Non-Gaussian belief space planning as a convex program

Robert Platt Jr. and Russ Tedrake

Abstract—Many robotics problems are challenging because it is necessary to estimate state based on noisy or incomplete measurements. In these partially observable control problems, the system must act in order to gain information while attempting to maximize the probability of achieving task goals. Recently, there has been an effort to solve partially observable problems by planning in belief space, the space of probability distributions over the underlying state of the system. However, one of the challenges is to plan effectively in high-dimensional belief spaces. This paper identifies a class of partially observable problems with state dependent noise for which planning can be expressed as a convex program. We also propose a method by which non-convex problems that are composed of a small number of convex “pieces” can be solved using integer programming. The convex sub-structure means that these integer programs are small and can be solved efficiently using off-the-shelf solvers. We also extend the integer programming formulation to incorporate chance constraints on the probability of colliding with an obstacle. Our approach is demonstrated in the context of micro air vehicle single- and multi-beacon navigation problems.

I. INTRODUCTION

The problem of controlling partially observable systems in real-valued state, action, and observation spaces 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 using laser range or visual data. Similarly, it can be difficult for an autonomous quad-rotor micro air vehicle to localize itself using local sensing. Since state is never known exactly in these problems, the objective of control should be to maximize the probability that the robot achieves its goals. This is in contrast to fully observable problems where the control objective is simply to reach a particular goal state.

Partially observable control problems are often formulated as Partially Observable Markov Decision Processes (POMDPs) [1], [2]. Unfortunately, the problem of finding an optimal policy for a POMDP has been shown to be computationally intractable (PSPACE complete) [3]. Moreover, even many algorithms for approximating POMDP solutions do not scale well to large problems in real-valued state, action, and observation spaces. Recently, there has been a growing body of work that explores belief space planning approaches to the problem that avoid representing entire policies in belief space. For example, in prior work, we and others have used trajectory optimization to find plans in a Gaussian belief space based on an assumption that maximum likelihood observations would be perceived in the future [4], [5]. van der Berg et al. proposed a version of this approach that creates plans without making the maximum likelihood assumption [6]. This work has a lot in common with some of the early control work [7]. Recently, we have extended this approach to non-Gaussian distributions [8], [9]. Another set of approaches are based on finding paths through state space road maps. For example, Prentice and Roy search a probabilistic road map for paths with high information content [10]. Van der Berg et al. search a rapidly exploring random tree [11]. Bry and Roy use an optimal version of the Rapidly Exploring Random Tree known as RRT* [12].

One of the main challenges in belief space planning is identifying an appropriate planning algorithm. Neither of the two main approaches described above is ideal. Local optimization methods only find local solutions. It may be necessary to try the optimization from many different initial seeds in order to develop confidence in the quality of the resulting solution. Roadmaps in state space are not ideal either because these trajectories will almost certainly not be optimal belief space trajectories (even when the trajectories are optimal in state space). Additionally, since belief space planning methods implicitly require re-planning when the system diverges from the planned trajectory, it is important to be able to re-plan quickly.

This paper applies convex optimization to the planning problem. Convex optimization is potentially very attractive because it always finds globally optimal solutions and can typically be expected to plan quickly. The question is whether it is possible to express belief space planning problems of reasonable complexity as a convex program. It turns out that it is possible to express some belief space planning problems as convex programs. In this paper, we limit consideration to problems with unit observation dynamics and state dependent noise. Although these convex problems are too simple to be practically important in themselves, they can be used as “building blocks” to solve more complex problems. The idea is that interesting practical problems can be solved by combining the solutions to a small number of convex problems. In this paper, we use integer programming to combine the convex programs. Although integer programming is an NP-complete problem, a number of branch-and-bound methods exist that make it practical when the number of integer variables is small. In Section IV, we identify a class of belief space planning problems that can be solved using convex programming. In Section V, we propose a mixed integer programming method to solve non-convex problems that are comprised of a small number of convex pieces. Finally, in Section VI, we incorporate a method for imposing chance constraints.
constraints on obstacle collision into the problem.

II. PROBLEM STATEMENT

We consider a class of belief space control problems that are closely related to partially observable Markov decision processes (POMDPs). It is assumed that the underlying system is Markov where, on each time-step, the system takes an action. Although state is not observed directly, the system makes a partial or noisy observation on each time-step. Let state, \( x \in \mathcal{X} \), action, \( u \in \mathcal{U} \), and observation, \( z \in \mathcal{Z} \), be Euclidean column vectors. The process dynamics are linear-Gaussian of the form,

\[
x_{t+1} = Ax_t + Bu_t + v_t,
\]

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

\[
z_t = x_t + w_t(x_t),
\]

where \( w_t(x_t) \) is zero-mean Gaussian noise with state-dependent covariance, \( W(x_t) \). We require \( W(x_t)^{-1} \) to be piecewise matrix convex [13] in \( x \) (matrix convexity implies that \( a' W(x)^{-1} a \) is a convex function for any constant vector, \( a \)). 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 \( O_1, \ldots, O_q \subset \mathcal{X} \) 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 \( \theta \):

\[
P( x_t \in \bigcup_{n=1}^q O_n ) \leq \theta.
\]

In contrast with the general POMDP formulation that allows for an arbitrary objective function, this paper restricts consideration to a finite horizon problem where the objective is to reach a desired goal state, \( x_{goal} \), with high confidence at time \( T \). We assume that we are given a prior probability distribution, \( b_1(x) \), that describes what is known about the initial system state at time 1. As discussed in the next section, we refer to this distribution as a belief state. Let \( b_T(x) \) denote the distribution (i.e. belief state) over state at time \( T \) given all actions taken and observations perceived since time 1. Our objective is to minimize the differential entropy of \( b_T \) while reaching \( x_{goal} \) with the mode of \( b_T \):

\[
\text{Minimize} \quad - \int_{x \in \mathcal{X}} b_T(x) \log(b_T(x)) \quad \text{subject to} \quad \max_{x \in \mathcal{X}} b_T(x) = x_{goal}.
\]

III. BAYESIAN FILTERING

In partially observable systems, it is common to reason in terms of belief state [2], [1]. A belief state is a probability distribution over the underlying state of the system that incorporates information from prior actions and observations:

\[
b_t(x) = P(x|u_1:t - 1, z_{1:t}).
\]

Although the underlying system is non-Markov, a process expressed over belief states is Markov because the belief state is a sufficient statistic for the past history of actions and observations. This makes it convenient to plan solutions to partially observable problems in belief space (the space of belief states). However, this requires us to identify a representation of belief state. For problems expressed in a continuous state space, this can be challenging because the corresponding belief state is a probability distribution over a real-valued space. In this paper, we represent the distribution as a set of support points and corresponding weights using a version of particle filtering known as sequential importance sampling (SIS) [14].

At time \( t \), the distribution, \( b_t(x) \), is approximated by a set of \( k \) support points \( x_{tk}^1, \ldots, x_{tk}^k \) and the corresponding un-normalized weights, \( w_{tk}^1: \)

\[
b_t(x) = \sum_{i=1}^k w_{tk}^i \delta(x - x_{tk}^i),
\]

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,

\[
x_{tk+1}^i \sim N(Ax_t^i + Bu_t, V),
\]

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:

\[
w_{tk+1}^i = w_{tk}^i N(z_{t+1} - h(x_{tk+1}), Q).
\]

In this paper, we will track un-normalized weights. As a result, the above measurement update can be written:

\[
w_{tk+1}^i = w_{tk}^i \exp \left( - \frac{1}{2} \left\| z_{t+1} - h(x_{tk+1}) \right\|_Q^2 \right),
\]

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

IV. BELIEF SPACE PLANNING

It is extremely difficult to find optimal solutions to partially observable control problems because the control policy needs to be expressed over a belief space which is potentially very high dimensional. An alternative is to compute plans in belief space and re-plan when necessary [4], [8]. However, in order to plan, it is necessary to make assumptions regarding future stochasticity. Given that the system starts in a particular belief state at time \( t \) (represented by support points, \( x_{tk}^1 \) and weights \( w_{tk}^1 \)), we assume that stochasticity in future process dynamics and in future observation dynamics will unfold in a way that is determined ahead of time. Before planning, a sequence of future process noise, \( v_{tk}^1 \), is drawn from the process noise distribution for each support point, \( x_{tk}^i \). Also, it is assumed that future observations will be generated as if the system were actually in the currently most likely
state, $x^i_t$, where $\zeta = \max_{i \in [1,k]} w^i_t$. On all future time steps, $\tau > t$, it is assumed that $x^\zeta_\tau = x^\zeta_t$ (recall the unit observation dynamics of Equation 2). Pulling it all together, belief space planning is performed by assuming a process update,

$$x_{t+1}^\zeta = Ax_t^\zeta + Bu_t + \bar{v}_t^\zeta,$$

and an observation update,

$$u_{t+1}^\zeta = w_t^\zeta \exp \left( -\|x_{t+1}^\zeta - x^i_{t+1}\|^2_Q(x^i_{t+1}) \right),$$

We do not add “determinized” process noise to the hypothesis, $u$, Given the belief state at time $t$, we must find a sequence of actions, $u_{t:T-1}$, that minimizes the entropy at time $T$ (Equation 4) while satisfying the time $T$ constraint (Equation 5) and adhering to the belief dynamics constraints (Equations 10 and 11) and the chance constraints (Equation 3). In [9] and [8], we solved the trajectory planning problem using sequential quadratic programming on a direct transcription of the problem. In this paper, we identify a class belief space trajectory planning problems that can be expressed as convex programs and solved without resorting to methods that are only locally optimal. Upon first approaching the problem, one might identify the following problem variables: $x^i_t$, $w^i_t$, and $u_t$, for all $\tau \in [t,T]$ and $i \in [1,k]$. However, notice that the weight update in Equation 11 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, $y^i_t = \log(w^i_t)$ rather than the weights themselves. Equation 11 becomes:

$$y_{t+1}^i = y_t^i - \|x^\zeta_{t+1} - x^i_{t+1}\|^2_Q(x^\zeta_{t+1}).$$

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

$$x^\zeta_{t+1} - x^i_{t+1} = A^i(x_t^\zeta - x_t^i),$$

where $x_t^i, i \in [1,k]$ are the support points for the prior distribution and therefore constant. As a result, the second term of Equation 12 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:

$$y_{t+1}^i \geq y_t^i - \|A^i(x^\zeta_t - x_t^i)\|^2_Q(x^\zeta_{t+1}).$$

The objective is to minimize the differential entropy of the belief state on the final time step (Equation 4). However, since it is not convenient to minimize this quantity (it is not linear or quadratic), we instead minimize $\frac{1}{k} \sum_{i=1}^{k} y_t^i$. The problem becomes:

$$\begin{equation}
\begin{aligned}
\text{Minimize} & \quad \frac{1}{k} \sum_{i=1}^{k} y_t^i + \alpha \sum_{\tau=t}^{T-1} u_\tau u_\tau^T \\
\text{subject to} & \quad x_{\tau+1}^i = A x_{\tau}^i + B u_\tau + \bar{v}_\tau, i \in [1,k] \\
& \quad y_{\tau+1}^i \geq y_t^i - \|A^i(x^\zeta_{\tau} - x_{\tau}^i)\|^2_Q(x^\zeta_{\tau+1}) \\
& \quad x_{1}^i = x_{1}, u_{1}^i = 1, i \in [1,k] \\
& \quad \bar{v}_\tau = x_{goal}.
\end{aligned}
\end{equation}$$

$\alpha$ 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 13) 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 9. Problem 1 is a convex program when $Q(x)^{-1}$ is matrix concave.

**B. Experiment 1: Light-Dark Domain**

For example, Figure 1 illustrates the solution to Problem 1 for the a variant on the “light-dark” domain introduced in [4]. The cost function in this problem (Equation 14) uses $\alpha = 0.4$. In this problem, the process dynamics have unit transition matrices, $A = I$ and $B = I$ in Equation 8, with Gaussian process noise with a variance of 0.01. The observation dynamics are defined in Equation 9 with the following hyperbolic state dependent noise,

$$Q(x) = \frac{1}{1+2x}, \quad x \geq 0.$$
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.

C. Experiment 2: Single-Beacon Domain

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 \( x_{\text{beacon}} = (2, 1.5)^T \). The covariance of the observation noise varies as a hyperbolic function of the L2 distance from the beacon center:

\[
Q(x) = \frac{1}{6 - \|x - x_{\text{beacon}}\|_2^2}, \quad \|x - x_{\text{beacon}}\|_2 < 6. \tag{15}
\]

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.

V. 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 is an NP-complete problem, there are good branch-and-bound methods available that can be used to efficiently solve problems with “small” numbers of integer variables (the exact number of integer variables that is feasible depends upon problem constraints).

A. Mixed Integer formulation

Suppose that state space contains \( q \) non-overlapping convex polyhedral regions, \( R_1 \cup \cdots \cup R_q \subset \mathcal{X} \) 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: \( \gamma_{i,t}^j \) and \( \lambda_{i,t}^j \), \( \forall j \in [1, q], t \in [1, T - 1] \). 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 \( \gamma_{i,t}^j = 1 \) transitions from zero to one denotes the time when the hypothesis state enters region \( j \). Similarly, the time step on which \( \lambda_{i,t}^j = 1 \) 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:

\[
\gamma_{i,t+1}^j \geq \gamma_{i,t}^j, \forall j \in [1, q], t \in [1, T - 1] \tag{16}
\]

and

\[
\lambda_{i,t+1}^j \geq \lambda_{i,t}^j, \forall j \in [1, q], t \in [1, T - 1]. \tag{17}
\]

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:

\[
\lambda_{i,T-1}^j \leq \gamma_{1,T-1}^j, \forall i \in [1, q]. \tag{18}
\]

The constraint that the regions are non-overlapping in time (the hypothesis state may not be in two regions at once) is:

\[
\sum_{j=1}^q (\gamma_{1,T-1}^j - \lambda_{1,T-1}^j) \leq 1. \tag{19}
\]

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, \( b_j(x) \), encoded by the sample points and weights. First, we need to constrain the hypothesis state to be within the region \( R_j \) when \( x \in [1, q^j - \lambda_{i,t}^j] \). Suppose that each region, \( R_j \), is defined by at set of \( \mu_j \) hyperplanes, \( r_j^1, \ldots, r_j^{m_j} \), such that \( x \in R_j \) iff \( \langle r_j^m \rangle x \leq b_j, \forall m \in [1, m_j] \). When \( \gamma_{i,t}^j - \lambda_{i,t}^j = 1 \), we enforce at \( x_{\text{beacon}} \) in \( R_j \) using the so-called “big-M” approach [15]:

\[
\forall m \in [1, m_j], \quad (r_j^m)^T x_{\text{beacon}} \leq b_j + M(1 - (\gamma_{i,t}^j - \lambda_{i,t}^j)), \tag{20}
\]
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 $\gamma_{1:T-1}^{i} - \lambda_{1:T-1}^{i} = 1$, we need to apply a constraint of the form of Equation 13. This is also accomplished using the big-M approach:

$$y_{t+1}^{i} \geq y_{t}^{i} - \|A_{t}^{i}(x_{t}^{i} - x_{t}^{i})\|_{Q_{t}}(x_{t+1}^{i}) - M(1 - (\gamma_{t}^{i} - \lambda_{t}^{i})).$$

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

$$y_{t+1}^{i} \geq y_{t}^{i} - \|A_{t}^{i}(x_{t}^{i} - x_{t}^{i})\|_{Q_{t}}(x_{t+1}^{i}) - M\sum_{j=1}^{q} (\gamma_{t}^{j} - \lambda_{t}^{j}).$$

(22)

B. Experiment 3: Multi-Beacon Domain

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 Equation 15, but it uses an L1 norm instead of an L2 norm. If the L1 distance between the system and any beacon, $x_{j}^{beacon}, j \in [1, q]$, is less than one ($\|x - x_{j}^{beacon}\| < 1$), then the covariance of observation noise is:

$$Q_{j}(x) = \frac{1}{1 + \rho - \|x - x_{j}^{beacon}\|},$$

(23)

where $\rho = 0.01$. When $\|x - x_{j}^{beacon}\| > 1, \forall j \in [1, q]$, 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.

VI. 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. When belief state is represented by support points and associated real-valued weights, chance constraints are always non-convex. This is in contrast to the case where the belief distribution is represented by a Gaussian or where it is represented by unweighted support points [16], [17]. Essentially, a set of support points is constrained to be within the feasible region or have cumulative weights below a desired weight threshold. This “or” condition makes the problem non-convex, even when the feasible region is convex. Fortunately, this problem is easily solved by introducing only a few additional integer variables and the method neatly fits with existing work on the subject of using integer programming to encode chance constraints.

In the literature, chance constraints are typically used in systems where uncertainty is described by a Gaussian distribution [16], [18], [6]. In these cases, the chance constraint can be expressed as a constraint on the mean of the distribution as a function of the covariance [13]. In general, it is possible to find optimal free trajectories using mixed integer linear programming in the context of polyhedral obstacles. This technique was pioneered by [15] in the context of the fully observable problem. The basic idea is that a single binary variable is allocated for each possible time-step in the trajectory, each possible obstacle, and each possible face on the obstacle. The “big-M” constraint is used to enforce the constraints. Naturally, 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 [19], 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 [17].

A. Mixed Integer Formulation

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, \( O_1, \ldots, O_q \), then this condition is (Equation 3):

\[
P \left( x \in \bigcup_{j=1}^{q} O_j \right) \leq \theta.
\]

Suppose that each obstacle, \( j \in [1, q] \), is defined by \( \nu_j \) hyperplanes where the \( m^{th} \) hyperplane has a normal \( o_{j}^m \) and an intercept \( e_{j}^m \). The condition that support point \( i \) be outside the \( m \) hyperplane in obstacle \( j \) is:

\[
(o_{j}^m)^T x_i^t \leq e_{j}^m.
\]  

The condition that the normalized weight of support point \( i \) be less than \( \theta/k \) is:

\[
\frac{\exp(y_i^t)}{\sum_{n=1}^{k} \exp(y_i^n)} \leq \frac{\theta}{k}.
\]  

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 25 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:

\[
y_i^t \leq \log(\theta) + \frac{1}{k} \sum_{n=1}^{k} y_i^n.
\]  

At this point, we are in a position to enumerate the set of constraints formed by Equations 24 and 26 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 \( \nu_j + 1 \) new binary variables, \( \psi_{i,j,m}^t \), \( m \in [1, \nu_j + 1] \). Essentially, these \( \nu_j + 1 \) binary variables enable us to formulate the following OR condition: either Equation 24 is true for some hyperplane, \( m \in [1, \nu_j] \) (indicating the point is outside of the obstacle), or Equation 26 is true (indicating a below-threshold weight). Using the “big-M” approach, we add the following constraints:

\[
\sum_{m=1}^{\nu_j+1} \psi_{i,j,m}^t \geq 1,
\]

\[
(o_{j}^m)^T x_i^t \leq e_{j}^m + M(1 - \psi_{i,j,m}^t), m \in [1, \nu_j],
\]

and

\[
y_i^t \leq \log(\theta) + \frac{1}{k} \sum_{n=1}^{k} y_i^n + M(1 - \psi_{i,j,\nu_j+1}^t).
\]  

B. Experiment 4: A Polyhedral Obstacle

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 IV-B. 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 IV-B. 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. Figure 4(a) shows the initial trajectory found using a null active constraint set. The trajectory found is nearly the same as in Section IV-B. The magenta “x”s in Figure 4 illustrate the support points in the initial distribution. The red “x”s show the support points that violated chance constraints on this iteration. Notice the large number of chance constraint violations. After adding constraints at only two samples and time steps, the optimizer finds the trajectory shown in Figure 4(b). After five more iterations of iterative deepening, we arrive at the trajectory in Figure 4(c). At this point, the number of violations has significantly dropped. Finally, after 25 iterations, we arrive at the trajectory in Figure 4(d). 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.

It is important to understand the differences between what the above formulation achieves and other approaches to chance constraints that are based on Monte Carlo representations [17]. The key point is that we are enforcing chance constraints in a partially observable problem where it is possible to gain information by making observations. Whereas the work of [17] only allows for the possibility of losing information via process noise, the example above demonstrates a system that is gaining information while simultaneously avoiding obstacles.

VII. CONCLUSIONS

In this paper, we express the problem of planning in partially observable systems as a convex program. Convex programs can be solved quickly (on the order of milliseconds) and with guaranteed convergence to global optima. This is very important for belief space planning because the idea of planning in belief space (rather than finding a policy) rests on the assumption that it will be possible to re-plan if the system diverges from the originally planned trajectory. What is needed is a planning approach that can be quickly re-run under new initial conditions as the situation requires. Convex optimization fits these requirements well. Our approach is based on the idea of planning in belief space by determining future stochasticity developed in [8], [4]. After identifying a class of convex belief space planning problems, we show that non-convex problems that can be expressed as a combination of a small number of convex “pieces” can be solved using mixed integer programming. Because of the convex sub-structure, these mixed integer programs have few integer variables and can be solved efficiently. We also extend the approach to incorporate chance constraints.

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


