Abort and Retry in Grasping

Alberto Rodriguez*, Matthew T. Mason*, Siddhartha S. Srinivasa†, Matthew Bernstein* and Alex Zirbel*
* Carnegie Mellon University — The Robotics Institute
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
albertor@cmu.edu
† Intel Labs Pittsburgh
4720 Forbes Avenue, Pittsburgh, PA 15213, USA
siddhartha.srinivasa@intel.com

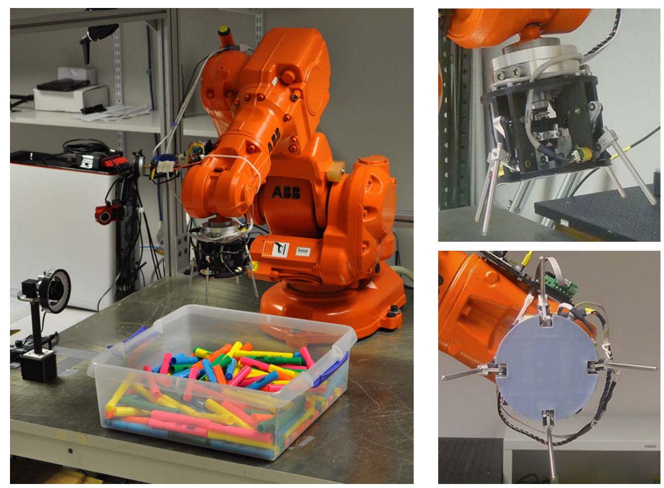
***Abstract*— Iteration is often enough for a simple hand to accomplish complex tasks, at the cost of an increase in the expected time to completion. We propose to streamline this idea by allowing a simple hand to abort early and retry grasps as soon as it realizes that the task is likely to fail. This paper presents two key contributions. First, we learn a probabilistic model of the relationship between the likelihood of success of a grasp and its grasp signature—the trace of the state of the hand along the entire grasp motion. Second, we model the iterative process of early abort and retry as a Markov chain and optimize the expected time to completion of the grasping task by effectively thresholding the likelihood of success. Our experiments show that early abort and retry significantly increases the efficiency of a simple approach to grasping with a simple hand.**

I. INTRODUCTION

Simple hands are characterized by simple mechanical designs and simple control strategies, both of which compromise the potential generality of the hand. In practice, and based on observations of humans using simple tools and effectors, simple hands offer broader manipulation capabilities than any autonomous system has yet demonstrated.

After arguing the case for simplicity in [1], and with the aim of demonstrating manipulation capabilities with simple hands, we approached the problem of singulating objects out of a bin in [2]. The approach in [2] has three key elements:

- **Simple mechanism**: The simple hand in Fig. 1. It has thin cylindrical fingers compliantly coupled to a single actuator, arranged symmetrically around a low friction circular palm.
- **Simple control**: Contrary to the more traditional approach where robotic hands try to "put the fingers in the right place", we close the hand and "let the fingers fall where they may." We expect the fingers either to drive the object to a stable pose or to reject it, effectively reducing the space of possible outcomes of the grasp. This simplifies the relationship between the signature of a grasp and its outcome, facilitating the creation of a data-driven model.
- **Iteration**: To address the expected failures of the simple approach, we use the iteration scheme in Fig. 2 until successfully singulating an object.



(a) (b)

Fig. 1. (a) Bin-picking scenario. A robotic manipulator drives a simple hand in and out of a bin full of identical objects. (b) P2, a simple hand prototype with four fingers compliantly coupled to a single actuator. Motor and fingers have absolute encoders that provide the full state of the hand.

This work was sponsored in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-10-2-0016, DARPA, and NSF. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, DARPA, or the U.S. Government.

The simplicity of the approach often comes at the cost of increasing the expected time to a successful grasp. We propose in this paper to reduce that expected time by using proprioceptive feedback to predict failure early during the execution of the grasp and possibly abort and restart the procedure. To predict failure we keep track of the instantaneous likelihood of success. Our goal is to show that we can optimize the threshold on the instantaneous likelihood of success to improve the performance of the simple approach in terms of expected time to a successful grasp.

We maintain the same scenario as in our previous work, Fig. 1, a bin-picking task. Relevant to our approach is the concept of grasp signature, the trace of the state of the hand along the entire grasp motion. In earlier work we

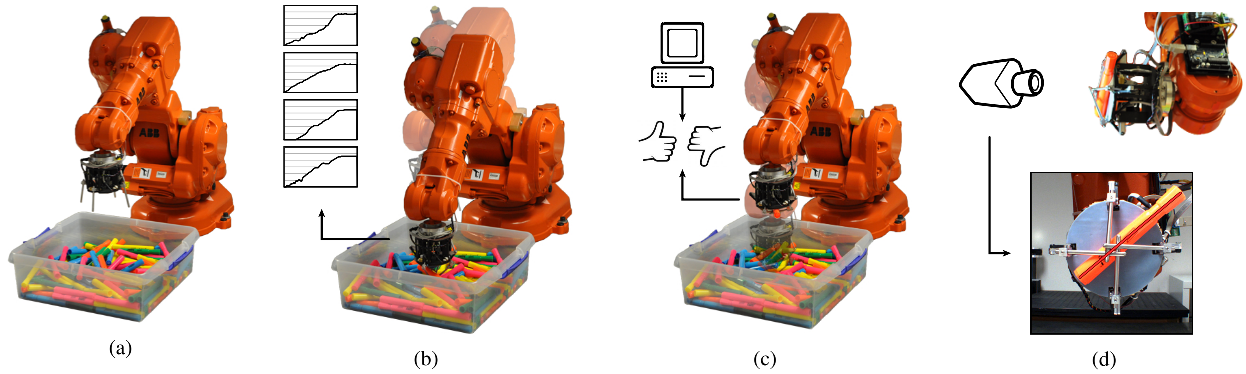


Fig. 2. Iterative framework: The grasp is iterated until the system confidently detects success in the task of singulating an object from the bin. (a) At the beginning of the grasp the hand is above the bin. (b) The interaction of the hand with the environment is logged during the entire grasp process, constituting the grasp signature. (c) During run time, a data-driven model classifies the grasp as success or failure. In the case of failure, the grasp restarts. (d) During the training phase, the success/failure learning process is supervised by a vision system that provides ground truth on the number and location of markers grasped.

learned a discriminative model of the relationship between the signature and outcome of a grasp. In this paper we instead learn a probabilistic model to keep track of the instantaneous probability of success all along the grasp process. This way we are able to detect grasps that are so likely to fail that aborting and restarting will reduce the expected time to success. In Sect. III we introduce in a more detailed manner the concepts of grasp signature, expected time to a successful grasp and likelihood of success.

In Sect. IV we model the iterative process of early abort and retry as a Markov chain and derive an analytical expression for the expected time to a successful grasp. We then use the model in Sect. V to optimize the abort thresholds to minimize the expected time to a successful grasp.

Our experiments show that early abort and retry significantly increases the efficiency of the system. The implementation of the system is detailed in Sect. VI. We conclude in Sect. VII describing how the overhead of the iterative framework—expected time to a successful grasp minus the nominal length of the grasp—evolves with the number of possible abort points along the grasp process.

II. PREVIOUS WORK

This paper focuses on a bin-picking scenario, a challenging grasping task combining high clutter with high pose uncertainty, for decades the focus of numerous research efforts [3], [4]. One of the main goals is to show that we can use proprioceptive feedback to improve the capabilities of a simple hand. In-hand sensor information has previously been shown to improve grasping performance [5].

We use a data-driven methodology for failure detection and abort optimization based on the signature of the grasp. Data-driven approaches have previously been proposed for failure detection in different contexts, including tool breakage detection in milling operations [6], [7], machine vibration analysis [8] or failure detection in assembly operations [9], [10].

In the process of detecting failure, we estimate the outcome of the grasp based on kinesthetic sensor data. Bicchi, Salisbury and Brock [11] explored a similar problem: assuming known finger shape and location, they estimate the contact point from a measured applied wrench, a technique known as *intrinsic contact sensing*. This contact information can be used to infer the pose of a known shape.

Also relevant is the related problem of inferring the object location from kinesthetic or contact sensor data, studied by Siegel [12], Jia and Erdmann [13], [14] and Wallack and Canny [15]. While all these works are based on analytical models of contact and grasp mechanics, in this paper we use a statistical data-driven approach to create a model of the relationship between the signature of the grasp process and its outcome, thereby bypassing the intermediate estimation of contact points. This way we incorporate numerous sources of information very challenging to capture otherwise, including the effect of the grasping motion and that of surrounding clutter. Laaksonen, Kyrki and Kragic [16] compared the effectiveness of different statistical data-driven methods for online estimation of grasp stability, based on both kinesthetic and contact sensor data.

The framework proposed in this paper for optimizing the series of abort conditions has some similarities with the cascades of classifiers proposed in computer vision to speed up the detection rate of object classifiers without compromising performance [17], [18]. A series of incrementally more computationally expensive classifiers trade off the cost and risk of taking a decision or letting the following classifier in the cascade do it.

III. GRASP SIGNATURE AND EARLY FAILURE DETECTION

A. Grasp Signature

By *grasp signature* we mean the trace of the state of the hand along the entire grasp motion as perceived by the hand's own sensors. The signature can be composed of, but not limited to, time-stamped data from joint encoders, tactile sensors, and torque sensors. This work builds on the assumption that the grasp signature encodes enough information to characterize the outcome of a grasp.

The compliant simple hand used in this work, P2, has absolute motor and fingers encoders that allow us to recover the full kinematic state of the hand. Figure 3 shows a side by side comparison of the finger encoder signatures of examples of successful and failed grasps.

In previous work [2] we demonstrated with P2 that it is possible to accurately detect correct singulation in a bin picking task based only on grasp signature information.

In that experiment, a robotic manipulator blindly drives the hand in and out of the bin full of objects following a preprogrammed path. After each retrieval, an offline learned model decides whether the system is confident enough about correct singulation of an object or if the robot should restart the procedure.

B. Expected Time to a Successful Grasp

In the iterative framework in Fig. 2, we define the *expected time to a successful grasp* τ, as the average number of grasp attempts needed to obtain a successful grasp of the object, multiplied by the time span of the grasp T.

In [2] we show how the system precision (false negative rate) can be improved by tuning the weights of positive and negative training examples, at the cost of increasing the false positive rate. This has the unsought consequence of increasing the probability of iteration f of the system, leading to an increase in the expected time to a successful grasp.

In this paper we propose to abort grasps that the system predicts likely to fail as a technique to reduce the expected time to a successful grasp while still maintaining high values for the precision of the system.

The expected time to a successful grasp of the system in Fig. 2 is a function of the probability of iteration, $\tau = \frac{T}{f}$. Analogously, we show in Sect. IV that the expected time with early abort is a function of the probability of early failure.

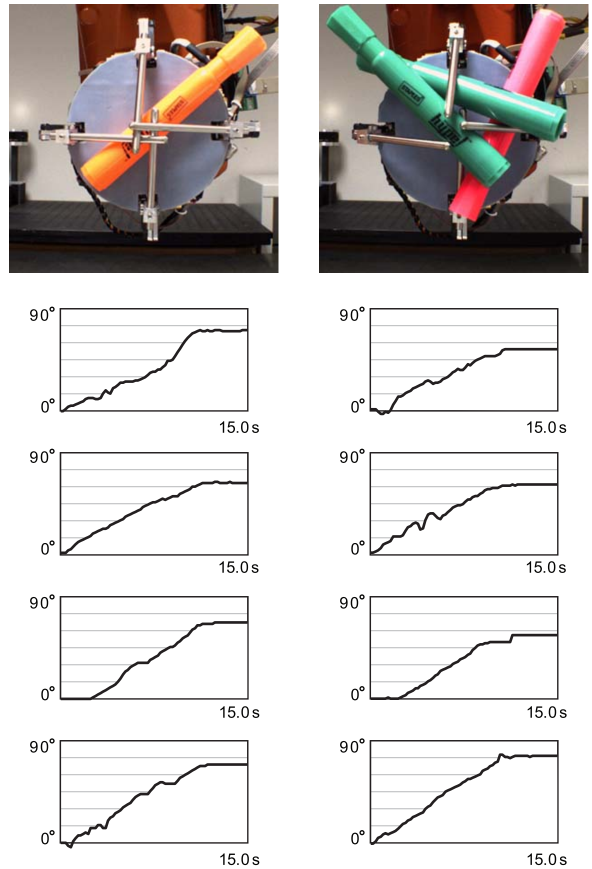
C. Success Probability

We are interested in confidently predicting failure early during the execution of the grasp. This will enable aborting grasps early and consequently reducing the cost of iteration.

To aid in the decision of whether or not to abort a grasp at instant t, we learn a probabilistic model of the relationship between the first t seconds of the signature and the final outcome of the grasp. This model provides us with an estimation of the instantaneous probability of success at instant t. This contrasts with our previous approach in [2] where we learn a discriminative model to signal success at the end of the grasp.

The specifics of how to learn the model are detailed in Sec. VI-C. For the following sections we will assume that for a given a set of K grasp signatures $\{g_i(t)\}_{1...K}$ and correspondent labels $\{l_i\}_{1...K} \in \{+,-\}$ we are able to learn a probabilistic model of their relationship:

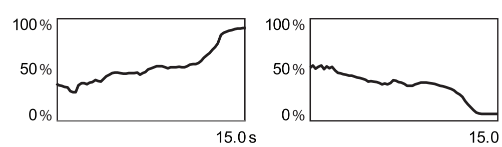
$$\mathcal{M} : g([0,t]) \mapsto p(t)$$



(a) (b)

Fig. 3. Side by side comparison of the grasp signature (4 finger encoders) of a typical (a) successful and (b) failed grasp. The fingers begin the grasp perpendicular to the palm (0°) and reach the final position shown in the figures.

where $g(t)$ is the signature of a grasp execution and $p(t)$ is the *success probability signal*, the instantaneous probability of success of the grasp as predicted by the model $\mathcal{M}$. As an example, Fig. 4 shows the success probability signal for the successful and failed grasp signatures in Fig. 3.



(a) (b)

Fig. 4. Evolution of the probability of success for the successful (a) and failed (b) examples illustrated in Fig. 3. At the beginning of the grasp, the system is uncertain about the possible outcome, but gradually becomes more confident about the outcome.

With the estimation of the instantaneous probability of success $p(t)$ in hand, we introduce the *cut-off probability signal* $\pi(t)$ that specifies the threshold for when to abort a grasp execution.

It is impractical to keep track of the value of the probability of success in a continuous manner. In

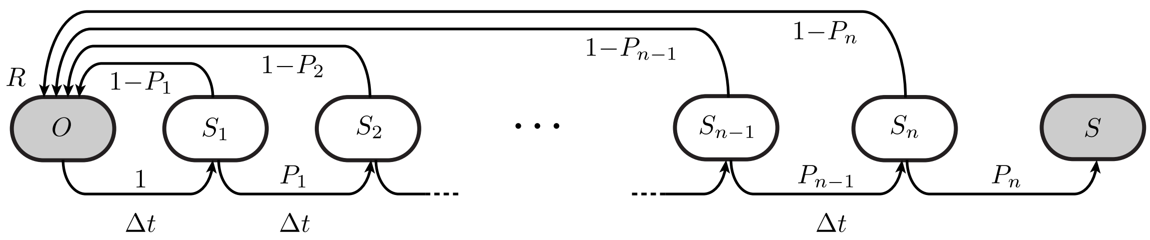


Fig. 5. Markov Chain that models a system with n abort points at instants $t_i = i \cdot \Delta t$. O is the initial state, $S_1 \dots S_n$ the abort points and S the end state. The system reaches the end state if and only if it is not discarded by any of the classifiers at the abort points. The cost of transition is equal to Δt while we assume the abort cost independent of the state and equal to R. At each state S_i the grasp continues with probability P_i and aborts with probability $1 - P_i$.

practice we discretize the grasp into n time slices $[t_0, t_1], [t_1, t_2] \dots [t_{n-1}, t_n]$ and train n independent probabilistic classifiers at instants $\{t_i\}_{1 \dots n}$. As the grasp progresses, they output a series of estimated success probabilities $p_1 \dots p_n$ that get compared against n probability thresholds $\pi_1 \dots \pi_n$:

$$\text{If } p_i < \pi_i \ \rightarrow \ \text{ABORT at } t_i \tag{1}$$

A unique contribution of this paper is to show how to model the execution of such a system with n possible aborting points, Sect. IV, and how to optimize the cut-off probabilities to minimize the expected time to a successful grasp, Sect. V.

IV. MODELING ABORT AND RETRY

In this section we model the steady-state behavior of the system proposed in Sect. III-C with n possible abort points. We will call O the initial state of the system before the beginning of the grasp motion, S_i the abort points along the duration of the grasp at instants t_i, $i = 1 \dots n$, and S the final success state.

The system can be represented by the set of states and transitions in the chain in Fig. 5. Notice that the system satisfies the Markov property, i.e., the following state depends only on the current state and not on the past. All states except the initial one trivially satisfy the property since, beginning from S, there is only one possible history of transitions to get to them. If we assume statistical independence between successive repetitions of the experiment, the initial state also satisfies the Markov property.

The behavior of a time-homogeneous Markov chain is represented by the transition probabilities between states. In our case we introduce the probabilities P_i: probability of transition from state S_i to state S_{i+1}. At each state S_i the abort probability is then $1 - P_i$. P_n is the fraction of grasp attempts that reach the end of the execution and are actually classified as good grasps.

The Markov chain model proves useful for the purpose of analyzing the steady-state behavior of the system, and in particular, the expected time to a successful grasp. We assume all abort points to be equispaced in time, with a constant spacing of Δt. We also suppose that the cost in time of aborting, R, is constant and independent of the state of the system. The following proposition gives a closed form expression for the expected time to success as a function of the transition probabilities.

Proposition 1 (Expected time to a successful grasp): In the iterative system of Fig. 5 with n equispaced abort points and transition probabilities $P_1 \dots P_n$, the expected time to success τ can be expressed as:

$$\tau = \Delta t \left[\frac{1 + \sum_{i=1}^{n-1} \left(\prod_{j=1}^{i} P_j \right)}{\prod_{i=1}^{n} P_i} \right] + R \left[\frac{1 - \prod_{i=1}^{n} P_i}{\prod_{i=1}^{n} P_i} \right] \tag{2}$$

Proof: We introduce the intermediate variables τ_0, $\tau_1 \dots \tau_n$ to represent the expected time to success from each one of the states of the system O, $S_1 \dots S_n$ correspondingly. Notice that τ_0 is by definition equal to the expected time to a successful grasp τ. We will prove the general term by induction on the number of time slices n.

For the case $n = 1$, the Markov chain reduces to the one in Fig. 6:

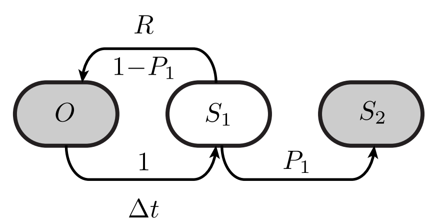


Fig. 6. Markov chain of the system for the case of $n = 1$. In this case, the only possible abort point t_1 is at the end of the grasp. It models the original framework in Fig. 2.

In steady-state the expected times τ_0 and τ_1 are related by the equations:

$$\left\{ \begin{array}{lcl} \tau_0 & = & \Delta t + \tau_1 \\ \tau_1 & = & (1 - P_1)(\tau_0 + R) \end{array} \right. \tag{3}$$

Solving the system (3) for τ_0, we get:

$$\tau_0 = \Delta t \cdot \frac{1}{P_1} + R \cdot \frac{1 - P_1}{P_1} \tag{4}$$

which satisfies the general term.

Now we assume that the general term is correct for the case of $n - 1$ abort points and we prove for the case of n. The Markov chain with n abort points in Fig. 5 is equivalent to the simplified chain in Fig. 7 where the first $n-1$ states are combined into a macro initial state O^* with transition cost τ^*.

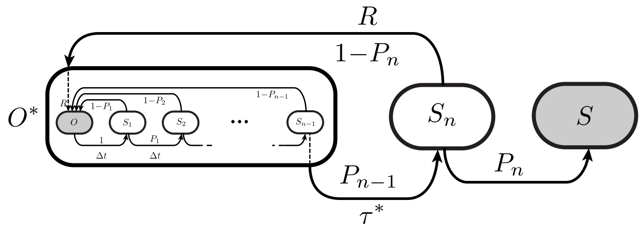


Fig. 7. Equivalent Markov chain for the case of n abort points. The first $n-1$ abort states can be included in a macrostate O^*. The new transition cost τ^* from O^* to S_n is the expected time to a successful grasp of the subsystem with $n-1$ abort states plus Δt, the original transition cost from S_{n-1} to S_n.

The new initial state behaves internally as a system with $n-1$ abort points. By induction, the transition cost from O^* to S_n is:

$$\tau^* = \Delta t \left[\frac{1+\sum_{i=1}^{n-2}\left(\prod_{j=1}^{i} P_j\right)}{\prod_{i=1}^{n-1} P_i} \right] + R \left[\frac{1-\prod_{i=1}^{n-1} P_i}{\prod_{i=1}^{n-1} P_i} \right] + \Delta t$$

The simplified equivalent system in Fig. 7 has the same structure as in the case of $n=1$, therefore, using (4) the expected time to a successful grasp can be computed as:

$$\begin{aligned}
\tau_0 &= \tau^* \cdot \frac{1}{P_n} + R \cdot \frac{1-P_n}{P_n} = \\
&= \Delta t \left[\frac{1+\sum_{i=1}^{n-2}\left(\prod_{j=1}^{i} P_j\right)}{\prod_{i=1}^{n} P_i} \right] + \\
&\quad + R \left[\frac{1-\prod_{i=1}^{n-1} P_i}{\prod_{i=1}^{n} P_i} \right] + \Delta t \cdot \frac{1}{P_n} + R \cdot \frac{1-P_n}{P_n} = \\
&= \Delta t \left[\frac{1+\sum_{i=1}^{n-1}\left(\prod_{j=1}^{i} P_j\right)}{\prod_{i=1}^{n} P_i} \right] + R \left[\frac{1-\prod_{i=1}^{n} P_i}{\prod_{i=1}^{n} P_i} \right]
\end{aligned}$$

which concludes the proof. ■

In the next section we see how the expression for τ in Proposition 1 simplifies the estimation of the expected time to a successful grasp of the system.

V. OPTIMIZING ABORT AND RETRY

As detailed in Sect. III-C, the learning system comprises n predictive models that produce success probability estimates p_i at the abort points t_i. Each state S_i decides whether to abort by comparing probability p_i with threshold π_i. In this section we show how to optimize the thresholds $\pi_1 \dots \pi_n$ to minimize the expected time to a successful grasp.

To optimize the expected time τ, we need to study how variations in the thresholds π_i affect τ. An online experimental approach would be impossibly time consuming, requiring numerous experiments to estimate the expected time to success for every trial value of the thresholds.

Instead of the experimental approach, we combine the analytical expression of Proposition 1 with offline experiments. Given a candidate set of thresholds π_i, we can use offline experimental data to estimate transition probabilities P_i, and then apply equation (2) to estimate τ. The transition probabilities can be estimated experimentally by running K grasp executions and computing:

$$P_i = P\,[\ p_i > \pi_i\ ] = \frac{\text{Grasps reach } S_{i+1}}{\text{Grasps reach } S_i} \tag{5}$$

It is key to notice that, when using the transition probabilities as an intermediate step to evaluate τ, it is not necessary to run the experiment again even if the values of the cut-off probabilities change. Equation (2) allows a more efficient strategy. Assuming that the learned predictive models do not change, we can reapply the abort condition in (1) for the same K grasp executions, but now with the new cut-off probabilities $\pi_1 \dots \pi_n$. We then reestimate the transition probabilities with (5). As a consequence, once we have captured the complete signatures of K grasp executions, the optimization process can be done completely offline and without any more required grasp execution.

In order to optimize (2) analytically, we would need to control the transition probabilities. However, we only have direct control over the cut-off probabilities. The relationship between π_i and P_i is complex, because P_i depends on the value of all previous thresholds $\pi_1 \dots \pi_i$ and on the actual data. An analytical expression of τ is terms of $\pi_1 \dots \pi_n$ is not feasible, so we use a direct search method to optimize it, given that the evaluation of the cost function is fast.

In Sect. VII we detail the implementation of the optimization and the results obtained, in particular how the expected time decreases with the number of abort points n.

VI. IMPLEMENTATION

A. System Architecture

The system implementation has a modular design, based on the ROS (Robot Operating System) architecture [19]. Each subsystem is contained within a separate node, with messages being passed between the nodes containing both sensory data and commands.

A finite state machine governs the overall system. The state machine implements the Markov chain in Fig. 5, cycling through each one of the steps of the grasp and allowing for easy modification of the grasp behavior. Different nodes within the system include:

- **Main Controller**: Primary node which implements the state machine.
- **Robot Controller**: Controls the position of the industrial manipulator.
- **Grasp Controller**: Controls the motor in the hand, and broadcasts motor and finger encoder positions.
- **Vision Interface**: Aggregated vision routines to provide ground truth for the learned models both on the number of markers grasped and their position within the hand.
- **Learning Interface**: Receives motor and encoder readings, and broadcasts success probabilities.

B. Vision System

The data-driven approach used to model the probability of success in Sect. VI-C requires first running a large number of grasp executions and logging their signature and outcome. The vision system is meant to provide feedback both in terms of the number of objects grasped and the location of the objects within the palm of the gripper, making the overall system self-supervised.

We have implemented a vision system tailored to the specific application and object (highlighter marker) using Willow Garage's OpenCV vision processing library [20]. It is composed of the following steps:

1) **Background subtraction:** Prior to the testing process, we capture an image of the hand with no markers. We then black out all areas of the image reasonably similar to the calibration image. This removes extraneous noise (color of the robot arm, objects in background, etc).
2) **Find color regions:** Since the highlighter markers are brightly colored, we threshold the image to determine regions of color. We then clean those same regions by removing small clusters of color.
3) **Find edges and lines:** We recognize markers by their straight edges using the Canny edge detector near color regions. We then use the Hough line detector to determine prominent lines in the image and assume that the long edges of each marker are among those.
4) **Most likely position for a marker:** Each detected line is scored proportional to the amount of color to each one of its sides. Iteratively we detect the most likely edge of a marker and subtract the color labeled region until insufficient color is left in the image for another marker to exist.

Fig. 8 shows an example of the output of each one of the four substeps of the vision system.

We evaluated the vision system with 266 images captured in successive trials. In the task of classifying the grasp outcome between the cases of 0, 1 or more markers, the algorithm was able to correctly classify all images except one where a marker was caught in the unlikely position of pointing directly at the camera. The accuracy of the vision algorithm is high enough to treat its output as ground truth for the posterior learning system.

Within the class of multiple markers, the vision system correctly distinguishes between 2, 3, 4 or more markers in 126 out of 133 images in the dataset. Though this is not immediately useful for the present application, it shows that the system is reliable enough for future work.

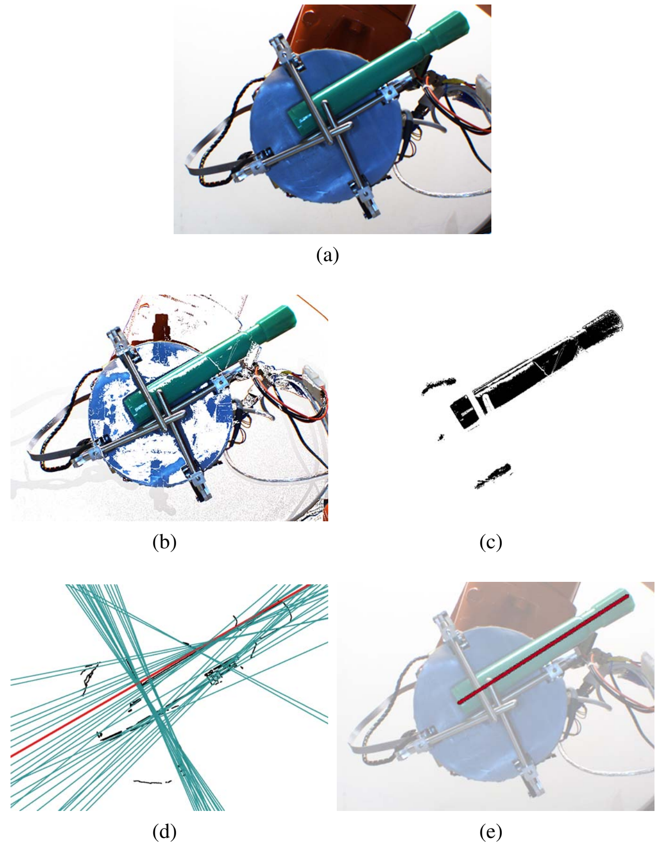


Fig. 8. The vision system outputs the position and orientation of the marker within the palm of the hand. (a) Example of input image to the vision system. (b) Filtered image after background subtraction. (c) Image after color region filtering. (d) Edge and line detectors. (e) Most likely position of the marker.

C. Learning System

As detailed in Sect. III-C, the objective of the learning system is to model the relationship between the signature of a grasp and the evolution of the success probability $p(t)$. The system decides to abort and retry a grasp by setting a threshold on that probability.

For that, we learn a probabilistic model of that relationship at predetermined points in time along the grasp signature. Among the different available techniques for probabilistic classification we choose *Relevance Vector Machines* (RVM) [21], which employ a similar formulation as Support Vector Machines, but use a Bayesian inference to provide probabilistic classification. We use the implementation provided in the Dlib Machine Learning library [22].

Prior to training the RVMs, we use *Principal Component Analysis* (PCA) [23] to reduce the dimensionality of the grasp signature. At each abort point t_i, we compress the section of the signature $[0, t_i]$. PCA finds a linear transformation of the signature into a smaller number of linearly uncorrelated features while retaining most of the original variability across the set of signatures.

After compression of the signatures with PCA, we use half of the training data to learn the RVMs. The other half will be used in Sect. VII to optimize the cut-off probabilities. Figure 4 shows an example of the evolution of the estimation of the success probability provided by the trained RVMs.

VII. RESULTS

In order to optimize the cut-off probabilities $\pi_1 \dots \pi_n$ we first capture the signatures of $K = 200$ grasp executions. Out of those we draw randomly $\frac{K}{2}$ that we use to train the the probabilistic classifiers as detailed in Sect. VI-C. We use then the other $\frac{K}{2}$ to optimize the probability thresholds.

TABLE I

NORMALIZED EXPECTED TIME TO A SUCCESSFUL GRASP.

n	τ	Improvement
1	2.17	-
2	2.12	4.3%
4	1.98	16.5%
8	1.91	22.0%
16	1.58	50.4%

For any given value of the cut-off probabilities, we make use of (2) to efficiently evaluate the expected time to a successful grasp. We use the *ga* optimizer provided by Matlab, choosing the expected time as the cost function to optimize and the cut-off probabilities as the set of parameters.

We normalize all obtained expected times by T, the time span of the grasp, so that $\tau = 1$ is the asymptotically optimal solution. In that case $\tau = \frac{1}{f}$ is the expected time in the original framework without early abort, where f is the success ratio in the original system. The improvement when using early abort is measured as the percentage of decrease of the expected time from the baseline $\frac{1}{f}$ towards the optimal.

Table I details the variation of the normalized expected time to a successful grasp with the number of abort points n after optimization. The case $n = 1$ is the baseline to compare with (system without early abort). Around $n = 16$ the optimization problem gets too big to be addressed by the off-the-shelf optimizer *ga* in Matlab. It is sufficient, though, to demonstrate that early abort reduces the expected time to a successful grasp, in the studied case with an improvement of up to 50%.

VIII. DISCUSSION AND FUTURE WORK

In this paper we introduce the concept of early abort and retry in the context of grasping in a bin picking task. We allow a hand to abort and retry the grasp as soon as it is confident that it will fail. In doing so, we improve the efficiency of the system with respect to earlier work and allow a simple hand to be competent in solving a complex task.

The main contribution of this paper is to show that we can model a iterative system with fixed abort points as a Markov chain and use it to optimize the expected time to successful completion of the desired task.

Although we have focused on grasping in a bin-picking scenario, the proposed methodology generalizes to any process generating a signature that correlates with the potential success or failure of the execution. Automated assembly is an example of application that would benefit from early abort to improve their performance.

Our long-term goal is to demonstrate broad manipulation capabilities with simple hands. In earlier work, we approached the bin-picking problem with a blind policy driving the hand. Early abort is a step forward by introducing a binary policy that at each instant allows the hand either to abort or to continue with the execution. In our process to gradually complexify the grasp policy, we intend in future work to learn an optimal singulating policy from a small parametrized set.

REFERENCES

[1] M. T. Mason, S. S. Srinivasa, A. S. Vazquez, and A. Rodriguez, “Generality and Simple Hands,” Robotics Institute, Carnegie Mellon University, Technical Report CMU-RI-TR-10-40, 2010.

[2] A. Rodriguez, M. T. Mason, and S. S. Srinivasa, “Manipulation Capabilities with Simple Hands,” in *International Symposium on Experimental Robotics (ISER)*, 2010.

[3] R. Tella, J. Birk, and R. Kelley, “General Purpose Hands for Bin-Picking Robots,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 12, no. 6, pp. 828–837, 1982.

[4] K. Boehnke, “Object Localization in Range Data for Robotic Bin-Picking,” in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2007, pp. 572–577.

[5] A. M. Dollar, L. P. Jentoft, J. H. Gao, and R. D. Howe, “Contact sensing and grasping performance of compliant hands,” *Autonomous Robots*, vol. 28, no. 1, pp. 65–75, Aug. 2010.

[6] S. Cho, S. Asfour, A. Onar, and N. Kaundinya, “Tool Breakage Detection Using Support Vector Machine Learning in a Milling Process,” *International Journal of Machine Tools and Manufacture*, vol. 45, no. 3, pp. 241–249, 2005.

[7] Y.-W. Hsueh and C.-Y. Yang, “Prediction of tool breakage in face milling using support vector machine,” *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 9-10, pp. 872–880, 2007.

[8] D. M. Tax, A. Ypma, and R. P. Duin, “Support Vector Data Description Applied to Machine Vibration Analysis,” in *ASCI’99*, vol. 54, no. 1, Heijen, Netherlands, Jan. 1999.

[9] K. Althoefer, B. Lara, Y. H. Zweiri, and L. D. Seneviratne, “Automated Failure Classification for Assembly with Self-Tapping Threaded Fastenings Using Artificial Neural Networks,” *Journal of Mechanical Engineering Science*, vol. 222, no. 6, pp. 1081–1095, 2008.

[10] A. Rodriguez, D. Bourne, M. T. Mason, G. F. Rossano, and J. Wang, “Failure Detection in Assembly : Force Signature Analysis,” in *IEEE Conference on Automation Science and Engineering (CASE)*, 2010.

[11] A. Bicchi, J. K. Salisbury, and D. L. Brock, “Contact Sensing from Force Measurements,” *The International Journal of Robotics Research*, vol. 12, no. 3, p. 249, 1993.

[12] D. Siegel, “Finding the Pose of an Object in a Hand,” in *IEEE International Conference on Robotics and Automation*, 1991, pp. 406–411.

[13] Y. Jia and M. A. Erdmann, “Geometric Sensing of Known Planar Shapes,” *The International Journal of Robotics Research*, vol. 15, no. 3, pp. 365–392, 1996.

[14] ——, “Pose and Motion from Contact,” *International Journal of Robotics Research*, vol. 18, no. 5, pp. 466–490, 1999.

[15] A. S. Wallack and J. F. Canny, “Generalized Polyhedral Object Recognition and Localization Using Crossbeam Sensing,” *The International Journal of Robotics Research*, vol. 16, no. 4, pp. 473–496, 1997.

[16] J. Laaksonen, V. Kyrki, and D. Kragic, “Evaluation of Feature Representation and Machine Learning Methods in Grasp Stability Learning,” in *IEEE International Conference on Humanoid Robots*, 2010, pp. 112–117.

[17] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[18] M. Dundar and J. Bi, “Joint Optimization of Cascaded Classifiers for Computer Aided Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[19] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “ROS: an Open-Source Robot Operating System,” in *International Conference on Robotics and Automation*. IEEE, 2009.

[20] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[21] M. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *The Journal of Machine Learning Research*, vol. 1, no. 3, pp. 211–244, Aug. 2001.

[22] D. King, “Dlib-ml: A Machine Learning Toolkit,” *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[23] L. Smith, “A tutorial on principal components analysis,” USA, 2002.