Simultaneous Localization and Grasping as a Belief Space Control Problem

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. We formalize the problem of achieving this as simultaneous localization and grasping (SLAG). The objective of SLAG is to reach what is very likely to be a good grasp configuration given initial uncertainty by actively perceiving the environment. Given this formalization, SLAG can be viewed as an instance of the belief space control problem. This paper applies a new approach to belief space control that is effective in non-Gaussian belief spaces [17]. The approach is demonstrated in a SLAG problem where a robot must locate a cardboard box using a wrist-mounted laser scanner and grasp it. If a second box occludes the box to be grasped, the robot must either move the laser scanner or push the occluding box out of the way. We empirically demonstrate that our approach is capable of solving this problem and present results over many trials.

1 Introduction

Robot grasping in perfectly-observed environments is a well-studied problem. Given perfect information regarding the position of an object to be grasped and potential obstacles, a number of techniques exist that can be used to plan obstacle-free manipulator trajectories that reach a grasp configuration [1, 23, 19, 14]. However, it is difficult to apply these ideas directly to real-world grasping problems because the

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA, e-mail: {rplatt,lpk,tlp,russt}@csail.mit.edu

perfect information assumption is rarely met. For example, it is possible in some situations to estimate the pose and shape of an object to be grasped using overhead cameras. However, these estimates can be noisy or erroneous due to occlusions or a lack of distinctive visual features. In general, it is not reasonable to expect that it will always be possible to design a fixed sensing strategy that is robust for perceiving the exact shape and pose of all relevant objects in the environment. Sometimes, it is necessary to have the capability to adjust the perception strategy automatically based on new information. For example, the system designer may not be able to predict all the different ways that an object to be grasped is partially or fully occluded by other objects. After perceiving how a target object is occluded, it may be necessary for the system to decide how to avoid the occlusion, remove it, or otherwise localize the target object. This paper is concerned with the problem of automatically generating plans or policies for simultaneously localizing and grasping objects presented in difficult-to-perceive configurations.

Grasping under uncertainty is challenging because it is necessary for the system to reason about what it "knows" as well as about how to perform the task itself. The challenges here are a bit different than in the simultaneous localization and mapping (SLAM) problem. SLAM is essentially an estimation problem – the agent must localize its own position and estimate the geometry of a map. In contrast, grasping under uncertainty requires the robot to solve a control problem as well: the robot must act specifically in order to gain information. Recent work has proposed framing grasping under uncertainty as a partially observable Markov decision process (POMDP) [7]. POMDP solvers find policies defined in *belief space*, the space of probability distributions over the underlying system state. An optimal policy maximizes expected reward. Unfortunately, finding optimal solutions to POMDPs has been shown to be PSPACE-hard in the worst case [15]. Polynomial time approximations to the optimal solution can be found using point-based POMDP solvers [10, 20]. However, most of these algorithms require the state, action, and observation spaces to be discretized. Moreover, as a group, point-based POMDP algorithms are not generally tractable over long time horizons. As a result, the application of standard point-based algorithms requires the grasping problem to be discretized and adapted by a human designer so that the resulting planning problem can be solved by the planner.

Rather than adapting the grasping problem so that it can be solved using point-based POMDP algorithms, this paper applies a new hypothesis based belief space control algorithm proposed by the authors [17] that is capable of solving instances of the grasping problem without modification. First, we identify the particular problem of interest: *simultaneous localization and grasping* (SLAG). In contrast to the standard definition of the grasp problem which assumes it is possible to make a perfectly accurate measurement of grasp problem geometry, SLAG incorporates perception into the problem statement. The SLAG problem is to grasp an object with a guaranteed minimum probability of failure given models of the process (action) dynamics and observation dynamics. Given initial uncertainty in the state of the grasp scenario (for example, uncertainty in the position or shape of the object to be grasped), a solution to the SLAG problem will, before grasping, act in order to

perceive the environment in order to localize the object and bring the probability of grasp failure below a specified threshold. SLAG can be viewed as a particular instance of the belief space control problem where the objective is to reach a belief state where the probability that an object has been successfully grasped is high. Unlike our prior work and other belief space planning algorithms currently in the literature [18, 4, 24], this approach is guaranteed, under certain conditions, to reach the belief space goal in finite time [17]. This guarantee is valid even when belief state is non-Gaussian. Moreover, the planning step in our algorithm solves a lower-dimensional problem than the corresponding planning step in other belief space planning algorithms such as [18, 4]. After introducing the SLAG problem, this paper reviews the planning algorithm, provides a simple simulated illustration, and then applies it to a box grasping task where it is sometimes necessary to push an occluding object out of the way in order to properly localize and grasp the target box.

2 Problem Statement

This paper is concerned with the set of grasp problems where it is necessary both to perceive and grasp a given object. We assume that grasping may be modeled as a discrete-time Markov process where the process and observation dynamics of the system are completely known but some elements of the state vector may be uncertain. Let $x \in \mathbb{R}^n$ describe *grasp state*, the pose and shape of all bodies that may mechanically affect the grasp interaction (we assume that all relevant quantities can be embedded in Euclidean space). Grasp state may include the pose and shape of: the object to be grasped, potential obstacles, objects that can potentially become occlusions or otherwise affect perception, and the manipulator itself. From a practical perspective, the system designer must select a compact representation of grasp state that affords a Markov process model. Let $u \in \mathbb{R}^l$ be an action vector describing the robot command inputs. The process dynamics,

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

model how grasp state changes as a function of action and is assumed to be deterministic and non-linear. For example, if grasp mechanics are modeled as a first-order mechanical process, u would be a vector of desired arm, hand, and/or head velocities or forces and the process dynamics would model first-order pushing or grasping mechanics of the manipulator [12, 13]. We assume that grasp state is initially uncertain and that it is not directly perceived. Instead, the system must infer grasp state based on observations, $z \in \mathbb{R}^m$, made at each time step according to a known non-linear and stochastic observation function,

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

where v_t is zero-mean Gaussian noise with covariance, Q. For example, for laser range sensors, h describes the expected laser range measurements as a function of grasp state based on a model of the sensor. For force sensors, h describes the expected loads as a function of state based on a model of grasp mechanics.

The objective of *simultaneous localization and grasping* (SLAG) is to achieve a grasp configuration while bounding the probability of failure. In this context, the set of "grasp configurations" is a state or set of grasp states, $\mathscr{G} \subset \mathbb{R}^n$, identified by the system designer. It may denote the set of grasp states for which force closure [21] exists about the target object, the set of form closure states [3], or perhaps just a small region about a specific hand-object configuration. We assume that the system tracks a probability density function, $\pi(x;b)$, over grasp state. The parameter vector, $b \in \mathscr{B}$, is called *belief state*; the space of all belief states, $\mathscr{B}$, is known as *belief-space*. It is possible to track belief state over time as a function of actions and observations using Bayes filtering (written below assuming deterministic process dynamics):

$$\pi(f(x,u_t);b_{t+1}) = \frac{\pi(x;b_t)P(z_{t+1}|x,u_t)}{P(z_{t+1})}. \tag{3}$$

Starting from $\pi(x;b_1)$, the SLAG problem is to reach a configuration where the posterior distribution, $\pi(x;b_T)$, is very peaked about he grasp configurations, $\mathscr{G}$.

Problem 1 (Simultaneous localization and grasping).
Given:

1. a specification of grasp state, $x \in \mathbb{R}^n$;
2. models of the process and observation dynamics, f and h, of the grasp system;
3. a prior probability density function, $\pi(x;b_1)$, over grasp state;
4. a desired set of grasp configurations, $\mathscr{G} \subset \mathbb{R}^n$;
5. a desired probability of grasp success, $\omega > 0$;

take a sequence of actions, $u_1, \ldots, u_{T-1}$, such that

$$\Theta(b_T, \mathscr{G}) = \int_{x \in \mathscr{G}} \pi(x;b_T) \geq \omega, \tag{4}$$

where $\pi(x;b_T)$ denotes the posterior probability distribution over grasp state.

Equation 4 formalizes the requirement that the belief distribution be peaked about the goal state. We require more than ω probability mass to be within $\mathscr{G}$.

3 Belief space control algorithm

SLAG can be viewed as an instance of belief space control. This section reviews the belief space control algorithm proposed in [17] and provides a simple example.

3.1 Optimization problem

Belief space control casts the problem of achieving a desired posterior distribution, $\pi(x; b_T)$, in terms of finding a plan or policy through belief space that reaches a particular belief-state (or region of belief space). Reaching a goal belief state is challenging because the "dynamics" of the Bayes update (Equation 3) can be very stochastic, especially in high-entropy regions of belief space. Our 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. 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. Given a prior belief state, b_1, define a "hypothesis" state at the maximum of the distribution, $x^1 = \arg\max_{x \in \mathbb{R}^n} \pi(x; b_1)$. Then, sample $k-1$ states from the prior distribution, $x^i \sim \pi(x; b_1), i \in [2,k]$, such that the probability density function evaluated at each sample is greater than a specified threshold, $\pi(x^i; b_1) \geq \varphi \geq 0$, and at least two of the the k samples are unique. We search for a sequence of actions, $u_{1:T-1} = \{u_1, \ldots, u_{T-1}\}$, that result in as wide a margin as possible between the observations that would be expected if the system were in the hypothesis state and the observations that would be expected in any other sampled state. As a result, a good plan enables the system to "confirm" that the hypothesis state is in fact the true state or to "disprove" the hypothesis state. If, during plan execution, the hypothesis state is disproved, then the algorithm selects a new hypothesis on the next re-planning cycle. To be more specific, let $F_t(x, u_{1:t-1})$ be the state at time t if the system begins in state x and takes actions $u_{1:t-1}$. Recall that the expected observation upon arriving in state x_t is $h(x_t)$. Therefore, the expected sequence of observations is:

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

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

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

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

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 linearizing the system about the points and taking an expectation over isotropic Gaussian distributions with variance ρ centered on the points:

$$\begin{aligned}\bar{J}(x^1,\ldots,x^k,u_{1:T-1}) &= \sum_{i=2}^{k} \mathbb{E}_{y^i\sim N(\cdot|x^i,\rho),y^1\sim N(\cdot|x^1,\rho)} N\left(\mathbf{h}(y^i,u_{1:T-1})|\mathbf{h}(y^1,u_{1:T-1}),\mathbb{Q}\right) \\ &= \sum_{i=2}^{k} N\left(\mathbf{h}(x^i,u_{1:T-1})|\mathbf{h}(x^1,u_{1:T-1}),\Gamma(x,u_{1:T-1})\right), \end{aligned} \tag{5}$$

where

$$\Gamma(x,u_{1:T-1}) = 2\mathbb{Q}+\rho\mathbf{H}(x,u_{1:T-1})\mathbf{H}(x,u_{1:T-1})^T+\rho\mathbf{H}(x^1,u_{1:T-1})\mathbf{H}(x^1,u_{1:T-1})^T, \tag{6}$$

and we have defined the gradient of $\mathbf{h}$ to be $\mathbf{H}(x,u_{1:T-1}) = \partial\mathbf{h}(x,u_{1:T-1})/\partial x$. The objective of planning is to find a horizon-T trajectory, $u_{1:T-1}$, that minimizes Equation 5.

3.2 Planning algorithm

In order to find plans that minimize Equation 5, it is convenient to restate the problem in terms of finding paths through a parameter space. First, notice that Equation 5 is dominated by the exponentials in the Gaussian terms. Therefore, it is sufficient to minimize

$$J(x^1,\ldots,x^k,u_{1:T-1}) = \frac{1}{k}\sum_{i=1}^{k} e^{-\Phi(x^i,u_{1:T-1})}, \tag{7}$$

where $\Phi(x,u_{1:T-1}) = \|\mathbf{h}(x,u_{1:T-1})-\mathbf{h}(x^1,u_{1:T-1})\|^2_{\Gamma(x,u_{1:T-1})}$ and we denote the weighted 2-norm as $\|x\|^2_A = x^TA^{-1}x$. Given the low-rank structure of Equation 6 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 $\phi(x,y) = \frac{1}{2}\|h(x)-h(y)\|^2_{\gamma(x,y)}$, $\gamma(x,y) = 2Q+\rho H(x)H(x)^T+\rho H(y)H(y)^T$, and $H(x) = \partial h(x)/\partial x$. As a result, we can upper-bound the cost, J (Equation 7), by

$$\begin{aligned} J(x^1,\ldots,x^k,u_{1:T-1}) &\leq \frac{1}{k}\sum_{i=1}^{k} e^{-\sum_{t=1}^{T-1}\phi(F(x^i,u_{1:t-1}),F(x^1,u_{1:t-1}))} \\ &= \frac{1}{k}\sum_{i=1}^{k}\prod_{t=1}^{T-1} e^{-\phi(F(x^i,u_{1:t-1}),F(x^1,u_{1:t-1}))}. \end{aligned} \tag{8}$$

The planning problem can now be written in terms of a path through a parameter space, $(x_t^1,\ldots,x_t^k,w_t^1,\ldots,w_t^k)\in\mathbb{R}^{(n+1)k}$, 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 2.

$$\text{Minimize} \quad \frac{1}{k}\sum_{i=1}^{k}\left(w_T^i\right)^2+\alpha\sum_{t=1}^{T-1}u_t^2 \tag{9}$$

$$\text{subject to} \quad x_{t+1}^i=f(x_t^i,u_t), i\in[1,k] \tag{10}$$

$$w_{t+1}^i=w_t^i e^{-\phi(F(x^i,u_{1:t-1}),F(x^1,u_{1:t-1}))}, i\in[1,k] \tag{11}$$

$$w_1^i=1, x_1^i=x^i, i\in[1,k], x_T^1\in\mathscr{G}. \tag{12}$$

The objective function (Equation 9) includes a quadratic cost on action (scaled by α) in order to favor short paths. Equation 10 encodes the constraints caused by the process dynamics. Equation 11 incorporates a weight update that incrementally constructs the product in Equation 8. Equation 12 initializes the weights to one, initializes the sample set, and incorporates the final value constraint. Problem 2 can be solved using a number of planning techniques such as rapidly exploring random trees [11], differential dynamic programming [9], or sequential quadratic programming [2]. In this paper, we solve Problem 2 using trajectory optimization. We use sequential quadratic programming to solve the direct transcription of Problem 2. This solution will be denoted

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

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

The overall algorithm is outlined in Algorithms 1 and 2. We assume that the user provides an implementation of Bayes filtering that is used by the algorithm to track grasp state (step 8 of Algorithm 1). Steps 2 and 3 of Algorithm 1 sample points $x^2,\ldots,x^k$ from the distribution and set the hypothesis state, x^1, to the maximum of the prior distribution. Step 4 of Algorithm 1 calls the CREATEPLAN function (Algorithm 2). CREATEPLAN has two steps. First, it solves Problem 2 with the final value (last condition, Equation 12) constraint. Then, CREATEPLAN calculates a corresponding belief trajectory forward by assuming that the hypothesis state is equal to the true state. If the resulting trajectory does not reach a belief state that satisfies the *while* loop condition in step 1 of Algorithm 1, then CREATEPLAN solves Problem 2 again, this time without the final value constraint. Steps 6 through 12 of Algorithm 1 execute the plan. Step 8 updates the belief state given the new action and observation using $b_{t+1}=G(b_t,u_t,z_{t+1})$, the Bayes filter implementation chosen by the designer. Step 9 breaks plan execution when the actual belief state departs too far from the nominal trajectory and ensures that each iteration of the *while* loop gains information. The outer *while* loop terminates when the Equation 4 is satisfied.

Input : initial belief state, b, goal region, $\mathscr{G}$, planning horizon, T, and belief-state update, G.

while $\Theta(b,\mathscr{G}) \leq \omega$ **do**
  $x^1 = \arg\max_{x \in \mathbb{R}^n} \pi(x;b)$;
  $\forall i \in [2,k], x^i \sim \pi(x;b) : \pi(x^i;b) \geq \varphi$;
  $\underline{b}_{1:T}, u_{1:T-1} =$ `CreatePlan(`$b, x^1, \ldots, x^k, x_g, T$`);`
  $b_1 = b$;
  for $t \leftarrow 1$ **to** $T-1$ **do**
    execute action u_t, perceive observation z_{t+1};
    $b_{t+1} = G(b_t, u_t, z_{t+1})$;
    if $D_1\left[\pi(x;b_{t+1}), \pi(x;\underline{b}_{t+1})\right] > \theta$ **and** $J(x^1,\ldots,x^k,u_{1:t-1}) < 1-\rho$ **then**
      break
    end
  end
  $b = b_{t+1}$;
end

Algorithm 1: Belief-space re-planning algorithm

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

$u_{1:T-1} =$ `DirTran(`$x^1,\ldots,x^k,\mathscr{G},T$`);`
$\underline{b}_1 = b$; $\forall t \in [1:T-1]$, $\underline{b}_{t+1} = G(\underline{b}_t, u_t, h(x_t^1))$;
if $\Theta(b,\mathscr{G}) \leq \omega$ **then**
  $u_{1:T-1} =$ `DirTran(`$x^1,\ldots,x^k,T$`);`
  $\underline{b}_1 = b$; $\forall t \in [1:T-1]$, $\underline{b}_{t+1} = G(\underline{b}_t, u_t, h(x_t^1))$;
end

Algorithm 2: CREATEPLAN procedure

A couple of points regarding the correctness and computational complexity of Algorithm 1 bear noting. First, the approach can be shown to be asymptotically correct [17]. In particular, if the Bayes filter selected by the designer is exact and if DIRTRAN finds plans with $J < 1$ each time it executes, then the mode of the belief state asymptotically converges to the true state with probability one. Second, the dimensionality of the planning sub-problem (Problem 2) is bi-linear in number of samples and the dimensionality of the underlying state space (Problem 2 is expressed in $\mathbb{R}^{k(n+1)}$). This is an improvement over other belief space planning methods [18, 4] which solve planning sub-problems in a parameter space that is quadratically larger than the dimensionality of the underlying state space.

3.3 Illustration

Figures 1 and 2 illustrate belief space planning for part of a simple SLAG scenario. Figure 1 shows a horizontal-pointing laser mounted to the end-effector of a two-link robot arm. The end-effector is constrained to move only vertically along the dotted

line. The laser points horizontally and measures the range from the end-effector to whatever object it "sees". There are two boxes and a gap between them. Box size, shape, and relative position are assumed to be perfectly known along with the distance of the end-effector to the boxes. The only uncertain variable in this problem is the vertical position of the end-effector measured with respect to the gap position. This defines the one-dimensional state of the system and is illustrated by the vertical number line in Figure 1. The objective is to localize the vertical end-effector with respect to the center of the gap (state) exactly and move the end-effector to the center of the gap. The control input to the system is the vertical velocity of the end-effector.

Figure 2(a) illustrates an information-gathering trajectory found by DIRTRAN that is expected to enable the Bayes filter to determine whether the hypothesis state is indeed the true state while simultaneously moving the hypothesis to the goal state (end-effector at the center of the gap). The sample set used to calculate the trajectory was $x^1,\ldots,x^k = 5,2,3,4,6,7,8$, with the hypothesis sample located at $x^1 = 5$. The action cost used while solving Problem 2 was $\alpha = 0.0085$. DIRTRAN was initialized with a random trajectory. The additional small action cost smooths the trajectory by pulling it toward shortest paths without changing information gathering or goal directed behavior much. The trajectory can be understood intuitively. Given the problem setup, there are two possible observations: range measurements that "see" one of the two boxes and range measurements that "see" through the gap. The plan illustrated in Figure 2(a) moves the end effector such that different sequences of measurements would be observed depending upon whether the system were actually in the hypothesis state or in another sampled state.

Figures 2(b) and (c) show the nominal belief-space trajectory and the actual trajectory, respectively, in terms of a sequence of probability distributions superimposed on each other over time. Each distribution describes the likelihood that the system started out in a particular state given the actions taken and the observations perceived. The nominal belief-space trajectory in Figure 2(b) is found by simulating the belief-space dynamics forward assuming that future observations will be generated by the hypothesis state. Ultimately, the planned trajectory reaches a belief state distribution that is peaked about the hypothesis state, x^1 (the red "X"). In contrast, Figure 2(c) illustrates the actual belief-space trajectory found during execution. This trajectory reaches a belief state distribution peaked about the true state (the cyan "X"). Whereas the hypothesis state becomes the maximum of the nominal distribution in Figure 2(b), notice that it becomes a minimum of the actual distribution in Figure 2(c). This illustrates the main idea of the algorithm. Figure 2(b) can be viewed as a trajectory that "trusts" that the hypothesis is correct and takes actions

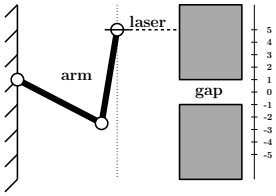

Fig. 1 SLAG scenario. The robot must simultaneously localize the gap and move the end-effector in front of the gap.

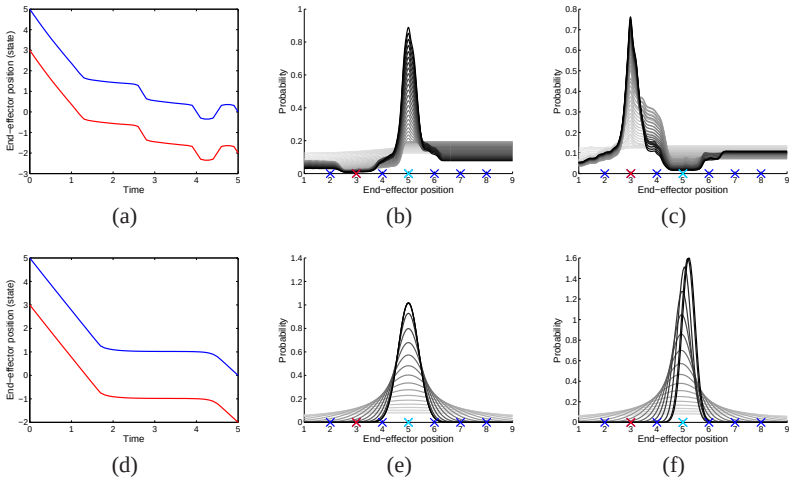

Fig. 2 Illustration of CREATEPLAN. (a) An information-gathering trajectory (state as a function of time) found using direct transcription. Blue denotes the trajectory that would be obtained if the system started in the hypothesis state. Red denotes the trajectory obtained starting in the true state. (b) The planned belief-space trajectory illustrated by probability distributions superimposed over time. Distributions early in the trajectory are light gray while distributions late in the trajectory are dark. The seven "X" symbols on the horizontal axis denote the positions of the samples (red denotes the true state while cyan denotes the hypothesis). (c) The actual belief-space trajectory found during execution. (d-f) Comparison with the EKF-based method proposed in [18]. (d) The planned trajectory. (e) The corresponding nominal belief-space trajectory. (f) Actual belief-space trajectory.

that confirm this hypothesis. Figure 2(c) illustrates that even when the hypothesis is wrong, the distribution necessarily gains information because it "disproves" the hypothesis state (notice the likelihood of the region about the hypothesis is very low).

Figure 2 (d-f) compares the performance of our approach with the extended Kalman filter-based (EKF-based) approach proposed in [18]. The problem setup is the same in every way except that cost function optimized in this scenario is:

$$J(u_{1:T-1}) = \frac{1}{10}\left(\sigma_T^2\right)^T \sigma_T^2 + 0.0085 u_{1:T-1}^T u_{1:T-1},$$

where σ_T^2 denotes covariance. There are several differences in performance. Notice that the two approaches generate different trajectories (compare Figures 2(a) and (d)). Essentially, the EKF-based approach maximizes the EKF reduction in variance by moving the maximum likelihood state toward the edge of the gap where the gradient of the measurement function is large. In contrast, our approach moves around the state space in order to differentiate the hypothesis from the other samples in regions with a small gradient. Moreover, notice that since the EKF-based approach

is constrained to track actual belief state using an EKF Bayes filter, the tracking performance shown in Figure 2(f) is very bad. The EKF innovation term actually makes corrections in the wrong direction. However, in spite of the large error, the EKF covariance grows small indicating high confidence in the estimate.

4 SLAG application

We apply our approach to an instance of the SLAG problem where it is necessary to localize and grasp a box. Two boxes of unknown dimensions are presented to the robot. The objective is to localize and grasp the box which is initially found directly in front of the left paddle. This is challenging because the placement of the two boxes may make localization of the exact position and dimensions of the boxes difficult.

4.1 Problem setup

(a) (b) (c)

Fig. 3 Illustration of the grasping problem, (a). The robot must localize the pose and dimensions of 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).

Our robot, *Paddles*, has two arms with one paddle at the end of each arm (see Figure 3(a)). Paddles may grasp a box by squeezing the box between the two paddles and lifting. We assume that the robot is equipped with a pre-programmed "lift" function that can be activated once the robot has placed its two paddles in opposition around the target box. Paddles may localize objects in the world using a laser scanner mounted to the wrist of its left arm. The laser scanner produces range data in a plane parallel to the tabletop over a 60 degree field of view.

We use Algorithm 1 to localize 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. We reduce the dimensionality of the planning problem by introducing an initial perception step that localizes the depth and orientation of

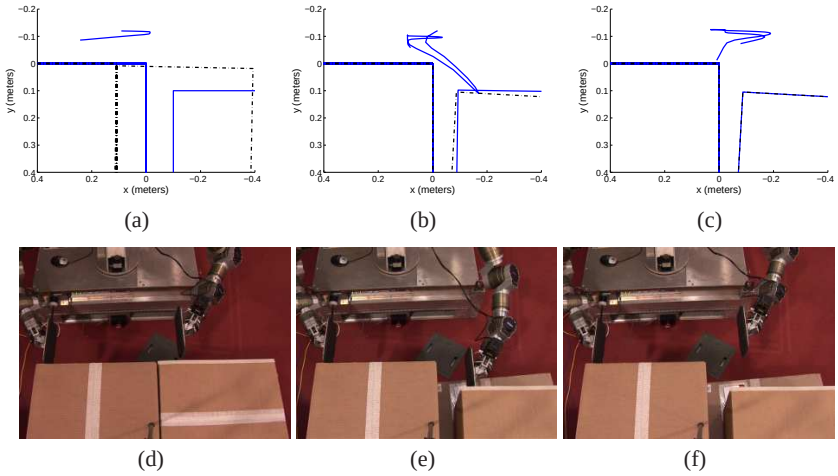

Fig. 4 Example of a box localization task. In (a) and (d), the robot believes the gap between the boxes is large and plans to localize the boxes by scanning this gap. In (b) and (e), the robot has recognized that the boxes abut each other and creates a plan to increase gap width by pushing the right box. In (c) and (f), the robot localizes the boxes by scanning the newly created gap.

the right box using RANSAC [5]. 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 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 3(b)), notice that this is more difficult when the boxes are pressed together (Figure 3(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 ends 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 [22]. The algorithm was parametrized by process dynamics that modeled

arms motions resulting from velocity commands and box motions produced by pushes from the arm. Box motions were modeled by assuming zero slip while pushing the box and assuming the center of friction was located at the center of the area of the box "footprint". The observation dynamics described 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.

Figure 4 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 4(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 4(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 4(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.

4.2 Localization 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 5(a) and (c)).

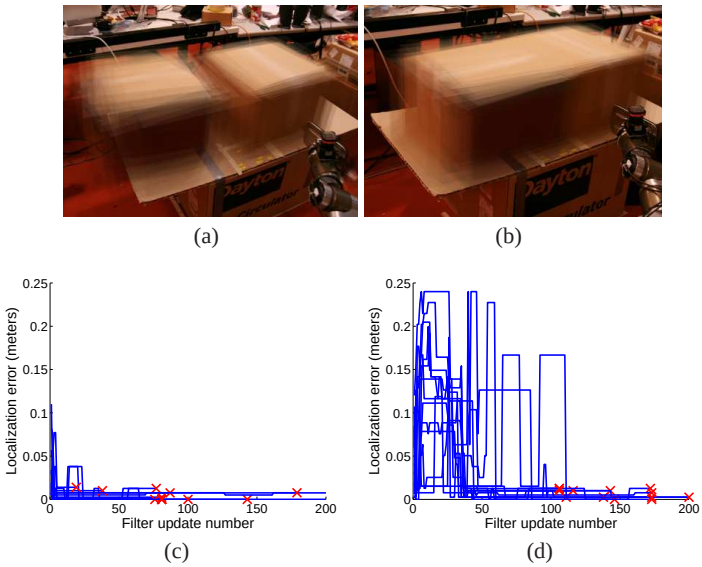

Fig. 5 "Easy" and "hard" experimental contingencies. (a) shows images of the 12 randomly selected "easy" configurations (both box configurations chosen randomly) superimposed on each other. (b) shows images of the 12 randomly selected "hard" configurations (boxes abutting each other). (c) and (d) are 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.

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 5(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 5(b)).

Figures 5(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. Sec-

ond, 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.

5 Discussion

When human designers 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 [16, 6]. This general approach is based on the premise that certain types of information will be accurately perceived at certain stages of the grasp process. However, the difficulty with which the research community has attempted to build robust grasping solutions suggests that we should re-examine this assumption. One way to improve grasp robustness is to perceive during the entire grasping process and not just at certain times. Another is for the robot to take actions that improve the information available to grasp synthesis. This paper explores these two ideas. We propose a reformulation of the grasping problem, known as simultaneous localization and grasping (SLAG), that incorporates perception into the grasping problem statement. SLAG is essentially a belief space control problem. This paper reviews a new belief space control algorithm developed by the authors [17], and applies it to a SLAG problem involving the grasping of cardboard boxes. The approach is empirically demonstrated to be capable of robustly grasping objects even when serious occlusions are present and even when it is necessary to interact with the objects in order to localize them properly.

References

1. D. Berenson, S. Srinivasa, and J. Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *Int'l Journal of Robotics Research*, March 2011.
2. J. Betts. *Practical methods for optimal control using nonlinear programming*. Siam, 2001.

3. A. Bicchi. On the closure properties of robotic grasping. *Int. J. of Robotics Research*, 14(4), 1995.
4. 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.
5. 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*, 24:381–395, 1981.
6. 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.
7. K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Grasping POMDPs. In *IEEE Int'l Conf. Robotics Automation*, April 2007.
8. K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Task-driven tactile exploration. In *Proceedings of Robotics: Science and Systems (RSS)*, 2010.
9. D. Jacobson and D. Mayne. *Differential dynamic programming*. Elsevier, 1970.
10. 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.
11. S. LaValle and J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, 2001.
12. M. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.
13. M. Mason and J. Salisbury. *Robot hands and the mechanics of manipulation*. MIT Press, 1985.
14. V. Nguyen. Constructing stable grasps in 3d. In *IEEE Int'l Conf. Robotics Automation*, volume 4, pages 234–239, March 1987.
15. C. Papadimitriou and J. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.
16. R. Platt, A. Fagg, and R. Grupen. Null space grasp control: theory and experiments. *IEEE Transactions on Robotics*, 26(2), 2010.
17. 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, Massachusetts Institute of Technology, 2011.
18. 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.
19. 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.
20. S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online planning algorithms for POMDPs. *The Journal of Machine Learning Research*, 32:663–704, 2008.
21. J. K. Salisbury, Jr. *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Stanford University, 1982.
22. L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.
23. A. Sudsang and J. Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *IEEE Int'l Conf. Robotics Automation*, volume 2, pages 1355–1360, May 1995.
24. J. Van der Berg, P. Abbeel, and K. Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. In *Proceedings of Robotics: Science and Systems (RSS)*, 2010.