Learning Grasp Strategies Composed of Contact Relative Motions

Robert Platt
Dextrous Robotics Laboratory
Johnson Space Center, NASA
robert.platt-1@nasa.gov

***Abstract*— Of central importance to grasp synthesis algorithms are the assumptions made about the object to be grasped and the sensory information that is available. Many approaches avoid the issue of sensing entirely by assuming that complete information is available. In contrast, this paper focuses on the case where force feedback is the only source of new information and limited prior information is available. Although, in general, visual information is also available, the emphasis on force feedback allows this paper to focus on the partially observable nature of the grasp synthesis problem. In order to investigate this question, this paper introduces a parameterizable space of atomic units of control known as *contact relative motions* (CRMs). CRMs simultaneously displace contacts on the object surface and gather force feedback information relevant to the object shape and the relative manipulator-object pose. This allows the grasp synthesis problem to be re-cast as an optimal control problem where the goal is to find a strategy for executing CRMs that leads to a grasp in the shortest number of steps. Since local force feedback information usually does not completely determine system state, the control problem is partially observable. This paper expresses the partially observable problem as a k-order Markov Decision Process (MDP) and solves it using Reinforcement Learning. Although this approach can be expected to extend to the grasping of spatial objects, this paper focuses on the case of grasping planar objects in order to explore the ideas. The approach is tested in planar simulation and is demonstrated to work in practice using Robonaut, the NASA-JSC space humanoid.**

I. Introduction

In many potential applications of robot grasping, approximate shape and pose parameters of the object to be grasped may be known ahead of time while exact parameters may be impossible to predict. For example, consider the problem of manipulating a cup or mug. The exact pose or geometry of the mug may be unknown, but the identity of the object as a mug may be perfectly clear from task context or from gross visual feedback. This characterization of the information available to the robot is particularly relevant to materials handling problems on the moon or Mars. Consider the task of grasping a cylindrical connector or a piece of tubing. While the robot may be ignorant of the exact diameter and pose, it may be evident that the object is a long cylinder of some kind. Similarly, a robot may know that a package is to be grasped by a U-handle even if the exact pose or geometry of the package is not known. In general, it is asserted that a large number of manipulation problems exist for which the solution space can be constrained by general information about the object or problem context.

Although this type of general information is frequently available, most current approaches to grasp synthesis do not leverage it to improve efficiency or robustness. First, consider planning approaches to grasp synthesis [1], [2], [3]. These approaches typically require a complete description of the object geometry before processing begins. Based on object geometry, a set of desired contact positions relative to the object (a contact configuration) that satisfies a grasp criterion is identified. Then, based on the object pose, the contact configuration is translated into a set of desired positions in the robot base frame. Finally, a position controller moves the manipulator contacts to this goal configuration. Approaches of this type assume that complete information about object pose and geometry is available; if only general information about the object is known, then additional techniques are needed to handle the uncertainty.

Grasp control methods are an alternative to grasp planning. Whereas planning approaches assume that the complete object geometry is known, grasp control approaches make only minimal assumptions (for example, that the object is convex) [4], [5]. Grasp control methods compensate for the dearth of prior information by using force feedback at the contacts. The manipulator is assumed to be equipped with sensors that measure the object surface normal at the contacts. The robot starts out in contact with the object. Based on force feedback, the controller displaces the contacts tangent to the object surface toward a quality grasp configurations. Ultimately, for arbitrary convex objects, the controller is guaranteed to reach a force closure grasp. In contrast to the grasp planning techniques described above, grasp control does not require a complete object model. However, grasp control methods do not use approximate information that may be known about the object to accelerate the process of finding good grasp configurations. Instead prior information can only be used to decide upon a grasp controller starting configuration.

Similar to grasp control, this paper explores an approach to displacing contacts toward grasp configurations based on force feedback measurements. However, the focus here is on learning how to sequence displacements in order to reach a grasp, rather than using a fixed policy. A parameterized space of contact relative motions (CRMs) is proposed that are the atomic units of control. CRMs simultaneously displace grasp contacts and recover force feedback information relevant to the state of the grasping task. Since a single observation

of force feedback need not uniquely determine the contact configuration, finding an optimal policy for executing CRMs is a partially observable problem. Optimal solutions to this problem simultaneously recover relevant force information and displace contacts toward grasp configurations. In this paper, the partially observable problem is modeled as a k-order Markov Decision Process and Reinforcement Learning techniques are applied. As a result, the robot is able to learn to grasp arbitrary classes of objects (not just convex objects) through interactive trial-and-error. No prior knowledge or object models are needed for this approach to work - all relevant information is recovered through interactive trial-and-error with the object. It should be noted that this approach is similar to recent work by Hsiao *et al.* who propose modeling the grasping synthesis problem as a partially observable markov decision process (POMDP) [6]. In that work, the underlying state space of a particular grasping problem is identified in a pre-processing step and Heuristic Search Value Iteration is used to find optimal solutions.

The layout of the paper is as follows. Section II introduces CRMs and proposes a specification for CRMs that operates with two contacts in the plane. The relevance of CRMs to grasping is demonstrated in an experiment where Robonaut uses a single CRM to grasp a box. Section III poses grasp synthesis as an optimal control problem and solves it as a k-order Markov Decision Process. This approach is tested in simulation for a planar grasping problem where force feedback from multiple time steps is required in order to resolve ambiguity in the contact configuration. After learning this grasp solution in simulation, the strategy is tested on Robonaut. Although this paper restricts formal consideration to planar objects, the approach can be expected to extend to spatial objects and more than two contacts with little modification.

II. Contact Relative Motions

When a robot is in contact with an object, a reference frame exists located at the point of contact that simultaneously describes both the local manipulator surface and the local object surface. This fact is relevant to grasping because it is possible for the robot to directly sense some of the axes of this shared reference frame using force sensing. A contact surface that is instrumented with one or more force sensors can identify the point of contact as well as the orientation of a tangent plane (five out of six axes defining pose) [7]. The idea of a CRM is to move the manipulator contacts relative to this shared reference frame. The advantage of CRMs over motions defined with respect to other types of sensor information (for example, with respect to a visual reference frame) is that more precise motions are possible. In open environments, while visual processing is a good source of qualitative information, it is almost always more accurate to measure local surface information using force sensing.

In this paper, contact relative motions (CRMs) are the atomic units of control. The manipulator must be in contact with the object before the displacement executes and the CRM must always re-establish contact before terminating. After using force sensing to measure local surface characteristics at the contacts, the CRM displaces the contacts in the shared reference frames. Note that the shared frames only contain information about the object surface geometry near the points of contact. Since the CRM displaces the contact away from this local reference frame, the new contact surface normals (after executing the CRM) depend on the shape of the object. The act of selecting a CRM so as to achieve a grasp necessarily involves an implicit prediction about what the new object surface normals will be after CRM execution based on a series of prior force feedback observations. In this paper, the job of making these predictions and selecting a CRM that leads to a grasp configuration is posed as a learning problem in Section III.

A. A CRM Specification For Two Contacts in the Plane

The general description of CRMs given above allows them to be implemented in a number of different ways. This paper proposes a set of CRMs for two contacts in the plane that is parameterized by three variables, c, f, and r, as follows. Suppose the two contacts are labeled a and b. The first parameter specifies which contact is to be moved. A single CRM may only move one contact, described by the binary parameter, $c \in \{a, b\}$. The second parameter specifies the reference frame in which the contact goal position is defined. Again, since there are only two contacts, this parameter, $f \in \{a, b\}$, is binary. The third parameter, $r \in \{-r_{max}, r_{max}\}$ is a real-valued scalar that specifies where in a planar reference frame the desired contact position will be.

These three parameters, $\mathcal{C} = (c, f, r)$, identify a CRM that moves the contact c toward a linear manifold of positions specified as follows. Consider the reference frame of the contact specified by the binary parameter, f. Suppose that contact f is located at $\mathbf{x}_f$. Let $\hat{n}_f$ describe the object surface normal at contact f pointing into the object surface (it is assumed that the force sensor is able to sense this surface normal directly). The line that intersects $\mathbf{x}_f$ and runs in the direction of the surface normal, $\hat{n}_f$, is parametrically described by, $\mathbf{x} = \mathbf{x}_f + t\hat{n}_f$, where t is the free parameter. We shift this line perpendicular to its axis by some amount, r. If we restrict ourselves to the plane, then r is a scalar. In the plane, this new shifted line is described by $\mathbf{x} = \mathbf{x}_f + t\hat{n}_f + r\,(\hat{n}_f \times \hat{z})$, where r is the amount by which the line is shifted and $\hat{z}$ is a normal vector perpendicular to the plane. This line describes a manifold of desired contact positions that the moving contact, c, approaches.

The trajectory that c takes to reach a point on $\mathbf{x} = \mathbf{x}_f + t\hat{n}_f + r\,(\hat{n}_f \times \hat{z})$ is as follows. First, c backs away from the surface along its local surface normal, $\hat{n}_c$, by some preset distance, d. Then it moves to the point on $\mathbf{x} = \mathbf{x}_f + t\hat{n}_f + r\,(\hat{n}_f \times \hat{z})$ furthest from the object surface (if $c = f$, then this is the smallest value of t that can be reached given the manipulator's aperture; otherwise it is the largest). Finally, the contact moves toward the object along $\mathbf{x} = \mathbf{x}_f + t\hat{n}_f + r\,(\hat{n}_f \times \hat{z})$ until contact is re-established.

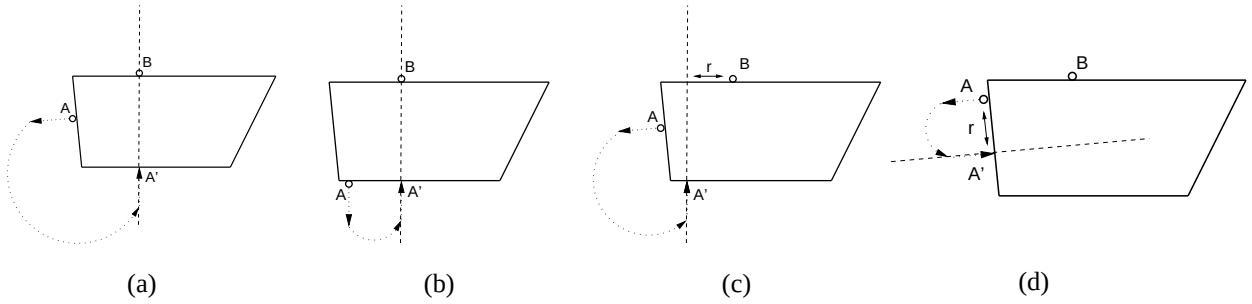


Fig. 1. Illustrations of CRMs. In these figures, contact a moves onto the linear manifold of goal positions illustrated by the vertical dotted lines. In each figure, contact A moves onto the goal manifold and terminates in contact with the object at A'. (a) and (b) illustrate the behavior of CRM, $\mathcal{C}_1 = (a, b, 0)$. (c) illustrates the behavior of $\mathcal{C}_2 = (a, b, r)$. (d) illustrates $\mathcal{C}_2 = (a, a, r)$.

Figures 1(a) and 1(b) illustrate $\mathcal{C}_1 = (a, b, 0)$. This CRM moves contact a with respect to the b reference frame. Because $r = 0$, the linear manifold of goal positions passes through the origin of the f contact frame. This manifold of goal positions is the same regardless of the initial position of contact a. Figures 1(a) and 1(b) show contact a moving onto this manifold from two different initial locations. Figure 1(c) illustrates the behavior of a different CRM, $\mathcal{C}_2 = (a, b, r)$. As in $\mathcal{C}_1 = (a, b, 0)$, contact a moves with respect to the b contact frame. However, now the manifold of goals positions is offset by a distance of r perpendicular to the contact normal. Finally, Figure 1(d) illustrates the behavior of $\mathcal{C}_3 = (a, a, r)$, a CRM that moves the a contact to a new position expressed in the reference frame of the same contact prior to moving. This CRM moves a to a new position offset by r units perpendicular to $\hat{n}_a$.

Figure 2(a) illustrates how this two-contact planar specification for CRMs translates into displacements of a humanoid robot hand. In this case, the four fingers are grouped together to act as a single virtual contact [8] and the thumb acts as the second contact. Figure 2(a) illustrates $\mathcal{C}_2 = (thumb, fingers, 0)$, where the thumb moves onto a linear manifold of positions that passes through the origin of the finger contact frame.

B. Experiment: Grasps Generated by a Single CRM

Figure 2(a) demonstrates that in some cases, an antipodal grasp (*i.e.* a two-contact opposition grasp) can be realized by executing a CRM that has a zero r parameter. In order for this to be feasible, the surface normal of one of the contacts must be co-linear with the surface normal of an opposing surface, *i.e.* is must be possible to achieve the antipodal grasp by moving just one contact. If this is the case, then a grasp can be achieved by executing CRMs $\mathcal{C}_1 = (a, b, 0)$ or $\mathcal{C}_1 = (b, a, 0)$ (for contacts labeled a and b). These CRMs will be referred to as "opposition CRMs" because they can lead to antipodal grasps.

Executing one of the above CRMs is an effective way to synthesize a grasp in cases where it is known that an antipodal grasp can be achieved by moving just one contact. Knowledge of this sort is not unusual. This is the case, for example, when it is known that two contacts are in contact with the sides of a cylinder or a regular prism with an even number of sides. For these objects, it is always possible to generate a grasp by removing either contact and placing it opposite the other contact.

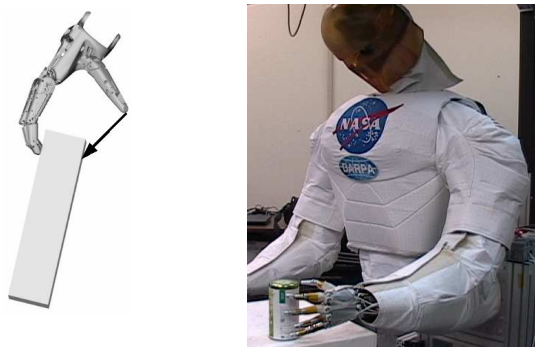


Fig. 2. (a) Illustration of the "opposition" CRM, $\mathcal{C}_2 = (thumb, fingers, 0)$. The Robonaut thumb moves to oppose the normal vector extending out from the fingers. (b) Robonaut, the NASA-JSC humanoid, grasping a can.

The practicality of using CRMs to synthesize grasps in these situations was tested using the NASA-JSC Robonaut, illustrated in Figure 2(b) [9]. Robonaut is a humanoid robot designed to assist astronauts perform manual maintenance and construction tasks in space and on planetary missions. It is equipped with twelve degree-of-freedom hands similar in shape, size, and dexterity to human hands. One of Robonaut's hands has recently been augmented with five fingertip load cells that measure six-axis loads applied at the tips using semiconductor strain gauges [10]. Since the load cells that were used in the current work were not compensated for temperature variation, they could not be used to measure absolute force magnitude. Nevertheless, good results were obtained using a first-order high-pass filter on the strain gauge output to eliminate low-frequency temperature variations. The high pass filter made it possible to measure short-time-constant changes in contact forces accurately. This enabled us to measure the change in force as the finger made contact with the surface,

(a) (b)

Fig. 3. The Robonaut hand before, (a), and after, (b), executing an "opposition" CRM. This illustrates the experimental scenario of Section II-B.

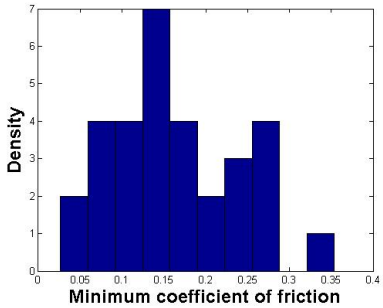


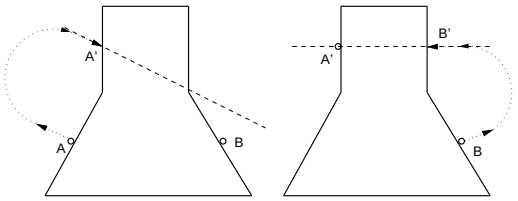
Fig. 4. Grasp quality histogram for 30 grasps. Grasp quality is measured in terms of the minimum coefficient of friction required for force closure given the contact configuration. Lower required friction values imply a better grasp.

and consequently to measure contact position and the local object surface normal.

In this experiment, an opposition CRM was used to grasp the same rectangular prism (a rectangular box) 30 times. On each grasp trial, the opposition CRM was executed starting in a configuration where the fingers were in contact with the long side of the box, similar to that shown in Figure 3(a). Figure 3(b) illustrates a typical final pose after CRM completion. Figure 4 shows a histogram of final grasp quality, measured by the minimum coefficient of Coulomb friction needed between the contacts and the object in order to achieve a force closure grasp. As the grasp moves further away from an antipodal grasp, larger coefficients of friction are needed in order to achieve force closure (*i.e.* to hold the object). Poor grasp configurations require a large coefficient of friction in order to grasp while good grasps require very little friction. The histogram shows that an opposition CRM is a practical approach to grasping in situations where it is known that an antipodal grasp is achievable by moving only one contact.

III. Learning sequences of CRMs for grasping

As noted above, executing an opposition CRM only results in a grasp when an antipodal grasp is achievable by moving only one contact. When this is not the case, as illustrated in Figure 5(a), multiple CRMs must be executed in sequence. Consider the positions of contacts a and b in Figure 5(a). In order to realize an antipodal grasp from this configuration, both contacts must be moved. This can be achieved by executing $\mathcal{C}_1 = (a, a, r)$ first (moving contact a to a' in Figure 5(a))



(a) (b)

Fig. 5. A grasp synthesis problem that requires at least two CRMs in order to reach a final grasp state. In (a), the two initial contact locations are labeled a and b. (a) illustrates how $\mathcal{C}_1 = (a, a, r)$ moves a to a'. (b) illustrates how subsequently executing $\mathcal{C}_2 = (b, a, 0)$ moves b to b'.

(a) (b)

Fig. 6. Pictures of the Robonaut hand in perceptually aliased configurations. Although the hand-object relationship in these two pictures is qualitatively different, the robot's observations of contact force feedback are approximately the same.

followed by $\mathcal{C}_2 = (b, a, 0)$ (illustrated in Figure 5(b)).

Even when it is possible to realize a grasp in one step, perceptual uncertainty may require the robot to execute multiple CRMs before reaching a grasp. When elements of the set of observations do not uniquely correspond to particular contact configurations relative to the object, *perceptual aliasing* can occur. Perceptual aliasing is the condition that a single observation can be generated by more than one underlying system state [11]. In these cases, it is necessary to recall prior observations in order to resolve the ambiguity. This is illustrated in Figures 6(a) and (b). In both figures, Robonaut's fingertip load cells make approximately the same measurements. Nevertheless, the orientation of the box in Figure 6(a) is 90 degrees different from its orientation in Figure 6(b). These two contact configurations are perceptually aliased.

Problems such as the above where observations are perceptually aliased have been studied as partially observable Markov Decision Processes (POMDPs). A POMDP encodes the underlying problem as a Markov Decision Process (MDP). However, the agent does not directly sense the underlying state of the world; instead, the it makes observations that improve its estimate of the state of the system. In the following, the problem of grasping with CRMs is first formulated as a POMDP. Then, a solution is proposed that treats the problem as a k-order Markov Decision Process (MDP).

A. Action Space and Observation Space

The action space corresponds to the space of CRMs that has already been defined. Let $C = \{a, b\}$ be the set of contacts (assuming the two-contact CRM specification) and let $R = [-r_{max}, r_{max}]$ be the location of the goal manifold in the specified contact reference frame. Then the action space is:

$$A = C \times C \times R. \tag{1}$$

As has already been stated, this paper focuses on the situation where the robot may only make observations derived from force feedback, not visual information. This information is summarized in terms of the squared magnitude of the frictionless force residual and the frictionless moment residual calculated over the two contacts. For two contacts, the squared magnitude of the frictionless force residual is:

$$\rho = \left(\sum_{i=1}^{2} \hat{n}_i\right)^T \left(\sum_{i=1}^{2} \hat{n}_i\right), \tag{2}$$

where $\hat{n}_i$ is the sensed surface normal at the i^{th} contact. This is the square of the magnitude of the net force that two contacts would apply to the object if they each applied unit forces normal to the object surface (*i.e.* under a unit force frictionless assumption). In order to calculate ρ, it is necessary to calculate the local contact surface normal at each of the contacts based entirely on contemporaneous sensor information. As noted earlier, one method is to use the approach of Bicci *et al.* to calculate object surface normal from load cell information [7].

The frictionless moment about the j^{th} contact is calculated in a similar way:

$$m^j = \sum_{i=1}^{2} \vec{r}_{ij} \times \hat{n}_i, \tag{3}$$

where $\hat{n}_i$ is the surface normal at the i^{th} contact and $\vec{r}_{ij}$ is a vector pointing from contact j to contact i. Since $\vec{r}_{ii} \times \hat{n}_i = 0$, m^j essentially measures moment about the j^{th} contact. The robot makes observations of the moment about each contact. For the two-contact system ($C = \{a, b\}$) considered here, the quantities of interest are m^a and m^b. Since we have restricted consideration to the plane, both $\hat{n}_i$ and $\vec{r}_{ij}$ exist in the plane and only the z component of moment contains any information. Hence, the relevant observations derived from the frictionless moment residual are the scalars m_z^a and m_z^b.

Finally, in addition to the frictionless force residuals and frictionless moments, the robot also observes error conditions encountered by the CRMs during execution. In general, $E = \{\epsilon_1, \ldots, \epsilon_k\}$ denotes the set of possible error conditions that a CRM might encounter. For the present, E is restricted to indicate the presence or absence of only two error conditions, ϵ_1 and ϵ_2. The first error condition, ϵ_1, indicates that the CRM could not reach its associated linear manifold of goal configurations because of aperture limitations. The second condition, ϵ_2, indicates that the manifold of goal configurations was not reached because of a collision with the object on a non-contact part of the manipulator (*i.e.* a collision at the palm). The absence of any error is indicated by $E = \epsilon_0$. Combining these various sources of information, the robot observes an element from

$$(\rho, m_z^a, m_z^b, \epsilon) \in \mathcal{O}, \tag{4}$$

where $\mathcal{O} = \mathcal{R} \times \mathcal{R} \times \mathcal{R} \times E$ when it executes a CRM.

B. Learning a Policy

The above observation of force residual and moment information does not instantaneously determine the complete state of the hand-object system. Instead, problems involving this sort of incomplete information are partially observable and are typically studied as Partially Observable Markov Decision Processes (POMDPs).

Two general approaches to solving POMDPs are generative-model approaches and history-based approaches [12]. In generative approaches, it is assumed that the agent is aware of the underlying structure of the MDP. The agent's past observations are summarized by a distribution that estimates the probability that the robot is in each possible underlying state. The agent then solves an optimization problem in the space of all possible distributions (the belief space). While this is an exact solution method for POMDPs, the resulting optimization problem is usually high-dimensional. In addition, it requires foreknowledge of the underlying system structure – information that is not readily available in the grasp domain. History-based approaches attempt to resolve the perceptual aliasing problem by storing a partial history of previous observations and actions that can resolve perceptual ambiguities. History-based approaches do not require an *a-priori* model of the underlying system. However, they may require the agent to store significant amounts of redundant information.

This paper takes a history-based approach to solving the partially observable grasp synthesis problem. In particular, the system is approximated as a k-order Markov Decision Process. An internal state of the robot is constructed from a history of the last k actions and observations,

$$s_t = (o_t, a_{t-1}, o_{t-1}, \ldots, o_{t-k+1}, a_{t-k}). \tag{5}$$

The internal state space is $S_k = \mathcal{O}_t \times A_{t-1} \times \ldots \times \mathcal{O}_{t-k+1} \times A_{t-k}$. The optimization problem is now solved as a fully observable MDP using the constructed internal state representation of Equation 5 and the action space described by Equation 1.

With a history-based system using a fixed time window, it is frequently necessary to trade off the amount of perceptual aliasing against the size of k. In order to keep the state space small (and therefore keep the problem computationally tractable), it is convenient to make k as small as possible. In this paper's grasping experiments, a value of $k = 2$ is used. The drawback of this approach is that the k-order system may "forget" important information because it does not store the full history of actions and observations. This would lead to perceptual aliasing. In general, the smaller the value of k, the more perceptual aliasing there will be.

Reinforcement Learning (RL) is used to find a policy that solves the POMDP directly in the space of S_k and A for two principle reasons. First, RL learns all relevant problem structure. There is no need to explicitly estimate or model underlying states, actions, or transition probabilities. Instead, based on the assumption of the k-order states, it is possible to estimate transition probabilities using straightforward maximum likelihood estimates. Second, RL has been shown to work robustly in non-stationary domains (domains where the underlying transition probabilities are not constant.) To the RL agent, the non-stationary transition probabilities caused by perceptual aliasing will appear to an RL agent as a stochastic or non-stationary transition function. On-policy versions of RL such as SARSA have been shown to work well for these problems [13], [14].

C. Experiment: Learning a Grasp Policy in a Perceptually Aliased Space

An experiment was performed where a policy for grasping a prismatic rectangular box using a two-contact manipulator was learned in a planar simulation. Then, one of the trajectories generated by the policy was demonstrated on the physical Robonaut system. Learning the grasp policy was non-trivial for two reasons. First, because of the limited aperture of the manipulator, it was only possible to form an opposition grasp on the rectangle by making contact on each of the long sides; the rectangle was too long for the manipulator to grasp it lengthwise. Second, in some contact configurations, the force feedback did not uniquely determine which contact was touching which side of the rectangle. As a result, the robot did not know which contact should be moved without considering a history of actions and observations.

1) Learning in Simulation: The robot had four CRM actions available to it. Two of the CRMs were "opposition" CRMs as described in Section II-B. Labeling the two contacts a and b, these CRMs were $\mathcal{C}_1 = (a, b, 0)$ (move the a contact opposite the b contact) and $\mathcal{C}_2 = (b, a, 0)$ (move the b contact opposite the a contact). The remaining two CRMs displaced a contact in its own reference frame by a predetermined amount. $\mathcal{C}_3 = (a, a, r)$ displaced the a contact to a fixed orthogonal distance r in the reference frame of the a contact. Similarly, $\mathcal{C}_4 = (b, b, r)$ displaced the b contact to a fixed orthogonal distance r in its own reference frame. Taken together, the space of available actions was:

$$A = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}.$$

The robot made observations of the form of the tuple in Equation 4, $(\rho, m_z^a, m_z^b, \epsilon) \in \mathcal{O}$. In this experiment, the space of observations was manually discretized by allowing each of the three variables, ρ, m_z^a, and m_z^b to take only two possible values each. As described in Section III-A, the error variable took on one of three values. Combining the above, each observation had $2 \times 2 \times 2 \times 3 = 24$ possible values. The complete second-order system state was given by the following tuple, describing the history of the last two actions and observations,

$$s_t = (o_t, a_{t-1}, o_{t-1}, a_{t-2}).$$

As a result, the second order state space had $24 \times 4 \times 24 \times 4 = 9216$ theoretically-possible states. However, because the latent structure of the problem made many theoretically conceivable sequences of action and observation impossible in practice, the actual state space was likely much smaller.

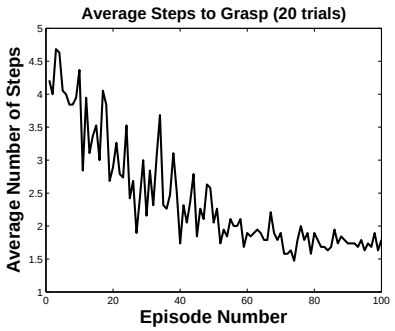


Fig. 7. Learning curve showing the number of steps needed to grasp the object as a function of earning episode averaged over 20 trials.

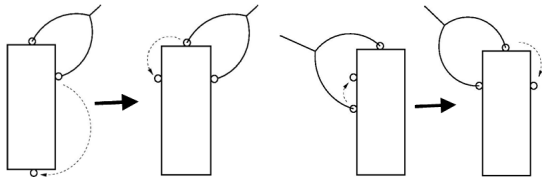


Fig. 8. Two grasp strategies learned in simulation. In (a), the robot is initially unsure about which contact should be opposed against the other. The robot arbitrarily chooses one contact to oppose. If that fails, then it opposes the other contact. In (b), the robot knows that the contact that is far away from the corner of the box must be on the long side. Therefore, it moves that contact closer to the corner and subsequently opposes the other contact.

The robot learned a policy for grasping the planar rectangle using RL. SARSA was used with a learning rate of 0.3, a discount factor of 0.9, and a reward of -1 in all states. All states were initialized with an optimistic initial value of zero. On each episode of learning, the robot started in a random starting contact configuration. An episode terminated when the robot reached an equilibrium grasp configuration or after ten actions.

Figure 7 shows the average number of steps needed to grasp the object averaged over 20 trials as a function of episode. As the number of episodes increased and the system acquired commensurately more experience, performance improved until a policy was learned that grasped the rectangle an average of 1.8 steps. Two of the grasp strategies learned are illustrated in Figures 8a and 8b. If the robot started in a configuration such that the two contacts were near a corner on orthogonal sides, then the robot used the strategy in Figure 8a. In this case, it was impossible to know based only on the current observation which contact was on the short side and which was on the

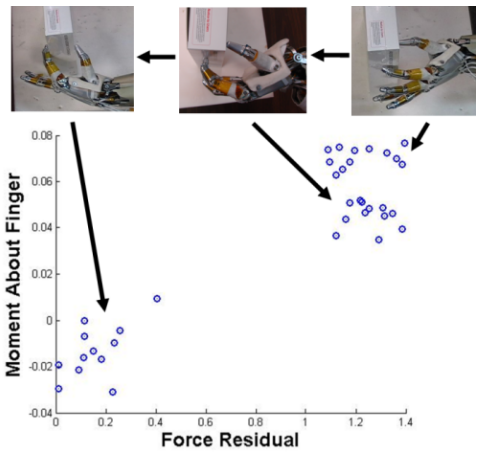


Fig. 9. Trajectory taken by the Robonaut hand as it executed the learned grasp policy for the rectangular box. The horizontal axis illustrates frictionless force residual. The vertical axis illustrates frictionless moment about the fingers. Each dot represents the configuration of the system at some point during policy execution for one of the eleven trials. The top right cluster corresponds to initial configuration. The cluster below that corresponds to the configuration after executing the first CRM. The lower left cluster corresponds to the configuration after the second CRM has executed. The clusters are associated with the pictures as indicated.

long side. The learned strategy chose a contact to oppose at random (the value of each action was approximately equal). If the CRM worked, then the episode terminated. Otherwise, the policy noted the failure and completed the grasp by opposing the other contact.

The situation was different for a contact configuration where the distance of one contact from the corner exceeded the length of the short side of the rectangle, as illustrated in Figure 8b. For the rectangle, this "distance from the corner" was encoded by the magnitude of the frictionless moment in the observation vector. This observation immediately disambiguated which contact was on the long side and which was on the short side. Nevertheless, it was not possible to move the contact on the short side into opposition immediately because it would cause non-contact surfaces of the manipulator (the "palm") to collide with the object. Instead, the learned policy first moved this contact closer to the corner using $\mathcal{C}_3$ or $\mathcal{C}_4$. Then the policy opposed the other contact using $\mathcal{C}_1$ or $\mathcal{C}_2$

Figure 7 shows that an optimal grasp strategy was learned within about 60 episodes. This learning time can be shortened with more sophisticated versions of RL including using eligibility traces or by performing dynamic programming iterations such as in DYNA-Q [15].

2) Testing on Robonaut: The learned strategy illustrated in Figure 8(b) was tested on Robonaut. The Robonaut hand was treated as a two-contact planar manipulator. The thumb acted as one of the contacts and the four fingers were grouped together to become a second virtual contact [16], [8]. In order to continue to use the planar approximation, the hand was constrained to move roughly parallel to the ground plane and interacted only with the prismatic rectangle (the box).

The trajectory of the Robonaut hand as it executed the two-CRM sequence is illustrated at the top of Figure 9. Robonaut starts in the configuration illustrated on the right where the distance of the thumb from the corner along the long side of the box exceeds the length of the short side. The picture in the center of the three at the top of Figure 9 illustrates the intermediate configuration where the thumb has moved closer to the corner, thereby enabling the fingers to oppose the thumb in its new location. Finally, the picture on the top left illustrates the final configuration of the manipulator.

The repeatability of this policy on the physical system was tested in an experiment where the above grasp strategy was executed eleven times. The plot in Figure 9 illustrates the trajectory of the manipulator as a function of frictionless force residual and moment. The horizontal axis is the measured frictionless force residual between the two contacts. The vertical axis is the measured frictionless moment of the thumb tip about the fingertips. Each point in the space corresponds to the state of the manipulator before the first action, before the second action, or after the second action on one of the eleven trials. Note that there are three clusters in the space. The cluster in the upper right ($\rho \sim 0.7$ and $m \sim 1.2$) illustrates the initial contact configurations where the distance of the thumb from the corner exceeds the length of the short side of the box. The cluster directly below that ($\rho \sim 0.5$ and $m \sim 1.2$) illustrates the intermediate configurations where the thumb has moved closer to the edge. Finally, the cluster in the lower left illustrates opposition configurations where both frictionless force residual and moment are close to zero. These results indicate that the learned policy transferred to the physical robot system in a consistent way.

IV. Discussion

The approach to grasping proposed in this paper occupies an interesting place in the pantheon of grasp synthesis methods. Most grasp synthesis approaches make very strict assumptions about the kind of foreknowledge available to the system. Grasp planning approaches typically assume that complete information about object geometry and relative hand pose is available. In contrast, grasp control methods assume that nothing is known about the object (aside from a general convexity assumption). This paper takes an approach related to grasp control while also using prior information to accelerate grasp synthesis.

In this paper, the manipulator reaches a grasp configuration by executing a sequence of CRMs according to a policy learned using Reinforcement Learning (RL). Although more experiments are needed to bear out these claims, it is expected that the policy that RL learns will depend on the "allowed" space of problem variation. For example, if the shape of the object is assumed to be constant in a particular situation, then RL may formulate the policy that grasps only the given object. However, if the shape of the object is allowed to vary within constraints, then RL can be expected to learn a policy that works throughout the space of object variation. For example, if the robot is grasping a cup of unknown radius and height, then the learned policy must reach grasp configurations for any

shaped cup. In order to achieve this, the learned policy may need to differentiate various classes of cups based on force feedback.

A key advantage of using RL to learn these grasp policies is that it is not necessary to encode the geometry of the object, what parameters are free to vary, or exactly what information is relevant to the problem. If the RL learning agent is presented with problem instances drawn uniformly from the space of variations (*i.e.* the robot is presented with randomly shaped cups presented in arbitrary orientations), then it will learn policies that minimize the average time/cost to grasp completion over all variations. So, if RL finds that a particular CRM sequence effectively grasps in the majority of problem instances and that the cost of grasp failure is small, then the agent will learn to use that sequence initially in all cases and only execute other strategies when the first strategy is found to fail. However, if the infrequent failures come with a high cost, then the agent will take steps in every case to determine ahead of time which situation it is in and then act appropriately.

V. Conclusion

This paper proposes a grasp synthesis strategy based on two ideas: contact relative motions (CRMs) and the notion of grasp synthesis as a partially observable optimal control problem. CRMs are units of control where a desired contact displacement is expressed in a shared robot-object reference frame. When the robot is in contact with the object, it is able to sense the local contact reference frame (shared between the object and the robot) directly. Since force feedback is usually more precise than visual feedback regarding the local object surface, CRMs are a mechanism for generating precise contact displacements over the local object surface. An experiment is presented that demonstrates that in certain situations, executing a single CRM is a practical way to synthesize a grasp that does not require precise foreknowledge of object parameters or manipulator pose.

This paper also explores the case where it is not possible to grasp an object by executing only one CRM and a sequence of CRMs is necessary instead. In these cases, selecting the correct sequence of CRMs may depend on the ability of the robot to sense the shape of the object. Sensing is a key issue for grasp synthesis techniques because it is frequently not possible to sense all aspects of object pose and geometry as accurately as desired. This paper focuses on issues related to incomplete sensing by allowing the robot to consider only force feedback and not visual information. Since the problem of selecting the correct sequence of CRMs is partially observable, this paper formulates it as a Partially Observable Markov Decision Process (a POMDP) where the robot makes a series of observations derived from force feedback alone. We solve the POMDP as a k-order Markov Decision Process using Reinforcement Learning. The advantage of this approach is that the robot needs no foreknowledge of the underlying structure of the grasp problem in order to find an optimal solution. Reinforcement Learning simply searches for policies that optimize the optimize the chance of grasp success in the space of variation to which the robot is exposed.

References

[1] L. Han and J. Trinkle, "Dextrous manipulation by rolling and finger gaiting," in *IEEE Int'l Conf. Robotics Automation*, vol. 1, May 1998, pp. 730 – 735.

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

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

[4] R. Platt, "Learning and generalizing control-based grasping and manipulation skills," Ph.D. dissertation, University of Massachusetts, September 2006.

[5] J. Coelho and R. Grupen, "A control basis for learning multifingered grasps," *Journal of Robotic Systems*, 1997.

[6] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Grasping pomdps," in *IEEE Int'l Conf. Robotics Automation*, April 2007.

[7] A. Bicchi, J. Salisbury, and D. Brock, "Contact sensing from force measurements," *International Journal of Robotics Research*, vol. 12, no. 3, 1993.

[8] C. MacKenzie and T. Iberall, *The Grasping Hand*. North-Holland, 1994.

[9] R. Ambrose, H. Aldridge, R. Askew, R. Burridge, W. Bluethman, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: Nasa's space humanoid," *IEEE Intelligent Systems Journal*, 2000.

[10] R. Platt, M. Chu, M. Diftler, T. Martin, and M. Valvo, "A miniature force sensor for prosthetic hands," in *Workshop on Robotic Systems for Rehabilitation, Exoskeleton, and Prosthetics; Robotics, Science, and Systems, Philadelphia, PA*, 2006.

[11] D. H. Ballard, *Natural Computation*. MIT Press, 1997.

[12] M. Littman, R. Sutton, and S. Singh, "Predictive representations of state," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 14, 2001, pp. 1555–1561.

[13] A. Barto, R. Sutton, and C. Anderson, "Neuron-like elements that can solve difficult learning control problems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 13, no. 5, pp. 834–846, 1983.

[14] M. Littman, "Memoryless policies: theoretical limitations and practical results," in *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, 1994.

[15] R. Sutton and A. Barto, *Reinforcement Learning, An Introduction*. MIT Press, 1998.

[16] R. Platt, A. H. Fagg, and R. Grupen, "Extending fingertip grasping to whole body grasping," in *IEEE Int'l Conference on Robotics and Automation*, 2003.