

A Data-Driven Statistical Framework for Post-Grasp Manipulation

Robert Paolini, Alberto Rodriguez, Siddhartha S. Srinivasa and Matthew T. Mason

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 of the task in terms of grasp state accuracy. 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 8000 grasp trials, and over 1000 trials each of placing, dropping and insertion.

1 Introduction

Knowledge of the grasp state is often critical to any subsequent manipulation task. Intuitively, harder tasks demand a more accurate estimation of the state of a grasp than simpler ones. For example, balancing a cylinder on a table requires more accuracy than dropping it into a hole. More generally, consider a manipulator, an object to manipulate, a task, and a 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 best suited to model the execution of tasks that require grasping an object prior to execution, i.e., post-grasp manipulation tasks. We address the problem by separating it into two independent

Robert Paolini and Alberto Rodriguez e-mail: {rpaolini, albertor}@cmu.edu
Siddhartha S. Srinivasa and Matthew T. Mason e-mail: {siddh, matt.mason}@cs.cmu.edu
The Robotics Institute, Carnegie Mellon University, Pittsburgh, USA

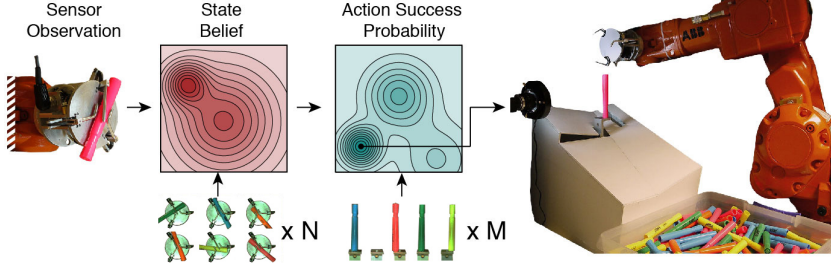


Fig. 1 Procedure to choose the optimal action to accomplish a manipulation task. First, we learn the belief of the state of the system from sensor readings. Based on that belief, we then estimate the probability of success of available actions and choose on the best action to take. Both the state estimation and task requirements are learned using real data.

steps. First, estimate the state of the grasp with in-hand sensors, 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 distribution in task action space, enabling us to find not only the optimal action, but to understand just how likely that action is to succeed.

Figure 1 illustrates the process for placing an object. Sensors in the hand provide information of the grasp state. First, we estimate the probability distribution of the pose of the object in the hand. Second, we predict the probability of success of each available action. Both of these are computed based on data-driven models. Finally, we 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. The experimental setup in Figure 1 consists of a simple gripper [14, 17] 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.

For the experiments in this paper, the manipulated objects are highlighter markers. The chosen state representation is that of a symmetrical cylinder in the plane as parametrized by the polar coordinates $x = (r, \theta)$ of its axis. That is $x \in X = \mathbb{R} \times \text{SO}(2)$.

2 Related Work

In this paper, we study how uncertainty affects post-grasp manipulation. While post-grasp manipulation has not been extensively explored, grasp planning with uncertainty is a well studied problem. Brost [3] found planar grasps that succeed even with large uncertainty in object position by exploiting friction. Goldfeder and Allen [9]

approached the problem of grasp planning from a data-driven perspective. Goldberg and Mason [8] proposed a Bayesian framework to model uncertainty propagation in manipulation plans and apply it to grasping. Stulp et al. [19] learned motion primitives to optimize the chance of grasping an object with Gaussian uncertainty on its location.

The closest work to this paper is probably by Brost and Christiansen [4], who provided a framework for probabilistic analysis of manipulation tasks to overcome the shortcomings of the worst-case-configuration-space approach to manipulation. They applied the framework to plan grasps of planar objects with a parallel jaw gripper. Dogar and Srinivasa [6] applied a similar idea to clutter and uncertainty in the context of push-grasping.

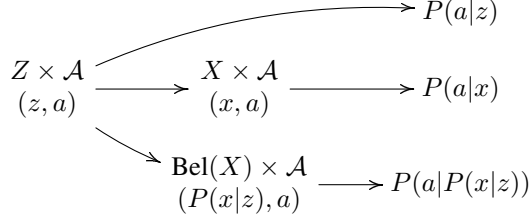
Some work has been done on analyzing the grasp outcome as well. Morales et al. [15] used real grasps on a collection of objects to predict the reliability of the grasp process. Kang and Goldberg [12] used a random sequence of parallel-jaw grasps to classify grasped objects using a Bayesian process.

There is plenty of work on statistical frameworks to model uncertainty. POMDPs [5] (Partially Observable Markov Decision Processes) are a general framework to optimally plan under uncertainty by tracking the distribution of a system under some provided state representation. Hsiao et al. [10] 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. PSRs [1, 20] (Predictive State Representation) 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. Lavalley [13] introduced information-spaces to formalize the process of propagating uncertainty along motion strategies.

In the context of post-grasp manipulation, Jiang et al. [11] 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. [7] 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.

3 Statistical Framework

Our goal in this paper is to find the action a from a set of available actions \mathcal{A} that, given sensor inputs $z \in \mathcal{Z}$, maximizes the expected performance of accomplishing a task. The following diagram illustrates three different paths to approach the problem:



The first path proposes to model the performance of an action directly as a function of sensor observations. How likely a specific action is to succeed and the action to execute are decided based upon the history of sensor readings. It makes the least assumptions about the system but is also the most difficult to implement, since the complexity of the model depends strongly on the dimension of the sensor space, which might be large.

In the second path, sensor inputs z are first projected into a more compact representation of state, noted here by x . 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 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. Note that it also fails to address uncertainty in the system induced by noisy sensors.

In this paper, we implement the third path, which encapsulates uncertainty by representing the system by its state belief $P(x|z)$ rather than just by its most likely value x . By maintaining the distribution of all possible poses of the object, we can later make a more informed 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 from the belief $P(x|z)$. We simplify this problem by marginalizing the probability of success of an action $P(a|z)$ with respect to the true state of the system x :

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

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

We will show later that this assumption is key to enable 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 tight the grasp is, which might be relevant to determine the outcome of an action.

Equation (1) models the performance of an action as a function of both $P(x|z)$ and $P(a|x)$. These characterize the *sensing capabilities* of the gripper and the *precision task requirements* for a successful task execution, respectively. The following

subsections detail the approach to model them, as well as the process to combine them to give an accurate estimation of $P(a|z)$.

3.1 Learning Sensing Capabilities

The shape of the posterior distribution $P(x|z)$ of the grasp state depends on several factors, including the geometries of the manipulator and 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 in general, different grasps yield different shaped belief distributions. We will pay special attention to the sharpness of those distributions, as an indicator of the confidence we get on the pose of the object.

In this section, we describe the process to model $P(x|z)$ from data. Learning $P(x|z)$ directly is usually expensive in terms of the amount of data required, 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 of the system. The likelihood is the distribution of sensor readings given the true state of the system, which is usually unimodal and we assume here to follow a Gaussian distribution $P(z|x) \sim \mathcal{N}(z; \mu(x), \sigma^2(x))$. This leads us to the following equation for our posterior distribution:

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)} \simeq \mathcal{N}(z; \mu(x), \sigma^2(x)) \cdot \frac{P(x)}{P(z)} \quad (2)$$

where $P(x)$ is the prior state distribution, and both μ and σ are functions of the true state of the system x . Given that $P(z)$ is independent of x , we can obviate it and normalize $P(x|z)$ a posteriori. We now detail the process to estimate the prior $P(x)$ and posterior $P(x|z)$ distributions from a collected dataset $C_1 = \{(z^i, x^i)\}_i$ of pose/sensor readings pairs. Figure 3 shows the data collected for 2000 grasps.

3.1.1 Prior Distribution

The prior distribution $P(x)$ is the distribution of the state of the system before considering any information in the sensor readings. We regress $P(x)$ by estimating the density of the pose of the object in state space. We use Kernel Density Estimation that models $P(x)$ as a sum of kernels:

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

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 in Botev et. al [2].

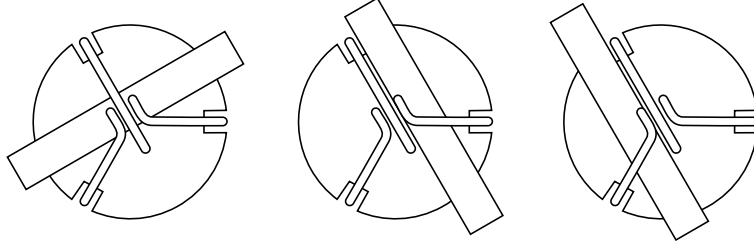


Fig. 2 The three most stable configurations of the object/gripper pair used our experiments. We label them from left to right as clusters I, II and III.

In this paper, the state of the system is represented by the coordinates (r, θ) of the axis of a grasped cylindrical object. The prior grasp distribution in Figure 3 shows three main clusters. These correspond to the three most stable grasps yielded by the combined geometries of object and gripper, shown in Figure 2. The expectation is for grasps in cluster II and III to be the most stable and informative, since one of the fingers acts as an alignment feature.

3.1.2 Posterior Distribution

Equation (2) expresses the posterior distribution as a function of μ and σ . Here we use Gaussian Processes (GP) [16] to regress them as functions of x . For that we again use the dataset C_1 . The process is detailed in the following steps:

1. Use a GP on the first half of the data points in C_1 to estimate the mean of the likelihood or observation model $P(z|x)$, $\mu : X \rightarrow \mathcal{Z}$. This provides the most likely set of observed sensor readings for every possible state of the system x . This implies training one independent GP for every sensory input, as a function of r and θ . Note that we will get a better estimation 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 second half of the dataset C_1 with the sensor readings $z_0^i = \mu(x^i)$ predicted by the learned observation model, and the squared error yielded by that prediction $\Delta^2 z^i = (z^i - z_0^i)^2$, $C_1^+ = \{(z^i, x^i, z_0^i, \Delta^2 z^i)\}_i$
3. Use GPR on C_1^+ to regress the variance of the observation model $\sigma^2 : X \rightarrow \Delta^2 \mathcal{Z}$. 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 $\mathcal{N}(z; \mu(x), \sigma^2(x))$. Figure 3 shows the likelihood $P(z|x)$ for an example grasp and the corresponding posterior distribution $P(z|x)$.

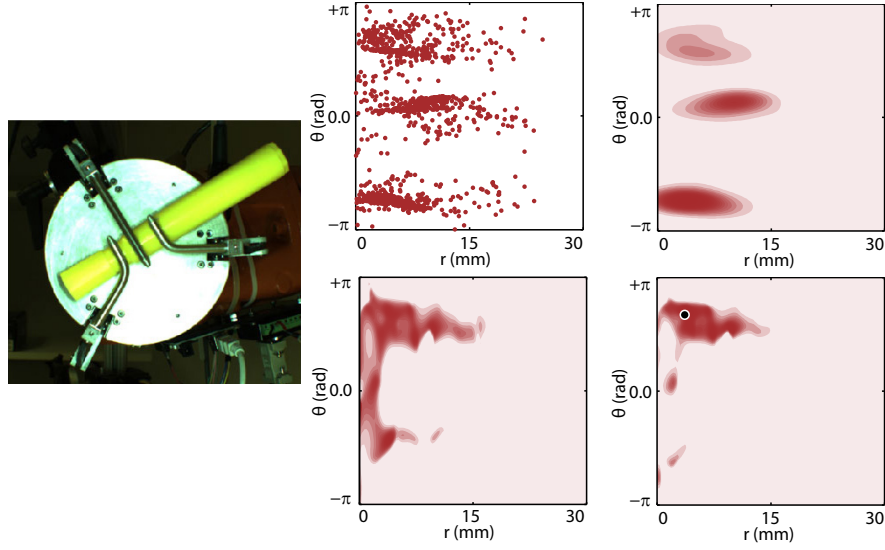


Fig. 3 [Top-center] State from 2000 grasps collected to model the *sensing capabilities* of the manipulator. [Top-right] Prior state distribution $P(x)$. [Bottom-center] Likelihood of pose of the object for an example grasp in cluster I. [Bottom-right] Corresponding posterior distribution $P(x|z)$ for the example grasp, and the most likely pose of the object.

3.2 Learning Task Requirements

We now model the probability of success of an action, $P(a|x)$. This will tell us how accurate our estimation of the state of the grasp must be for an action to successfully execute a task. While not required for our framework, we choose to state parameterize the set of actions. For example, for the task of placing a cylindrical object, if assumed to be at pose p , we design an action a_p that turns the cylinder so it is upright with respect to the ground, and then set it down. Note that a_p is an action parameterized by a chosen state p .

In general, the success of an action depends both on the action a_p itself and the true state of the system x . Since we assume state parametrized actions, we assume that the probability of success only depends on the difference $(x - p)$. For example, when placing a cylinder whose estimated axis is 1 degree 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} , so that:

$$P(a_p = 1|x) = \phi_{a_p}(x) = \phi(x - p) \quad (4)$$

The use of state parameterized actions allows us to randomly sample the space of mismatches $(x - p)$ by choosing to execute the action a_p with $p = x + \epsilon$, where

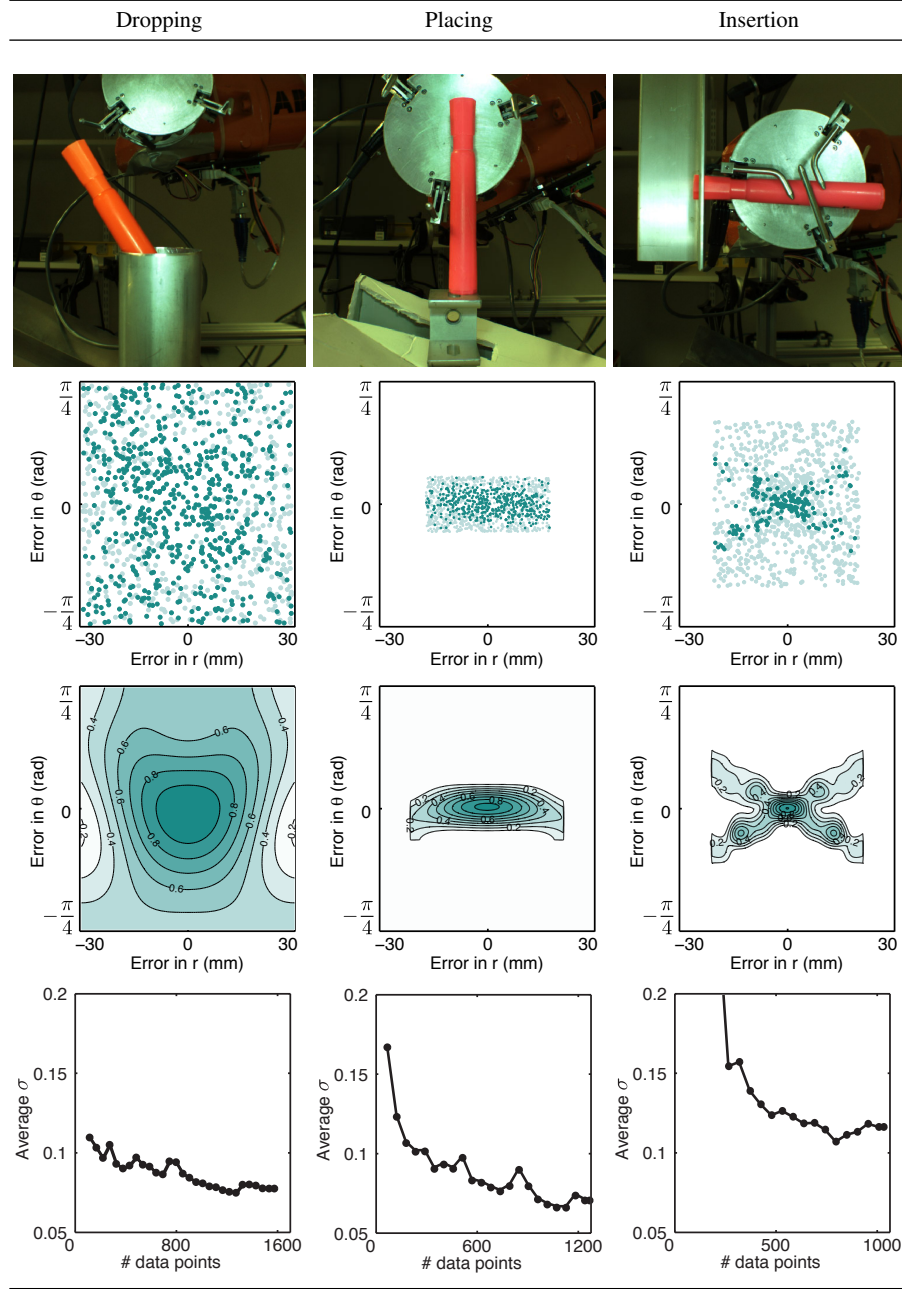


Fig. 4 [1st Row] Examples of successful executions of dropping, placing, and insertion. [2nd Row] Datapoints from perturbed task experiments. Dark points are successes and light ones are failures. Notice that the magnitude of the perturbation noise is different for each task. [3rd Row] Distribution of task requirements $P(a_p = 1|x)$ for dropping, placing and insertion, as a function of the error in state estimation. [4th 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 stop criteria.

ϵ is a uniformly distributed error in the space of system states instead of the optimal one a_x .

We next detail the process of estimating the function $\phi(\epsilon)$ from a dataset $C_2 = \{(z^i, x^i, \epsilon^i, y^i)\}_i$, where ϵ^i is the error in system state and y^i is the success/failure output of the trial. For each task, we uniformly sample ϵ^i from $\mathcal{E} = [-\Delta r_{\max}, \Delta r_{\max}] \times [-\Delta \theta_{\max}, \Delta \theta_{\max}]$, where Δr_{\max} and $\Delta \theta_{\max}$ are chosen to be large enough where we expect failure for anything outside of that range. We use a GP on C_2 to regress the Bernoulli parameter ϕ with the outcome of over 1000 executions of each task.

Figure 4 shows the requirements $\phi(x - p)$ for three different manipulation tasks: dropping, placing and insertion of a highlighter marker with a simple hand. As $|x - p|$ increases, our likelihood of task success decreases, which is as expected. Note that for the dropping task the probability decreases much slower than 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. We can be more inaccurate and still succeed at dropping. Insertion is also interesting, as it resembles the shape of an X. This can be explained by noticing that if we incorrectly try to insert the marker too high, but also tilted downward, the end of the marker still manages to fit in the hole. Generating these task requirement distributions allows us to gain key insight into how robust our post-grasp manipulation tasks are.

In general, the more data, the more accurate the regressed distributions of task requirements are. The magnitude of the variance returned by the Gaussian Process Regression 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 4 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:

$$\begin{aligned} P(a_p = 1|z) &= \int_X P(a_p = 1|x)P(x|z)dx \\ &= \int_X \phi(x - p)\mathcal{N}(z; \mu(x), \sigma(x)^2) \frac{P(x)}{P(z)} dx \\ &\simeq \int_{\mathcal{E}} \phi(\epsilon)\mathcal{N}(z; \mu(p + \epsilon), \sigma(p + \epsilon)^2) \frac{P(p + \epsilon)}{P(z)} d\epsilon \end{aligned} \quad (5)$$

where we apply the change of variables $\epsilon = x - p$.

In the experiments we approximate the integral numerically. If we grid the space of mismatches between real state and estimated state into $N_r \times N_\theta$, and call the deviation from the center ϵ_{ij} , then we can approximate the integral in (5) by the following convolution:

$$P(a_p = 1|z) \simeq \sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} \phi(\epsilon_{ij}) \mathcal{N}(z; \mu(p + \epsilon_{ij}), \sigma(p + \epsilon_{ij})) \frac{P(p + \epsilon_{ij})}{P(z)} \Delta A \quad (6)$$

where

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

Depending on the maximum value of $P(a|z)$, we can decide either to execute the task with the optimal action or to abort the execution. Figure 5 shows the predicted task success distribution for an example grasp.

Note that our framework allows for the complete decoupling of sensing capabilities and task requirements. For a given hand and object, once we have determined its sensing capabilities, whenever we have a new task, we only have to compute the task requirements, and then we can follow the above analysis to compute our overall probability. Imagine another scenario where we have an industrial robot learning in a room for days at a time. If a mobile manipulator robot learns its sensing capabilities, it can directly use the task requirements learned by the industrial robot to immediately predict its likelihood of success. If we had chosen to go directly from sensors to actions, each time we wanted to learn a new task, we would have to start all over again, without being able to reuse any learned models.

4 Experimental Validation

To validate our framework, we use the same hand and object to complete three different tasks. This requires one training set for the *sensing capabilities* of the hand, $P(x|z)$, and estimating the *task requirements*, $P(a|x)$, for each one 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.

Figure 6 compares experimental results to model predictions for each of our tasks. 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 40% of the time, the average experimental success rate for those grasps should ideally be 40%. Each task is executed 500 times for validation, and the most likely action is estimated from information provided by the three finger encoders. For all three tasks, the experimental probability follows the predicted probability, supporting the validity of the framework.

Depending on the adequacy of the in-hand sensors to capture the real state of the system and the difficulty of the specific task, each task has a region in which most predicted probabilities fall. Dropping is the easiest task, followed by placing, followed by insertion. This could have been predicted by looking at the task requirements and noting that dropping has the widest distribution of success in the

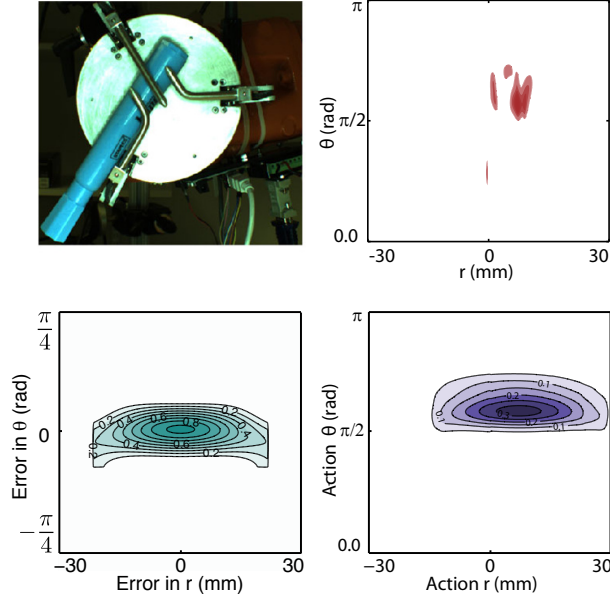


Fig. 5 [Top-left] Example grasp. [Top-right] Posterior distribution $P(x|z)$ of the pose of the object for the example grasp. Note that, for visualization purposes, we have rearranged the set of possible values for the pose of the marker from $\theta \in [-\pi, \pi]$ to $\theta \in [0, \pi]$, and allowed the distance r to take on negative values. [Bottom-left] Estimated task requirements, $P(a_p = 1|x)$, for the task of placing. [Bottom-right] Estimated probability of success for all placing actions, $P(a_p|z) \forall p$.

presence of pose error. It is clear from these experiments 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 6 shows the precision-recall curves of the three tasks. As expected, when we choose to execute a task only when our expected probability is above a certain threshold, our average success rate of the task increases. By choosing where to set the 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 likelihood of success of a post-grasp manipulation task. We contribute the following three steps:

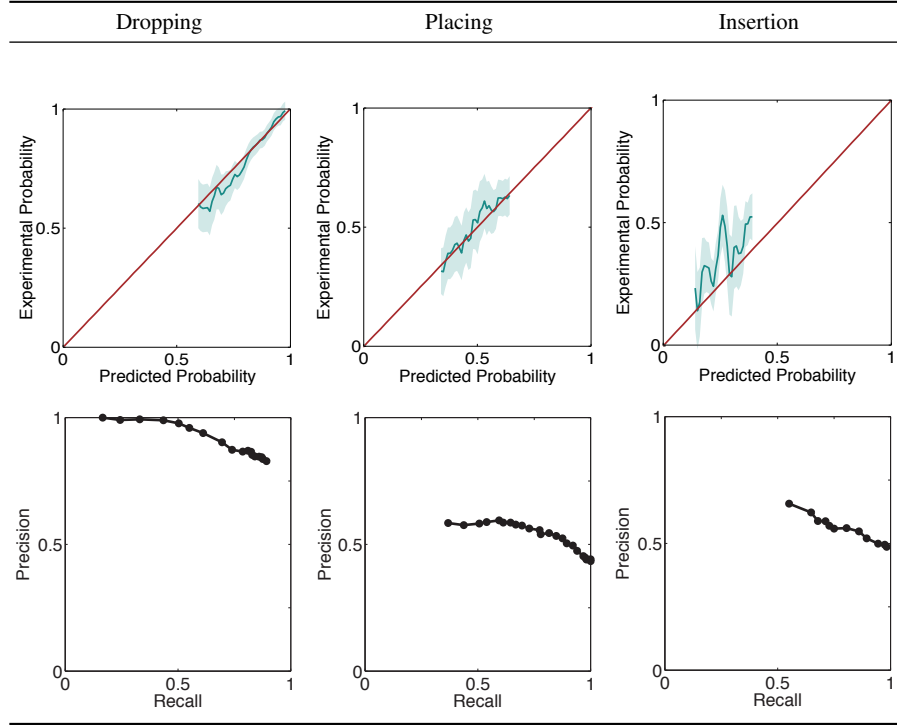


Fig. 6 [Top] Comparison between the experimental probability of success and the predicted one. The shaded region is a 95% confidence interval of the estimation of the Bernoulli parameter, according to a binomial distribution. The plots show that the predictions follow the experimental observations quite well. [Bottom] Precision-recall curves of the success in task execution when conditioning the task execution to the predicted probability go above a certain thresholding. The plot shows that we can increase our success rate by rejecting low probability grasps in the three tasks.

Sensing Capabilities: Using state and sensor pairs, we regress the posterior distribution of the state of an object. Generating a probability distribution of object location allows us to understand which grasps give good object localization and which do not, possibly informing both hand design and grasping strategies. We could attempt to find strategies to reduce uncertainty along the most troublesome directions and design hands that don't have grasps where object pose uncertainty is high.

Task Requirements: By perturbing our state, we learn a function for how our object state accuracy affects our task success probability. Generating a probability function of task success for a given object state error would be an excellent tool for identifying weaknesses in task execution. We could discover, for example, that placing fails when our marker is in a certain location, so adding a move to fix this could greatly improve our overall task success rate.

Task Success Distribution: By combining the two functions above, we can generate a function of how likely an action is to succeed at a task for any new sensor input. Note that if we have the same hand but different tasks, we only have to learn a new task requirement function. If we are executing the same task with different hands, we only need to learn its sensing capabilities. Knowing the probability of succeeding at a task before we execute it has many benefits. We can choose to abort in order to increase our overall success rate and not take unnecessary risks when success is vital. We should even be able to find an optimal policy to minimize mean time to success, similar to [18]. Also, because we calculate the entire probability distribution in action space, this would allow us to choose the best action even in the presence of constraints or other cost functions.

We performed over 8000 grasps, and 1000 trials each for placing, dropping, and insertion.

6 Discussion

Assumptions: While the framework discussed in Section 3 is quite general, we make several simplifying assumptions to actually implement it on a real system. Although already mentioned in the paper, we summarize them here. First, we assume that the probability of success of an action is conditionally independent of our sensors given the true state of the system. This is reasonable when we have a good representation; however, if it is incomplete, our estimate of $P(a|x)$ may be incorrect. For example, in the dropping task, the location of the center of mass of the marker relative to the center of the hand seemed to greatly affect the probability of success. Given that our state representation is only the axis that the marker lies on, the center of mass is not observable. This discrepancy was simply treated as extra noise in our system, and we were still able to accurately predict our probability of dropping success. Greater accuracy could have been obtained by adding this extra dimension in our state, although this would have also required us to collect much more data.

Our second assumption is that given an object pose, each of our sensors was unimodal and normally distributed. This is very dependent on the hand, object, and sensors in question. If the sensors being used cannot fully identify the object pose, one might consider adding more sensors. Note that even without this Gaussian assumption, one could still find the distribution $P(x|z)$, but comparatively more data would be required to accurately learn this distribution.

Finally, we assume that the set of actions for executing a task are state parameterized. While this is true in most cases, for the instances where it is not, in order to find $P(a|x)$, for every object state, we would need to sample the space of all actions *and* pose errors, again requiring much more data.

Robustness: In collecting all of the data for the robot to learn, it was clear that having a robust system is important but also extremely challenging. For our post-

grasp manipulation tasks we had to focus on three different aspects: object acquisition, task execution, and post-task reset. With thousands of experiments needed, a human could not sit there handing the robot objects, as this would not only be time-consuming but also introduce bias into our system, since the robot might always grasp the object in the same way. 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 one third 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 designing robust experiments, and this should not be overlooked when attempting to use our framework in a real setting.

Statistical Analysis: Another area which we would like to focus more effort on is that of understanding the statistical significance of our distributions. We suggested the use of the covariance learned using a Gaussian Process as a stopping criterion, which worked well in practice, but a more thorough analysis is needed.

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.

Acknowledgements This work was supported by National Science Foundation [NSF-IIS-0916557], the Defense Advanced Research Projects Agency [DARPA-BAA-10-28] and Army Research Laboratory [W911NF-10-2-0016]. This work does not necessarily reflect the position or the policy of the U.S. Government, DARPA or ARL. No official endorsement should be inferred.

References

1. Boots, B., Siddiqi, S.M., Gordon, G.J.: Closing the Learning-Planning Loop with Predictive State Representations. *The International Journal of Robotics Research* **30**(7), 954–966 (2011)
2. Botev, Z.I., Grotowski, J.F., Kroese, D.P.: Kernel density estimation via diffusion. *The Annals of Statistics* **38**(5), 2916–2957 (2010)
3. Brost, R.: Automatic Grasp Planning in the Presence of Uncertainty. In: *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 1575–1581. Institute of Electrical and Electronics Engineers (1986)
4. Brost, R.C., Christiansen, a.D.: Probabilistic Analysis of Manipulation Tasks: A Conceptual Framework. *The International Journal of Robotics Research* **15**(1), 1–23 (1996)

5. Cassandra, A.R., Kaelbling, L.P., Littman, M.L.: Acting Optimally in Partial Observable Stochastic Domains. In: AAAI (1994)
6. Dogar, M., Srinivasa, S.S.: A Framework for Push-Grasping in Clutter. In: Robotics: Science and Systems (RSS) (2011)
7. Fu, J., Srinivasa, S., Pollard, N., Nabbe, B.: Planar batting under shape, pose, and impact uncertainty. In: IEEE International Conference on Robotics and Automation (ICRA) (2007)
8. Goldberg, K., Mason, M.T.: Bayesian grasping. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1264–1269 (1990)
9. Goldfeder, C., Allen, P.K.: Data-driven grasping. *Autonomous Robots* **31**(1), 1–20 (2011)
10. Hsiao, K., Kaelbling, L., Lozano-Pérez, T.: Robust grasping under object pose uncertainty. *Autonomous Robots* **31**(2), 253–268 (2011)
11. Jiang, Y., Lim, M., Zheng, C., Saxena, A.: Learning to place new objects in a scene. *The International Journal of Robotics Research* (2012)
12. Kang, D., Goldberg, K.: Sorting parts by random grasping. *IEEE Transactions on Robotics and Automation* **11**(1), 146–152 (1995)
13. Laval, S.M., Hutchinson, S.A.: Evaluating Motion Strategies under Nondeterministic or Probabilistic Uncertainties in Sensing and Control. In: IEEE International Conference on Robotics and Automation (ICRA), April, pp. 3034–3039 (1996)
14. Mason, M.T., Rodriguez, A., Srinivasa, S.S., Vazquez, A.S.: Autonomous Manipulation with a General-Purpose Simple Hand. *The International Journal of Robotics Research* **31**(5), 688–703 (2012)
15. Morales, A., Chinellato, E., Fagg, A.H., del Pobil, A.P.: Using Experience for Assessing Grasp Reliability. *International Journal of Humanoid Robotics* **1**(4), 671–691 (2004)
16. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. Press, MIT (2006)
17. Rodriguez, A., Mason, M.T., Srinivasa, S.S.: Manipulation Capabilities with Simple Hands. In: International Symposium on Experimental Robotics (ISER) (2010)
18. Rodriguez, A., Mason, M.T., Srinivasa, S.S., Bernstein, M., Zirbel, A.: Abort and Retry in Grasping. In: IEEE International Conference on Intelligent Robots and Systems (IROS) (2011)
19. Stulp, F., Theodorou, E., Buchli, J., Schaal, S.: Learning to Grasp under Uncertainty. In: IEEE International Conference on Robotics and Automation (ICRA) (2011)
20. Wingate, D.: Exponential family predictive representations of state. Ph.D. thesis, University of Michigan (2008)