^2 \geq \varepsilon^2$ . Below, we show that these regions always exist for action sequences with a sufficiently small associated cost,  $J(t+T)$  (Equation 9).

**Lemma 3.** *Given a linear-Gaussian system with state dependent noise that satisfies Equations 1 and 2 and a set of  $k$  initial support points and weights,  $x_1^i, i \in [1, k]$  and  $w_1^i, i \in [1, k]$ , let  $\mathbf{u}$  be a sequence of  $T-1$  actions with cost  $J = \frac{1}{k} \sum_{i=1}^k \log w_T^i$ . Then,  $\exists j \in [1, k]$  such that  $\forall r \in \mathbb{R}^+, \forall \delta_1, \delta_2 \in B_n(r)$ :*

$$\|F_{\mathbf{u}}(x_1^1 + \delta_1) - F_{\mathbf{u}}(x_1^j + \delta_2)\|_{\mathbb{Q}_{\mathbf{u}}(x_1^j)}^2 \geq \left[\sqrt{-J} - 2r\sqrt{\lambda}\right]^2,$$

where  $B_n(r) = \{x \in \mathbb{R}^n; x^T x \leq r^2\}$ , and  $\lambda$  denotes the maximum eigenvalue of  $G^T \mathbb{Q}_{\mathbf{u}}(x_1^i)^{-1} G$  where  $G = (I(A)^T (A^2)^T \dots (A^{T-1})^T)^T$ .

*Proof.* A cost  $J$  implies that there is at least one support point,  $j \in [1, k]$ , such that  $\log w_T^j \leq J$ . This implies that  $\|G(x_1^j - x_1^j)\|_{\mathbb{Q}_{\mathbf{u}}(x_1^j)}^2 \geq J$ . Also, we have that  $\forall \delta_1, \delta_2 \in B_n(r), \|G(\delta_1 + \delta_2)\|_{\mathbb{Q}_{\mathbf{u}}(x_1^j)}^2 \leq 4r^2\lambda$ . Applying Lemma 4 in [9] gives us the conclusion of the Lemma.

In order to show convergence of Algorithm 1, we show that the likelihood of a region about the true state increases by a finite amount on each iteration of the outer *while* loop. Our proof only holds for deterministic systems (however, experimental evidence suggests that the approach is applicable to stochastic systems).

**Theorem 1.** *Suppose we have:*

1. a system described by Equations 1 and 2 where the process dynamics are required to be deterministic;
2. a prior distribution,  $b_1$ ;
3. a trajectory,  $\mathbf{u}_{\tau-1}$ , defined over  $k \geq 2$  support points with cost,  $J_{\tau-1}$ .

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

$$\exp \left[ 2\eta^2 \gamma^2 \left( 1 - e^{-\frac{1}{2}(\sqrt{-\log J} - 2r\sqrt{\lambda})^2} \right) \right] \quad (13)$$

relative to its value at time 1, where  $\eta = 1/\sqrt{(2\pi)^n |\mathbb{Q}_{\mathbf{u}}(x_1^1)|}$  is the Gaussian normalization constant,  $\gamma = \varepsilon \text{Vol}_n(r)$ ,  $\text{Vol}_n(r)$  is the volume of the  $r$ -ball in  $n$  dimensions, and  $\lambda$  is the maximum eigenvalue of  $G^T \mathbb{Q}_{\mathbf{u}}(x_1^1)^{-1} G$  where  $G = (I(A)^T (A^2)^T \dots (A^{T-1})^T)^T$ .

*Proof.* Lemma 3 gives us two samples,  $x^i$  and  $x^1$  such that  $\|F_{\mathbf{u}}(x_1^1 + \delta_1) - F_{\mathbf{u}}(x_1^j + \delta_2)\|_{\mathbb{Q}_{\mathbf{u}}(x_1^j)}^2$  is lower-bounded. Lemma 2 gives us a lower bound of  $D_1(q_y, p)$  by setting  $\varepsilon = \sqrt{-\log J} - 2r\sqrt{\lambda}$ . Lemma 1 gives us the conclusion of the Theorem by noting that  $D_1(q_{\kappa}, q_x) = 0$  when  $x = \kappa$ .

Since Theorem 1 shows that the probability of a region about the true state increases on each iteration of the *while* loop of Algorithm 1, we know that the algorithm eventually terminates in the belief space goal region, as stated in the following Theorem.

**Theorem 2.** *Suppose we have the system described by Equations 1 and 2 where the process dynamics are deterministic. Suppose that  $\forall x, x_g \in \mathbf{R}^n$ , there always exists a solution to Problem 1 from  $x$  to  $x_g$  such that  $J_T < \rho < 0$ . Then Algorithm 1 reaches the goal region of belief space and terminates with probability 1 when a sufficiently accurate implementation of Bayesian filtering is used to track belief state.*

*Proof.* Since the convex planner is globally optimal, we know that it will find a feasible plan with  $J_t < 0$  if one exists. According to Theorem 1, this implies that the likelihood of an  $r$ -ball for some  $r \in \mathbb{R}^+$  about the true state is expected to increase by a finite amount on each iteration of the *while* loop. Eventually, the goal region must contain at least  $\omega$  probability mass and the algorithm terminates in the goal region of belief space.

## 4 Non-Convex Observation Noise

The fact that some belief space planning problems can be solved efficiently as a convex program is important because it suggests that it might also be possible to solve efficiently non-convex problems comprised of a small number of convex parts. One way to accomplish this is to use mixed integer programming. Although integer programming (IP) is an NP-complete problem, there are branch-and-bound methods available that can be used to efficiently solve problems with “small” numbers of integer variables with good average-case running times. Moreover, since the complexity of the IP is insensitive to the dimensionality of the underlying state space or the belief space representation, this method can be used to solve high dimensional planning problems that cannot be solved in other ways.

### 4.1 Mixed Integer formulation

Suppose that state space contains  $q$  non-overlapping convex polyhedral regions,  $R_1 \cup \dots \cup R_q \subset \mathcal{X}$  such that  $Q(x)^{-1}$  is matrix concave in each region. In particular, suppose that,

$$Q(x) = \begin{cases} Q_1(x) & \text{if } x \in R_1 \\ \vdots & \\ Q_q(x) & \text{if } x \in R_q \\ Q_b(x) & \text{otherwise} \end{cases}$$

where  $Q_j^{-1}, i \in [1, k]$  is the convex noise function in region  $R_j$  and  $Q_b$  is the ‘‘background’’ noise function. For this noise function, Problem 1 is non-convex. To solve the problem using mixed integer programming, we define  $2 \times q \times (T - 1)$  binary variables:  $\gamma_t^j$  and  $\lambda_t^j, \forall j \in [1, q], t \in [1, T - 1]$ . These binary variables can be regarded as bit-vectors,  $\gamma_{1:T-1}^j$  and  $\lambda_{1:T-1}^j$ , that denote when the hypothesis state enters and leaves region  $j$ . Each bit-vector is constrained to be all zero, all one, or to change from zero to one exactly once. This representation of the transition time from one region to the next uses extra bits but will make it easier to express the constraints in the following. The time step on which  $\gamma_{1:T-1}^j$  transitions from zero to one denotes the time when the hypothesis state enters region  $j$ . Similarly, the time step on which  $\lambda_{1:T-1}^j$  transitions from zero to one denotes the time when the hypothesis state leaves region  $j$ . These constraints on the form of the bit-vectors are expressed:  $\gamma_{t+1}^j \geq \gamma_t^j, \forall j \in [1, q], t \in [1, T - 1]$  and  $\lambda_{t+1}^j \geq \lambda_t^j, \forall j \in [1, q], t \in [1, T - 1]$ . We also constrain the system to enter and leave each region exactly once and that the times when the hypothesis state is within different regions must be non-overlapping. This constraint is:  $\lambda_{1:T-1}^j \leq \gamma_{1:T-1}^i, \forall i \in [1, q]$ . The constraint that the regions are non-overlapping in time (the hypothesis state may not be in two regions at once) is:  $\sum_{j=1}^q (\gamma_{1:T-1}^j - \lambda_{1:T-1}^j) \leq 1$ .

Now that the constraints on the bit-vectors themselves have been established, we need to make explicit the relationship between the bit-vectors and the distribution,  $b_t(x)$ , encoded by the sample points and weights. First, we need to constrain the hypothesis state to be within the region  $R_j$  when  $\gamma_t^j - \lambda_t^j = 1$ . Suppose that each region,  $R_j$ , is defined by a set of  $\mu_j$  hyperplanes,  $r_j^1, \dots, r_j^{\mu_j}$ , such that  $x \in R_j$  iff  $(r_j^m)'x \leq b_j, \forall m \in [1, \mu_j]$ . When  $\gamma_t^j - \lambda_t^j = 1$ , we enforce at  $x_t^j \in R_j$  using the so-called ‘‘big-M’’ approach [13]:

$$\forall m \in [1, \mu_j], \quad (r_j^m)'x_t^\zeta \leq b_j + M(1 - (\gamma_t^j - \lambda_t^j)), \quad (14)$$

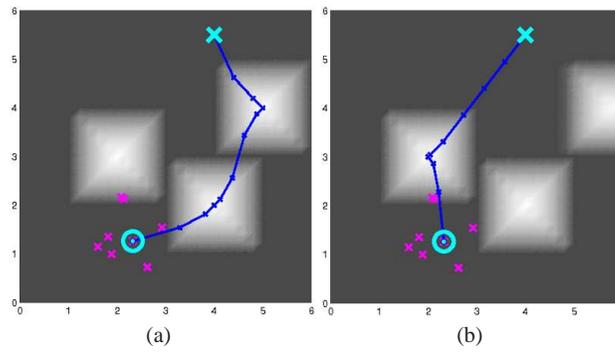
where  $M$  is defined to be a scalar large enough to effectively relax the constraint. Also, when the hypothesis state is in a given region, we need to apply the corresponding noise constraint. That is, when  $\gamma_{1:T-1}^j - \lambda_{1:T-1}^j = 1$ , we need to apply a constraint of the form of Equation 8. This is also accomplished using the big-M approach:

$$y_{t+1}^i \geq y_t^i - \|A^t(x_1^\zeta - x_1^i)\|_{Q_j(x_{t+1}^\zeta)} - M(1 - (\gamma_t^j - \lambda_t^j)). \quad (15)$$

When the hypothesis state is outside of all regions, then the background noise model is applied:

$$y_{t+1}^i \geq y_t^i - \|A^t(x_1^\zeta - x_1^i)\|_{Q_b(x_{t+1}^\zeta)} - M\left(\sum_{j=1}^q (\gamma_t^j - \lambda_t^j)\right). \quad (16)$$

## 4.2 Example: Multi-Beacon Domain



**Fig. 3** Multi-beacon problem. Three beacons are located at the centers of each of the white regions. Observation noise is proportional to the shading in the figure: lower about the beacons and higher further away. (a) and (b) illustrate the trajectories found by the planner in two scenarios where the beacons are placed in slightly different configurations.

We have explored the mixed-integer approach in the context of a multi-beacon localization problem. This problem is essentially a combination of three convex single-beacon problems and is illustrated in Figure 2. The scenario is as follows: a maneuverable aerial vehicle (such as a mini quad-rotor) is flying through the environment. The objective is to reach a neighborhood around a designated position with high confidence. Localization information is provided by beacons scattered throughout the environment. At each time step, the beacons report an estimate of the vehicle's position. The covariance of these estimates is state dependent: noise is small near the beacons but large further away. The covariance noise function is similar to that used in the single beacon domain, but it uses an L1 norm instead of an L2 norm. If the L1 distance between the system and any beacon,  $x_{beacon}^j, j \in [1, q]$ , is less than one ( $\|x - x_{beacon}^j\|_1 < 1$ ), then the covariance of

observation noise is:  $Q_j(x) = 1/(1 + \rho - \|x - x_{beacon}^j\|_1)$ , where  $\rho = 0.01$ . When  $\|x - x_{beacon}^j\|_1 > 1, \forall j \in [1, q]$ , observation noise is constant and equal to  $1/\rho$ .

The sampled distribution was comprised of 8 samples. The prior distribution was sampled from a Gaussian with a standard deviation of 0.25. The trajectory was 24 time steps long. The cost function in this experiment used  $\alpha = 0.25$ . The experiment used a variation of the bit-vector strategy described above where each bit was associated with four adjacent time steps. As a result, each obstacle was associated with two bit-vectors that were each 6 bits long. In total, there were 36 binary variables in the problem. The resulting problem was a mixed integer quadratic program (MIQP). As in the light-dark experiment, we solved it using IBM CPLEX on a dual-core 2.53GHz Intel machine with 4G of RAM. It took roughly 30 second to solve each of the MIQPs in this section.

The results are illustrated in Figure 3. The shading in the figure denotes the observation noise about each beacon. We performed the experiment for two different contingencies where the beacons were arranged slightly differently. In the first, Figure 3(a), the upper right beacon was positioned closer to the goal whereas in the second, Figure 3(b), the upper right beacon was positioned further away. Notice the effect on the resulting solution. The position of this second beacon influences the direction of the entire trajectory and illustrates that the MIQP is finding globally optimal solutions to Problem 1.

## 5 Chance Constraints

Chance constraints are an important part of belief space planning. It is assumed that obstacles are present in the state space that are to be avoided. The probability that the planned trajectory will cause the system to collide with an obstacle on any given time step is constrained to be less than a desired threshold. One approach to finding feasible trajectories, introduced by [13], is to use mixed integer linear programming in the context of polyhedral obstacles. However, the large number of integer variables needed results in a large integer program that can be difficult to solve. A key innovation, known as iterative deepening [5], improves the situation dramatically by adding constraints and integer variables as needed rather than all at once. This helps because most constraints in obstacle avoidance problems are inactive – they can be removed without affecting the solution. Blackmore has used this idea to solve chance constrained problems where belief state is represented by unweighted support points and where there are no observations [3]. The key innovation of this section is to extend this approach to avoiding obstacles to partially observable problems where sensing is active.

Our problem is to find a plan such that on every time step, the probability that the system has collided with an obstacle is less than a threshold,  $\theta$ . If there are  $q$  polyhedral obstacles,  $O_1, \dots, O_q$ , then this condition is (Equation 3):

$$P\left(x \in \cup_{j=1}^q O_j\right) \leq \theta.$$

Suppose that each obstacle,  $j \in [1, q]$ , is defined by  $v_j$  hyperplanes where the  $m^{\text{th}}$  hyperplane has a normal  $o_j^m$  and an intercept  $c_j^m$ . The condition that support point  $i$  be outside the  $m$  hyperplane in obstacle  $j$  is  $(o_j^m)^T x_t^i \leq c_j^m$ . The condition that the normalized weight of support point  $i$  be less than  $\theta/k$  is:

$$\frac{\exp(y_t^i)}{\sum_{n=1}^k \exp(y_t^n)} \leq \frac{\theta}{k}. \quad (17)$$

If the weight of each support point that is inside an obstacle is less than  $\theta/k$ , then we are guaranteed that the total weight of the contained support points is less than  $\theta$ . Unfortunately, since Equation 17 turns out to be a concave constraint, we cannot directly use it. Instead, we note that a simple sufficient condition can be found using Jensen's inequality:

$$y_t^i \leq \log(\theta) + \frac{1}{k} \sum_{n=1}^k y_t^n. \quad (18)$$

At this point, we are in a position to enumerate the set of constraints for all possible time steps, support points, obstacles, and obstacle hyperplanes. However, since each of these constraints is associated with a single binary variable, it is infeasible to solve the corresponding integer program for any problem of reasonable size. Instead, we use iterative deepening to add only a small set of active constraints.

Iterative deepening works by maintaining a set of active constraints. Initially, the active set is null. On each iteration of iterative deepening, the optimization problem is solved using the current active set. The solution is checked for particles that intersect an obstacle with a weight of at least  $\theta/k$ . Violations of the obstacle chance constraint are placed in a temporary list that contains the time step and the sample number of the violation. The violations are ranked in order of the amount by which they penetrate the obstacle. Constraints (and the corresponding binary variables) are added corresponding to the most significant violation. If sample  $i$  collided with obstacle  $j$  at time  $t$  on a particular iteration of iterative deepening, we create  $v_j + 1$  new binary variables,  $\psi_t^{i,j,m}$ ,  $m \in [1, v_j + 1]$ . Essentially, these  $v_j + 1$  binary variables enable us to formulate the following OR condition: either the sample is feasible for some hyperplane,  $m \in [1, v_j]$  (indicating the point is outside of the obstacle), or Equation 18 is true (indicating a below-threshold weight). Using the "big-M" approach, we add the following constraints:

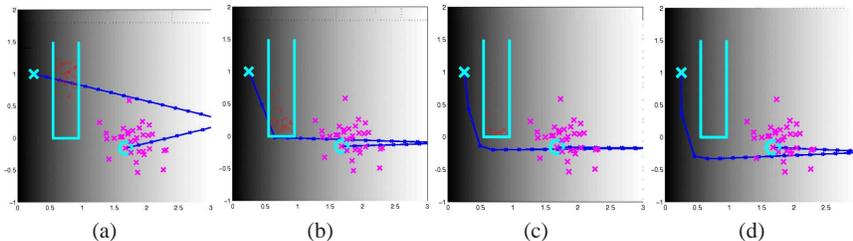
$$\sum_{m=1}^{v_j+1} \psi_t^{i,j,m} \geq 1, \quad (19)$$

$$(o_j^m)^T x_t^i \leq c_j^m + M(1 - \psi_t^{i,j,m}), m \in [1, v_j], \quad (20)$$

and

$$y_t^i \leq \log(\theta) + \frac{1}{k} \sum_{n=1}^k y_t^n + M(1 - \psi_t^{i,j,v_j+1}). \quad (21)$$

The constraint in Equation 19 forces at least one of the  $v_j + 1$  binary variables to be one. This forces at least one of the constraints in Equations 20 and 21 to *not* be relaxed. Essentially, the support point cannot be inside *all* faces of the obstacle while also having a large weight. This process of adding constraints via iterative deepening continues until no more violations are found. In principle, this procedure could become intractable if it is necessary to add all possible constraints in order to bring the violations list to null. In practice, however, this rarely happens.



**Fig. 4** Chance-constrained trajectories found on iterations 1 (a), 3 (b), 9 (c), and 25 (d) of iterative deepening. The obstacle is shown in cyan. The magenta “x”s show the initial distribution. The red “x”s in collision with the obstacle show the list of chance constraint violations at that particular iteration.

Figure 4 illustrates our approach to chance constraints in the presence of a single obstacle. With the exception of the obstacle, the domain is nearly identical to the light-dark domain in Section 3.3. The distribution is encoded using 32 weighted samples. There are 24 time steps. The cost function and the state-dependent noise are the same as those used in Section 3.3. The chance constraint threshold is  $\theta = 0.1$  – the probability that the system may be in collision with the obstacle cannot exceed 0.1 on a single time step. (The effective chance constraint may be tighter because of the sufficient condition found using Jensen’s inequality.) The sequence of sub-figures in Figure 4 roughly illustrates progress on different iterations of iterative deepening. Even though this problem requires  $32 \times 24 \times 3 = 2304$  binary variables in the worst case, we have found an optimal solution after adding only  $25 \times 3 = 75$  binary variables.

## 6 Conclusions

In this paper, we express the problem of planning in partially observable systems as a convex program. Since convex programs can be solved quickly (on the order of milliseconds) and with guaranteed convergence to global optima, this makes the approach suitable for receding horizon control. In consequence, we are able to identify a new class of partially observable problems that can be solved efficiently

and with provable correctness guarantees. We extend our approach to non-convex problems using mixed-integer programming. Although we are using integer programming, the approach is still appropriate for real-time applications by solving the integer part of the program in advance and optimizing the convex part of the program on-line.

## References

1. S. Arulampalam, S. Maskell, Gordon N., and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001.
2. D. Bertsekas. *Dynamic Programming and Optimal Control: 3rd Edition*. Athena Scientific, 2007.
3. L. Blackmore, M. Ono, A. Bektassov, and B. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, June 2010.
4. A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE Int'l Conf. on Robotics and Automation*, 2011.
5. M. Earl and r. D'Andrea. Iterative milp methods for vehicle control problems. In *IEEE Conf. on Decision and Control*, pages 4369–4374, December 2004.
6. 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.
7. C. Papadimitriou and J. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.
8. 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.
9. R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake. A hypothesis-based algorithm for planning and control in non-gaussian belief spaces. Technical Report CSAIL-TR-2011-039, Massachusetts Institute of Technology, 2011.
10. R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake. Non-gaussian belief space planning: Correctness and complexity. In *IEEE Int'l Conference on Robotics and Automation*, 2012.
11. 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.
12. 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.
13. A. Richards and J. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Controls Conference*, 2002.
14. J. van den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using differential dynamic programming in belief space. In *Int'l Symposium on Robotics Research*, 2011.