Simultaneous Localization and Grasping Using Belief Space Planning

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

***Abstract*— Most approaches to grasp planning assume that the configurations of the object to be grasped and any potential obstacles are known perfectly. As a result, implementations of these "perfect information" approaches to grasp synthesis are necessarily preceded by a perception stage where the the state of the object and obstacles are estimated. Unfortunately, small perceptual errors during the perception stage can cause even good grasp plans to fail. A potentially more robust approach would be to combine perception and grasp synthesis in a single process. Our paper formalizes this as the simultaneous localization and grasping (SLAG) problem. Our formalization casts SLAG as an instance of the belief space planning problem where the robot must reach a configuration where it is very likely that the robot has achieved the task (*i.e.* grasp) objectives. We apply an approach to belief space planning that is effective in non-Gaussian belief spaces. The approach is demonstrated in a SLAG problem where a robot must locate a cardboard box using a wrist-mounted laser scanner. If a second box occludes the box to be grasped, the robot must push the occlusion out of the way. We empirically demonstrate that our approach is capable of solving this problem and present results over many trials.**

I. Introduction

Creating robot grasping or manipulation plans in perfectly-observed environments is a well-studied problem. Given perfect information regarding the position of an object to be grasped and the associated obstacles, a number of techniques exist that can be used to plan arm motions [1], [2] and grasp configurations [3], [4], [5] that will accomplish the grasp. However, it is difficult to apply these ideas directly to real-world grasping problems because the perfect information assumption is rarely met. For example, it is possible in some situations to estimate object pose and shape using overhead cameras. However, it is common for these methods to fail due to occlusions, shape uncertainty, or a lack of distinctive perceptual features. In general, it is sometimes necessary to interact with the world in order to gain the sensor information needed to perceive the locations of objects to be grasped accurately enough. For example, it might be necessary to move a sensor or to push an object in order to perceive the information needed to grasp sufficiently accurately.

One way to formalize the problem of grasping under uncertainty is as a partially observable Markov decision process (POMDP) [6]. Unfortunately, finding optimal solutions to POMDPs has been shown to be PSPACE hard in the worst case [7]. Polynomial time approximations to the optimal solution can be found using point-based POMDP solver algorithms [8], [9]. However, most of these algorithms require discretization of the state, action, and observation spaces. The application of these techniques to grasping requires significant abstraction of the underlying grasp synthesis problem [10]. Instead, this paper applies recent ideas from the belief space planning literature [11], [12]. A belief-state is a probability distribution over an underlying state space that represents the state of information of the robot. The goal of belief space planning is to terminate in a belief state where a greater-than-threshold amount of probability mass is contained in a goal region. This paper summarizes a new approach to belief space planning that is capable of planning in high-dimensional non-Gaussian belief spaces and applies it to the grasp synthesis problem. We apply this approach to a simultaneous localization and grasping problem where a robot must take actions to estimate the position and shape of an object during the grasp synthesis process. We call this the SLAG problem (simultaneous localization and grasping).

This paper investigates a particular example where robot must localize a cardboard box using a wrist-mounted laser scanner while moving into a grasp configuration. In our scenario, the algorithm generates robot arm trajectories that gain information by "scanning" the boxes using the laser scanner. However, the possibility that an additional box has been placed adjacent to the first box complicates localization. The additional box could prevent the laser scanner from perceiving the extent of the box to be grasped. If the algorithm finds that the first box cannot be localized accurately due to the second box, it may choose to push the second box out of the way in order to facilitate localization. The algorithm continues to execute until the likelihood of having incorrectly localized the first box drops below a threshold. The approach is tested over a range of randomly selected box configurations that are initially unknown to the system.

II. SLAG as a belief-space planning problem

We view simultaneous localization and grasping (SLAG) as an instance of the belief space planning problem. The goal of belief space planning is to reach a belief state where the likelihood that task goals have been meet exceeds a given threshold. In SLAG, we are interested reaching a belief state where we are very confident that the grasp has succeeded. This might be accomplished in different ways. For example, the robot might become more certain of a successful grasp by actively perceiving the location of the object. Alternatively, a robot might move in such a way that the entire prior probability distribution is "captured" within the grasp.

The authors are with the Computer Science and Artificial Intelligence Laboratory at MIT.

A. System

Consider a discrete-time system with continuous non-linear deterministic process dynamics,

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

defined over a state space embedded in an n-dimensional Euclidean space, $\mathcal{X} \subseteq \mathbb{R}^n$, and a continuous action space, $\mathcal{U}$. We will find it useful to integrate the process dynamics over time. Let $F(x, u_{1:\tau-1})$ be the state at time τ if the system begins in state x at time 1 and takes actions $u_{1:\tau-1} = \{u_1, \ldots, u_{\tau-1}\}$. Denote the true state of the system at time t as κ_t. In the context of SLAG, state denotes the configuration of the object to be grasped relative to the robot manipulator. We are also given continuous non-linear stochastic observation dynamics,

$$z_t = h(x_t) + v_t, \tag{2}$$

where the observations are embedded in an m-dimensional Euclidean space, $\mathcal{Z} \subseteq \mathbb{R}^m$, and $v_t \sim N(0, Q)$ is zero-mean Gaussian noise with variance Q. The Jacobian matrix of h at x will be denoted $H(x) = \partial h(x)/\partial x$. In SLAG, the observation function describes the range, visual, or tactile measurements we expect to perceive.

B. Bayesian filtering

Although the true system state is not directly observed, it is possible to track a probability distribution over state using Bayesian filtering. At time t, suppose that x is a random variable that denotes system state with probability distribution $P(x)$. If the system takes action u, and subsequently observes the measurement z, then the probability distribution becomes:

$$P(x'|z, u) = \frac{1}{P(z)} P(z|x') \int_x P(x'|x, u) P(x). \tag{3}$$

In this paper, $P(x)$ is described by a probability density function (a pdf), $\pi(x; b) : \mathcal{X} \to \mathbb{R}^+$ with parameters $b \in \mathcal{B}$. The parameters, b, of the pdf are called the *belief state* and the parameter space, $\mathcal{B}$, is called the *belief-space*. Equation 3 can be re-written using the deterministic process dynamics and the belief-space parametrization:

$$\pi(f(x, u_t); b_{t+1}) = \frac{\pi(x; b_t) P(z_{t+1}|x, u_t)}{P(z_{t+1})}. \tag{4}$$

In general, it is impossible to implement Equation 4 exactly using a finite-dimensional parametrization of belief-space. However, a variety of approximations exist in practice [13]. In particular, particle filtering is a popular choice for filtering applications in manipulation []. This paper assumes that the system designer will use whichever representation of belief-space and implementation of Bayes filtering that is most appropriate in the application scenario. The implementation of Equation 4 used by the designer will be written, $b_{t+1} = G(b_t, u_t, z_{t+1})$.

C. Objective

Starting from an initial belief state, b_1, the objective of belief-space planning is to achieve task objectives with a given minimum probability. Specifically, we want to reach a belief state, b_τ, at some time $\tau > 0$ and $r > 0$ such that

$$\int_{x \in B_r} \pi(x + x_g; b_\tau) > \omega,$$

where $B_r = \{\delta \in \mathcal{X}, \|\delta\|^2 \leq r^2\}$ denotes the r-ball in $\mathcal{X}$, $x_g \in \mathcal{X}$ denotes the goal state, and ω denotes the minimum probability of success. In the case of SLAG, x_g might describe a relative hand-object configuration that constitutes a good grasp (prior to grasping) and B_r describes the radius about that goal state from which the grasp will still succeed.

III. Planning algorithm

This section describes an extension of the approach proposed in [12] to non-Gaussian belief spaces. The algorithm iteratively creates and executes a series of belief-space plans. A re-planning step is triggered when, during plan execution, the true belief state diverges too far from the nominal trajectory.

A. Creating plans

The key to the approach is a mechanism for creating horizon-T belief-space plans that guarantees that new information is incorporated into the belief distribution on each planning cycle. The basic idea is as follows. Given a prior belief state, b_1, define a "hypothesis" state be at the maximum of the distribution,

$$x^1 = \arg\max_{x \in \mathcal{X}} \pi(x; b_1).$$

Then, sample $k - 1$ states from the prior distribution,

$$x^i \sim \pi(x; b_1), i \in [2, k], \tag{5}$$

such that the pdf at each sample is greater than a specified threshold, $\pi(x^i; b_1) \geq \varphi \geq 0$, and there are at least two unique states (including α). We search for a sequence of actions, $u_{1:T-1}$, 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, consider that the expected observation (Gaussian observation noise) upon arriving in state x_t at time t is $h(x_t)$ (Equation 2). If the system starts in state x and takes a sequence of actions, $u_{1:t-1}$, then we expect to see the following sequence of observations (written as a column vector):

$$hF(x, u_{1:t-1}) = \left(h(F(x, u_1))^T, \ldots, h(F(x, u_{1:t-1}))^T\right)^T.$$

We are interested in finding a sequence of actions over a planning horizon T, $u_{1:T-1}$, that maximizes the squared observation distance,

$$\|hF(x^i, u_{1:T-1}) - hF(x^1, u_{1:T-1})\|^2_{\mathbb{Q}},$$

summed over all $i \in [2, k]$, where $\|a\|_A = \sqrt{a^T A^{-1} a}$ denotes the Mahalanobis distance and $\mathbb{Q} = diag(Q, \ldots, Q)$ denotes a block diagonal matrix of the appropriate size composed of observation covariance matrices. The wider the observation distance, the more accurately Bayes filtering will be able to determine whether or not the true state is near the hypothesis in comparison to the other sampled states.

Notice that the expression for observation distance is only defined with respect to the sampled points. Ideally, we would like a large observation distance between a region of states about the hypothesis state and regions about the other samples. Such a plan would "confirm" or "disprove" regions about the sampled points - not just the zero-measure points themselves. We incorporate this objective to the first order by minimizing the gradient of the measurements, $HF(x^i, u_{1:T-1})$ for all $i \in [1, k]$, where we define

$$HF(x, u_{1:T-1}) = \left(H(F(x, u_1))^T, \ldots, H(F(x, u_{1:T-1}))^T\right)^T$$

to be the column of measurement function Jacobian matrices at each state along the trajectory given that we execute $u_{1:T-1}$. These dual objectives, maximizing measurement distance and minimizing measurement gradient, are simultaneously optimized by minimizing the following cost function:

$$\bar{J}(x^{1:k}, u_{1:T-1}) = \frac{1}{k} \sum_{i=1}^{k} e^{-\Phi(x^i, u_{1:T-1})}, \tag{6}$$

where $x^{1:k} = \{x^1, \ldots, x^k\}$ and

$$\Phi(x, u_{1:T-1}) = \|hF(x, u_{1:T-1}) - hF(x^1, u_{1:T-1})\|^2_{\Gamma(x, u_{1:T-1})}.$$

The weighting matrix (*i.e.* the covariance matrix) above is defined

$$\begin{aligned}\Gamma(x, u_{1:T-1}) =& 2\mathbb{Q} + HF(x, u_{1:T-1}) HF(x, u_{1:T-1})^T \\ &+ HF(x^1, u_{1:T-1}) HF(x^1, u_{1:T-1})^T.\end{aligned}$$

In order to find plans that minimize Equation 6, it is convenient to restate the problem in terms of finding paths through a parameter space. Notice that for any positive semi-definite matrix, A, and vector, x, we have $x^T A x \geq x^T \bar{A} x$, where $\bar{A}$ is equal to A with all the off-diagonal terms set to zero. Therefore, we have the following lower-bound,

$$\Phi(x^i, u_{1:t-1}) \geq \sum_{t=1}^{T-1} \phi(F(x^i, u_{1:t-1}), F(x^1, u_{1:t-1})),$$

where

$$\phi(x, y) = \frac{1}{2} \|h(x) - h(y)\|^2_{\gamma(x,y)}$$

and

$$\gamma(x, y) = 2Q + H(x) H(x)^T + H(y) H(y)^T.$$

As a result, we can upper-bound the cost, $\bar{J}$ (Equation 6), by

$$\begin{aligned}\bar{J}(x^{1:k}, u_{1:T-1}) &\leq \frac{1}{k} \sum_{i=1}^{k} e^{-\sum_{t=1}^{T-1} \phi(F(x^i, u_{1:t-1}), F(x^1, u_{1:t-1}))} \\ &= \frac{1}{k} \sum_{i=1}^{k} \prod_{t=1}^{T-1} e^{-\phi(F(x^i, u_{1:t-1}), F(x^1, u_{1:t-1}))}.\end{aligned} \tag{7}$$

As a result, the planning problem can be written in terms of a path through a parameter space, $(x_t^1, \ldots, x_t^k, w_t^1, \ldots, w_t^k) \in \mathbb{R}^{2k}$, where x_t^i denotes the state associated with the i^{th} sample at time t and the weight, w_t^i, denotes the "partial cost" associated with sample i. This form of the optimization problem is stated as follows.

Problem 1:

$$\text{Minimize} \quad \frac{1}{k} \sum_{i=1}^{k} \left(w_T^i\right)^2 + \alpha \sum_{t=1}^{T-1} u_t^2 \tag{8}$$

$$\text{subject to} \quad x_{t+1}^i = f(x_t^i, u_t), i \in [1, k] \tag{9}$$

$$w_{t+1}^i = w_t^i e^{-\phi(F(x^i, u_{1:t-1}), F(x^1, u_{1:t-1}))}, i \in [1, k] \tag{10}$$

$$x_1^i = x^i, w_1^i = 1, i \in [1, k] \tag{11}$$

$$x_T^1 = x_g. \tag{12}$$

Notice that we have included a quadratic action cost term in the cost function (Equation 8) scaled by α in order to favor short paths. Equation 9 encodes the constraints caused by the process dynamics. Equation 10 incorporates a weight update that incrementally constructs the product in Equation 7. Equations 11 and 12 incorporate the initial and final value constraints, respectively. Problem 1 can be solved using a number of planning techniques such as rapidly exploring random trees [14], differential dynamic programming [15], or sequential quadratic programming [16]. We use sequential quadratic programming to solve the direct transcription of Problem 1. The direct transcription solution will be denoted

$$u_{1:T-1} = \textsc{DirTran}(x^1, \ldots, x^k, x_g, T), \tag{13}$$

for samples, $x^1, \ldots, x^k$, goal state constraint, x_g, and time horizon, T. Sometimes, we will call DIRTRAN without the final value goal constraint (Equation 12). This will be written, $u_{1:T-1} = \textsc{DirTran}(x^1, \ldots, x^k, T)$. It is important to recognize that the computational complexity of planning depends only on the number of samples used (the size of k in step 3 of Algorithm 1) and not strictly on the dimensionality of the underlying space. This suggests that the algorithm is potentially efficient in high-dimensional belief spaces. It is a subject for future work to determine how algorithm performance falls off as the dimensionality of the underlying space increases.

B. Re-planning

After creating a plan, our algorithm executes it while tracking belief state using the specified belief-state update, G. If

the true 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. Steps 2 through 4 sample k states from the distribution with the hypothesis state, $x^1 = \arg\max_{x \in \mathcal{X}} \pi(x; b)$, located at the maximum of the prior distribution. The prior likelihood of each sample is required to be greater than a minimum threshold, $1 > \varphi \geq 0$. In step 5, CREATEPLAN creates a horizon-T plan, $u_{1:T-1}$, that optimizes Equation 6 and generates a nominal belief-space trajectory, $\bar{b}_{1:T}$. Steps 7 through 13 execute the plan. Step 9 updates the belief state given the new action and observation using the Bayes filter implementation chosen by the designer. Step 10 breaks plan execution when the actual belief state departs too far from the nominal trajectory, as measured by the KL divergence, $D_1\left[\pi(\bullet; b_{t+1}), \pi(\bullet; \bar{b}_{t+1})\right] > \theta$. The second condition, $\bar{J}(x^1, \ldots, x^k, u_{1:t-1}) < 1-\rho$, guarantees that the *while* loop does not terminate before a (partial) trajectory with cost $\bar{J} < 1$ executes. The outer *while* loop terminates when there is a greater than ω probability that the true state is located within r of the goal state:

$$\Theta(b, r, x_g) = \int_{\delta \in B_r} \pi(x_g + \delta; b) > \omega,$$

where B_r is the r-ball, $B_r = \{\delta \in \mathcal{X} : \|\delta\|^2 \leq r^2\}$.

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

1 **while** $\Theta(b, r, x_g) \leq \omega$ **do**
2 $\quad x^1 = \arg\max_{x \in \mathcal{X}} \pi(x; b)$;
3 $\quad \forall i \in [2, k], x^i \sim \pi(x; b) : \pi(x^i; b) \geq \varphi$;
4 $\quad \bar{b}_{1:T}, u_{1:T-1} =$ `CreatePlan`$(b, x^1, \ldots, x^k, x_g, T)$;
5 $\quad b_1 = b$;
6 $\quad$ **for** $t \leftarrow 1$ **to** $T-1$ **do**
7 $\quad\quad$ execute action u_t, perceive observation z_{t+1};
8 $\quad\quad$ $b_{t+1} = G(b_t, u_t, z_{t+1})$;
9 $\quad\quad$ **if** $D_1\left[\pi(x; b_{t+1}), \pi(x; \bar{b}_{t+1})\right] > \theta$ **and** $\bar{J}(\mathcal{G}, u_{1:t-1}) < 1 - \rho$ **then**
10 $\quad\quad\quad$ break
11 $\quad\quad$ **end**
12 $\quad$ **end**
13 $\quad b = b_{t+1}$;
14 **end**

Algorithm 1: Belief-space re-planning algorithm

Algorithm 2 shows the CREATEPLAN procedure called in step 4 of Algorithm 1. Step 1 calls DIRTRAN parametrized by the final value constraint, x_g. If DIRTRAN fails to satisfy the goal state constraint, then the entire algorithm terminates in failure. Step 2 creates a nominal belief-space trajectory by integrating the user-specified Bayes filter update over the planned actions, assuming that observations are generated by the hypothesis state. If this nominal trajectory does not terminate in a belief state that is sufficiently confident that the true state is located within r of the hypothesis, then a new plan is created in steps 4 and 5 that is identical to the first except that it does not enforce the goal state constraints. This allows the algorithm to gain information incrementally in situations where a single plan does not lead to a sufficiently "peaked" belief state. When the system gains sufficient information, this branch is no longer executed and instead plans are created that meet the goal state constraint.

Input : initial belief state, b, sample set, $x^1, \ldots, x^k$, goal state, x_g, and time horizon, T.
Output: nominal trajectory, $\bar{b}_{1:T}$ and $u_{1:T-1}$

1 $u_{1:T-1} =$ `DirTran`$(x^1, \ldots, x^k, x_g, T)$;
2 $\bar{b}_1 = b$; $\forall t \in [1 : T-1],\ \bar{b}_{t+1} = G(\bar{b}_t, u_t, h(x_t^1))$;
3 **if** $\Theta(b, r, x_g) \leq \omega$ **then**
4 $\quad u_{1:T-1} =$ `DirTran`$(x^1, \ldots, x^k, T)$;
5 $\quad \bar{b}_1 = b$; $\forall t \in [1 : T-1],\ \bar{b}_{t+1} = G(\bar{b}_t, u_t, h(x_t^1))$;
6 **end**

Algorithm 2: CREATEPLAN procedure

IV. ILLUSTRATION OF SLAG

We evaluated this approach to SLAG in the context of a robot box grasping task.

A. Scenario

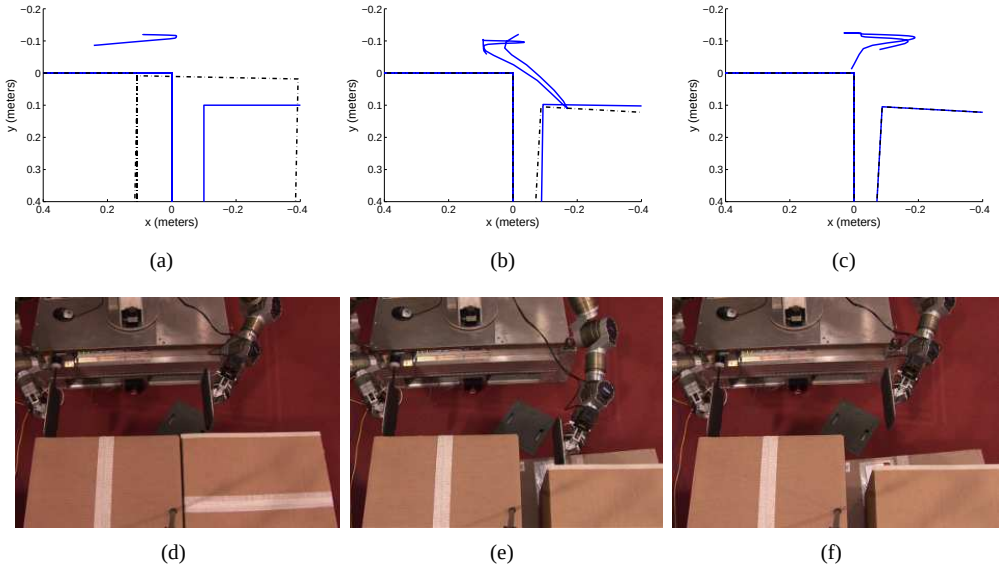
We evaluate Algorithm 1 in the context of the SLAG problem illustrated in Figure 1(a). Two boxes of unknown dimensions are presented to the robot. It is assumed that one of these boxes can be found in front of the left paddle. The objective is for robot to localize this box and move the left paddle into a "pre-grasp" configuration where the tip of the paddle is aligned with and just touching the left corner of the box. It is assumed that a mechanism exists to move the right paddle into opposition with the left as long as the left paddle is positioned correctly (for example, by closing slowly from a safe distance until a force is perceived). The robot is equipped with a laser scanner mounted to the wrist of the robot left arm that can be used to localize the boxes. The laser scanner produces range data in a plane over a 60 degree field of view. The left paddle of the robot is capable of reaching and pushing the left box.

B. Implementation

The algorithm localizes the planar pose of the two boxes (parametrized by a six-dimensional underlying metric space). The boxes are assumed to have been placed at a known height. The straightforward application of Algorithm 1 to our box problem is to create plans in a belief space defined over the six-dimensional underlying space. However, we reduce the dimensionality of the planning problem by introducing an initial perception step that localizes the depth and orientation of the right box using RANSAC [17]. From a practical perspective, this is a reasonable simplification because RANSAC is well-suited to finding the depth and orientation of a box that is assumed to be found in a known region of the laser scan. The remaining (four) dimensions that are not localized using RANSAC describe the horizontal dimension of the right box location and the three-dimensional pose of

(a) (b) (c)

Fig. 1. Illustration of the grasping problem, (a). The robot must localize the boxes using the laser scanner mounted on the left wrist. This is relatively easy when the boxes are separated as in (b) but hard when the boxes are pressed together as in (c).



(a) (b) (c)

(d) (e) (f)

Fig. 2. Example of a box localization task. In (a) and (d), the robot believes the gap between the boxes is large and plans to localize the boxes by scanning this gap. In (b) and (e), the robot has recognized that the boxes abut each other and creates a plan to increase gap width by pushing the right box. In (c) and (f), the robot localizes the boxes by scanning the newly created gap.

the left box. These dimensions are localized using a Bayes filter that updates a histogram distribution over the four-dimensional state space based on laser measurements and arm motions measured relative to the robot. The histogram filter is comprised of 20000 bins: 20 bins (1.2 cm each) describing right box horizontal position times 10 bins (2.4 cm each) describing left box horizontal position times 10 bins (2.4 cm each) describing left box vertical position times 10 bins (0.036 radians each) describing left box orientation. While it is relatively easy for the histogram filter to localize the remaining four dimensions when the two boxes are separated by a gap (Figure 1(b)), notice that this is more difficult when the boxes are pressed together (Figure 1(c)). In this configuration, the laser scans lie on the surfaces of the two boxes such that it is difficult to determine where one box end and the next begins. Note that it is difficult to locate the edge between abutting boxes reliably using vision or other sensor modalities – in general this is a hard problem.

Our implementation of Algorithm 1 used a set of 15-samples including the hypothesis sample. The algorithm controlled the left paddle by specifying Cartesian end-effector velocities in the horizontal plane. These Cartesian velocity commands were projected into the joint space using standard Jacobian Pseudoinverse techniques [18]. The algorithm was parametrized by accurate analytical models of the process and observation dynamics. The process dynamics described how the arms moved in response to the velocity commands. They also modeled the expected box motions produced by arm pushing trajectories based on assumptions that the paddle does not slip while pushing the box and assumptions regarding the location of the box center of friction. The observation dynamics describe the set of range measurements expected in a given paddle-box configuration. For planning purposes, the observation dynamics were simplified by modeling only a single forward-pointing scan rather than the full 60 degree scan range. However, notice that since this is a conservative estimate of future perception, low cost plans under the simplified observation dynamics are also low cost under the true dynamics. Nevertheless, the observation model used for *tracking* (step 8 of Algorithm 1) accurately described measurements from all (100) scans over the 60 degree range. The termination threshold in Algorithm 1 was set to 50% rather than a higher threshold because we found our observation noise model to overstate the true observation

noise.

Our hardware implementation of the algorithm included some small variations relative to Algorithm 1. Rather than monitoring divergence explicitly in step 9, we instead monitored the ratio between the likelihood of the hypothesis state and the next most probable bin in the histogram filter. When this ratio fell below 0.8, plan execution was terminated and the *while* loop continued. Since the hypothesis state must always have a maximal likelihood over the planned trajectory, a ratio of less than one implies a positive divergence. Second, rather than finding a non-goal directed plan in steps 3-5 of Algorithm 2, we always found goal-directed plans.

C. Example trajectory

Figure 2 illustrates an example of an information-gathering trajectory. The algorithm begins with a hypothesis state that indicates that the two boxes are 10 cm apart (the solid blue boxes in Figure 2(a)). As a result, the algorithm creates a plan that scans the laser in front of the two boxes under the assumption that this will enable the robot to perceive the (supposed) large gap. In fact, the two boxes abut each other as indicated by the black dotted lines in Figure 2(a). After beginning the scan, the histogram filter in Algorithm 1 recognizes this and terminates execution of the initial plan. At this point, the algorithm creates the pushing trajectory illustrated in Figure 2(b). During execution of the push, the left box moves in an unpredicted way due to uncertainty in box friction parameters (this is effectively process noise). This eventually triggers termination of the second trajectory. The third plan is created based on a new estimate of box locations and executes a scanning motion in front of the boxes is expected to enable the algorithm to localize the boxes with high confidence.

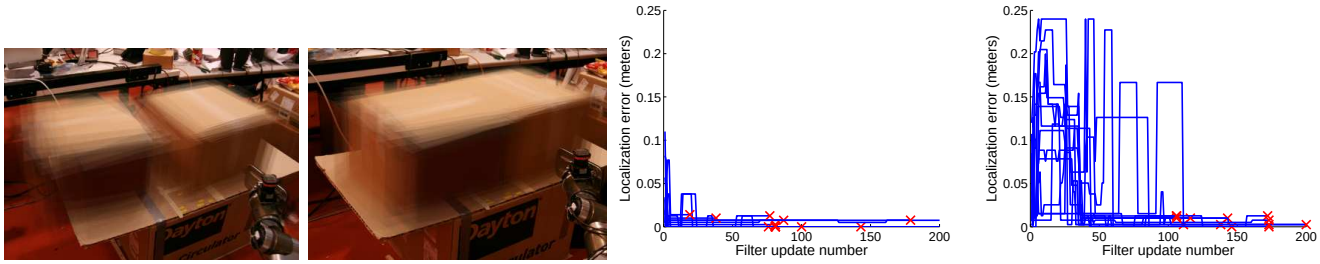
D. Expected Performance

At a high level, the objective of SLAG is to robustly localize and grasp objects even when the pose or shape of those objects is uncertain. We performed a series of experiments to evaluate how well this approach performs when used to localize boxes that are placed in initially uncertain locations. On each grasp trial, the boxes were placed in a uniformly random configuration (visualized in Figures 3(a) and (c)). There were two experimental contingencies: "easy" and "hard". In the easy contingency, both boxes were placed randomly such that they were potentially separated by a gap. The right box was randomly placed in a 13×16 cm region over a range of 15 degrees. The left box was placed uniformly randomly in a 20×20 cm region over 20 degrees measured with respect to the right box (Figure 3(a)). In the hard contingency, the two boxes were pressed against each other and the pair was placed randomly in a 13×16 cm region over a range of 15 degrees (Figure 3(b)).

Figures 3(c) and (d) show right box localization error as a function of the number of updates to the histogram filter since the trial start. 12 trials were performed in each contingency. Each blue line denotes the progress of a single trial. The termination of each trial is indicated by the red "X" marks. Each error trajectory is referenced to the ground truth error by measuring the distance between the final position of the paddle tip and its goal position in the left corner of the right box using a ruler. There are two results of which to take note. First, all trials terminate with less than 2 cm of error. Some of this error is a result of the coarse discretization of possible right box positions in the histogram filter (note also the discreteness of the error plots). Since the right box position bin size in the histogram filter is 1.2 cm, we would expect a maximum error of at least 1.2 cm. The remaining error is assumed to be caused by errors in the range sensor or the observation model. Second, notice that localization occurs much more quickly (generally in less than 100 filter updates) and accurately in the easy contingency, when the boxes are initially separated by a gap that the filter may used to localize. In contrast, accurate localization takes longer (generally between 100 and 200 filter updates) during the hard contingency experiments. Also error prior to accurate localization is much larger reflecting the significant possibility of error when the boxes are initially placed in the abutting configuration. The key result to notice is that even though localization may be difficult and errors large during a "hard" contingency, all trials ended with a small localization error. This suggests that our algorithm termination condition in step 1 of Algorithm 1 was sufficiently conservative. Also notice that the algorithm was capable of robustly generating information gathering trajectories in all of the randomly generated configurations during the "hard" contingencies. Without the box pushing trajectories found by the algorithm, it is likely that some of the hard contingency trials would have ended with larger localization errors.

V. Conclusion

When humans try to reconstruct the process of grasping introspectively from their own experiences, it is natural to decompose grasp synthesis into two stages: a perception stage followed by a grasping stage. Sometimes, a second haptic refinement stage is added to an initial gross perception stage [19], [20]. This general approach is based on the premise that certain types of information will be accurately perceived at certain stages of the grasp process. There is typically no convenient recourse if one of the perception stages should fail. The difficulty with which the research community has attempted to build robust grasping solutions suggests that we should re-examine this assumption. Certainly, grasp robustness might be improved if perception occurred during the entire grasping process and not just at certain times. Another possibility is for the robot to take actions that improve the information available to grasp synthesis. Our paper explores this approach in the context of a new problem statement, simultaneous localization and grasping (SLAG). Essentially, SLAG incorporates the perception process into the problem. It turns out that this can be conveniently expressed in terms of a belief space planning problem. In belief space planning, the objective is to reach a configuration where the algorithm strongly believes that it has achieved task (*i.e.* grasp) objectives. This paper



(a) (b) (c) (d)

Fig. 3. "Easy" and "hard" experimental contingencies. (a) shows images of the 12 randomly selected "easy" configurations (both box configurations chosen randomly) superimposed on each other. (b) shows images of the 12 randomly selected "hard" configurations (boxes abutting each other). (c) and (d) are plots of error between the maximum a posteriori localization estimate and the true box pose. Each line denotes a single trial. The red "X" marks denote localization error at algorithm termination.

describes a new sampled-based approach to planning in non-Gaussian belief spaces. Then, the approach is explored in the context of a particular SLAG problem where a robot must localize one of two boxes that are placed in front of it in unknown configurations. The algorithm generates information gathering trajectories that move the arm in such a way that a laser scanner mounted on the end-effector is able to localize the two boxes. The algorithm potentially pushes the boxes as necessary in order to gain information.

REFERENCES

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

[2] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int'l Journal of Robotics Research*, vol. 20, no. 5, 2001.

[3] A. Sudsang and J. Ponce, "New techniques for computing four-finger force-closure grasps of polyhedral objects," in *IEEE Int'l Conf. Robotics Automation*, vol. 2, May 1995, pp. 1355–1360.

[4] J. Ponce, S. Sullivan, A. Sudsang, J. Boissonnat, and J. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *Int. J. Rob. Res.*, 1996.

[5] V. Nguyen, "Constructing stable grasps in 3d," in *IEEE Int'l Conf. Robotics Automation*, vol. 4, March 1987, pp. 234–239.

[6] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

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

[8] 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.

[9] 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.

[10] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Task-driven tactile exploration," in *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[11] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Efficient planning in non-gaussian belief spaces and its application to robot grasping," in *Proceedings of the International Symposium on Robotics Research (submitted)*, 2011.

[12] 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.

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

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

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

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

[17] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[18] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, 2000.

[19] R. Platt, A. Fagg, and R. Grupen, "Null space grasp control: theory and experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, 2010.

[20] K. Hsiao, S. Chitta, M. Ciocarlie, and G. Jones, "Contact-reactive grasping of objects with partial shape information," in *IEEE Int'l Conf. on Intelligent Robots and Systems*, 2010.