Article

A data-driven statistical framework for post-grasp manipulation

Robert Paolini[1], Alberto Rodriguez[2], Siddhartha S. Srinivasa[1] and Matthew T. Mason[1]

Abstract

Grasping an object is usually only an intermediate goal for a robotic manipulator. To finish the task, the robot needs to know where the object is in its hand and what action to execute. This paper presents a general statistical framework to address these problems. Given a novel object, the robot learns a statistical model of grasp state conditioned on sensor values. The robot also builds a statistical model of the requirements for a successful execution of the task in terms of uncertainty in the state of the grasp. Both of these models are constructed by offline experiments. The online process then grasps objects and chooses actions to maximize likelihood of success. This paper describes the framework in detail, and demonstrates its effectiveness experimentally in placing, dropping, and insertion tasks. To construct statistical models, the robot performed over 8,000 grasp trials, and over 1,000 trials each of placing, dropping, and insertion.

Keywords

Robotic manipulation, grasping, post-grasp manipulation, robot learning, grasp estimation, data-driven models

1. Introduction

We study the problem of post-grasp manipulation, where a robotic manipulator performs a task with a grasped object, such as setting down a mug, inserting a key into a hole, or flipping a pancake with a spatula. In each of these examples, knowing the pose of the object in the robot's hand, the *grasp state*, is often critical. Intuitively, harder tasks demand a more accurate estimate of the grasp state than simpler ones. For example, in Figure 1, balancing a pen on a table requires more accuracy than dropping it into a container.

More generally, consider a manipulator, an object to manipulate, a task, and a parameterized set of actions designed to accomplish the task. In this paper we build a data-driven framework to automate the process of deciding whether the task is solvable with the available hardware and set of actions, and find the action most likely to succeed.

The statistical framework proposed in this paper is suited to model post-grasp manipulation tasks such as those described above. We model these tasks by breaking them into two independent steps. First, estimate the state of the grasp with available sensor information, and second, model the accuracy requirements that the particular task imposes on our state estimation. This separation yields the benefit that we can use the same model of state estimation for different tasks, and the same model of task requirements for different manipulators. Using this framework, each sensor reading generates a probability function in task action space, enabling us not only to find the optimal action, but to understand just how likely that action is to succeed.

Figure 2 illustrates the process for the task of placing an object. First, we use sensors in the hand to estimate the probability distribution of the pose of the grasped object, which will be referred to as the belief state. Second, we predict the probability of succeeding at a task given the pose uncertainty. Both of these are computed based on data-driven models. Finally, we combine these probability functions to predict the probability of success of each available action, and choose the action most likely to succeed.

In this paper we test the framework with three different manipulation tasks: placing an object, dropping it into a hole, and inserting it, all three described in Section 1.2. The experimental setup in Figure 2 consists of a simple gripper (Rodriguez et al., 2010; Mason et al., 2012) mounted on a robotic arm that iteratively grasps an object from a bin, estimates the distribution of the pose of the object, computes the probability of success for all available actions, chooses the optimal one, and executes it.

[1]Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, USA
[2]Department of Mechanical Engineering at Massachusetts Institute of Technology, Cambridge, MA, USA

Corresponding author:
Robert Paolini, Robotics Institute at Carnegie Mellon University, Newell Simon Hall, Pittsburgh, PA 15213, USA.
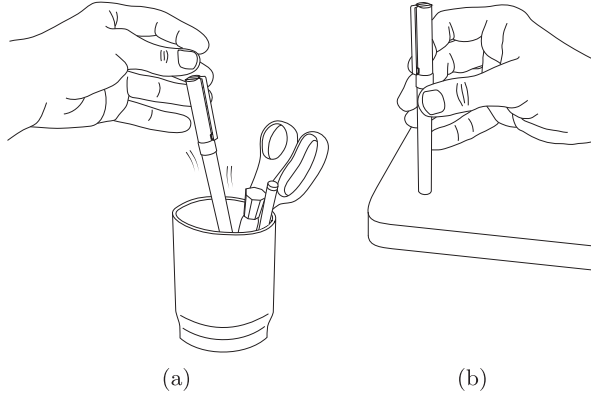Email: rpaolini@cmu.edu

(a) (b)

Fig. 1. Two manipulation tasks (a) dropping and (b) balancing, with different required accuracy of the pose of the manipulated object.

1.1. *Motivational simple example*

Let us first look at the simple example in Figure 3. A two-fingered planar gripper holds a rectangular object to insert it into a hole. We assume the object always makes face contact with the palm of the gripper, so that once it is in the hand, it has only one degree of freedom, which we call grasp state x. One of the fingers of the gripper has a noisy angle sensor z. The angle z informs us of the position of the rectangular block. The question we study in this paper is, given a sensor reading z, where should we move the gripper (how do we choose action a) to maximize the probability of success of inserting the object, and how likely are we to succeed. Note that there are two main factors that influence the probability of success:

- How accurately can we estimate the pose of the object? In the rest of the paper, we will refer to this factor as the *sensing capabilities* of the hand.
- What is our margin for error when inserting the object? In the rest of the paper we will refer to this factor as the *task requirements*.

Both are empirically modeled with the output of tailored experiments.

If there is no noise in the sensor reading z, and the geometries of both hand and object are known, we know with zero uncertainty the state of the grasp x, i.e. the location of the object. While not true in general, for the simple example in Figure 3 there is a one-to-one mapping between z and x, and we can therefore recover x perfectly. However, with the addition of noise, the estimate of object pose becomes a distribution $P(x|z)$, rather than an isolated configuration. Figure 4 shows how that distribution changes for three different sensor noise models $P(z|x)$. Section 3.1 details the process to derive the belief state $P(x|z)$ from the observation model $P(z|x)$.

The second factor to consider is the margin of error the task allows. Note that we can effectively change that margin of error by varying the shape of the hole. Figure 5(a) shows
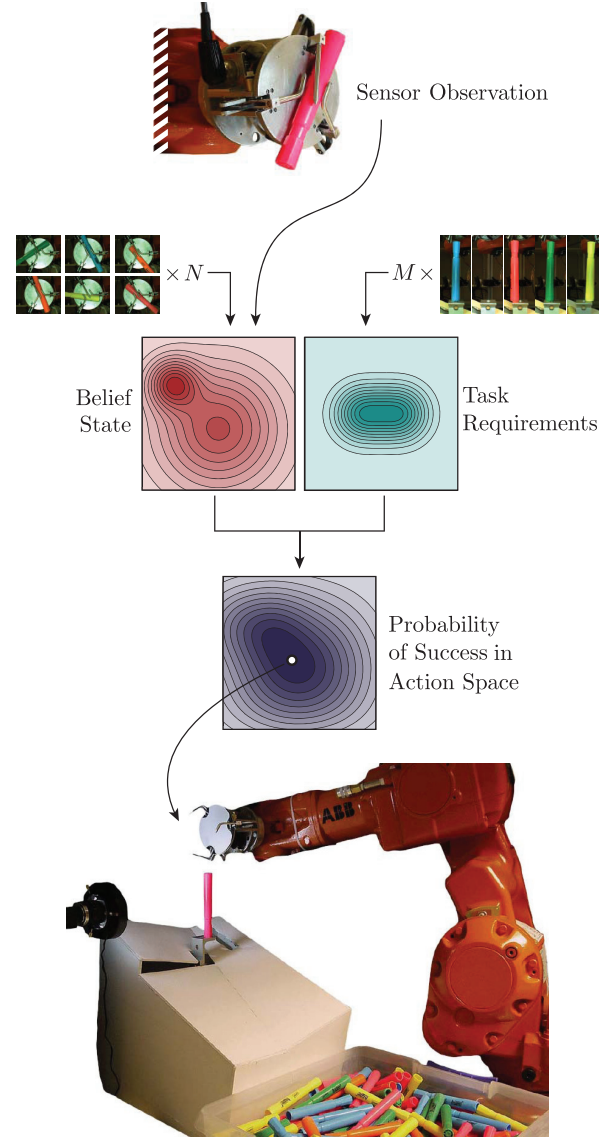

Fig. 2. Procedure to choose the optimal action to accomplish a manipulation task. First, we estimate the belief state of the grasp, that is, a probability distribution of grasp state from sensor readings. Second, we learn how robust our task is to state uncertainty. Finally, we combine them to estimate the probability of success of all available actions and choose the best one.

three differently shaped holes that induce three different task requirements. The larger the hole, the less accurately we need to know the object location. In this case, the action is parameterized by a, the gripper position we choose. Figure 5(b) shows that the probability of successful insertion varies with the error ϵ in the estimate of the object pose.

We can now combine the probabilistic models for the sensing capabilities of the gripper (Figure 4(b)) and task requirements (Figure 5(b)) to find the optimal placement a of the gripper to maximize the probability of successful insertion. Figure 6 shows the result for the three sensor noise models in Figure 4(a) and the three holes in Figure 5(a). Note that even when the estimation of the pose is
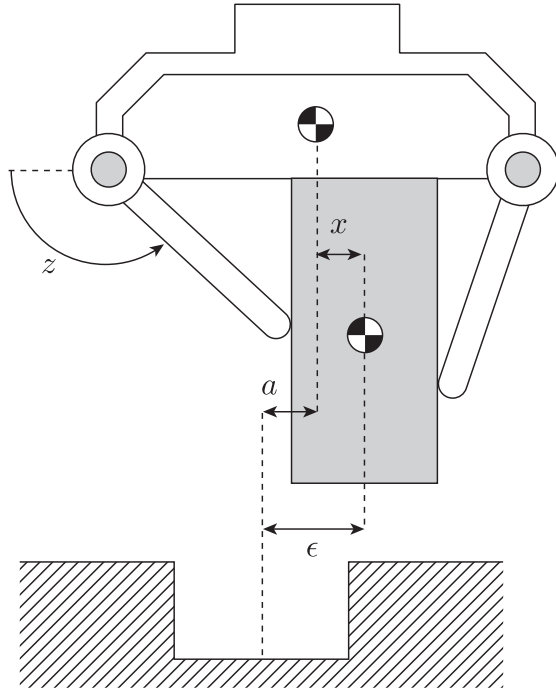
Fig. 3. Motivational simple example in Section 1.1. A two-fingered planar gripper grasps a rectangular object with the goal of inserting it into a hole. The horizontal freedom of the part is parameterized by x. One of the two fingers has a noisy angle sensor z, which we use to estimate the location of the object in the hand. We note with action a the horizontal position of the gripper that we choose to vertically insert the object in the hole. Finally, ϵ represents the error in a with respect to the optimal value that would align the centers of object and hole. The goal is to estimate the likelihood of successful insertion using action a given the sensor reading z.

uncertain, if the hole is wide the chance of success remains high. Likewise, if the hole is a perfect fit for the part, even the smallest amount of noise in the sensor reading will make it impossible to insert the part in the hole.

The example is a simple model to illustrate the principles. Real scenarios are more complex and assumptions are often violated, making it difficult to solve analytically or through simulation. We leverage analytical models to define the structure of the statistical framework, but the actual models are learned directly from observed data. Our goal is for a robot to learn these probabilistic models on its own so it can predict the likelihood of success for a given post-grasp manipulation task.

1.2. Experimental manipulation tasks

We evaluate the proposed framework on three different real manipulation tasks: placing, dropping, and inserting an object. In all three cases the object is a highlighter
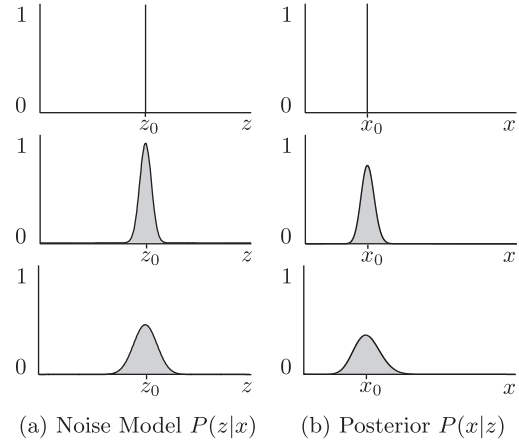

(a) Noise Model $P(z|x)$ (b) Posterior $P(x|z)$

Fig. 4. Sensor model for the gripper in the motivational example in Figure 3. (a) The observation or noise model $P(z|x)$ is the distribution of possible values z we read from the sensor, assuming the object is at x. The level of noise in the sensor determines the sharpness of the distribution $P(z|x)$. (b) The posterior distribution $P(x|z)$ of the pose of the object x is obtained by inverting the observation model. If the sensor has no noise (top row), for the simple gripper in Figure 3 which has no sensor aliasing, we recover the position of the object with zero uncertainty. The higher the level of noise, the less certain we are about the location of object in the hand.
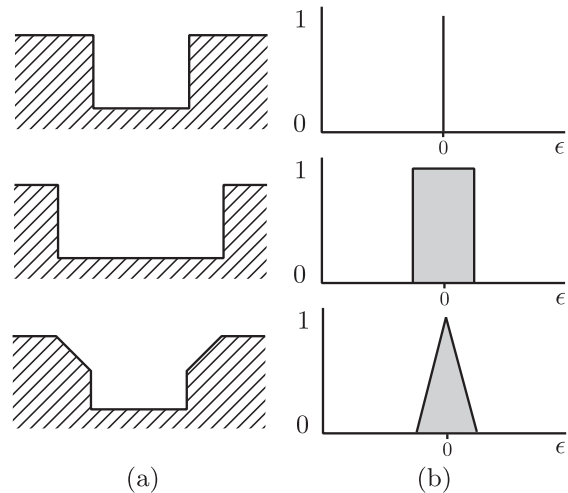

(a)   (b)

Fig. 5. (a) Three different hole shapes. (b) Precision required in the estimation of the pose of the object for a successful insertion, induced by the three holes. Here $P(\text{Success}|\epsilon)$ is a model of the probability of successful insertion as a function of the error ϵ in that estimate. The top hole has the exact same size as the part, which requires perfect accuracy of object position, the middle hole is wider, and requires less accuracy. The corners of the bottom hole are chamfered, which also allows for some deviation where, for the purpose of this example, we assume a continuous degradation of the probability of success as the error increases.

marker, such as that shown in Figure 7(a). The gripper used in the paper is prototype 3 of the MLab Hand
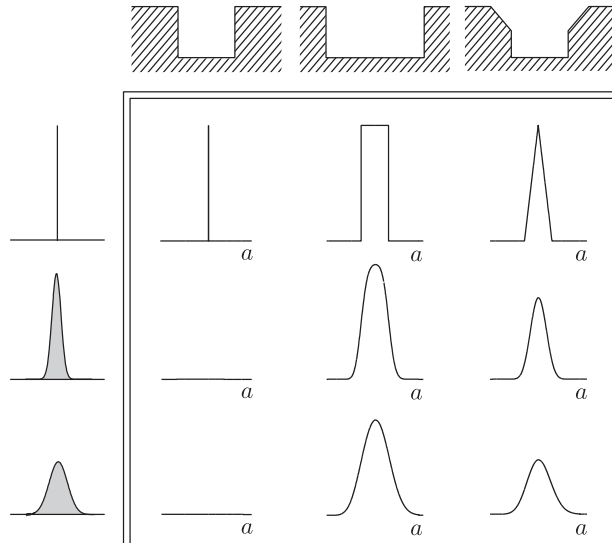
Fig. 6. Probability of successful execution of the simple task in Figure 3 as a function of the chosen action *a*. The matrix shows the probability of success over choice of action for different combinations of noise models (rows) and hole shapes (columns). As expected, both the *sensing capabilities* of the hand, and the *task requirements* affect the predicted probability of success. Accurate sensors (top row) allow us to reliably execute difficult tasks, and simple tasks (central column) allow for noisier sensors.

Fig. 7. (a) A highlighter marker used as an object in the experiments in the paper. (b) Prototype 3 of the MLab Hand.

(Rodriguez et al., 2010; Mason et al., 2012), the simple gripper shown in Figure 7(b). It has three fingers all compliantly connected to a single actuator. When the hand grasps an object, the fingers "fall where they may" around the object. Both the fingers and the actuator have encoders, and by looking at those sensor values, we estimate the pose of the grasped object. In this paper we use the encoders in the three fingers as the input to the learning system.

The placing task consists of picking a marker from a bin full of markers, and balancing it vertically on top of a platform. As illustrated in Figure 8, we use an intermediate step where we push the marker against the platform, to make sure that there is enough clearance to execute the placing action.

The second experimental task is to drop a marker inside a relatively large hole, as illustrated in Figure 9. We will see that the experiments corroborate the intuition that dropping is simpler than placing, in the sense that it can get away with a more noisy estimation of the pose of the marker.

Finally, insertion is a horizontal version of the peg-in-hole problem, where the hand tries to insert the grasped marker into a relatively small hole, as shown in Figure 10. Again, experiments will corroborate that insertion is a more demanding manipulation task than either placing or dropping.

Note that, in the three cases, the goal is to execute the task and accurately predict the probability of success using
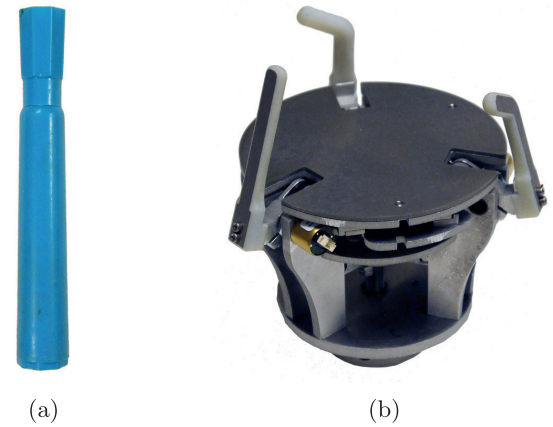

Fig. 8. Diagram of the strategy used to place a highlighter marker. (1) The hand picks a marker out of the bin, and estimates its location. The rest of the strategy is open loop. (2) Reorientation of the hand so that the marker is perpendicular to and centered with the placing platform. (3) Push against the platform to center the location of the marker along its axis. (4) Rotation of $180°$ and recentering with respect to the platform. (5) Push again against the platform. (6) Release marker.
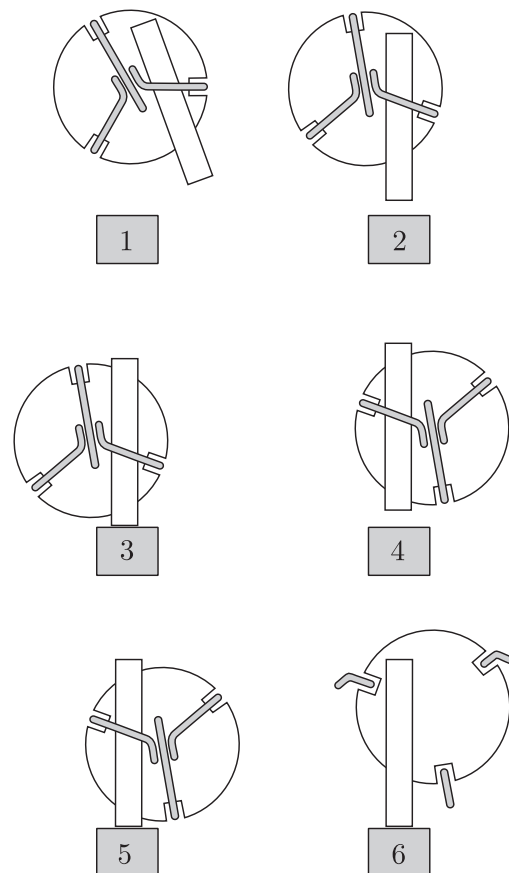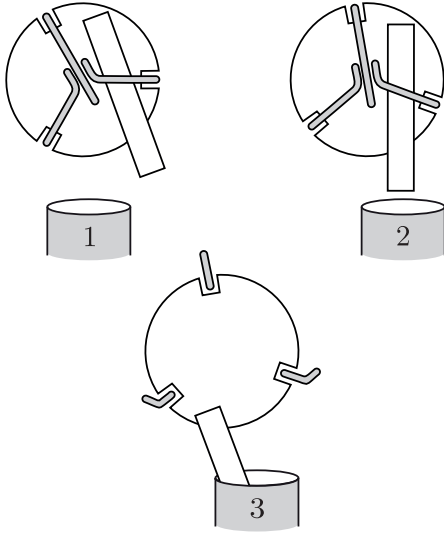
Fig. 9. Diagram of the strategy used to drop a highlighter marker into a hole. (1) The hand picks a marker out of the bin, and estimates its location. The rest of the strategy is open loop. (2) Reorientation of the hand so that the marker is centered with the hole. (3) Release marker.

only feedback from in-hand sensors. For the experiments in this paper we assume that the highlighter marker always lies flat against the palm of the hand.

The configuration space of a cylindrical object lying flat on the palm is four dimensional: the two polar coordinates of the axis of the cylinder, the location of the cylinder along the axis, and the rotation with respect to that same axis.

However, in this paper we only consider the first two dimensions, since the last two dimensions are unobservable to the sensors, and not relevant for the execution of the three tasks. Note that for placing and insertion we included an intermediate pushing step to explicitly reduce the uncertainty of the cylinder along its axis (see Figures 8 and 10). This reduction has computational and data-requirement benefits, as discussed later in Section 6.4, and is beneficial for visualization purposes and clarity of exposition.

The chosen state representation is then that of a symmetrical cylinder in the plane as parameterized by the polar coordinates $x = (r, \theta)$ of its axis, as in Figure 11. That is $x \in X = \mathbb{R} \times \mathbb{SO}(2)$.

1.3. Paper outline

We break up the rest of the paper as follows. Section 2 reviews previous work. Section 3 gives an overview of the proposed statistical framework. Section 3.1 explains how we learn the sensing capabilities of the hand. Section 3.2 explains how we learn the task requirements for the three different post-grasp manipulation tasks: placing, dropping into a hole, and insertion. Section 3.3 shows how to combine these probabilistic models to predict success. Section 4 presents experiments that validate the proposed framework, and Sections 5 and 6 conclude and discuss future directions.
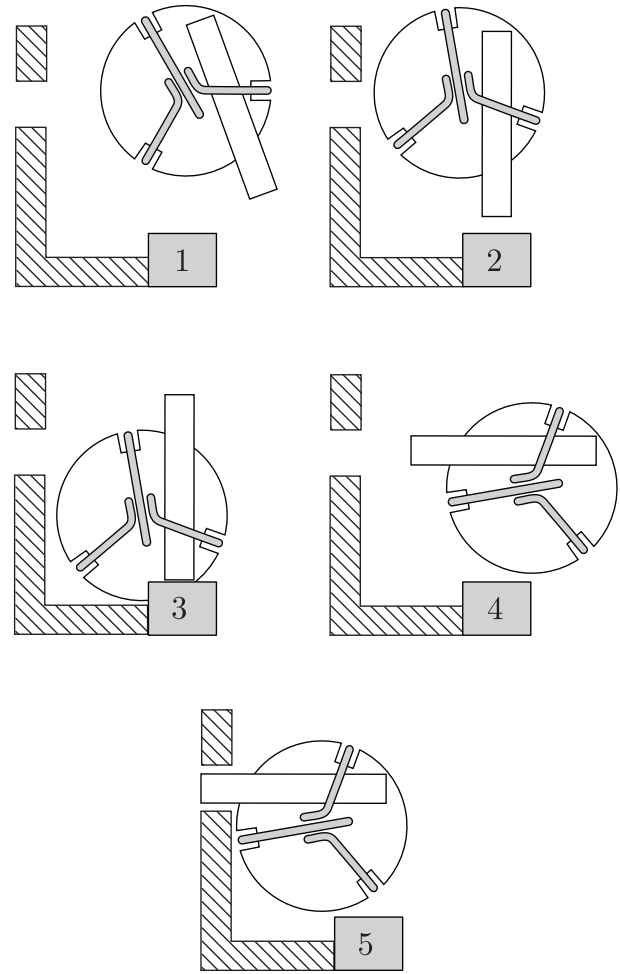

Fig. 10. Diagram of the strategy used to insert a highlighter marker into a hole. (1) The hand picks a marker out of the bin, and estimates its location. The rest of the strategy is open loop. (2) Reorientation of the hand so that the marker is perpendicular to and centered with the placing platform. (3) Push against the platform to center the location of the marker along its axis. (4) Alignment of the axis of the marker with the axis of the hole. (5) Insert marker.
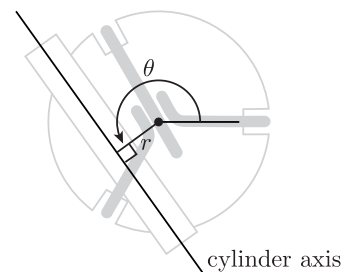

Fig. 11. Parameterization of the pose of the cylindrical marker in the hand. We assume that the marker is always flat against the palm, and parameterize its location by the polar coordinates (r, θ) of its axis. We will ignore the exact location of the marker along the axis, which is unobserved both by the parameterization and the sensors in the hand.

2. Related work

This paper is a revision and extension of Paolini et al. (2012), and is part of the "Simple Hands" project described in Mason et al. (2012) and Rodriguez et al. (2010, 2011).

This paper develops a statistical approach to model grasping and manipulation, with a focus on how uncertainty affects post-grasp manipulation. The importance of uncertainty in manipulation has long been recognized. Incorporating stochastic models into modeling, perception and control was attempted even in the 1970s, for example using Kalman filters in industrial assembly (Simunovic, 1979; Whitney and Junkel, 1982). For more recent work using either Kalman filters or particle filters see (Gadeyne et al., 2005).

Numerous early experiments illustrated the necessity of modeling uncertainty. Most notable was Inoue's peg insertion work (Inoue, 1974) which inspired the *pre-image backchaining* approach (Lozano-Perez et al., 1984; Erdmann, 1986). Preimage backchaining adopted a *possibilistic* approach, representing the robot's belief state by a set of possible configurations. Later work extended the approach to probabilistic models (LaValle and Hutchinson, 1998).

The 1980s and early 1990s saw several projects exploring grasping and manipulation under uncertainty, using both possibilistic and probabilistic models (Brost, 1988; Goldberg, 1990; Goldberg and Mason, 1990; Brost and Christiansen, 1996; Christiansen, 1992; Christiansen and Goldberg, 1995; Mason, 1986; Trinkle et al., 1988; Peshkin and Sanderson, 1988; Brost, 1991; Erdmann and Mason, 1988; Christiansen, 1990). Among these, the closest to the present work are probably Goldberg and Mason (1990), Christiansen and Goldberg (1995), and Brost and Christiansen (1996), which develop Bayesian decision-theoretic techniques, applied to planar grasping problems and the problem of an object sliding in a tilting tray. Dogar and Srinivasa (2011) applied similar ideas to clutter and uncertainty in the context of push-grasping. Kang and Goldberg (1995) used a random sequence of parallel-jaw grasps to classify grasped objects using a Bayesian process.

Goldfeder and Allen (2011) approached the problem of grasp planning from a data-driven perspective.

There is a substantial literature on statistical frameworks to model uncertainty. Partially observable Markov decision processes (POMDPs) (Cassandra et al., 1994) are a general framework that describes the current problem well. Hsiao et al. (2011) used a POMDP framework to track the belief of the pose of an object and tactile exploration to localize it by planning among grasping and information-gathering trajectories. Predictive state representations (PSRs) (Wingate, 2008; Boots et al., 2011) are also introduced as a general framework to learn compact models directly from sequences of action–observation pairs without the need for a hand-selected state representation. LaValle and Hutchinson, (1998) advocate information spaces to formalize the process of propagating uncertainty along motion strategies. Grupen and Coelho (2002) explored the application of optimal control policies in information space, derived from

changes in observable modes of interaction. Platt (2007) has worked on Markov decision process planning, with actions expressed relative to contact locations, and on compliant hand motion. Petrovskaya et al. (2006) have worked on belief state estimation for uncertain manipulation task geometry.

Stulp et al. (2011) learned motion primitives to optimize the chance of grasping an object with Gaussian uncertainty on its location.
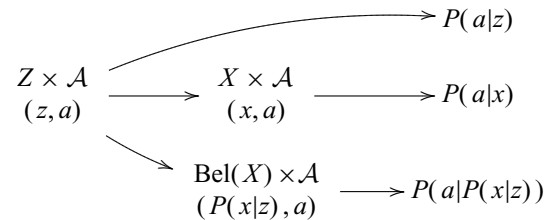
Some work has been done on analyzing the grasp outcome as well. Morales et al. (2004) used real grasps on a collection of objects to predict the reliability of the grasp process.

Balasubramanian et al. (2012) noted that different tasks lead humans to different initial grasps, and Faria et al. (2012) were able to estimate the best part of an object to grasp based on the task using human trials.

In the context of post-grasp manipulation, Jiang et al. (2012) looked at scenes to determine good locations to place objects. However, they did not study how robust the final process of actually placing an object is, which is the subject of our work. Fu et al. (2007) addressed the problem of batting an object to a goal in the presence of uncertainty. They first maximized information gain in an observation step, and then chose the action most likely to succeed. Holladay et al. (2013) used inverse motion planning to determine the optimal placement of a robot's other hand to increase the probability of successfully placing objects.

3. Statistical framework

Given a manipulation task and a sensor observation $z \in Z$ of the state of the task, our goal in this paper is to find the action a from a set of available actions $\mathcal{A}$ that maximizes the expected performance of accomplishing the task. The following diagram illustrates three different strategies to approach the problem:

$$
\begin{array}{c}
Z \times \mathcal{A} \longrightarrow X \times \mathcal{A} \longrightarrow P(a|x) \\
(z,a) \qquad\qquad (x,a) \\
\searrow \\
\mathrm{Bel}(X) \times \mathcal{A} \longrightarrow P(a|P(x|z)) \\
(P(x|z),a)
\end{array}
$$

with $P(a|z)$ reached from $Z \times \mathcal{A}$.

The first and most straightforward strategy is to model the performance of an action directly as a function of sensor observations. The decision on what action to execute and how likely it is to succeed is based upon the history of sensor readings. It makes the least assumptions about the system but also uses the least knowledge about the structure of the problem. It is also the most difficult to implement, since the complexity of the model depends strongly on the dimension of both the sensor and action space, which might be large.

The second strategy introduces an intermediate step where sensor inputs z are first projected into a more compact representation of state, denoted here by x, and all information not captured by that representation is assumed to be irrelevant for planning optimal actions. In this work, we chose x to be the pose of the grasped object. The probability of success of an action is then modeled as a function of the most likely pose of the object x rather than the sensor observations z directly. The intermediate representation x potentially reduces the model complexity, since the dimension of state space is generally smaller than that of sensor space. On the other hand, it introduces the possibility of information loss or lack of observability. It also fails to address uncertainty in the system induced by noisy sensors.

In this paper, we implement a third approach, which encapsulates uncertainty by representing the system by its belief state $P(x|z)$ rather than just by its most likely value x. By explicitly considering uncertainty in the state of the task we can make a more informed and accurate prediction on the probability of success of a given action.

The dimension of the space of belief distributions $\text{Bel}(X)$ is too large to model the probability of success of an action $P(a|z)$ directly as a function of the belief $P(x|z)$. We can alleviate this problem by marginalizing the probability of success of an action $P(a|z)$ with respect to the true state of the system x:

$$P(a|z) = \int_X P(a|z,x) \cdot P(x|z) \, dx$$
$$= \int_X P(a|x) \cdot P(x|z) \, dx \qquad (1)$$

where, in the last step, we make the assumption that the state representation x is informative enough that the output of an action is conditionally independent of sensor observations z, given the true state x.

This assumption enables the computation of the probability of success $P(a|z)$. Note, however, that for some tasks the pose of an object is not always fully representative of the grasp state. For example, in a compliantly actuated gripper, the state of the actuators also contains information on how stiff the grasp is, which is not captured by the pose of the object and might be relevant to determine the outcome of an action.

It is key to note that (1) divides the problem of modeling the performance of an action $P(a|z)$ into two simpler ones: modeling the distributions $P(x|z)$ and $P(a|x)$. Respectively, these represent the *sensing capabilities* of the gripper and the *task requirements* for a successful task execution. The following subsections detail the approach to model them, as well as the process to combine them to give an accurate estimate of $P(a|z)$.

3.1. Sensing capabilities

The shape of the belief state $P(x|z)$ depends on several factors, including the geometries of the manipulator and
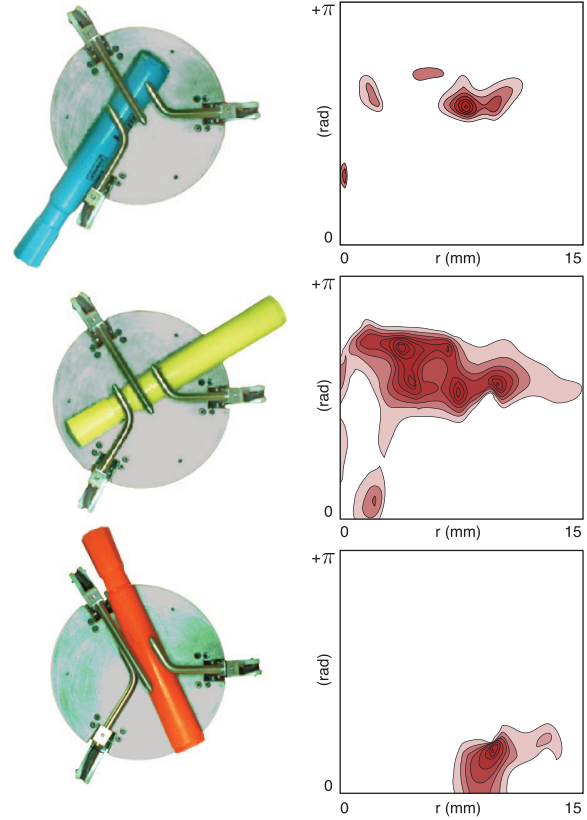

Fig. 12. Three grasps of a highlighter marker, and the corresponding estimated beliefs of the pose of the object. Note that grasps where the object is localized or constrained by geometric features of the hand yield sharper beliefs. These tend to correspond to more stable grasps.

object, the location and type of sensors, and the type of grasp. Assuming fixed geometries for the manipulator, object, and sensors, we will see that different grasps yield differently shaped beliefs.

We will pay special attention to the sharpness of the belief as an indicator of the confidence we get on the pose of the object. As illustrated in Figure 12 the choice of grasp has an important effect on that confidence. We will say that *some grasps are more informative than others*.

In this section, we describe the process to model $P(x|z)$ from experimental data. Learning $P(x|z)$ directly is usually data intensive, since it can be arbitrarily shaped and the complexity of the model depends on the dimension of sensor space. To simplify the process, we use Bayes rule to flip the conditioning in $P(x|z)$ to $P(z|x)$, the likelihood or observation model of the system.

$P(z|x)$ is the distribution of sensor readings given the true state of the system. Unlike the posterior distribution $P(x|z)$, which can be arbitrarily complex due to possible lack of observability or sensor aliasing, the likelihood $P(z|x)$ tends to be simpler and we assume here to follow a Gaussian distribution $P(z|x) \sim \mathcal{N}(z; \mu(x), \sigma^2(x))$. In order to make the

learning feasible, we also assume independence between sensors. This leads us to the following equation for the posterior distribution:

$$
\begin{aligned}
P(x|z) &= \frac{P(z|x)\,P(x)}{P(z)} \\
&\simeq \mathcal{N}(z;\mu(x),\sigma^2(x)) \cdot \frac{P(x)}{P(z)} \\
&\simeq \frac{P(x)}{P(z)} \cdot \prod_{k=1}^{L} \mathcal{N}(z_k;\mu_k(x),\sigma_k^2(x)) \\
&\sim P(x) \cdot \prod_{k=1}^{L} \mathcal{N}(z_k;\mu_k(x),\sigma_k^2(x))
\end{aligned}
\tag{2}
$$

where $P(x)$ is the state distribution prior to any sensor observations, both μ_k and σ_k are functions of the true state of the system x, and L is the number of sensor dimensions. Since $P(z)$ is independent of x, we omit it and normalize $P(x|z)$ a posteriori. In the rest of the paper whenever we use the expression $\mathcal{N}(z;\mu(x),\sigma^2(x))$, we will refer to the decomposition induced by the independence between sensors assumed in (2).

We now detail the process of estimating the prior distribution $P(x)$ and the observation model $P(z|x)$ from a collected dataset $C_1 = \{(z^i,x^i)\}_i$ of pose/sensor readings pairs. Figure 14 shows the data of 2,000 grasps collected for C_1 (see the dataset in Multimedia Extension 1).

3.1.1. Learning the prior distribution $P(x)$.
The prior distribution $P(x)$ is the distribution of the state of the system before considering sensor information. In our case, it is a reflection of the distribution of stable grasps yielded by the combined geometries of object and gripper. Figure 13 illustrates the three most stable configurations or grasp types for the hand and object used in this paper. The expectation is that the prior distribution $P(x)$ will cluster around those three grasp types.

We regress $P(x)$ by estimating the density of the pose of the object in state space. We use kernel density estimation to model $P(x)$ as a sum of kernels:

$$
P(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x-x^i}{h}\right)
\tag{3}
$$

where K is a Gaussian kernel, h is the bandwidth parameter and x^i are the state points in the dataset C_1. The bandwidth parameter is chosen automatically to minimize the mean integrated squared error following the algorithm of Botev et al. (2010). Figure 14 illustrates the learned prior distribution. As expected, it shows three clusters corresponding to grasp types I, II, and III in Figure 13.

3.1.2. Learning the observation model $P(z|x)$.
Equation (2) yields an approximation of the observation model and expresses it in terms of functions $\mu_k(x)$ and $\sigma_k(x)$:
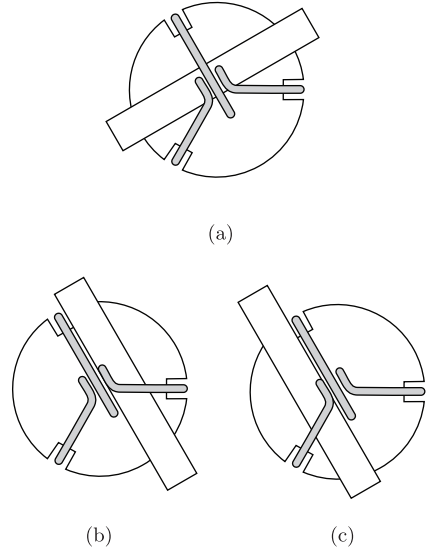

Fig. 13. The three most stable configurations of the object/gripper pair used in our experiments: (a) grasp type I; (b) grasp type II; (c) grasp type III.
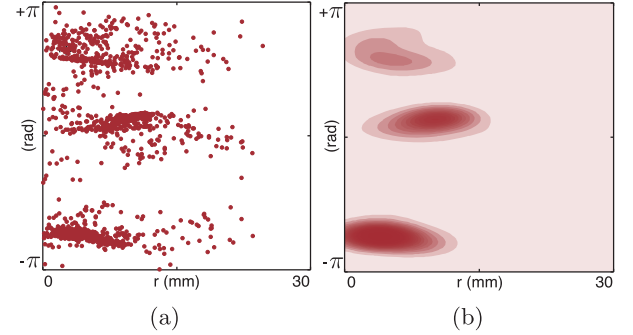

Fig. 14. (a) Distribution of grasp states in dataset C_1. Each dot corresponds to the final pose of the marker after an experimental grasp. (b) Corresponding prior distribution $P(x)$ estimated with kernel density estimation. The three clusters in the distributions correspond to the three expected grasp types in Figure 13.

$$
P(z|x) \approx \prod_{k=1}^{L} \mathcal{N}(z_k;\mu_k(x),\sigma_k^2(x))
\tag{4}
$$

We use Gaussian processes (GPs) (Rasmussen and Williams, 2006) to regress functions $\mu_k(x)$ and $\sigma_k(x)$ for each sensor. Throughout this paper, we use the Gaussian Processes for Machine Learning toolbox (Rasmussen and Nickish, 2010), which also makes use of a non-linear optimization solver by Carbonetto (2007). For that we again use the dataset C_1. The process is detailed in the following steps.

1. Use a GP on half of the data points in C_1 to estimate the mean of the observation model $P(z|x)$, $\mu_k : X \longrightarrow \mathcal{Z}_k$. This implies training one independent GP for every sensory input, as a function of r and θ, which computes
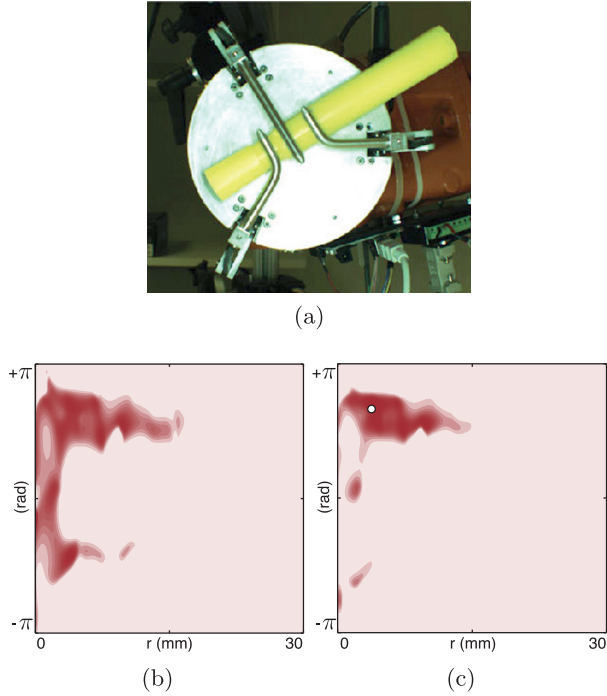
(a)

(b)   (c)

Fig. 15. (a) Example grasp with its corresponding estimated (b) likelihood $P(z|x)$ and (c) posterior distribution $P(x|z)$. The dot corresponds to the most likely pose of the object.

the most likely sensor values for every possible state of the system x. Note that we will get a better estimate of the observation model for the regions of the state space that are most often observed, since those areas will be more populated with the collected data.

2. Complement the other half of the dataset C_1 with the sensor readings $\bar{z}^i = \mu(x^i)$ predicted by the learned observation model, and the squared error yielded by that prediction $\Delta^2 z^i = (z^i - \bar{z}^i)^2$, $C_1^+ = \{(z^i, x^i, \bar{z}^i, \Delta^2 z^i)\}_i$.

3. Use Gaussian process regression (GPR) on C_1^+ to regress the variance of the observation model σ_k^2 : $X \longrightarrow \Delta^2 \mathcal{Z}_k$. Again, this implies training one independent GP for every sensor in the system.

By following these steps, we can now estimate $P(z|x)$ as in (4) (see Multimedia Extension 2 for example code). Figure 15(b) and (c) illustrate the estimated observation model and posterior distribution $P(x|z)$ for the example grasp in Figure 15(a).

3.2. Task requirements

We now model the probability of success of an action, $P(a|x)$. This will tell us how accurate the estimation of the state of the grasp must be for an action to successfully execute the task.

While not required in general, we choose to state parameterize the set of actions. For example, for the task of placing a cylindrical object, we design an action a so that given the true state of the grasp it first turns the cylinder so it is upright with respect to the ground, and then sets it down.

We will note an action parameterized by state as a_p, where p indicates the state of the system for which the action was designed.

In general, the success of an action depends both on the specific action a_p itself and the true state of the grasp x. However, in order to reduce the complexity of the process, we will assume that, if the true state of the system is x, the probability of success of action a_p only depends on $(x - p)$, the difference between the real and assumed state of the grasp. For example, when placing a cylinder whose estimated axis is $1°$ off from its true state, we are more likely to succeed than if we try to place an object several degrees off.

We model the outcome of an action a_p as a Bernoulli random variable of parameter ϕ_{a_p} that depends on the true state x, so that

$$P(a_p = 1|x) = \phi_{a_p}(x) = \phi(x - p) \qquad (5)$$

The use of state parameterized actions also allows us to introduce controlled noise in the space of mismatches $\epsilon = (x - p)$. To learn $\phi(\epsilon)$, during each task execution, if the true state of the grasp is x, instead of choosing the action a_x, we execute action a_p with $p = x + \epsilon$, where ϵ is a uniformly distributed error in the space of system states.

We now detail the process of estimating the task requirements model $\phi(\epsilon)$ from a dataset $C_2 = \{(\epsilon^i, y^i)\}_i$, where ϵ^i is the error in system state and $y^i \in \{0, 1\}$ is the success/failure output of the trial (see the dataset in Multimedia Extension 3). For each task, we uniformly sample ϵ from $\mathcal{E} = [-\Delta r_{max}, \Delta r_{max}] \times [-\Delta \theta_{max}, \Delta \theta_{max}]$. We choose Δr_{max} and $\Delta \theta_{max}$ to be large enough to cover the range of errors we care about, and assume everything falling outside that range to be a failure.

We use a Gaussian Process to regress the function $\phi(\epsilon)$ from dataset C_2 containing the outcome of more than 1,000 executions for three manipulation tasks: placing, dropping, and insertion (see Multimedia Extension 4 for example code). Although the number of successes of a Bernoulli trial is a Binomial distribution, as the number of trials gets large, this approximates a normal distribution, which is the assumption a GP makes. As $y^i \in \{0, 1\}$, this allows us to directly estimate the probability of success for a given amount of error. The results are illustrated in Figure 16. As expected, when $|x - p|$ increases, the likelihood of task success decreases. Note that for the dropping task the probability decreases much slower than for placing or insertion. This indicates that dropping a marker into a hole is easier than balancing it on a platform or inserting into a small hole. The task tolerates more error.

The task requirements for insertion resemble the shape of an X. This can be explained by noting that if we incorrectly try to insert the marker too high, but also tilt it downward, the end of the marker still manages to fit in the hole. In addition to enabling accurate estimates of the probability of successful task execution, computing these task requirement distributions also give us insight in to what types of errors the task execution is robust against.
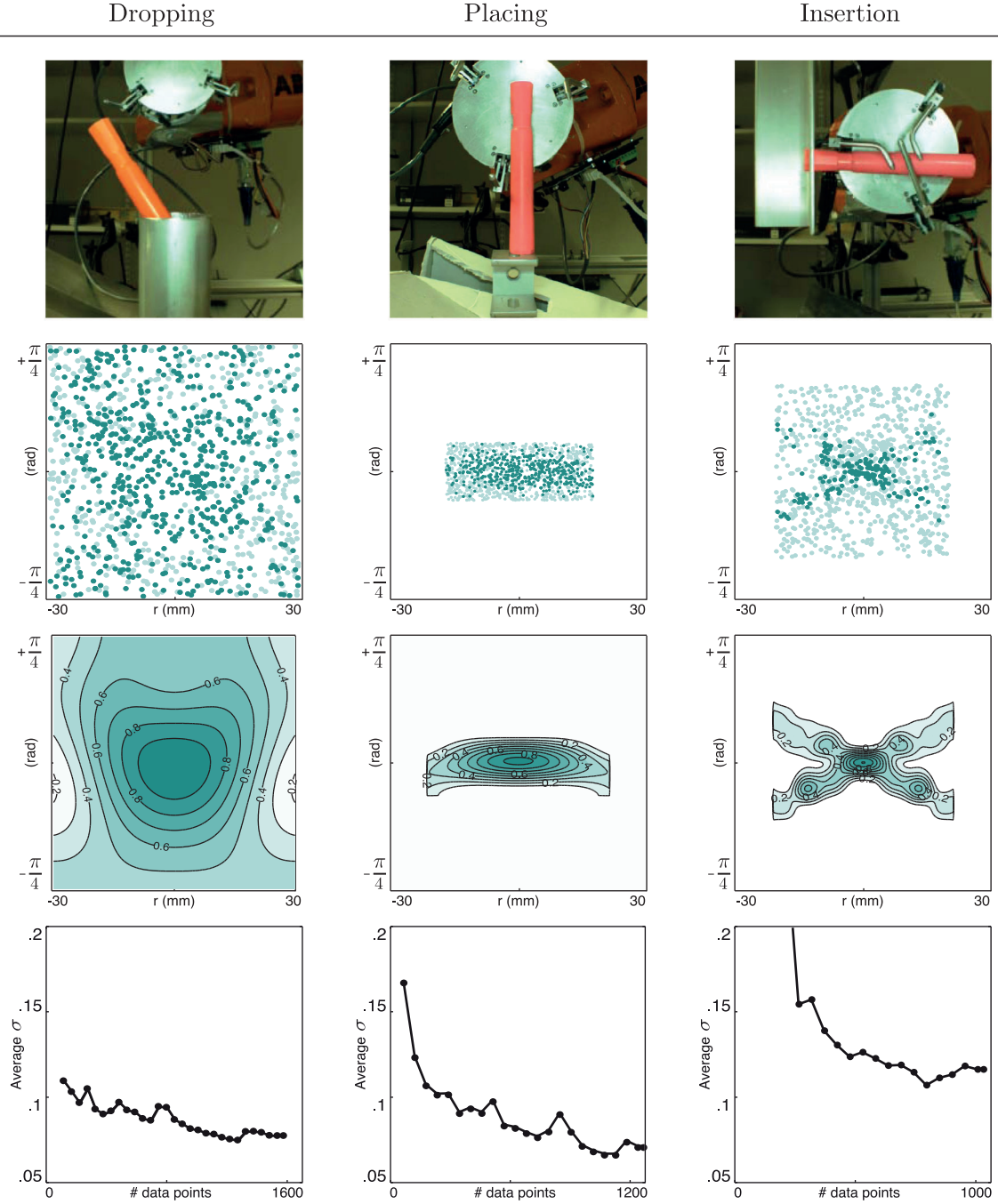
Fig. 16. Learned task requirements for three manipulation tasks: dropping, placing and insertion. Second row: Dataset C_2 of task execution with perturbed states. Dark points are successes and light ones are failures. Note that the range of perturbations is different for each task. Third row: Distribution of task requirements $P(a_p = 1|x)$ as a function of the error in state estimation. Fourth row: Average standard deviation of the regression of the Bernoulli parameter of $P(a_p = 1|x)$ obtained with a GP. This is used as a rough estimate of the convergence of the algorithm and stoping criteria.

In general, the more data we use the more accurate the regressed distributions of task requirements are. The magnitude of the variance returned by the GPR can be used to define a stopping criteria. In our case, we use the average standard deviation to assess how certain we are about the learned distribution. The bottom graphs in Figure 16 shows how the average standard deviation changes with the number of experiments for each task.

3.3. Matching task requirements with sensing capabilities

Here we combine the models of $P(x|z)$ and $P(a|x)$ to estimate the probability of success of an action a_p. For that, we extend (1) as follows:

$$
\begin{aligned}
P(a_p = 1|z) &= \int_X P(a_p = 1|x) P(x|z) \, dx \\
&= \int_X \phi(x - p) \mathcal{N}(z; \mu(x), \sigma^2(x)) \frac{P(x)}{P(z)} \, dx \\
&\simeq \int_{\mathcal{E}} \phi(\epsilon) \mathcal{N}(z; \mu(p+\epsilon), \sigma^2(p+\epsilon)) \frac{P(p+\epsilon)}{P(z)} \, d\epsilon
\end{aligned}
$$
(6)

where we apply the change of variables $\epsilon = x - p$, and $\mathcal{N}(z; \mu(x), \sigma^2(x))$ represents the decomposition in (2).

In the experiments we approximate the integral numerically (see Multimedia Extension 5 for example code). We grid the space of mismatches between real and estimated states into $N_r \times N_\theta$. Letting ϵ_{ij} be the corresponding error, we can approximate the integral in (6) as

$$
\begin{aligned}
P(a_p = 1|z) \simeq \sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} \phi(\epsilon_{ij}) \mathcal{N}(z; \mu(p+\epsilon_{ij}), \\
\sigma^2(p+\epsilon_{ij})) \frac{P(p+\epsilon_{ij})}{P(z)} \Delta A
\end{aligned}
$$
(7)

where ΔA is

$$
\Delta A = \frac{4 \Delta r_{max} \Delta \theta_{max}}{(N_\theta - 1)(N_r - 1)}
$$

Once we compute $P(a|z)$ we find its maximum in action space to choose the optimal action to execute. Depending on that maximum probability we can decide either to execute the task with the optimal action or to abort the execution. Figure 17 shows the complete process for an example grasp.

The presented framework decouples the learning of the sensing capabilities of the hand from the learning of the requirements of the task. For a given hand and object, we only need to compute its sensing capabilities once. For any given new task, we only have to compute the task requirements, and then follow the described procedure to estimate the overall probability.

Another possible scenario where the decoupling between both models is useful is in sharing models of task requirements between different robots. For example, an industrial robot could learn in a room for days at a time, and a mobile manipulator could reuse those learned models.

4. Experimental validation

To validate the proposed framework, we use the hand and object in Figure 7 to complete three different manipulation tasks. This requires one training set for the *sensing capabilities* of the hand, $P(x|z)$, and three training sets to estimate
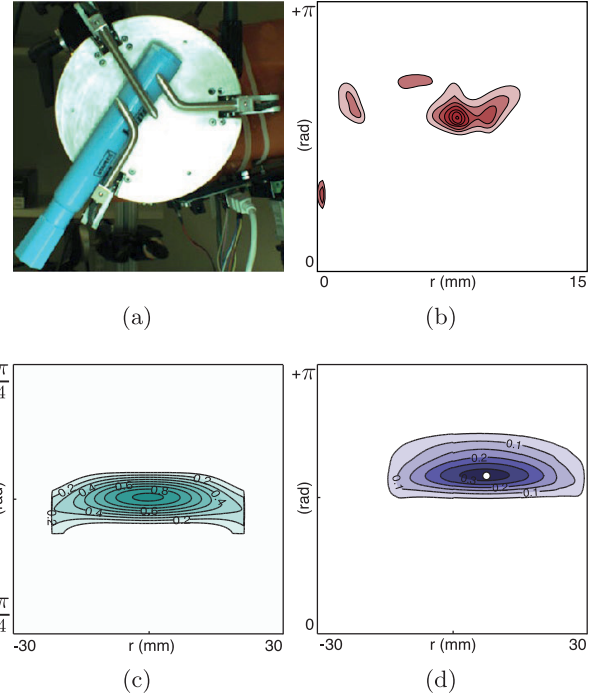

Fig. 17. Complete process to compute the probability of success at placing a marker. (a) Example grasp. (b) Learned belief $P(x|z)$ of the pose of the object. (c) Learned task requirements, $P(a|x)$, for the placing task. (d) Estimated probability of success for the parameterized set of placing actions. The white dot corresponds to the optimal action to execute.

the *task requirements*, $P(a|x)$, one for each of the tasks. After learning these functions, for any new grasp, we can predict the action most likely to successfully execute a task and its expected probability of success.

We are interested in evaluating the accuracy of the predicted probability of success. For that we execute each task 500 times according to the action most likely to succeed as predicted by the learned models. After each execution, we note down both the predicted probability of success and the actual outcome of the experiment (see the data in Multimedia Extension 6).

To test the validity of the predictions, we group grasps by their predicted task success probability and compare it with their correspondent experimental success rate. For example, if we take all grasps that were predicted to succeed at an action around 60% of the time, the average experimental success rate for those grasps should ideally be around that same 60%.

Figure 18 compares the experimental success ratios with the predictions by the learned models. We see that for the three tasks, the experimental probability follows the predicted probability, supporting the validity of the framework. Our results for insertion do not follow the predicted probability as closely as dropping or placing. It is possible that the
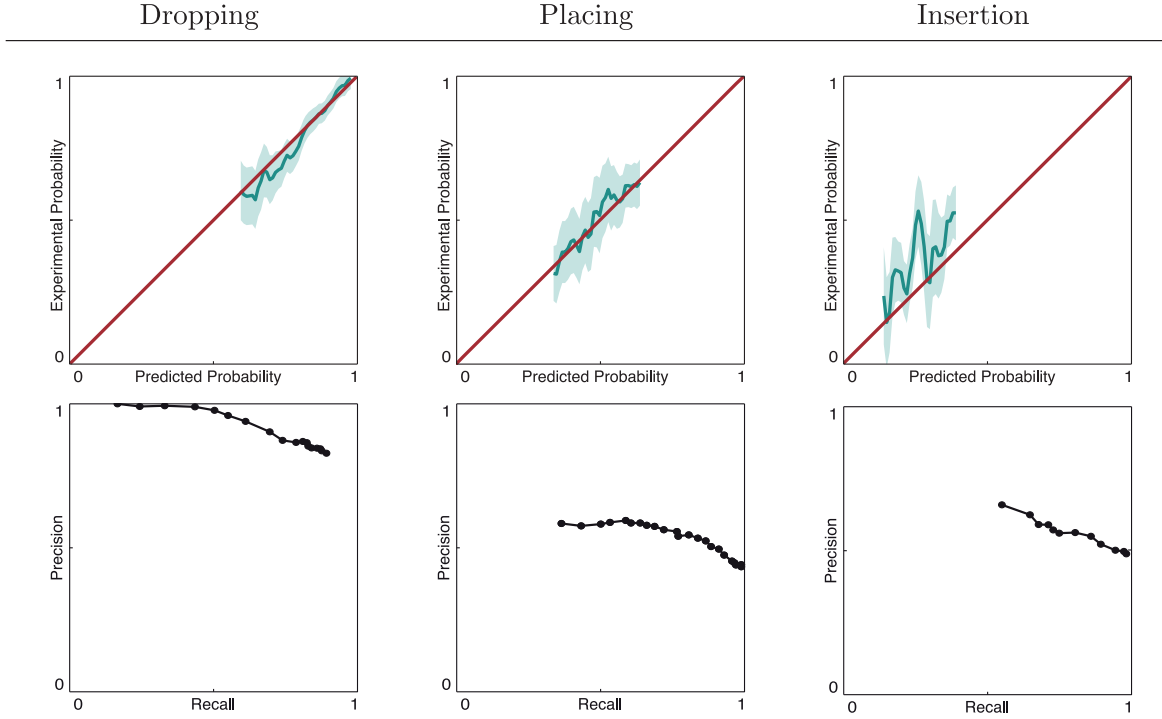
Dropping	Placing	Insertion

Fig. 18. Top: Comparison between the experimental and predicted probability of success. The shaded region is a 95% confidence interval of the estimation of the Bernoulli parameter, according to a binomial distribution. The plots show that the predictions follow the experimental observations quite well. Bottom: Precision-recall curves of the success in task execution formed by only considering task executions whose predicted probabilities were above certain thresholds. The plot shows that we can increase the success rate in task execution by rejecting low-probability grasps in the three tasks.

insertion task violates some of our assumptions, or we simply did not collect enough data to accurately characterize the task.

Both the quality of the in-hand sensor feedback and the difficulty of the manipulation task determine the range of values of the probability of success. From Figure 18, we see that dropping is the easiest of the three tasks, followed by placing, and lastly insertion. This could have been predicted by looking at the task requirements and noting that dropping has the widest distribution of success in the presence of pose error. The plots in Figure 18 indicate that the proposed framework successfully predicts the probability of success of actions independently of the complexity of the task.

Predicting the probability of success of an action allows us to make an informed decision on what action to execute and improve the overall system performance. The bottom row of Figure 18 shows the precision-recall curves of the three tasks. They reveal that we can effectively increase the success rate in task execution if we decide to abort executions whose predicted probability of success are below a certain threshold. By changing that threshold, we can move along the precision-recall curve, and achieve a desired performance. Dropping increases from 80% to near 100% success, placing increases from 40% to 60%, and insertion increases from 50% to 60%.

5. Conclusion

In this paper, we introduce a general statistical framework to model the problem of post-grasp manipulation. The framework is composed of the following three steps.

- **Off-line learning of sensing capabilities.** We learn a model to estimate the belief state of the grasp from in-hand sensor information. The training process creates two models: first, a prior distribution of the expected final grasp states; and, second, an observation model of the hand/object pair that relates grasp states to expected sensor readings. Both models are constructed from experimental data, and the combination allow us to estimate the distribution of the state of the grasp $P(x|z)$ from sensor readings.

- **Off-line learning of task requirements.** We also learn a model of the object pose accuracy required to execute a specific post-grasp manipulation task. The model captures how the probability of successful task execution degrades as a function of object pose error. We train this model by systematically introducing controlled perturbations in the state of the grasp, and recording the relationship between the noise introduced and the success/failure outcome.

- **On-line estimation of the probability of success.** During execution time, we use the off-line learned models for the sensing capabilities of the hand and the task requirements to make accurate predictions of the probability of success. That prediction can be used, for example, to choose the optimal action to execute from a set of actions or to improve the overall performance of the system by deciding to abort runs when the predicted probability is too low.

We implemented the framework and tested it on three different post-grasp manipulation tasks: dropping, placing, and insertion. To validate the framework, the robot performed over 8,000 real grasps, and over 1,000 trials each of placing, dropping, and insertion.

6. Discussion

6.1. Insights

The proposed statistical framework is general, and is designed to be implemented on a real system. The unpredictability of the real world, especially when contact and physical interaction are involved, strengthens the case for a statistical framework.

The sensing capabilities of a hand/object pair help us understand which grasps are informative and which are not. This can inform both the design of hands, as well as the design of grasp policies, by being aware of what it means to be a good grasp from the point of view of grasp observability. Different hands and strategies can then be tested to see exactly which configurations give us the most certainty in object state, which would improve post-grasp manipulation tasks down the road.

Learning the accuracy requirements of a task by artificially adding noise is an excellent tool to identify weaknesses in task execution. We can discover, for example, that placing tends to fail when the object is in a certain pose, so adding a move to avoid that pose could improve the overall task success rate.

We can combine these two models, *sensing capabilities* and *task requirements*, to predict the probability of successful task execution. The separation of the two models allows us to mix and match different tasks and hands. If we reuse the same hand, for each new task, we only need to learn its task requirements. Conversely, if we are executing the same task with different hands, for each new hand, we only need to learn its sensing capabilities.

Estimating the probability of success is powerful, since it allows the robot to increase its overall task success rate by not taking unnecessary risks. We can, for example, find an optimal policy to minimize mean time to success in an abort and retry scheme, similar to Rodriguez et al. (2011). In addition, since we estimate the probability of success for the complete set of actions, we could consider optimization with different cost functions, or in the presence of constraints.

6.2. Implementation

Designing a robust system to collect all of the necessary data was both important and challenging. The amount of data required to learn probability functions forced us to carefully design experimental setups requiring minimal human intervention. For the three post-grasp manipulation tasks we had to focus on three different aspects: object acquisition, task execution, and post-task reset. Having a human in the loop to hand objects to the robot was not an option, since we needed to execute thousands of experiments and it would also possibly introduce bias in the process.

We solved the object acquisition problem by having a large bin of objects and training an open-loop grasping strategy that singulates a marker out of the bin approximately 40% of the time. For task execution, it was important to make sure that the hand did not collide with the environment, regardless of the pose of the marker or action chosen by the robot.

Finally, for resetting the system after task execution, different strategies were used depending on the task. For placing, the object was placed at the top of a ramp and knocked back into the bin. For dropping, after some constraining moves, the object was grasped out of the hole and dropped back into the bin. For insertion, the object was held the entire time and then dropped back in. It is important to note that a fair amount of time was spent in designing robust experiments, and this should not be overlooked when trying to collect data in a real setting.

6.3. Assumptions

While the proposed framework is general, we make a few simplifying assumptions to implement it on a real system. The goal of these assumptions is to reduce the total amount of data required to learn, with minimal sacrifices to functionality. Here is a summary of those assumptions.

- We assume that the probability of success of an action is conditionally independent of sensor readings given the true state of the system, $P(a|x, z) = P(a|x)$. This is reasonable when the chosen state is a good representation; however, if it is incomplete, the estimate of $P(a|x)$ may be suboptimal. For example, the location of the center of mass of the marker relative to the center of the hand seemed to have an effect on the probability of success of the dropping task. However, the state representation (r, θ) we choose in the paper does not capture it, and consequently it is not observable. This is treated as noise in the process, and although we are still able to give accurate predictions of the probability of dropping success, they could be better.
- We assume that the observation model $P(z|x)$ for an object/hand pair is unimodal and normally distributed. If the grasp is at a fixed and known state x, the distribution of readings z that we get from the sensors in the

hand is induced by the sensor noise, which we assume here to be Gaussian. Note that the framework still holds without the Gaussian assumption. It is, however, a common convenience to reduce the amount of data needed to learn the observation model.

- We assume independence between in-hand sensors. This is generally not true. Still it is a common simplification to reduce the complexity of the distribution to learn. It effectively restricts the type of distributions we can learn, since instead of learning a distribution in the joint space of all sensors, we learn a single dimensional distribution for each sensor and multiply them.

- Finally, we assume that the set of actions designed to execute a task is state parameterized. This is often true, but again it is a convenience that allows us to reduce the dimension of the model of task requirements. The framework still holds for sets of actions not parameterized by state. However, to estimate $P(a|x)$, we would need to sample the product space of actions and states, rather than just the space of mismatches between action and state.

6.4. Scalability

As this paper is based on data, we briefly discuss the scalability of this framework for varying dimensionality in sensor space, state space, and action space.

- **Sensor dimension.** If L is the number of sensors in the hand, we learn $2L$ GPs to compute the sensing capabilities of the hand, corresponding to the mean and variance of $P(z|x)$ for each sensor. Note that when we add additional sensors to our hand, while the number of GPs increases linearly, the amount of data used to train each GP is still the same, so it does not affect the accuracy of our estimates. If for example we remove a sensor, the belief is now much less peaked, because we would indeed know less about the state of the object. While the overall success rate may change based on the absence or addition of sensors, the *accuracy of the predictions* should not change.

- **State dimension.** Increasing the dimension of the state space on the other hand can have an effect on the accuracy. First, the sensing capabilities would now need to handle an additional dimension when computing the mean and variance of $P(z|x)$. It has been shown that the number of data points required to achieve the same accuracy for a GP grows exponentially with the state dimension (Györfi et al., 2002). In the case of sensing capabilities we can do slightly better, as often the state is clustered in small areas of the space, or has an underlying lower-dimensional representation. If the state is truly uniform, then data needs grow exponentially to maintain the same level of accuracy.

  In the case of task requirements, the data requirement also grows exponentially with dimension, since we are uniformly sampling errors in each dimension.

- **Action dimension.** If we assume actions are state parameterized, then the action dimension increases with the state dimension, and all of the statements above hold true. If instead, actions are not state parameterized, then increasing the dimension of the action space will again cause the data required to learn our task requirements to a given accuracy to grow exponentially. Note that increasing the action dimension will not affect sensing capabilities.

- **Computational dependence on dimension.** In addition to the data requirement, another factor that cannot be ignored is the computational complexity of calculating these probability functions. As the number of data points increases, the computation time for the GPs used to compute the sensing capabilities and task requirements suffer as $O(n^3)$. GPs are much more dependent on the number of data points compared with the dimension of each data point, so both sensor and state dimension do not have too much of an impact.

  Conversely, when trying to combine task requirements and sensing capabilities together, that integral is directly dependent on the state dimension. Currently, we are gridding up the state space finely in each dimension and numerically approximating the integral. This is exponential with dimension, and puts a large requirement on time and memory (if things are being precomputed). Given that we are computing the integrals of probability distributions, we should be able to lessen this requirement by using particle and Monte Carlo-based approaches. This will enable us to sample only in relevant regions and increase efficiency.

6.5. Future directions

One area we would like to explore in the future is understanding the statistical significance of the distributions the robot learns. During experiments, we had no analytical means of knowing when we had *enough* data. While we were able to show that our probability estimates improved with more data, more analysis is needed to quantify how data density affects performance.

Carrying statistical significance through all of the distributions would enable us to compute a confidence bound on our final probability distribution in action space. This would expand the usefulness of our framework and help us understand how collecting more data affects our estimates. Another direction that we are interested in exploring is using active learning to selectively sample so as to reduce overall data requirements.

Finally, when the robot encounters a situation where the probability of success is not acceptable, the only options to increase it are either to abort and retry, to design better actions, or to use better hardware. Instead we could consider

a scenario where, based on the computed probability distributions, the robot decides to execute extra actions aimed at improving the expected probability of success rather than aimed directly at solving the task.

In particular, we are interested in developing robots with regrasping capabilities. Inserting a key into a lock is nearly impossible if you are holding it by its teeth rather than its head. Being able to regrasp objects, either to reduce state uncertainty or change configuration, would greatly increase the capabilities of current autonomous robotic systems.

Funding

References

Balasubramanian R, Xu L, Brook P, Smith J and Matsuoka Y (2012) Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics* 28(4): 899–910.

Boots B, Siddiqi SM and Gordon GJ (2011) Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research* 30(7): 954–966.

Botev ZI, Grotowski JF and Kroese DP (2010) Kernel density estimation via diffusion. *The Annals of Statistics* 38(5): 2916–2957.

Brost RC (1988) Automatic grasp planning in the presence of uncertainty. *The International Journal of Robotics Research* 7(1): 3–17.

Brost RC (1991) *Analysis and Planning of Planar Manipulation Tasks*. Ph.D. thesis, Carnegie Mellon University School of Computer Science.

Brost RC and Christiansen AD (1996) Probabilistic analysis of manipulation tasks: a conceptual framework. *The International Journal of Robotics Research* 15(1): 1–23.

Carbonetto P (2007) A MATLAB interface for L-BFGS-B. Department of Computer Science, University of British Columbia.

Cassandra AR, Kaelbling LP and Littman ML (1994) Acting optimally in partially observable stochastic domains. In: *Proceedings of the twelfth National conference on artificial intelligence (AAAI 1994)*. Menlo Park, CA: AAAI Press, pp. 1023–1028.

Christiansen AD (1990) Manipulation planning from empirical backprojections. In: *International conference on robotics and automation*, Sacramento, CA, pp. 762–768.

Christiansen AD (1992) *Automatic acquisition of task theories for robotic manipulation*. Technical Report CMU-CS-92-111, Carnegie Mellon University, Pittsburgh, PA, USA.

Christiansen AD and Goldberg KY (1995) Comparing two algorithms for automatic planning by robots in stochastic environments. *Robotica* 13(06): 565–573.

Dogar M and Srinivasa SS (2011) A framework for push-grasping in clutter. In: *Robotics: Science and Systems (RSS)*.

Erdmann M (1986) Using backprojections for fine motion planning with uncertainty. *The International Journal of Robotics Research* 5(1): 19.

Erdmann MA and Mason MT (1988) An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation* 4(4): 369–379.

Faria DR, Martins R, Lobo J and Dias J (2012) Extracting data from human manipulation of objects towards improving autonomous robotic grasping. *Robotics and Autonomous Systems* 60(3): 396–410.

Fu J, Srinivasa S, Pollard N and Nabbe B (2007) Planar batting under shape, pose, and impact uncertainty. In: *IEEE International conference on robotics and automation (ICRA)*.

Gadeyne K, Lefebvre T and Bruyninckx H (2005) Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation. *The International Journal of Robotics Research* 24(8): 615–630.

Goldberg K and Mason MT (1990) Bayesian grasping. In: *IEEE International conference on robotics and automation (ICRA)*, pp. 1264–1269.

Goldberg KY (1990) *Stochastic Plans for Robotic Manipulation*. Ph.D. thesis (Technical Report CMU-CS-90-161), Carnegie Mellon University School of Computer Science.

Goldfeder C and Allen PK (2011) Data-driven grasping. *Autonomous Robots* 31(1): 1–20.

Grupen RA and Coelho JA (2002) Acquiring state from control dynamics to learn grasping policies for robot hands. *Advanced Robotics* 16(5): 427–443.

Györfi L, Kohler M, Krzyzak A and Walk H (2002) *A Distribution-free Theory of Nonparametric Regression* (*Springer Series in Statistics*). New York: Springer.

Holladay A, Barry J, Kaelbling LP and Lozano-Perez T (2013) Object placement as inverse motion planning. In: *IEEE conference on robotics and automation (ICRA)*. http://lis.csail.mit.edu/pubs/tlp/ICRA13_1509_FI.pdf.

Hsiao K, Kaelbling L and Lozano-Pérez T (2011) Robust grasping under object pose uncertainty.

Inoue H (1974) *Force Feedback in Precise Assembly Tasks*. Technical Report MIT AIM-308, MIT.

Jiang Y, Lim M, Zheng C and Saxena A (2012) Learning to place new objects in a scene. *The International Journal of Robotics Research* 31(9): 1021–1043.

Kang D and Goldberg K (1995) Sorting parts by random grasping. *IEEE Transactions on Robotics and Automation* 11(1): 146–152.

LaValle SM and Hutchinson SA (1998). An objective-based framework for motion planning under sensing and control uncertainties. *The International Journal of Robotics Research* 17(1): 19–42.

Lozano-Perez T, Mason MT and Taylor RH (1984) Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research* 3(1): 3–24.

Mason MT (1986) Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research* 5(3): 53–71.

Mason MT, Rodriguez A, Srinivasa SS and Vazquez AS (2012) Autonomous manipulation with a general-purpose simple hand. *The International Journal of Robotics Research* 31(5): 688–703.

Morales A, Chinellato E, Fagg AH and del Pobil AP (2004) Using experience for assessing grasp reliability. *International Journal of Humanoid Robotics* 1(4): 671–691.

Paolini R, Rodriguez A, Srinivasa SS and Mason MT (2012) A data-driven statistical framework for post-grasp manipulation. In: *International Symposium on Experimental Robotics (ISER)*.

Peshkin MA and Sanderson AC (1988) The motion of a pushed, sliding workpiece. *Journal of Robotics and Automation* 4(6): 569–598.

Petrovskaya A, Khatib O, Thrun S and Ng AY (2006) Bayesian estimation for autonomous object manipulation based on tactile sensors. In: *IEEE International conference on robotics and automation*, pp. 707–714.

Platt R (2007) Learning grasp strategies composed of contact relative motions. In: *7th IEEE-RAS International conference on humanoid robots*. IEEE, pp. 49–56.

Rasmussen CE and Hannes N (2010) Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research* 11: 3011–3015.

Rasmussen CE and Williams CKI (2006) *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Rodriguez A, Mason MT and Srinivasa SS (2010) Manipulation capabilities with simple hands. In: *International symposium on experimental robotics (ISER)*.

Rodriguez A, Mason MT, Srinivasa SS, Bernstein M and Zirbel A (2011) Abort and retry in grasping. In: *IEEE International conference on intelligent robots and systems (IROS)*.

Simunovic SN (1979) *An Information Approach to Parts Mating*. Sc.D. Thesis, Massachusetts Institute of Technology.

Stulp F, Theodorou E, Buchli J and Schaal S (2011) Learning to grasp under uncertainty. In: *IEEE International conference on robotics and automation (ICRA)*.

Trinkle JC, Abel JM and Paul RP (1988) An investigation of frictionless enveloping grasping in the plane. *The International Journal of Robotics Research* 7(3): 33–51.

Whitney DE and Junkel EF (1982) Applying stochastic control theory to robot sensing, teaching, and long term control. In: *American control conference*, pp. 1175–1183.

Wingate D (2008) *Exponential Family Predictive Representations of State*. Ph.D. thesis, University of Michigan.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Data	Data used to compute sensing capabilities.
2	Code	Sensing capabilities source code.
3	Data	Data used to compute requirements for dropping, placing, and insertion.
4	Code	Task requirements source code.
5	Code	Overall framework source code.
6	Data	Computed results for dropping, placing, and insertion.