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 proposes an approach to grasp synthesis expressed in terms of units of control that simultaneously change the contact configuration and sense information about the object and the relative manipulator-object pose. These units of control, known as *contact relative motions* (CRMs), allow the grasp synthesis problem to be recast 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. An experiment is described that uses Robonaut, the NASA-JSC space humanoid, to show that CRMs are a viable means of synthesizing grasps. However, because of the limited amount of information that a single CRM can sense, the optimal control problem may be partially observable. This paper proposes expressing the problem as a k-order Markov Decision Process (MDP) and solving it using Reinforcement Learning. This approach is tested in a simulation of a two-contact manipulator that learns to grasp an object. Grasp strategies learned in simulation are tested on the physical Robonaut platform and found to lead to grasp configurations consistently.**

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]. 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, the controller is guaranteed to reach a force closure grasp. While grasp control works well for unmodeled objects in unknown configurations, the process of descending the gradient based on sensed object surface normals can be time consuming if the controller begins in a contact configuration far from a good grasp. One way to accelerate grasping is to use general or approximate information to place the manipulator contacts near a good grasp configuration before the grasp controller executes. If the robot starts near a goal configuration, the grasp controller will finish quickly.

This paper presents an alternative approach to accelerating a force feedback grasp synthesis process. As in the grasp control paradigm, it is assumed that precise geometric and pose information is not available. Likewise, the method presented here will rely primarily on local contact information derived from force feedback rather than visual information. However, instead of making only incremental displacements tangent to the object surface, this paper proposes a set of *contact relative motions* that are used to make larger, non-incremental, contact displacements. In addition, this paper focuses on the problem of selecting an *optimal* contact displacement based on prior knowledge about the object and a history of force feedback information. Grasp synthesis is posed as an optimal control problem where the robot must select contact relative motions expected to lead to a grasp configuration in the shortest number of steps. Since a single observation of force feedback need not uniquely determine the contact configuration, the problem of selecting a contact relative motion is partially observable. This paper proposes solving the partially observable problem by modeling the system as a k-order Markov Decision Process.

The layout of the paper is as follows. Section II introduces the notion of a contact relative motion (CRM). The utility of CRMs for grasp synthesis is demonstrated in an experiment where Robonaut uses a single CRM to grasp a box. Section III proposes grasping as an optimal control problem and solves it as a k-order Markov Decision Process. This approach to solving the decision problem is tested in simulation for a 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.

II. Contact Relative Motions

In this paper, contact relative motions (CRMs) are the atomic units of control. Grasps are synthesized by sequencing appropriate contact relative motions. CRMs are displacements of the manipulator from one contact configuration to another. The manipulator must be in contact with the object before the displacement executes and the CRM must always re-establish contact before terminating. In order to use CRMs, the manipulator contacts must be equipped with sensors that measure the object surface normal at contact (typically fingertip force sensors). The robot can measure the pose of a locally planar surface under each contact in five dimensions: three position dimensions and two orientation dimensions. The only pose dimension that is not measured is the orientation of the contact about the surface normal. The set of goal configurations is defined in terms of these local contact reference frames.

The main advantage of using CRMs to grasp is that the resulting displacements are precisely aligned with the local object surface. However, there is no explicit assumption about the local surface geometry in the new location. The CRM is simply a well-defined relative displacement that delivers the contact to a new location as a function of the geometry of the object surface. Executing different CRMs results in characteristically different types of contact displacements.

A. CRMs Based on a Reference Moment

The general definition of CRMs given above allows CRMs to be implemented in a number of different ways. This paper focuses on CRMs for two contacts that are specified in terms of a desired moment and an assumption about the local surface normal after the contact is displaced. Starting from an arbitrary contact configuration, the "moment" CRM moves one contact toward a set of desired positions while leaving the other contact stationary. The set of desired positions is calculated to be those such that the moved contact would apply the reference moment if it made contact with the object at the assumed contact normal and applied a unit force. This displacement can be understood geometrically as follows. The local surface normal assumption specifies a line in Cartesian space that is intersected with the position of the stationary contact. The line is translated by some amount such that it realizes the desired moment about the stationary contact. In order to reduce the size of the CRM parameter space, the robot is constrained to assume that the surface normal of the moving contact after the motion will be either: 1. the same as it was before the motion, or 2. opposite that of the stationary contact.

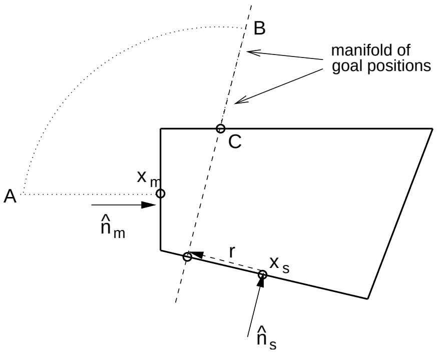


Fig. 1. The process of calculating a manifold of CRM goal positions. Initially, the two contacts are at $\vec{x}_s$ and $\vec{x}_m$. The dashed line indicates the set of goal positions. When the CRM executes, the moving contact, $\vec{x}_m$, lifts off the surface and follows the dotted line, finally re-establishing contact at point C.

Consider a moment CRM defined with respect to the stationary contact c_s and moving contact c_m. The Cartesian positions of the contacts before moving are $\vec{x}_s$ and $\vec{x}_m$; the unit normals are $\hat{n}_s$ and $\hat{n}_m$. It is assumed that after moving, $\hat{n}'_m$ will be equal to either: 1. the surface normal of the moving contact, $\hat{n}'_m = \hat{n}_m$, or 2. opposite the surface normal of the stationary contact: $\hat{n}'_m = -\hat{n}_s$. Finally, a desired moment $\vec{m}_d$ is specified. Note that the moment applied by a pure force is always perpendicular to that force. Therefore, $\vec{m}_d$ is required to be perpendicular to $\hat{n}'_m$. The set of positions, $\vec{r}$, for the moving contact that realizes the desired moment satisfies the following: $\vec{m}_d = \vec{r} \times \hat{n}$. Since $\vec{m}_d$ is orthogonal to $\hat{n}'_m$, one such position is $\vec{r} = \hat{n} \times \vec{m}_d$. The manifold of goal positions for c_m that will realize $\vec{m}_d$ is the line defined by the intercept $\vec{x}_s + \hat{n} \times \vec{m}_d$ and the unit vector $\hat{n}'_m$.

The set of goal positions for the moving contact does not give the manipulator an actual path for the contact to follow during displacement. In general, a CRM requires the manipulator to lift a contact off of the surface, move it to the line of goal positions without colliding the the object, and re-establish contact. One way of achieving this is as follows. First, the manipulator moves c_m to its widest aperture along $\hat{n}_m$. Then, while maintaining the widest aperture, the manipulator sweeps out an arc until it arrives at a point on $(\vec{x}_s + \hat{n} \times \vec{m}_d, \hat{n}'_m)$. Then, c_m follows the assumed inward surface normal, $\hat{n}'_m$ until contact is re-established.

The process of calculating the manifold of goal positions is illustrated in Figure 1. The positions of two contacts on a

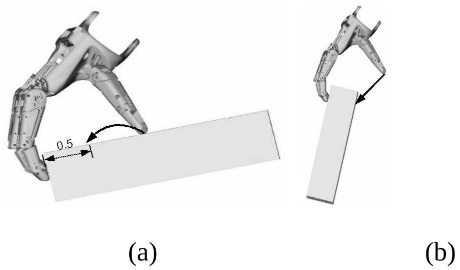


Fig. 2. The “corner” CRM, (a), and the “opposition” CRM, (b). (a) illustrates the motion calculated by a CRM parameterized by a moment of 0.5 and an assumption that the thumb object surface normal will be the same before and after displacement. (b) illustrates the motion calculated by a CRM parameterized by a zero moment reference and a surface assumption parallel to the surface normal of the stationary contact.

Fig. 3. Robonaut, the NASA-JSC humanoid, grasping a can.

planar quadrilateral before moving are $\vec{x}_s$ and $\vec{x}_m$ with unit normals $\hat{n}_s$ and $\hat{n}_m$. It is assumed that the moving contact will be placed on a face opposite that of the stationary contact: $\hat{n}'_m = -\hat{n}_s$. A desired moment is specified directed into the page. In Figure 1, the manifold is the dashed line parallel to $\hat{n}'_m = -\hat{n}_s$ that passes through the intercept, $\vec{r} + \vec{x}_s$, where $\vec{r} = \hat{n} \times \vec{m}_d$. First, c_m moves along the local surface normal away from the surface until the maximum aperture of the manipulator is reached (point A in Figure 1). Then, the moving contact sweeps an arc toward the manifold of goal positions while maintaining the largest aperture (point B). Finally, the moving contact moves forward along the assumed contact normal until it makes contact (point C).

Figure 2(a) illustrates a planar example of a moment CRM parameterized by a moment reference of 0.5. The thumb is the moving contact and the fingers remain stationary. It is assumed that the thumb surface normal is the same before and after motion (the thumb is assumed to be moving along a planar surface). In order to apply the positive 0.5 moment about the fingers, the thumb must move such that the distance between the thumb and the fingers perpendicular to the surface that the thumb moves on is 0.5. In the case of the rectangular object illustrated in Figure 2(a), this CRM moves the thumb 0.5 units from the corner.

Figure 2(b) shows another example. As above, the thumb is the moving contact and the fingers are stationary. In this example, the reference moment is zero and it is assumed that the thumb surface normal after the displacement will be opposite the surface normal at the fingers. These parameters define a line of goal positions parallel to the finger surface normal and passing through the fingertip contacts.

CRMs are useful for grasp synthesis because they are able to resolve the uncertainty in object pose and geometry precisely. Typically, robot motions are defined either in the robot base frame or with respect to a visually localized object. Unfortunately, it is difficult to determine object pose and geometry precisely using vision. In addition, even when precise information is available, kinematic calibration inaccuracies in the manipulator can prevent the robot from reaching a desired grasp configuration. Contact relative motions use local sensors (force sensors) at the contacts to precisely control the relative manipulator-object motion. This gives the robot the capability to achieve precise grasps. Depending upon what may be known *a priori* about the object to be grasped and the manipulator starting configuration, it may be possible to grasp an object by executing a single CRM.

B. Experiment: Grasps Generated by a Single CRM

The example of Figure 2(b) shows that in some cases, a grasp can be synthesized by executing a single CRM. The figure illustrates an “opposition” CRM where the moving contact is placed in opposition with the stationary one. This corresponds with the moment CRM where a zero desired moment is specified and the assumption is made that the moving contact will re-establish contact on a face opposite (that is, a face with an opposite surface normal) that of the stationary contact.

Executing this CRM can effectively be used to synthesize a grasp when it is known that an anti-podal solution exists opposite one of the contacts. This is the case, for example, when it is known that two contacts are currently 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 placing a contact opposite (in an anti-podal configuration) another contact on the object. The CRM can also be used for non-regular objects when it has been established that an opposing face exists for one of the contacts.

The efficacy of using this CRM to synthesize an anti-podal grasp was tested using Robonaut [5]. Robonaut is a humanoid robot designed to assist astronauts perform manual maintenance and construction tasks in space and on planetary missions. See Figure 3. 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.

In this experiment, the reliability of using the “opposition” CRM to synthesize grasps was tested by attempting to grasp a box 30 times. On each grasp trial, the manipulator was placed in an initial configuration similar to that shown in Figure 4(a), and the CRM was executed until a configuration similar to that shown in Figure 4(b) was reached. A distribution over quality

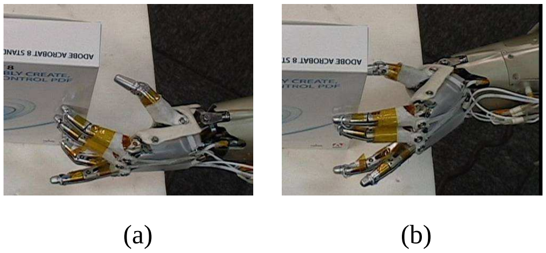

(a) (b)

Fig. 4. The Robonaut hand before, (a), and after, (b), executing an "opposition" CRM. This illustrates the experimental scenario of Section II-B.

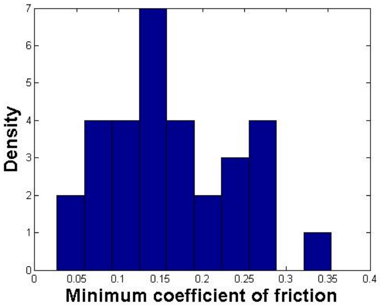


Fig. 5. Grasp quality histogram for 30 grasps. Grasp quality is measured in terms of the minimum coefficient of friction required for force closure given the contact configuration. Lower required friction values imply a better grasp.

for the resulting anti-podal grasps is illustrated in Figure 5. Grasp quality is measured by the minimum coefficient of Coulomb friction that is required between the contacts and the object in order to achieve a force closure grasp. Poor grasp configurations require a large coefficient of friction in order to grasp while good grasps require very little friction. This result demonstrates that it is possible to use a CRM to synthesize grasps repeatably.

III. Learning sequences of CRMs for grasping

It may not be possible to synthesize a grasp by executing a single CRM. For example, in the case of two contacts, there may not be any position that affords an anti-podal grasp opposite either of the contacts. Furthermore, even if such a grasp does exist, the robot may not know which contact should be moved and which should remain stationary. In these situations, a sequence of CRMs may be needed in order to realize a grasp.

The problem of finding the "right" sequence of CRMs to execute can be described as follows. Given a set of actions that the robot is allowed to execute (a set of CRMs) and a set of observations that the robot may make after executing each action, determine a strategy for executing CRMs that leads to a goal configuration in the shortest number of steps. It should be noted that 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 [6]. For these problems, it is necessary to use prior observations to resolve the ambiguity.

This approach to grasp synthesis is related to work by Coelho and Grupen who propose sequencing moment residual controllers in order to synthesize grasps [7]. However, in contrast to that work, the CRMs used in the current method allow the contacts to make non-incremental jumps from one configuration to another. In addition, the current approach addresses the partial observability problem explicitly. This section first describes a set of actions and observations that are used to define the partially observable grasp synthesis decision problem. Then, a strategy for solving it based on k-order Markov Process is described. Finally, the approach is tested in simulation and on Robonaut.

A. Action Set and Observation Set

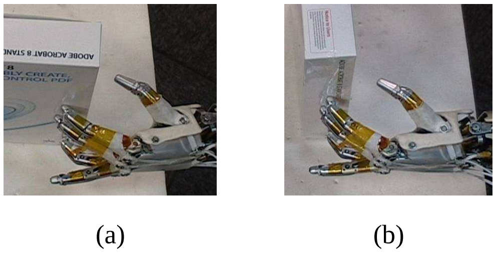
Th action set is described as follows. Recall that a moment CRM is parameterized by a reference moment, $\mathbf{m}_{ref}$, and an assumption regarding the surface normal used to calculate a line of goal positions. Note that the robot must also specify which contact will move and which will remain stationary. This results in the following factored action representation:

$$A = \alpha_{fin} \times \alpha_{opp} \times \alpha_{\mathbf{m}}, \qquad (1)$$

where the three action variables are defined below. For the case of a two-contact manipulator (the only case considered in this paper), $\alpha_{fin} \in (1, 2)$ encodes the choice of which finger to move. Recall that in order to constrain the CRM parameter space, Section II-A restricted the choice of the surface normal assumption to be the surface normal of one of the contacts before execution. For a two contact manipulator, this parameter takes values, $\alpha_{opp} \in (1, 2)$, denoting an assumption parallel to one of the two contact surface normals. Finally, $\alpha_{\mathbf{m}} \in R^2$ is a vector encoding the desired moment. This is a two-dimensional parameterization of a three-dimensional vector that is constrained to be orthogonal to the assumed surface normal. Because of the large parameter space associated with $\alpha_{\mathbf{m}}$, it may be convenient to prune this space or otherwise modify it in order to simplify the problem.

In order to focus the optimization problem, this paper only considers observations of CRM error states and observations derived from the local surface normals at the contacts. Typically, local surface normals can be derived from fingertip force feedback. Note that observations derived from visual processing are ignored.

An important question is how observations of raw contact position and surface normal data will be represented to the system. In general, it is desirable to represent sensory feedback in a space as similar to the task space as possible. An appropriate representation can reduce the complexity of the policy that the system must learn. For the grasp synthesis problem, this suggests that contact position and local surface normal data should be represented to the robot by variables related to grasp quality. The current work takes inspiration from grasp controller methods for this purpose [8], [9], [4]. Contact position and local surface normal information are used



(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.

to calculate frictionless force residual and frictionless moments over the set of contacts. Frictionless force residual and moment residual are closely related to grasp quality because together, they describe the contact configuration relative to frictionless equilibrium, a sufficient condition for grasp force closure.

For two contacts, the frictionless force residual is defined,

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

where $\vec{n}_i$ 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 frictionless assumption). The magnitudes of frictionless moments can also be calculated,

$$m^j = \left(\sum_{i=1}^{2} \vec{r}_{ij} \times \vec{n}_i \right)^T \left(\sum_{i=1}^{2} \vec{r}_{ij} \times \vec{n}_i \right), \tag{3}$$

where $\vec{r}_{ij}$ is a vector pointing from the Cartesian position of contact j to the position of contact i. For a two-contact manipulator, it is possible to measure the moment about each contact: m^1 and m^2. In addition to frictionless force residuals and frictionless moments, the robot also observes error conditions encountered by the CRMs during execution. Let $E \in (1, k)$ denote the set of possible error conditions that a CRM might encounter. In summary, each time a robot with a two-contact manipulator executes a CRM, it observes an element from

$$(\rho_t^f, m_t^1, m_t^2, e_t) \in \mathcal{O}, \tag{4}$$

where $\mathcal{O}$ is the set of all possible observations, $\mathcal{O} = R \times R \times R \times E$.

B. Learning a Policy

The problem of selecting an action that leads to a goal configuration based on the observations derived from contact positions, surface normals, and CRM failure modes is partially observable. In the context of grasp synthesis, this means that the configuration of the manipulator relative to the object cannot necessarily be determined on the basis of a single observation of force residual and moment residual about the contacts. 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.

Partially observable problems such as this are typically studied in the context of a Partially Observable Markov Decision Process (POMDP). 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. Two general approaches to solving POMDPs are generative-model approaches and history-based approaches [10]. In generative approaches, it is assumed that the agent is aware of the underlying structure of the MDP. Based on the agent's observations, the agent calculates a probability distribution over all possible underlying states. The agent then solves a high-dimensional optimization problem in the space of all possible distributions (the belief space). History-based approaches are simpler to implement and do not require a model of the underlying system. These approaches attempt to resolve the perceptual aliasing problem by storing a partial history of previous observations and actions that can resolve perceptual ambiguities. The drawback of this approach is the potentially large state space and the complexities of determining which parts of the entire history to remember.

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 $S = \mathcal{O}_1 \times A_2 \times \ldots \times \mathcal{O}_k \times A_{k+1}$, where A_i and $\mathcal{O}_{i-1}$ are the action and resulting observation i time steps ago. The optimization problem is now solved using the constructed internal state representation of Equation 5 as a fully observable MDP. Reinforcement Learning (RL) is a convenient approach to solving MDPs in unmodeled domains [11].

In order to keep the state space small (and therefore keep the problem tractable), it is convenient to make k as small as possible. In this paper's grasping experiments, a value of $k = 2$ is used. However, because the full history of actions and observations is not stored, the k-order system may "forget" important information and therefore potentially alias certain system configurations with respect to the internal state representation. In general, the smaller the value of k, the more perceptual aliasing there will be. 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. This residual perceptual aliasing will appear to an RL agent as a stochastic or non-stationary transition function. It has been shown that versions of RL that work well in stochastic domains (for example, SARSA [11]) can also work for non-stationary problems in some situations [12], [13]. Therefore, this paper uses SARSA to solve the k-order MDP.

Representing the problem in this way has several advantages. First, this approach provides a mechanism for the robot to reason about perceptions acquired *during* grasp synthesis itself. Typically, grasp planning occurs prior to grasp synthesis. Second, the grasp policies learned in this paper monitor progress toward the grasp and are capable of reacting to unexpected events. Finally, it is expected that the learned grasp policy may be as general or as specific as need be. If the robot is trained only on cylinder grasping problems, then the learned policy should reflect a cylinder assumption. However, if the system trains on a larger set of objects, the learned policy will implicitly disambiguate classes of objects before forming a grasp.

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

In order to validate this approach, learning experiments were performed in planar simulation. The resulting policies were tested on the physical Robonaut.

1) Learning in Simulation: In this experiment, a two-contact manipulator with limited aperture learned a policy for grasping a rectangle based on observations derived from force feedback. Learning 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 is 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.

Four CRMs were available to the robot to execute as actions. These were two "oppose" actions and two "move-to-corner" actions. The "oppose" actions were CRMs parameterized by a zero moment reference and an assumption that the moving contact would make contact with a surface parallel to the surface of the stationary contact (see Figure 2(b)). One "oppose" action was defined for each of the two contacts: $a_{opp}(1)$ or $a_{opp}(2)$. The "move-to-corner" actions were CRMs parameterized by a fixed positive moment reference and an assumption that the moving contact would re-establish contact on a surface parallel to that of the moving contact before displacement (see Figure 2a). The two "move-to-corner" actions allowed the robot to use this CRM to move either contact: $a_{cor}(1)$ or $a_{cor}(2)$. Taken together, the robot had access to the following action space:

$$A = \{a_{opp}(1), a_{opp}(2), a_{cor}(1), a_{cor}(2)\}.$$

The grasp synthesis problem was approximated by a second order MDP. Internal system state was given by the following tuple, describing the history of the last two actions and observations:

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

where each action a_t was drawn from A defined above. Each observation was a vector:

$$o_t = (\rho^f_t, m^1_t, m^2_t, e_t),$$

where ρ^f, m^1, and m^2 were manually discretized into two regions each. The error condition, e_t, had three possible values: $e_t \in (1, 2, 3)$. These three conditions corresponded to no error or one of two possible CRM failure modes. In the first failure mode, the aperture of the manipulator was not wide enough to allow the moving contact to reach the goal position. In the second failure mode, the palm of the manipulator collided with the object during the motion.

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 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

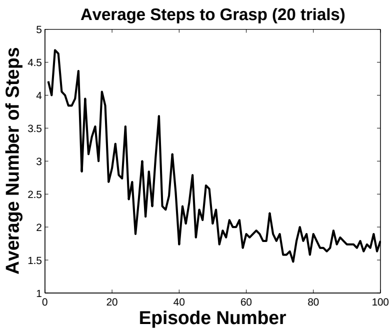


Fig. 7. Learning curve showing the number of steps needed to grasp the object as a function of earning episode averaged over 20 trials.

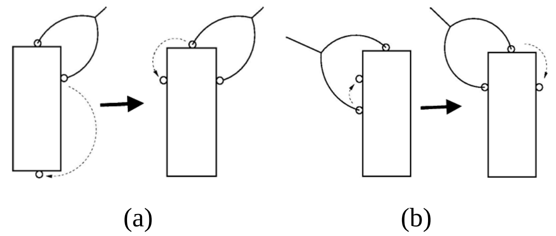


Fig. 8. Two grasp strategies learned in simulation. In (a), the robot is initially unsure about which contact should be opposed against the other. The robot arbitrarily chooses one contact to oppose. If that fails, then it opposes the other contact. In (b), the robot knows that the contact that is far 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.

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 starts in a configuration such that the two contacts are near a corner on orthogonal sides, then the robot uses the strategy in Figure 8a. In this case, it is impossible to know based only on the current observation which contact is on the short side and which is on the long side. The learned strategy chooses a contact to oppose at random (the value of each action was approximately equal). If the CRM works, then the grasp is complete. Otherwise, the policy notes the failure and completes the grasp by opposing the other contact.

The situation is different for a contact configuration where the distance of one contact from the corner exceeds the length of the short side of the rectangle, as illustrated in Figure 8b. For the rectangle, this "distance from the corner" is encoded in the magnitude of the frictionless moment in the observation vector. It immediately disambiguates which contact is on the long side and which is on the short side. However, it is not possible to move the contact on the short side into opposition immediately because this may cause the palm of the manipulator to collide with the object. Instead, the learned policy first moves this contact closer to the corner using one of the a_{cor} actions. Then the policy opposes the other contact.

Figure 7 shows that an optimal grasp strategy was learned within about 60 episodes. It is likely that this learning time could be shortened by using eligibility traces or by performing dynamic programming iterations such as in DYNA-Q [11].

2) Testing on Robonaut: The learned strategy illustrated in Figure 8(b) was tested on Robonaut. For this experiment, the Robonaut hand was approximated as a two contact manipulator where the four fingertips were grouped together and treated as a single *virtual finger* and the thumb tip acted as a second contact [14], [15]. The position of the virtual fingertip contact was the average of the constituent contacts; the object surface normal of the virtual fingertip was the average of the component surface normals.

It was possible to apply the policy learned for the planar rectangle in simulation to a real-world grasping experiment because of the CRMs used in the experiment moved the contacts roughly parallel to the ground plane (the plane orthogonal to the sides of the box). This results from the behavior of CRMs. Recall that a moment CRM moves one contact to a set of goal positions on a line parallel to the surface normal at one of the contacts. When parameterized by a reference moment perpendicular to the ground plane (as it was in this experiment), a CRM always moves a contact to a new position at the same elevation as one of the two contacts before displacement.

The Robonaut hand pictures at the top of Figure 9 illustrate the trajectory of the Robonaut hand as it executes the sequence of two CRMs. 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.

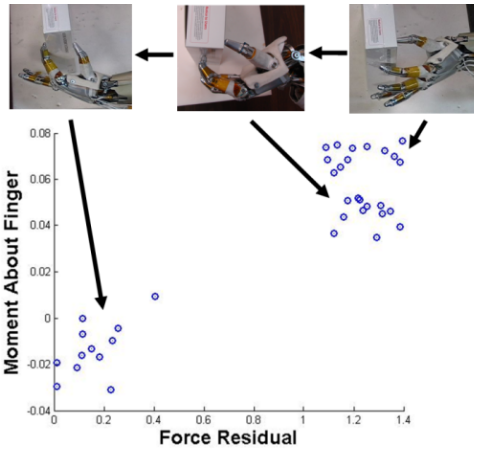


Fig. 9. Trajectory taken by the Robonaut hand as it executed the learned grasp policy for the rectangular box. The horizontal axis illustrates frictionless force residual. The vertical axis illustrates frictionless moment about the fingers. Each dot represents the configuration of the system at some 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.

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 ($\rho^f \sim 0.7$ 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 ($\rho^f \sim 0.5$ 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 transfered to the physical robot system in a consistent way.

IV. Conclusion

This paper proposes a grasp synthesis strategy based on two ideas: CRMs and the notion of grasp synthesis as an optimal control problem. CRMs are units of control where the contacts move from one contact configuration to another.

This displacement must be defined relative to local information sensed at the contacts. In this paper's experiments, this local feedback is derived from force sensors located on the manipulator contacts. An experiment is presented where the "opposition" CRM was used to generate an anti-podal grasp of a box. The results show that this CRM can repeatably generate high quality grasps.

Depending upon the object and starting configuration, it may not be possible to synthesize a grasp in a single step. In these situations, a sequence of CRMs must be executed. Finding a policy for executing CRMs can be formulated as a partially observable Markov Decision Process (POMDP), where local contact feedback and CRM failure modes constitute observations that the robot makes regarding the object and the current grasp configuration. This POMDP is solved by approximating the problem as a k-order Markov Decision Process (MDP).

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] 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.

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

[7] J. Coelho, J. Piater, and R. Grupen, "Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot," *Robotics and Autonomous Systems Journal, special issue on Humanoid Robots*, vol. 37, no. 2-3, November 2001.

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

[9] R. Platt, A. H. Fagg, and R. A. Grupen, "Nullspace composition of control laws for grasping," in *IEEE Int'l Conf. on Intelligent Robots and Systems*, 2002.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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