

## CHAPTER 4

# GRASP CONTROL

Creating force domain behavior requires control processes that optimize manipulator contact configuration based on the forces that can be applied. In the case of grasping, contact configuration must be optimized for grasp quality measures. If robust controllers can be defined that converge to good grasp configurations, even over limited domains of attraction, then these controllers can be sequenced or combined to create robust behavior over larger domains. This chapter focuses on defining controllers that control grasp quality. Starting with Coelho’s force and moment residual control primitives, a composite grasp controller is defined that executes both of these control primitives concurrently [60]. Next, a hybrid position and force controller is defined that acts in concert with the grasp controller to slide the contacts to good grasp configurations [61]. Finally, the set of potential grasp controllers is expanded by allowing controllers to be parameterized by *virtual contacts* that correspond to contact groups [57]. Experimental results are presented that demonstrate these controllers to be a practical and effective way of synthesizing grasps in poorly modeled domains.

### 4.1 Background

The control-based approach to solving force-domain problems taken in this thesis rests heavily upon Coelho’s force and moment residual controllers. These controllers minimize grasp error functions by displacing contacts on the surface of an object in response to local tactile feedback at the contacts. This section describes different approaches to tactile sensing and a method for displacing contacts on the surface of an object that acquires tactile feedback, called probing. Next, Coelho’s force residual and moment residual controllers that displace contacts into quality grasp configurations based on local tactile feedback are described.

#### 4.1.1 Sensing for Grasp Control

Grasp controllers assume that it is possible to sense local surface geometry in the neighborhood of each contact. This may be accomplished in a number of ways. For example, it may be possible to use computer vision to extract the relationship between object surface and grasp contact [1]. Notice that this vision problem is significantly easier than the general problem of reconstructing full object geometry. Unfortunately, it can be difficult to use vision for the purpose of “tactile sensing” without placing

fiducial marks on the fingertips [1]. Even worse, vision cannot be used when the point of contact is occluded by the hand or the object itself.

A more common approach to tactile sensing is to actually place sensors on the finger itself. Approaches to tactile sensing have been divided into two categories: “extrinsic contact sensing” (ECS) and “intrinsic contact sensing” (ICS) [75]. In ECS, the contact region is covered with some type of tactile “skin” that senses applied forces. Typical examples include force sensing resistors (FSRs) and quantum tunneling composite (QTC). When pressure is applied, the effective resistance of these materials changes and can be used to sense absolute pressure or a change in pressure. These materials are typically sandwiched in a contact array that localizes the pressure to multiple positions on a grid. A single contact location can be calculated by averaging the reported contact locations [75]. Recent examples of this type of sensing include the Robonaut “sensor glove” and the fingertips of the Shadow hand [43].

In contrast to ECS, ICS attaches the sensor to internal structural elements on the finger. The primary example of this approach uses load cells. It is often mechanically feasible to embed a small load cell in the load path between the fingertip and the rest of the finger. If there are no secondary load paths, this load cell can sense small forces and torques applied to the fingertip. Notice that force and torque information does not directly correspond to contact location. However, when it can be assumed that the sensed load was produced by a single point of contact (or contact region), Bicchi, Salisbury, and Brock propose an algorithm that calculates the point of contact (or average contact point) [9]. This approach only produces a unique contact point when it is known ahead of time that contact can only occur on a convex region of the finger. This approach to tactile sensing has a long history, starting with Salisbury [45, 75, 12]. This type of tactile sensing is used by Dexter, the UMass bimanual humanoid that is used in most of the experiments of this thesis.

One final approach to ICS worth mentioning uses IR-LEDs. A manipulator equipped with built-in IR-LEDs can be used to sense the surface normal of an object when in close proximity. This approach to sensing-for-grasping has been considered by Teichmann and Mishra [81].

#### 4.1.2 Displacing Grasp Contacts by Probing

In the “probing” approach to contact displacement, the manipulator iteratively makes light contact with the object, lifts, and displaces the contacts. The contacts are moved toward the object surface until they lightly touch. Fingertip load cells measure the load applied to the object. After touching, the manipulator contacts must be lifted and displaced a short distance along the object surface in the direction of the negative gradient of the grasp error function.

One way to realize this displacement is to execute a three step sequence: first, lift all contacts off the object; second, execute the position controller introduced in Section 3.1.1 to displace the contacts to the goal configuration; third, approach the object with all contacts along an estimated surface normal. Probably a more principled approach to contact displacement is to use a collision-free motion controller that treats the object surface like an obstacle [15]. In this approach, the motion

controller lifts the contacts off of the surface in order to avoid colliding with the object. Contact with the object is re-acquired only when approaching the goal.

### 4.1.3 Wrench Residual

Roughly speaking, the objective of grasp control is to displace the contacts toward force closure configurations (for more on *force closure*, see Section 2.2.1). This is accomplished by minimizing the net force and moment that would be applied by frictionless contacts applying unit forces when the mass of the object is ignored. Intuitively, contact configurations in frictionless equilibrium are those where the fingers can squeeze the object arbitrarily tightly without generating net forces or torques. These configurations are good grasps because the robot can increase frictional forces arbitrarily by squeezing the object. The grasp controller finds these grasps by minimizing the squared frictionless wrench residual,  $\epsilon_w = \rho^T \rho$ . Recall from Section 2.2.1 that *wrench* is a 6-dimensional generalized force consisting of force and moment. The frictionless wrench residual,  $\rho = \sum_{i=1}^k w_i$ , is the net wrench applied by  $k$  frictionless point contacts with unit magnitude.

If the grasp controller succeeds in reaching a frictionless zero net wrench configuration, the resulting grasp is guaranteed to be force closure. Several researchers, Ponce [77] for example, have shown that net zero wrench with at least three frictionless contacts (frictionless equilibrium) is a sufficient condition for force closure when the frictionless contacts are replaced with point contacts with friction. The same argument can be made for two-contact frictionless equilibrium when frictionless contacts are replaced with soft contacts. Since real-world contacts usually have friction, the frictionless equilibrium configurations that grasp controllers find are force closure configurations in the real world.

One approach to grasp control is to differentiate  $\epsilon_w = \rho^T \rho$  with respect to contact configuration and to displace grasp contacts accordingly. Unfortunately,  $\epsilon_w$  can have a number of non-zero local minima even for simple objects. This is because wrench has both a force component and a moment component. For convex objects, the force component of wrench error has only one minimal connected component in the contact configuration space. However, the component of wrench error contributed by the moment residual can have multiple minima. When both components are combined naively into a single error measure, the moment component introduces local minima. This makes the gradient of wrench residual error by itself unsuitable for a grasp controller potential function.

An alternative is to decompose  $\epsilon_w$  into separate force and moment residual error functions:

$$\epsilon_{fr} = \left( \sum_{i=0}^k \mathbf{f}_i \right)^T \left( \sum_{i=0}^k \mathbf{f}_i \right), \quad (4.1)$$

and

$$\epsilon_{mr} = \left( \sum_{i=0}^k \mathbf{r}_i \times \mathbf{f}_i \right)^T \left( \sum_{i=0}^k \mathbf{r}_i \times \mathbf{f}_i \right). \quad (4.2)$$

In these equations,  $\mathbf{r}_i$  is the location of the  $i^{th}$  contact point in an arbitrary object coordinate frame and  $\mathbf{f}_i$  is a unit vector (a frictionless force) normal to the object surface applied by the  $i^{th}$  contact. By defining the error function in terms of unit vectors of force, the grasp controller will prefer contact configurations where all contacts apply equal forces. Coelho’s approach avoids minima in the wrench residual error function by first descending the gradient of the force residual error  $\frac{\partial \epsilon_{fr}}{\partial \mathbf{f}}$  and subsequently descending the gradient of the moment error  $\frac{\partial \epsilon_{mr}}{\partial \mathbf{m}}$  [12]. The controller that descends  $\frac{\partial \epsilon_{fr}}{\partial \mathbf{f}}$  “funnels” the contact configuration away from local minima in the wrench residual error. As long as the moment residual controller does not ascend the force gradient, this approach will reach zero-wrench configurations.

#### 4.1.4 Calculating a Grasp Error Gradient

In order to create controllers that descend the force and moment residual error functions, it is necessary to express the gradients,  $\frac{\partial \epsilon_{fr}}{\partial \mathbf{f}}$  and  $\frac{\partial \epsilon_{mr}}{\partial \mathbf{m}}$ , in terms of surface coordinates. This enables the controllers to displace contacts so as to minimize these functions. In principle, expressing these gradients in surface coordinates requires information regarding the local surface geometry of the object in the neighborhood of each contact. Since this information may not be available, Coelho makes a constant instantaneous radius of curvature assumption in order to calculate the gradient of the force error function and an infinite plane assumption in order to estimate the gradient of the moment error function [13]. In particular, as a contact moves over the surface of the object, net force is assumed to change as if that surface were a unit sphere tangent to the object surface at the contact point. As a function of object surface coordinates,  $(\theta, \phi)$ , the frictionless force applied by the  $i^{th}$  contact is,

$$f(\theta_i, \phi_i) = \begin{pmatrix} -\cos(\theta_i) \cos(\phi_i) \\ -\sin(\theta_i) \cos(\phi_i) \\ -\sin(\phi_i) \end{pmatrix}. \quad (4.3)$$

The gradient with respect to surface coordinates is,

$$\frac{\partial f(\theta_i, \phi_i)}{\partial(\theta_i, \phi_i)} = \begin{pmatrix} \sin(\theta_i) \cos(\phi_i) & \cos(\theta_i) \sin(\phi_i) \\ -\cos(\theta_i) \cos(\phi_i) & \sin(\theta_i) \sin(\phi_i) \\ 0 & -\cos(\phi_i) \end{pmatrix}. \quad (4.4)$$

The gradient of the force residual function (Equation 4.1) can now be expressed in terms of the object surface coordinates of the  $i^{th}$  contact:

$$\frac{\partial \epsilon_{fr}}{\partial(\theta_i, \phi_i)} = 2 \left( \sum_{i=0}^k \mathbf{f}_i \right)^T \frac{\partial \mathbf{f}_i}{\partial(\theta_i, \phi_i)}. \quad (4.5)$$

Note that the object itself does not need to be a sphere; given an arbitrary contact displacement, the change in net force applied to the convex object will have the same sign as it would if the object were a sphere. This allows the force residual controller to calculate the direction of lower force residual without measuring the local surface curvature.

The moment residual controller calculates the gradient of net moment as if the object surface were a plane oriented tangent to the object at the contact point. The moment generated by such a contact is

$$m(\theta_i, \phi_i) = \begin{pmatrix} -r_\theta \sin(\phi_i) \cos(\theta_i) + r_\phi \sin(\theta_i) \\ -r_\theta \sin(\phi_i) \sin(\theta_i) - r_\phi \cos(\theta_i) \\ r_\theta \cos(\phi_i) \end{pmatrix}. \quad (4.6)$$

In this equation, the angles  $\theta_i$  and  $\phi_i$  encode the orientation of the plane. The planar surface coordinates,  $r_\theta$  and  $r_\phi$ , encode contact position on the tangent plane with the origin at the current contact point. The gradient of this is:

$$\frac{\partial m(r_{\theta_i}, r_{\phi_i})}{\partial (r_{\theta_i}, r_{\phi_i})} = \begin{pmatrix} -\cos(\theta_i) \sin(\phi_i) & \sin(\theta_i) \\ -\sin(\theta_i) \sin(\phi_i) & -\cos(\theta_i) \\ \cos(\phi_i) & 0 \end{pmatrix}. \quad (4.7)$$

It is now possible to calculate the gradient of the moment residual in terms of the surface coordinates of the  $i^{\text{th}}$  contact,  $r_\theta$  and  $r_\phi$ :

$$\frac{\partial \epsilon_{mr}}{\partial (r_{\theta_i}, r_{\phi_i})} = 2 \left( \sum_{i=0}^k \mathbf{m}_i \right)^T \frac{\partial \mathbf{m}_i}{\partial (r_{\theta_i}, r_{\phi_i})}. \quad (4.8)$$

As with the spherical assumption, the planar assumption allows the moment residual controller to displace contacts in the correct direction without considering local surface geometry.

These two control laws are combined to create  $C^*$ , a composite controller that first executes the force control law to convergence and subsequently executes the moment control law to convergence. Coelho shows that the  $C^*$  controller converges to optimal grasp configurations for regular convex objects for 2 and 3 contacts [13].

## 4.2 Null Space Composition of Force Residual and Moment Residual

Section 4.1 describes two closed-loop controllers that can be used to displace contacts into quality grasp configurations. These two controllers,  $\pi_{fr}$  and  $\pi_{mr}$ , descend the force and moment error functions,  $\epsilon_{fr}$  and  $\epsilon_{mr}$  respectively (see Equations 4.1 and 4.2). This thesis takes this approach a step further by proposing that  $\pi_{mr}$  execute in the null space of  $\pi_{fr}$  [60]. This is similar to Coelho's  $C^*$  controller. However, since  $C^*$  executes the force and moment residual controllers sequentially, it makes it difficult to ensure that both error functions are converged when the controller finishes. While Coelho has shown that the moment residual controller does not ascend the force residual error function on regular convex objects, this is not true in general. In contrast, the null space composition approach to executing the force and moment residual controllers ensures that once the force residual error function is minimized, the moment controller does not subsequently ascend this function.

In order to execute the moment residual controller in the null space of the force residual controller, it is necessary to represent the gradients of both controllers in the same coordinate frame. We will parameterize the surface of an arbitrary closed convex object using spherical coordinates,  $(\theta, \phi)^T$ . The gradient of Coelho's force residual controller (Equation 4.5) is already expressed in spherical coordinates. Let the configuration of the  $i^{th}$  contact be  $\psi_i = (\theta_i, \phi_i)^T$  and let  $\psi = (\psi_1, \psi_2, \dots, \psi_k)^T$  be the configuration of  $k$  contacts. The gradient of the force error function in spherical coordinates is

$$\frac{\partial \epsilon_{fr}}{\partial \psi} = \frac{\partial \epsilon_{fr}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \psi}, \quad (4.9)$$

where  $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_k)^T$  and  $\mathbf{f}_i$  is the frictionless force applied by the  $i^{th}$  contact.

This controller can be encoded using the control basis framework of Section 3.1. The force residual gradient is  $\nabla_f \phi_{fr} = \frac{\partial \epsilon_{fr}}{\partial \mathbf{f}}^T$ . The force residual sensor transform calculates net frictionless force applied at the contacts. In order to calculate this, it is first necessary to define the unit normal sensor transform,

$$\sigma_n(\Gamma_\sigma) = (\mathbf{n}_{\gamma_1}, \dots, \mathbf{n}_{\gamma_{|\Gamma_\sigma|}})^T, \quad (4.10)$$

where  $\mathbf{n}_{\gamma_i}$  is the surface normal at the  $i^{th}$  contact. Now, the frictionless force residual can be defined to be the sum of unit surface normals,

$$\sigma_{fr}(\Gamma_\sigma) = \sum_{\gamma_i \in \Gamma_\sigma} \sigma_n(\gamma_i). \quad (4.11)$$

The effector transform converts the force residual gradient into surface coordinates for the set of contacts in  $\Gamma_\tau$ :

$$\tau_{fr}(\Gamma_\tau) = \left( \frac{\partial \mathbf{f}}{\partial \psi_{\gamma_1}}, \dots, \frac{\partial \mathbf{f}}{\partial \psi_{\gamma_{|\Gamma_\tau|}}} \right) = \frac{\partial \mathbf{f}}{\partial \psi_{\Gamma_\tau}}. \quad (4.12)$$

By Equation 3.5, the force residual controller,  $\pi_{fr} = \phi_{fr}|_{\tau_{fr}(\Gamma_\tau)}^{\sigma_{fr}(\Gamma_\sigma)}$  is implemented by

$$\nabla_\psi \phi_{fr} = \frac{\partial \mathbf{f}}{\partial \psi_{\Gamma_\tau}}^T \nabla_f \phi_{fr}(\sigma_{fr}(\Gamma_\sigma)), \quad (4.13)$$

where the reference force is taken to be zero. Note Equation 4.9 and 4.13 are the same.

In order to express the gradient of the moment residual controller in spherical coordinates,  $(\theta_i, \phi_i)$ , it is necessary to convert from surface Cartesian coordinates,  $(r_{\theta_i}, r_{\phi_i})$ . We will use the small angle assumptions,  $r_\theta = r\theta$  and  $r_\phi = r\phi$ , where  $r$  is the radius of the sphere used by the force residual controller. Assuming a unit sphere, we get  $(r_\theta, r_\phi) \approx (\theta, \phi)$ . The gradient of the moment residual error function as a function of the position of the  $i^{th}$  contact can be expressed in spherical coordinates,

$$\frac{\partial \epsilon_{mr}}{\partial \psi} = \frac{\partial \epsilon_{mr}}{\partial \mathbf{m}} \frac{\partial \mathbf{m}}{\partial (\theta, \phi)}, \quad (4.14)$$

where  $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k)^T$  is a vector of the contact moments for  $k$  contacts expressed in the object reference frame and  $(\theta, \phi) = ((\theta_1, \phi_1), \dots, (\theta_k, \phi_k))$  is the vector of surface coordinates for  $k$  contacts.

The moment residual controller is encoded in the control basis framework as follows. The gradient of the moment residual potential function is  $\nabla_m \phi_{mr} = \frac{\partial \epsilon_{mr}}{\partial \mathbf{m}}^T$ . The moment residual sensor transform uses unit normal and position sensor transforms as components to calculate the net moment residual among the contacts in  $\Gamma_\sigma$ :

$$\sigma_{mr}(\Gamma_\sigma) = \sum_{\gamma_i \in \Gamma_\sigma} (\sigma_p(\gamma_i) \times \sigma_n(\gamma_i)). \quad (4.15)$$

The effector transform converts the moment residual gradient into surface coordinates for the set of controller resources,  $\Gamma_\tau$ ,

$$\tau_{mr}(\Gamma_\tau) = \left( \frac{\partial \mathbf{m}}{\partial \psi_{\gamma_1}}, \dots, \frac{\partial \mathbf{m}}{\partial \psi_{\gamma_{|\Gamma_\tau|}}} \right) = \frac{\partial \mathbf{m}}{\partial \psi_{\Gamma_\tau}}. \quad (4.16)$$

In control basis notation, this controller is  $\pi_{mr} = \phi_{mr}|_{\tau_{mr}(\Gamma_\tau)}^{\sigma_{mr}(\Gamma_\sigma)}$ , and is implemented by,

$$\nabla_\psi \phi_{mr} = \frac{\partial \mathbf{m}}{\partial \psi_{\Gamma_\tau}}^T \nabla_m \phi_{mr}(\sigma_{mr}(\Gamma_\sigma)), \quad (4.17)$$

where the reference moment is zero.

Now that the force residual and moment residual controllers have been encoded in the control basis framework, these two controllers can execute concurrently using the *subject-to* operator:

$$\pi_g|_{\Gamma_\tau}^{\Gamma_\sigma} \equiv \phi_{mr}|_{\tau_{mr}(\Gamma_\tau)}^{\sigma_{mr}(\Gamma_\sigma)} \triangleleft \phi_{fr}|_{\tau_{fr}(\Gamma_\tau)}^{\sigma_{fr}(\Gamma_\sigma)}. \quad (4.18)$$

The equivalence in the above equation denotes that the composite controller is abbreviated by  $\pi_g|_{\Gamma_\tau}^{\Gamma_\sigma}$ . Note that this requires both sensor transforms and effector transforms to be parameterized by the same controller resources. Applying Equation 3.30, the gradient of the composite controller is:

$$\nabla_\psi (\phi_{mr} \triangleleft \phi_{fr}) = \nabla_\psi \phi_{fr} + \mathcal{N}(\nabla_\psi \phi_{fr}^T) \nabla_\psi \phi_{mr} \quad (4.19)$$

where

$$\mathcal{N}(\nabla_\psi \phi_{fr}^T) \equiv I - (\nabla_\psi \phi_{fr}^T)^\# (\nabla_\psi \phi_{fr}^T)$$

projects column vectors into the null space of  $\nabla_\psi \phi_{fr}^T$ . This composite controller calculates the gradients of both the force residual and the moment residual in spherical object surface coordinates. The null space term,  $\mathcal{N}(\nabla_\psi \phi_{fr}^T)$ , projects the gradient of the moment residual controller into the null space of the gradient of the force residual controller. The moment residual gradient is stripped of any component that ascends or descends the force residual error function.

The null space composition of force residual and moment residual controllers has two advantages over executing these controllers sequentially. First, projecting the moment residual controller into the null space of the force residual controller prevents it from ascending the force residual error function. This makes it easier for the composite controller to reach zero wrench residual configurations on arbitrary convex objects. In addition, executing both controllers concurrently accelerates the grasping process. When both objectives are compatible, the composite grasp controller can descend both error functions simultaneously.

### 4.3 Displacing Grasping Contacts by Sliding

The probing approach to contact displacement described in Section 4.1.2 lifts the contacts off the object surface and moves them through the air to a new contact configuration. In the context of grasp control (either the force residual and moment residual controllers of Section 4.1 or the composite grasp controller of Section 4.2), each iteration of the grasp controller is associated with a single probe. On each iteration, the controller reads the tactile data once and makes a single displacement. The number of probes required to reach a good grasp configuration can be minimized by making large contact displacements on every probe. However, displacement step size cannot be increased too much because the system has no prior knowledge of the shape of the object surface. Since, in our experiments, each tactile probe and displacement takes an amount of time on the order of at least one second, the number of probes required to move into a good grasp configuration forms a lower bound on how quickly grasp controllers can synthesize grasps.

#### 4.3.1 Sliding Contacts

In order to synthesize grasps faster and acquire more tactile information, this section proposes a grasp controller that slides contacts along the object surface into good grasp configurations [61]. This can be accomplished using a form of hybrid position-force control where each contact applies a small force along the inward object surface normal in order to maintain contact with the object. While maintaining this force, the contacts are displaced along the local surface tangent in the direction of the negative gradient of the grasp controller. Typically, a hybrid force-position controller specifies orthogonal directions for position control and force control to operate in. After computing force and position control errors, these errors are mapped through matrices,  $S_p$  and  $S_f$ , that project the error terms into orthogonal spaces. Next, the errors are summed, multiplied by a position gain and projected into the robot joint space. In order for this controller to function properly, it is essential that the two constraint matrices,  $S$  and  $S'$ , have orthogonal row spaces. If they do not, position and force error terms will compete with each other, interfering with both objectives.

Instead of explicitly coding the two constraint matrices, this section uses the control basis framework to combine force and position objectives. Recall from Section 3.1.1 that the position controller,  $\phi_p|_{\tau_p(\Gamma_\tau)}^{\sigma_p(\Gamma_\sigma)}(\mathbf{x}_{ref})$ , descends a gradient in joint position space,

$$\nabla_q \phi_p = \tau_p(\Gamma_m)^T \nabla_x \phi_p(\mathbf{x}_{ref} - \sigma_p(\Gamma_m)), \quad (4.20)$$

where  $J^+$  represents either the transpose or the pseudo-inverse of the Jacobian. Given an appropriate choice for the position gain (see Section 3.1.1),  $K_p$ , this controller moves the  $k$  control resources,  $\Gamma_m = (\gamma_1, \dots, \gamma_k)$ , to the vector of reference positions,  $\mathbf{x}_{ref}$ . The force controller (from Section 3.1.2),  $\phi_f|_{\tau_f(\Gamma_m)}^{\sigma_f(\Gamma_m)}(\mathbf{f}_{ref})$ , also descends a gradient in joint position space,

$$\nabla_q \phi_f = \tau_f(\Gamma_m)^T \nabla_f \phi_f(\mathbf{f}_{ref} - \sigma_f(\Gamma_m)). \quad (4.21)$$

This controller moves the vector of control resources,  $\Gamma_m$ , to the vector of reference forces,  $\mathbf{f}_{ref}$ .

Since these position and force controllers have been formulated in terms of the control basis framework, they can be combined using the *subject-to* operator:

$$\phi_p|_{\tau_p(\Gamma_m)}^{\sigma_p(\Gamma_m)}(\mathbf{x}_{ref}) \triangleleft \phi_f|_{\tau_f(\Gamma_m)}^{\sigma_f(\Gamma_m)}(\mathbf{f}_{ref}). \quad (4.22)$$

This controller is implemented using Equation 3.30:

$$\nabla_q(\phi_p \triangleleft \phi_f) = \nabla_q\phi_f + \mathcal{N}(\nabla_q\phi_f^T)\nabla_q\phi_p. \quad (4.23)$$

This controller applies the reference force,  $\mathbf{f}_{ref}$ , as a first priority and uses excess degrees of freedom to move to the reference position,  $\mathbf{x}_{ref}$ . Note that this encoding of the hybrid controller eliminates the need to manually specify the constraint matrices,  $S$  and  $S'$ . Since the position objective is projected into the null space of the force objective, the two terms of Equation 4.23 are guaranteed to have orthogonal row spaces.

The expression of Equation 4.22 can be applied to arbitrary hybrid position-force control problems. In the case of sliding contacts over a surface, each contact must apply a small force along the inward object surface normal. Then the force reference can be expressed,  $\mathbf{f}_{ref} = \kappa_f\sigma_n(\Gamma_m)$ , where  $\kappa_f$  is a constant parameter specifying the magnitude of the desired force and  $\sigma_n(\Gamma_m)$  is the vector of unit surface normals defined in Equation 4.10. The sliding contact displacement controller is defined by substituting this force reference into Equation 4.22:

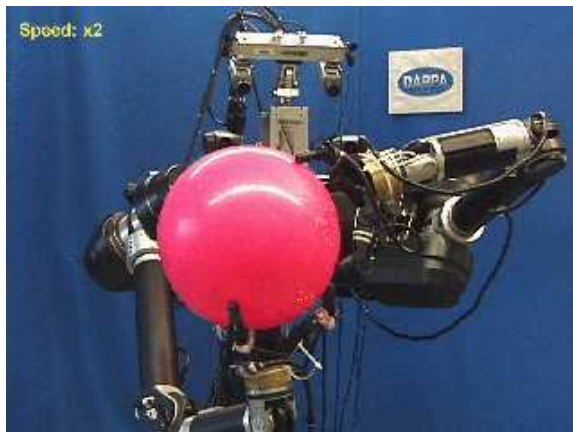
$$\pi_s|_{\Gamma_m}^{\Gamma_m}(\mathbf{x}_{ref}) \equiv \phi_p|_{\tau_p(\Gamma_m)}^{\sigma_p(\Gamma_m)}(\mathbf{x}_{ref}) \triangleleft \phi_f|_{\tau_f(\Gamma_m)}^{\sigma_f(\Gamma_m)}(\kappa_f\sigma_n(\Gamma_m)). \quad (4.24)$$

This controller slides the  $k$  contacts in  $\Gamma_m$  toward the vector contact reference positions encoded in  $\mathbf{x}_{ref}$ . Note that the abbreviation,  $\pi_s|_{\Gamma_m}^{\Gamma_m}(\mathbf{x}_{ref})$ , implicitly encodes the direction and magnitude of the force controller reference.

### 4.3.2 Posture Optimization During Sliding

As contacts slide over the object surface, the manipulator posture changes relative to the object so as to track the desired position. However, in some situations it may be desirable to maintain a particular posture with respect to the object. For example, Figure 4.1 illustrates Dexter holding a large ball in its right hand. The left hand is sliding over the ball surface toward the top. In order for Dexter's left hand to remain in contact with the ball, it is necessary for the hand to reorient so that the plane of the palm remains parallel to the object surface. In the case of Figure 4.1, if the hand orientation does not change, then contact cannot be maintained.

This section proposes executing the joint posture controller of Section 3.1.3 in the null space of (using the subject-to operator) the force controller in order to maintain a desired manipulator configuration relative to the object. Recall that the joint posture



**Figure 4.1.** The beach ball must remain within the workspace of the left hand contacts as the left hand slides over the ball surface.

controller,  $\phi_k|_{\tau_k(\Gamma_\tau)}^{\sigma_k(\Gamma_\sigma)}$ , servos a subset of the manipulator joints,  $\Gamma_\sigma$ , toward a reference configuration. The composite controller,

$$\phi_k|_{\tau_k(\Gamma_\tau)}^{\sigma_k(\Gamma_\sigma)} \triangleleft \phi_f|_{\tau_f(\Gamma_m)}^{\sigma_f(\Gamma_m)}(\kappa_f \sigma_n(\Gamma_m)), \quad (4.25)$$

uses the set of actuated joints,  $\Gamma_\tau$ , to reach a reference posture defined over  $\Gamma_\sigma$ . Without the primary force objective, the joint posture controller would simply move the joints in  $\Gamma_\sigma \cap \Gamma_\tau$  so as to realize the desired configuration for the joints in  $\Gamma_\sigma$ . However, since the force objective keeps the manipulator contacts on the surface of the object, the joint posture controller must move the non-optimized joints,  $\Gamma_\tau - \Gamma_\sigma$ , so as to maintain contact while the joints in  $\Gamma_\sigma$  move toward the reference configuration. In the example shown in Figure 4.1,  $\Gamma_\sigma$  might consist of the finger flexion joints on the left hand and  $\Gamma_\tau$  might include all joints in the left hand and arm. In addition, suppose that the reference posture,  $\mathbf{q}_{\text{ref}}$ , encodes finger flexion joint angles in the middle of their range. With this assignment to the variables, the composite controller of Equation 4.25 moves the arm so as to allow the finger flexion joints to reach the middle of their range without breaking contact with the object.

Integrating Equation 4.25 with the sliding controller of Equation 4.24, we get:

$$\pi_{sq}|_{\Gamma_m, \Gamma_\tau}^{\Gamma_m, \Gamma_\sigma}(\mathbf{x}_{ref}) \equiv \phi_p|_{\tau_p(\Gamma_m)}^{\sigma_p(\Gamma_m)}(\mathbf{x}_{ref}) \triangleleft \phi_k|_{\tau_k(\Gamma_\tau)}^{\sigma_k(\Gamma_\sigma)} \triangleleft \phi_f|_{\tau_f(\Gamma_m)}^{\sigma_f(\Gamma_m)}(\kappa_f \sigma_n(\Gamma_m)). \quad (4.26)$$

This controller, denoted by the abbreviation,  $\pi_{sq}|_{\Gamma_m, \Gamma_\tau}^{\Gamma_m, \Gamma_\sigma}(\mathbf{x}_{ref})$ , uses the degrees of freedom in  $\Gamma_\tau$  to optimize the posture of the joints in  $\Gamma_\sigma$  while sliding the contact resources,  $\Gamma_m$ , toward  $\mathbf{x}_{ref}$ .

### 4.3.3 Combining Grasping and Sliding

We would like to use this contact sliding controller to displace contacts during grasping. However, since the sliding controller is built on top of a position controller

that accepts only Cartesian references, the grasp controller must produce contact displacements in Cartesian coordinates. This can be accomplished by projecting the grasp displacements onto a plane tangent to the object surface at the point of contact. A gradient represented in surface coordinates can be projected onto the tangent plane by multiplying by a Jacobian,  $\frac{\partial \psi}{\partial \mathbf{x}}: \frac{\partial \epsilon}{\partial \mathbf{x}} = \frac{\partial \epsilon}{\partial \psi} \frac{\partial \psi}{\partial \mathbf{x}}$ . This Jacobian can be approximated using small angle assumptions:

$$\begin{aligned} \frac{\partial \psi}{\partial \mathbf{x}} &= \frac{\partial \psi}{\partial(r_\theta, r_\psi)} \frac{\partial(r_\theta, r_\psi)}{\partial \mathbf{x}} \\ &= \begin{pmatrix} \hat{\mathbf{r}}_\theta^T \\ \hat{\mathbf{r}}_\phi^T \end{pmatrix}, \end{aligned} \quad (4.27)$$

where  $\hat{\mathbf{r}}_\theta$  and  $\hat{\mathbf{r}}_\phi$  are orthogonal basis vectors in the tangent plane. Hence, the force residual and moment residual effector transforms for Cartesian displacement are:

$$\tau_{fr_x}(\Gamma_\tau) = \frac{\partial \mathbf{f}}{\partial \psi_{\Gamma_\tau}} \begin{pmatrix} \hat{\mathbf{r}}_\theta^T \\ \hat{\mathbf{r}}_\phi^T \end{pmatrix} \quad (4.28)$$

and

$$\tau_{mr_x}(\Gamma_\tau) = \frac{\partial \mathbf{m}}{\partial \psi_{\Gamma_\tau}} \begin{pmatrix} \hat{\mathbf{r}}_\theta^T \\ \hat{\mathbf{r}}_\phi^T \end{pmatrix}. \quad (4.29)$$

When these two effector transforms are substituted into the force residual and moment residual controllers of Section 4.2, the resulting controllers produce displacements in Cartesian space. The composite grasp controller is:

$$\pi_{gx}|_{\Gamma_\tau}^{\Gamma_\sigma} = \phi_{mr}|_{\tau_{mr_x}(\Gamma_\tau)}^{\sigma_{mr}(\Gamma_\sigma)} \triangleleft \phi_{fr}|_{\tau_{fr_x}(\Gamma_\tau)}^{\sigma_{fr}(\Gamma_\sigma)}, \quad (4.30)$$

where the abbreviation,  $\pi_{gx}|_{\Gamma_\tau}^{\Gamma_\sigma}$ , distinguishes this controller from  $\pi_g|_{\Gamma_\tau}^{\Gamma_\sigma}$ .

Now that we have defined a sliding controller and a variant of the composite grasp controller that produces a gradient in Cartesian space, a controller that slides contacts toward good grasp configurations can be defined:

$$\pi_{sgx}|_{\Gamma_\tau}^{\Gamma_\sigma} \equiv \pi_s|_{\Gamma_\tau}^{\Gamma_\sigma} \left( \pi_{gx}|_{\Gamma_\tau}^{\Gamma_\sigma} \right). \quad (4.31)$$

The main advantage of the sliding approach to grasp control, in contrast to the probing approach of Section 4.1.2, is that the grasp controller has access to much more tactile data. Instead of taking one sensory reading per probe, the sliding grasp controller can read new sensory information as often as needed. Since this grasp controller is constantly getting new data, it can potentially reach a good grasp configuration faster. Instead of being limited by the duration of the average probe, the speed of the sliding grasp controller is limited by the settle-time of the tactile sensors and the performance of the force controller. If the tactile sensors can update quickly and the force controller is able to maintain contact with the object surface, then the sliding grasp controller can be expected to find good grasp configurations very quickly.

An important consideration when selecting tactile feedback to use in a sliding grasp controller is the ability of the sensor to determine the object surface normal in

the presence of high tangential forces. This can be a particular problem with sliding control because the contacts will inevitably experience tangential forces caused by friction that opposes the direction of motion. If the contact normal is not isolated from the tangential components of forces, then the grasp controller will get poor information. This is another reason for using the fingertip load cells to determine contact locations and surface normals. The algorithm for contact localization using load cells by Bicchi, Salisbury, and Brock is able to cancel out these tangential forces [9].

## 4.4 Virtual Contacts

Up until this point, the grasp controller has been defined in terms of the force or moment residual calculated over a set of physical contacts. However, in addition to considering physical contacts, it is also possible to calculate grasp error functions with respect to *virtual contacts*. This idea was originally proposed by Arbib, Iberall, and Lyons in the context of a computational model of human grasps [3]. Instead of considering every possible grasp separately, they proposed a few fundamental grasp types (oppositions) that are parameterized by an appropriate set of contacts or fingers on the hand. In addition to being parameterized by physical fingers, oppositions can be parameterized by *virtual fingers*. A virtual finger is a group of hand surfaces that act together for the purposes of a particular grasp. For example, consider grasping an object between the thumb pad (the last phalange on the thumb) and the finger pads (the last phalange on the fingers). The grasp can be formed by using the index finger independently or by using the index and middle fingers together. When the index and middle fingers work together to form the grasp, they form a virtual finger (*i.e.* a virtual contact.)

### 4.4.1 Virtual Contacts Comprised of Multiple Physical Contacts

The control basis implementation of the grasp controller provides a quantitative way to realize virtual fingers. Up until this point, it has been assumed that a set of controller resources,  $\Gamma_i \subseteq \Gamma$ , that parameterize sensor and effector transforms was comprised of physical contacts. Sensor and effector transforms can be parameterized by virtual contacts by averaging the values of the sensor or effector transforms applied to the constituent contacts [57]. Given a sensor transform,  $\sigma_j$ , and a set of control resources,  $\alpha \subseteq \Gamma$ , the value of the sensor transform for the virtual contact resource,  $\gamma_\alpha$ , is the average of the sensor transform evaluated for the constituent contact resources,

$$\sigma_j(\gamma_\alpha) = \frac{1}{|\alpha|} \sum_{\gamma_i \in \alpha} \sigma_j(\gamma_i). \quad (4.32)$$

Hence, the position, force residual, or moment residual of a virtual contact is, respectively, the average position, force residual, or moment residual of the constituent contacts.



**Figure 4.2.** A grasp that uses a virtual contact. The two contacts on the left constitute a virtual contact that opposes the physical contact on the right.

The effector transform for a virtual contact acts on the control point calculated by the sensor transform parameterized by the virtual contact:

$$\begin{aligned}
 \tau_k(\gamma_\alpha) &= \frac{\partial \sigma_j(\gamma_\alpha)}{\partial \mathbf{y}_k} & (4.33) \\
 &= \frac{\partial}{\partial \mathbf{y}_k} \left[ \frac{1}{|\alpha|} \sum_{\gamma_l \in \alpha} \sigma_j(\gamma_l) \right] \\
 &= \frac{1}{|\alpha|} \sum_{\gamma_l \in \alpha} \frac{\partial \sigma_j(\gamma_l)}{\partial \mathbf{y}_k} \\
 &= \frac{1}{|\alpha|} \sum_{\gamma_l \in \alpha} \tau_k(\gamma_l) & (4.34)
 \end{aligned}$$

By averaging the component Jacobian matrices, Equation 4.34 displaces a control point located at the average position of the constituent contacts. For example, Figure 4.2 illustrates a grasp where the two fingertip contacts on the left act as a single virtual contact that opposes the third contact on the right. Let the physical contact on the right correspond to the physical contact resource,  $\gamma_1 \in \Gamma$ . Let the virtual contact,  $\gamma_\alpha$ , on the left correspond to the set of physical contact resources,  $\alpha = \{\gamma_2, \gamma_3\}$ . The force residual sensor and effector transforms for this contact configuration are:

$$\sigma_{fr}(\{\gamma_1, \gamma_\alpha\}) = \frac{1}{2} [\sigma_{fr}(\gamma_1) + \sigma_{fr}(\gamma_\alpha)] = \frac{1}{2} \left[ \sigma_{fr}(\gamma_1) + \frac{1}{2} (\sigma_{fr}(\gamma_2) + \sigma_{fr}(\gamma_3)) \right]$$

and

$$\tau_{fr}(\{\gamma_1, \gamma_\alpha\}) = \left( \frac{\partial \mathbf{f}}{\partial \phi_{\gamma_1}}, \frac{\partial \mathbf{f}}{\partial \phi_{\gamma_\alpha}} \right) = \left( \frac{\partial \mathbf{f}}{\partial \phi_{\gamma_1}}, \frac{1}{2} \left( \frac{\partial \mathbf{f}}{\partial \phi_{\gamma_2}} + \frac{\partial \mathbf{f}}{\partial \phi_{\gamma_3}} \right) \right).$$

The sensor transform calculates the force residual between  $\gamma_1$  and the average of the constituent contacts,  $\gamma_2$  and  $\gamma_3$ . Similarly, the effector transform calculates a Jacobian transpose for  $\gamma_1$  and  $\gamma_\alpha$ , where the effector transform of the virtual contact is the average of that for the two constituent contacts. This effector transform displaces the virtual contact by the average of the amounts by which it would have displaced the individual contacts.

In the context of grasping, virtual contacts are an important way to affect the grasp to which the controller converges. For example, in Figure 4.2, when a grasp controller utilizes the two fingers on the left as a virtual finger, the result is essentially a two-contact grasp: an opposition between the two fingers on the left and the finger on the right. Contrast this with a grasp where all three fingers oppose each other equally, thus forming an equilateral triangle. The remainder of this thesis will include many examples of grasp controllers parameterized with virtual fingers. This chapter combines two of the Barrett hand fingers into a virtual finger, as shown in Figure 4.2. Chapter 5 describes bimanual grasps where Dexter treats three fingers on the hand as a single virtual contact and forms two-handed grasps by controlling two virtual point contacts.

#### 4.4.2 Gravity as a Virtual Contact

In addition to the forces that can be applied by a group of physical contacts, the notion of a virtual finger can also represent forces applied by gravity. This allows grasps that rely on gravity, such as the platform or hook grasp, to be represented in a consistent way alongside grasps that rely on physical contact exclusively [40]. As a virtual contact,  $\gamma_g$ , gravity must always be understood in relation to an object that is being grasped. It applies a force at the center of mass directed in the negative  $z$  direction of the world frame:

$$\sigma_p(\gamma_g) \equiv \mathbf{x}_{CG}, \quad (4.35)$$

$$\sigma_f(\gamma_g) \equiv \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix}, \quad (4.36)$$

and

$$\sigma_n(\gamma_g) \equiv \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}. \quad (4.37)$$

The gravity virtual contact cannot parameterize the effector transform.

One complexity that arises when the gravity virtual contact parameterizes a grasp controller is that it is frequently more practical to synthesize a grasp by moving the object rather than moving the contacts relative to the object. Instead of displacing contacts relative to the surface of the object, the grasp controller should rotate the controllable contact resource (and the entire object along with it) so that the wrench

residual between gravity and the controlled contact resource tends toward zero. Note that this “rotation” version of the grasp controller assumes that the opposing contact is fixtured to the object. Therefore, this controller can only be used when the object is already grasped by some combination of contact resources. The subject of synthesizing grasps while holding an object is discussed in detail in Chapter 5. For the purposes of the current discussion, assume that the object does not drop while the contact is rotating into opposition with gravity.

Section 4.2 expressed the force residual and moment residual gradients in terms of spherical coordinates. Instead of projecting this gradient into Cartesian space, as in Section 4.3.3, the rotation version of the grasp controller is defined by converting the gradient into a rotation. Let  $\frac{\partial \psi_i}{\partial \mathbf{r}_i}$  be the gradient of spherical coordinates with respect to the  $i^{\text{th}}$  contact’s orientation, expressed in exponential coordinates. Then the effector transform that projects the force residual error into an angular velocity for the set of contact resources,  $\Gamma_\tau$ , is

$$\tau_{fr_r} = \frac{\partial \mathbf{f}}{\partial \psi_{\Gamma_\tau}} \frac{\partial \psi_{\Gamma_\tau}}{\partial \mathbf{r}_{\Gamma_\tau}}, \quad (4.38)$$

where  $\psi_{\Gamma_\tau} = (\psi_1, \dots, \psi_{|\Gamma_\tau|})^T$  and  $\mathbf{r}_{\Gamma_\tau} = (\mathbf{r}_1, \dots, \mathbf{r}_{|\Gamma_\tau|})^T$  are vectors of spherical coordinates and orientations for the contact resources in  $\Gamma_\tau$ . Likewise, the effector transform that projects the moment residual error into an angular velocity for the contact resources,  $\Gamma_\tau$ , is

$$\tau_{mr_r} = \frac{\partial \mathbf{m}}{\partial \psi_{\Gamma_\tau}} \frac{\partial \psi_{\Gamma_\tau}}{\partial \mathbf{r}_{\Gamma_\tau}}. \quad (4.39)$$

Using these rotational effector transforms, a rotational grasp controller can be defined,

$$\pi_{g\theta}|_{\Gamma_\tau}^{\Gamma_\sigma} \equiv \phi_{mr}|_{\tau_{mr_r}(\Gamma_\tau)}^{\sigma_{mr}(\Gamma_\sigma)} \triangleleft \phi_{fr}|_{\tau_{fr_r}(\Gamma_\tau)}^{\sigma_{fr}(\Gamma_\sigma)}. \quad (4.40)$$

This controller can be used as the reference for the orientation controller of Section 3.1.1,

$$\pi_{rg\theta}|_{\Gamma_\tau}^{\Gamma_\sigma} \equiv \phi_r|_{\tau_r(\Gamma_\tau)}^{\sigma_r(\Gamma_\sigma)} \left( \pi_{g\theta}|_{\Gamma_\tau}^{\Gamma_\sigma} \right). \quad (4.41)$$

This rotational grasp controller rotates the manipulator so as to oppose the controlled contacts,  $\Gamma_\tau$  with the gravitational virtual contact. If the gravitational contact resources,  $\gamma_g$ , is not among the contact resources that parameterize the sensor transform, then the gradient of Equation 4.41 is always zero.

## 4.5 Experiments

Experiments were conducted that characterize how robustly the sliding grasp controller,  $\pi_s|_{\Gamma_\tau}^{\Gamma_\sigma}$  ( $\pi_{gs}|_{\Gamma_\tau}^{\Gamma_\sigma}$ ), can synthesize grasps and the conditions under which it will converge to a good grasp (force closure) configurations. In a series of four experiments, the sliding grasp controller is executed a number of times from different starting configurations in each of four different grasp scenarios. All experiments were conducted on Dexter, the UMass bimanual humanoid robot described in Appendix C. Grasp



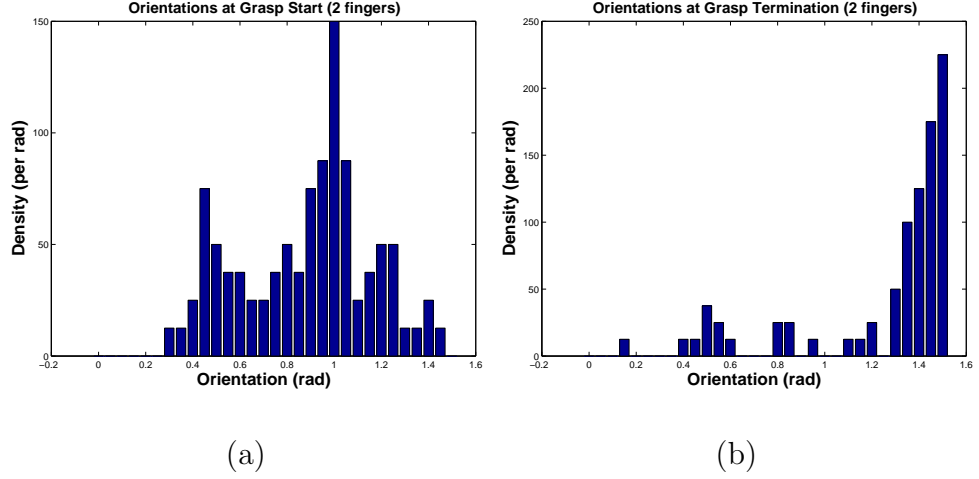
**Figure 4.3.** The sliding grasp controller,  $\pi_s|_{\Gamma_\tau}^{\Gamma_\tau} \left( \pi_{gx}|_{\Gamma_\tau}^{\Gamma_\sigma} \right)$ , was characterized for these three objects.

controller performance is characterized for the towel roll, squirt bottle, and detergent bottle shown in Figure 4.3. Since the grasp controller does not make any prior assumptions about object geometry, orientation, or pose, these experiments accurately reflect expected performance of the sliding grasp controller in uncontrolled and unmodeled domains. The results show the sliding grasp controller to be a practical way of synthesizing grasps in uncontrolled scenarios. The sliding grasp controller converges to grasp configurations with low force and moment residual errors from a variety of starting configurations. When a distribution over starting and ending configurations is calculated, the grasp controller is shown to funnel a large number of starting configurations toward a small set of good grasp configurations. Our experiments also show the sliding grasp controller to be susceptible to local minima caused by kinematic limitations of the manipulator. We propose avoiding these local minima by identifying domains of attraction that enable the grasp controller to find good grasps. Subsequent chapters of this thesis consider ways of ensuring that grasp controllers start execution within the correct domain of attraction.

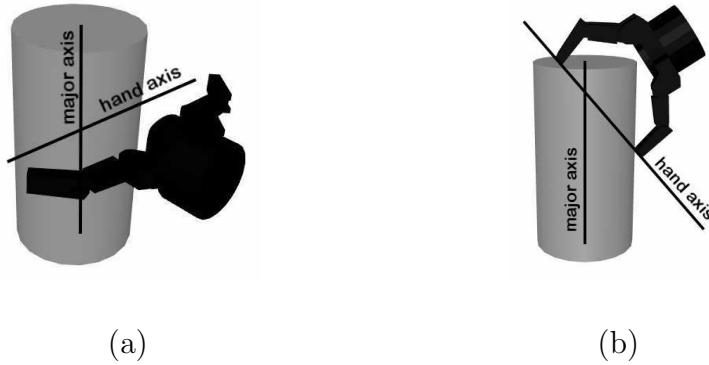
#### 4.5.1 Experiment 1: Grasping a Towel Roll Using Two Virtual Fingers

The first experiment characterizes sliding grasp controller performance by using Dexter to attempt to grasp the vertical towel roll (10cm in diameter and 20cm tall), shown in Figure 4.3(a), 58 times. In these grasp trials, the sliding grasp controller began execution from a variety of different manipulator configurations, illustrated in Figure 4.4(a). On each trial, the sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_{23}\}}^{\{\gamma_1, \gamma_{23}\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_{23}\}}^{\{\gamma_1, \gamma_{23}\}} \right)$ , executed until convergence or until grasp failure as determined by the human monitor. This controller was parameterized by two contact resources on Dexter’s Barrett hand. The Barrett hand has three fingers with corresponding contact resources,  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$ . In Experiment 1, contact resources,  $\gamma_2$  and  $\gamma_3$  were combined to form a single

virtual contact,  $\gamma_{23} = \{\gamma_2, \gamma_3\}$ . The sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_{23}\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_{23}\}} \right)$ , tended toward grasp configurations that opposed  $\gamma_1$  and  $\gamma_{23}$ .



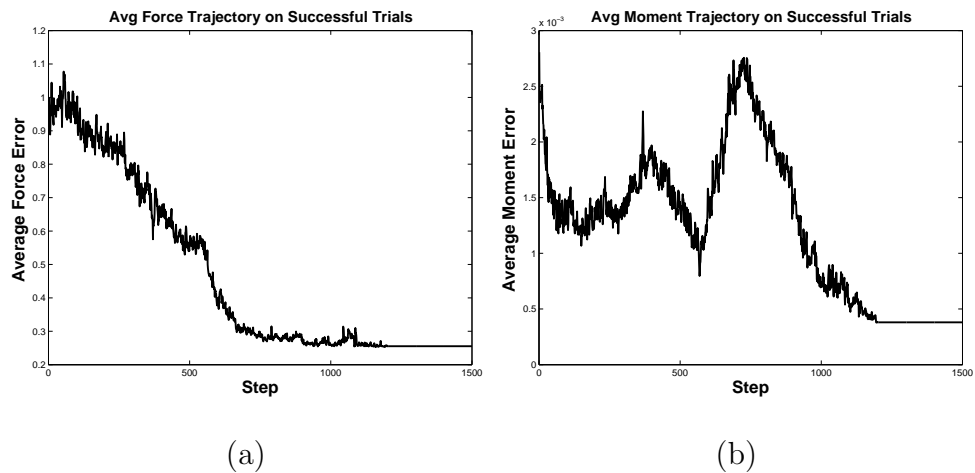
**Figure 4.4.** Experiment 1 (towel roll, two contacts): the distribution of contact orientations before, (a), and after, (b), the grasp controller has executed. Orientation is the angle between a line that passes between the two grasp contacts and the major axis of the object (see text).



**Figure 4.5.** Experiment 1 (towel roll, two contacts): (a) grasp configuration corresponding to a peak in Figure 4.4(b) near an orientation of  $\pi/2$  radians. (b) configuration corresponding to smaller peak near an orientation of 0.45 radians.

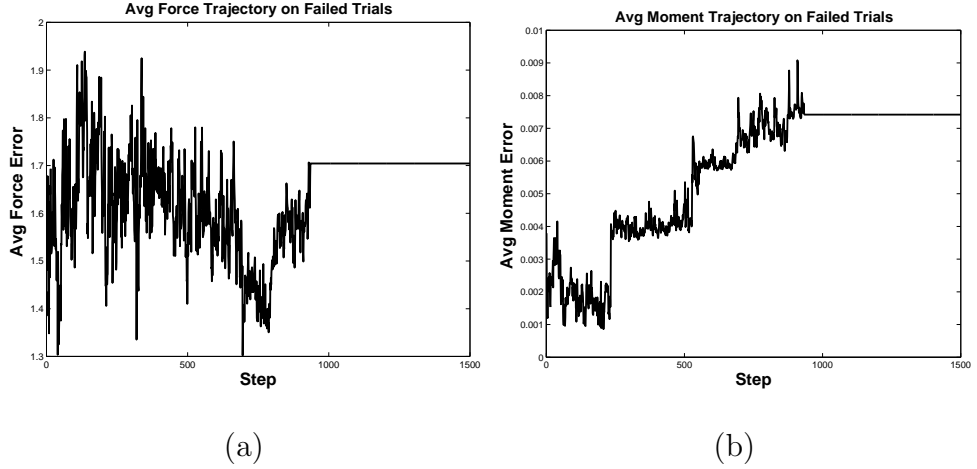
Experiment 1 shows that for the towel roll, the two-finger sliding grasp controller funnels the robot toward good grasp configurations. Figure 4.4(a) shows the density of

manipulator orientations before executing the grasp controller. The hand orientation is measured in terms of the line that connects the two virtual contacts. Orientation is the angle between this line and the towel roll major axis. Notice that the grasp controller begins execution in a variety of orientations. Figure 4.4(b) illustrates the density of manipulator configurations after grasp controller execution. Notice that the largest peak is near  $\pi/2$  radians. This corresponds to the configuration shown in Figure 4.5(a), where the line formed by the two contacts is perpendicular to the object major axis. Also, notice that there is a much smaller peak near 0.45 radians. This corresponds to the configurations shown in Figure 4.5(b), where one finger is on the top of the object and the other finger on the side.



**Figure 4.6.** Experiment 1 (towel roll, two contacts): average force residual, (a), and moment residual, (b), for the grasp trials that terminated near the peak at  $\pi/2$  in Figure 4.4(b).

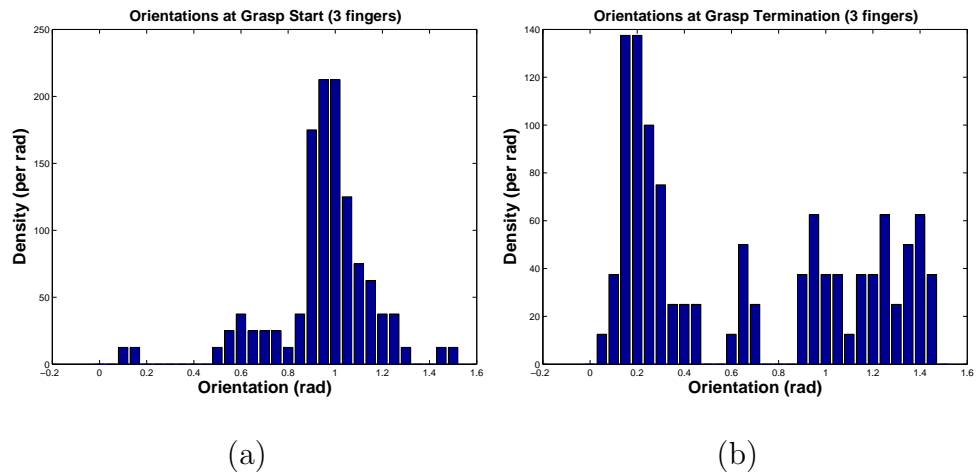
The average force and moment residual error trajectories for the grasp trials that comprise the peak near  $\pi/2$  in Figure 4.4(b) are illustrated in Figure 4.6. Figure 4.6(a) shows the average force residual error while Figure 4.6(b) shows the average moment residual error. Note that these grasps converge to low force residual and moment residual errors, corresponding to good grasp configurations. The horizontal axis in both figures is grasp controller iteration. Since the contacts are constantly touching the object, the grasp controller is able to update approximately once every 20ms (50Hz). The graphs illustrate that, on average, both force and moment errors converge in approximately 1000 iterations (20 seconds, not including the time taken to tare the fingertip load cells.) Notice that the relative priority of the force residual and the moment residual errors is evident from the figures. On average, force residual decreases monotonically from the start while moment residual only converges after force residual converges. At first, in order to reduce force residual, the controller sacrifices moment residual. Only after force residual converges does the controller also minimize moment residual.



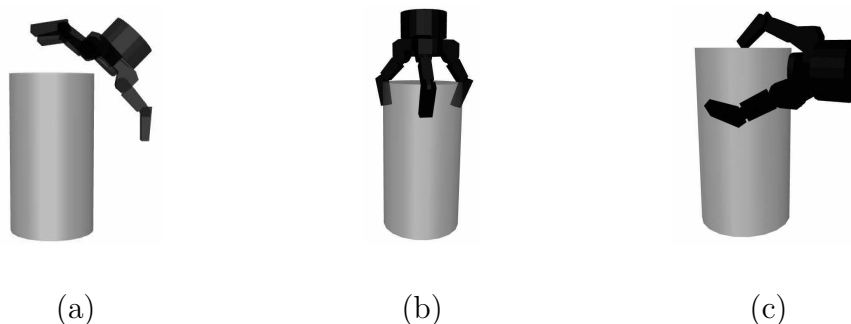
**Figure 4.7.** Experiment 1 (towel roll, two contacts): average force residual, (a), and moment residual, (b), for the grasp trials that terminated outside of the peak at  $\pi/2$  in Figure 4.4(b).

Figure 4.7 illustrates the average force and moment residual error trajectories for the grasp trails that did not peak around  $\pi/2$  in Figure 4.4(b). These plots are noisier because the average is taken over fewer samples because few grasp trials failed. Figure 4.7(a) shows that average force residual error never significantly decreases while Figure 4.7(b) shows that moment residual error actually increases. On these trials, the grasp controller failed to reach a good grasp configuration. The approximate physical configuration of the manipulator during these trials is illustrated in Figure 4.5(b). On these runs, the kinematic limitations of the fingers prevented the two contacts from reaching all the way around the object before the palm collided with the object. Since Dexter had no sensing on the palm, trials where the palm collided with the object were prematurely stopped by the human monitor.

These results show that the sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_{23}\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_{23}\}} \right)$ , converges to good grasp configurations on a vertical cylinder from a large number of starting orientations. These good grasp configurations are characterized by low force residual and moment residual errors. Nevertheless, it is possible for the controller to fail to reach a good grasp configuration due to the kinematic limitations of the manipulator. In the absence of these limitations (imagine “floating” contacts), the manipulator can be expected to converge to a good grasp configuration. However, the kinematics of the manipulator can effectively introduce local minima into the grasp artificial potential. Subsequent chapters of this thesis study how to start the sliding grasp controller from configurations where convergence can be expected.



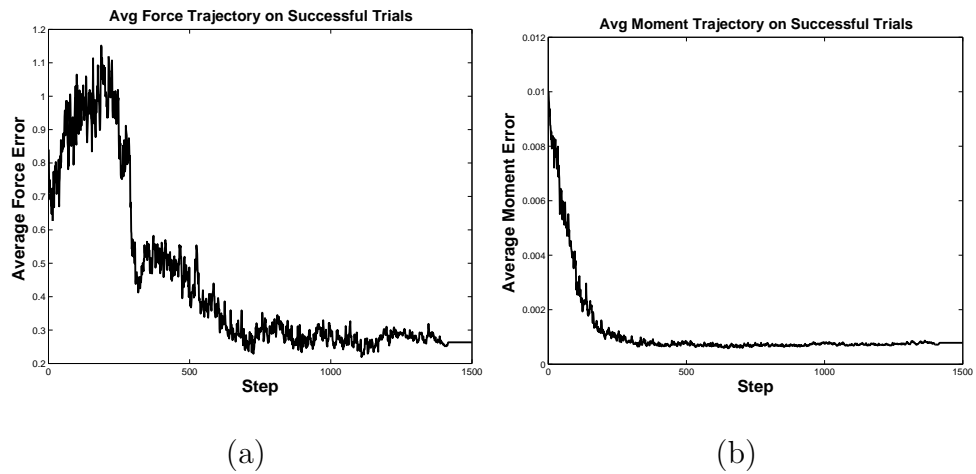
**Figure 4.8.** Experiment 2 (towel roll, three contacts): the distribution of contact orientations before, (a), and after, (b), the three-contact grasp controller has executed. Orientation is the angle between a normal to the plane of the three grasp contacts and the major axis (see text).



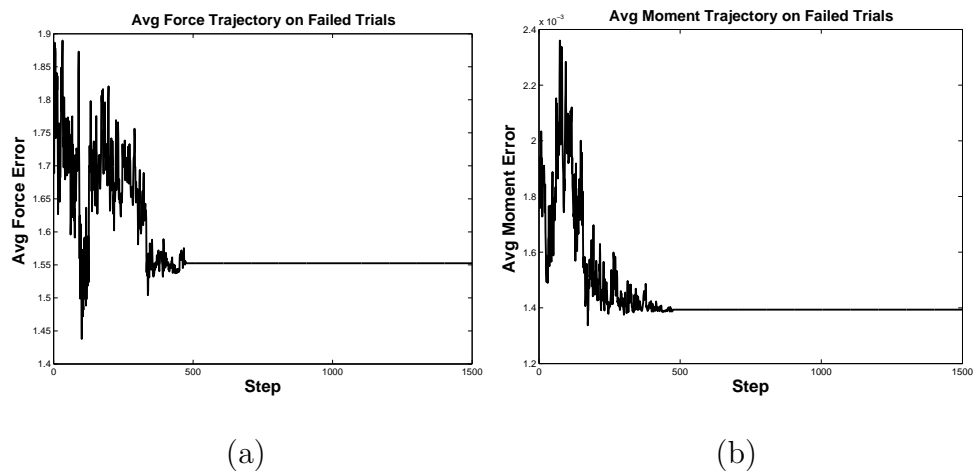
**Figure 4.9.** Manipulator configurations during Experiment 2. (a) shows the manipulator at a starting configuration near the peak in Figure 4.8(a). (b) shows the manipulator after the sliding grasp controller has executed and the manipulator has reached a globally optimal grasp. (c) shows the manipulator after grasp controller execution has reached a local minimum in the force residual error function.

#### 4.5.2 Experiment 2: Grasping a Towel Roll Using Three Virtual Fingers

The second experiment tested a three-contact parameterization of the sliding grasp controller. Dexter executed 61 reaches and grasps on the vertical towel roll shown in Figure 4.3(a). Figure 4.8(a) shows the density of manipulator orientations before the grasp controller executed. This figure measures orientation in terms of the plane



**Figure 4.10.** Experiment 2 (towel roll, three contacts): average force residual, (a), and moment residual, (b), for the grasp trials that terminated near the peak near 0 radians in Figure 4.4(b).



**Figure 4.11.** Experiment 2 (towel roll, three contacts): average force residual, (a), and moment residual, (b), for the grasp trials that terminated outside of the peak near 0 radians in Figure 4.4(b).

of the three contacts. Orientation is the angle between the normal of this plane and the major axis of the object. Figure 4.8(a) shows that the three-contact sliding grasp controller executed from a distribution of starting orientations with a strong peak near 1 radian. This 1 radian peak corresponds to a starting manipulator configuration like that shown in Figure 4.9(a).

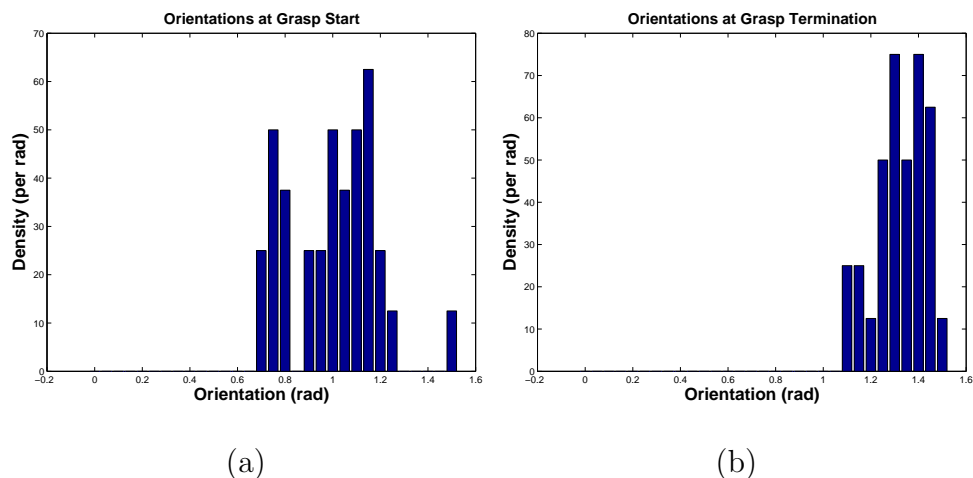
Figure 4.8(b) shows the density of manipulator orientations after executing the three-contact sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_2, \gamma_3\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_2, \gamma_3\}} \right)$ . This controller is parameterized by the three contact resources on each of the three fingers of the Barrett hand. For more on the Barrett hand, see Appendix C. Figure 4.8(b) shows that the sliding grasp controller funneled the distribution of manipulator configurations away from the peak near 1 radian to a peak near 0 radians. This peak corresponds to good grasp configurations where the Barrett hand is directly above and aligned with the towel roll, as shown in Figure 4.9(b). In addition, Figure 4.8(b) also shows that on a significant number of grasp trials, the sliding grasp controller converged to other orientations between 0.8 and  $\pi/2$  radians. One of these grasps is shown in Figure 4.9(c).

Figures 4.10 and 4.11 show the average force residual and moment residual error functions for grasps that terminated, respectively, in orientations near the 0 radian peak and between 0.8 and  $\pi/2$  radians in Figure 4.8(b). Note that the grasps that terminated near the 0 radian peak converge, on average, to force residual errors below 0.3 Newtons. In contrast, grasps that terminated between 0.8 and  $\pi/2$  radians converged to force residual errors closer to 1.55 Newtons. The grasps that terminate near the 0 radian peak are “successful” in the sense that they minimize the contact wrench residual and, therefore, for a positive coefficient of friction, are force closure configurations. In this configuration, the robot can resist large perturbing forces by applying arbitrarily large forces at the three contacts. In contrast, the grasps that terminate in orientations between 1 and  $\pi/2$  radians have large net force residuals because the contact on the top of the cylinder (see Figure 4.9(c)) is not opposed by either of the other two contacts. Note that even for the grasps that ultimately fail, the grasp controller minimizes the force residual and moment residual errors. Interestingly, the moment residual error converged to similarly low values in both cases. However, the contact configuration shown in Figure 4.9(c) is essentially a local minimum in the force residual error function. The problem is that, once one of the three contacts reaches the top of the cylinder, the grasp controller must ascend the force residual error function in order to move that contact onto the side of the cylinder. Viewed from the side, the cylinder is essentially a rectangle. Although floating contacts without kinematic constraints can grasp a rectangle by moving toward the corners, in this case, the fingers of the Barrett hand are not long enough to allow this. Therefore, a substantial number of grasp trials get “caught” in this local minimum and ultimately terminate with a high grasp error.

The results from Experiment 2 mirror those from Experiment 1. The three-contact sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_2, \gamma_3\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_2, \gamma_3\}} \right)$ , funnels the manipulator from a large number of poor grasp configurations toward good grasps characterized by low force residual and moment residual errors. Of particular note in this experiment is the significant number of grasp trials that ended in a local minimum characterized by a high force residual error. This is a result of the fact that a three-fingered manipulator can grasp a cylinder in two qualitatively different ways. One way to grasp the cylinder places all three fingers around the radius of the cylinder. The other way depends on rounded fingertips: it places one contact on the top of the cylinder and the other

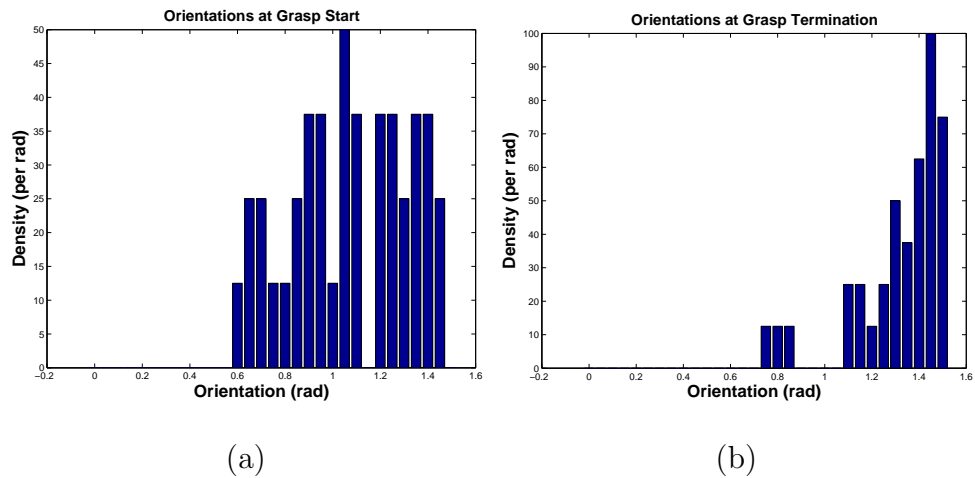
two contacts on the opposite corners, as if the robot were grasping a rectangle with three fingers. In the grasp trials that failed to reach a low force residual error, the controller got “caught” attempting to reach the “rectangle” grasp. Although this grasp is feasible in principle, kinematic limitations (to say nothing of the table the object is resting on) prevent this approach from succeeding.

### 4.5.3 Experiments 3 and 4: Grasping a Squirt Bottle and a Detergent Bottle

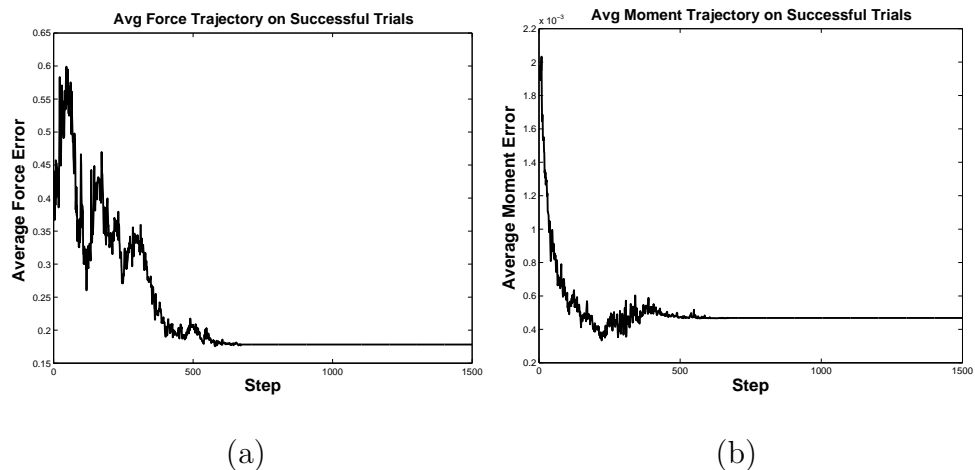


**Figure 4.12.** Experiment 3 (squirt bottle): the distribution of contact orientations before, (a), and after, (b), the grasp controller has executed.

Experiments 3 and 4 characterize the sliding grasp controller on a squirt bottle and a detergent bottle, respectively, as shown in Figures 4.3(b) and 4.3(c). The experimental procedure for these objects is roughly the same as in the earlier experiments. On each trial, Dexter reached toward the object and executed the two-contact sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_{23}\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_{23}\}} \right)$ . In Experiment 3, Dexter executed 28 grasps of the squirt bottle. In Experiment 4, Dexter executed 31 grasps of the detergent bottle. As in Experiment 1, this controller is parameterized by a physical contact resource,  $\gamma_1$ , and a virtual contact resource,  $\gamma_{23} = \{\gamma_2, \gamma_3\}$ . Figures 4.12 and 4.13 illustrate the manipulator configurations before and after executing the grasp controller for the squirt bottle and the detergent bottle, respectively. As in Experiment 1, orientation is measured to be the angle between the object major axis and the line connecting the two virtual contacts. Figures 4.12 and 4.13 show that in both experiments, the sliding grasp controller starts in a range of configurations and funnels the manipulator toward configurations where it is approximately perpendicular to the major axis of the object. Average grasp controller performance on both objects is illustrated in Figures 4.14 and 4.15.

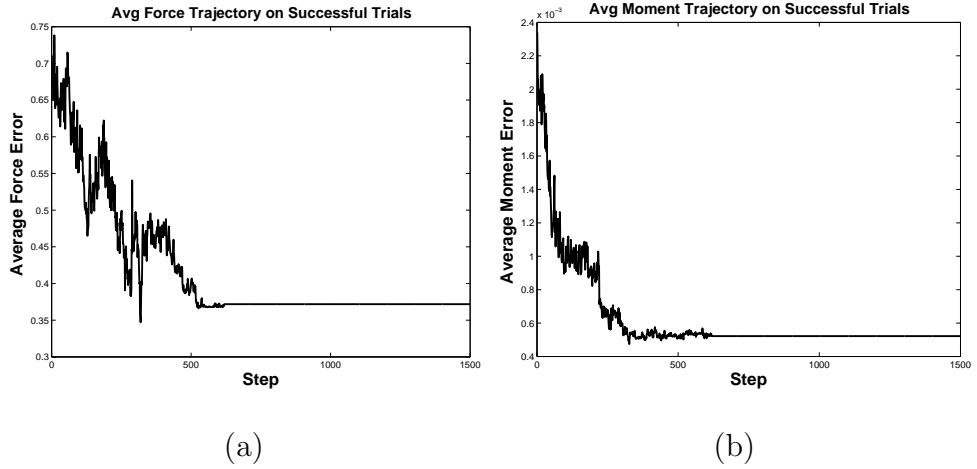


**Figure 4.13.** Experiment 4 (detergent bottle): the distribution of contact orientations before, (a), and after, (b), the grasp controller has executed.



**Figure 4.14.** Experiment 3 (squirt bottle): average force residual, (a), and moment residual, (b), for the grasp trials that terminated near the peak at  $\pi/2$  in Figure 4.4(b).

In contrast to Figures 4.4(b) and 4.8(b), the density functions in Figures 4.12(b) and 4.13(b) have only one mode. This indicates that for the squirt bottle and the detergent bottle, the sliding grasp controller always converged to a single range of configurations. Therefore, instead of plotting average force residual and moment residual for different grasp outcomes, Figures 4.14 and 4.15 are averaged over all grasp trials. These averages show that the sliding grasp controller converges to low wrench residual configurations for both objects. These are good grasp configurations



**Figure 4.15.** Experiment 4 (detergent bottle): average force residual, (a), and moment residual, (b), for the grasp trials that terminated near the peak at  $\pi/2$  in Figure 4.4(b).

because the manipulator is able to resist large perturbation forces by squeezing the object arbitrarily tightly.

Experiment 3 and 4 show that the sliding grasp controller,  $\pi_s|_{\{\gamma_1, \gamma_{23}\}} \left( \pi_{gx}|_{\{\gamma_1, \gamma_{23}\}} \right)$ , can be used to grasp objects used in everyday environments. Although the controller had no prior knowledge regarding the geometry of either the squirt bottle or the detergent bottle, it is shown to effectively funnel the manipulator to good grasp configurations. In fact, as a result of the narrow profile of these objects, the controller never gets caught in kinematically-induced local minima.

## 4.6 Summary

This chapter proposes several composite grasp controllers based on Coelho’s force residual and moment residual controllers. First, a composite controller is proposed that executes the moment residual controller in the null space of the force residual controller. This composite controller is more robust and faster than executing either of the constituent controllers separately. Next, a controller is proposed that concurrently executes a hybrid force and position controller and a grasp controller. The hybrid force-position controller slides the grasp contacts over the surface of the object toward good grasp configurations under the direction of grasp control. The problem of kinematic posture optimization during grasping is also considered. In addition, this chapter expands the set of contact resources by proposing that grasp controllers be parameterized by virtual contacts. A virtual contact combines sensory information from multiple physical contacts in order to create what, for the grasp controller, is a single logical contact. These controllers are characterized in a series of experiments where Dexter, the UMass bimanual humanoid, grasps an unmodeled

cylinder, squirt bottle, and detergent bottle using two and three fingers. Although it is possible for the controller to get caught in local minima, the results show that the sliding grasp controller effectively funnels the manipulator to good grasp configurations from a range of starting configurations. One way to avoid getting caught in a local minimum is to ensure that the manipulator is the domain of attraction of a good grasp configuration before executing the grasp controller. Chapters 6 and 7 pursue this idea using schema structured learning framework. In these chapters, the robot autonomously learns which instantiations of a reach controller are likely to deliver the system to configurations where the grasp controller converges to a good grasp.

## BIBLIOGRAPHY

- [1] Allen, P., Miller, A., Oh, P., and Leibowitz, B. Integration of vision, force, and tactile sensing for grasping. *Int'l Journal of Intelligent Mechatronics* 4, 1 (1999), 129–149.
- [2] Arbib, M. A. Schema theory. In *Encyclopedia of Artificial Intelligence (2nd Edition)*. Wiley-Interscience, 1992.
- [3] Arbib, M. A., Iberall, T., and Lyons, D. Coordinated control program for movements of the hand. *Exp. Brain Research* (1985), 111–129.
- [4] Arkin, R. *Behavior-Based Robotics*. MIT Press, 1998.
- [5] Atkeson, Chris, Moore, Andrew, and Schaal, Stefan. Locally weighted learning. *AI Review* 11 (April 1997), 11–73.
- [6] Atkeson, Chris, Moore, Andrew, and Schaal, Stefan. Locally weighted learning for control. *AI Review* 11 (April 1997), 75–113.
- [7] Bellman, R. E. *Dynamic Programming*. Princeton University Press, Princeton, 1957.
- [8] Bicchi, A. On the closure properties of robotic grasping. *Int. J. of Robotics Research* 14, 4 (1995).
- [9] Bicchi, A., Salisbury, J., and Brock, D. Contact sensing from force measurements. *International Journal of Robotics Research* 12, 3 (1993).
- [10] Brooks, R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2, 1 (March 1986), 14–23.
- [11] Burridge, R., Rizzi, A., and Koditschek, D. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research* 18, 6 (1999).
- [12] Coelho, J. *Multifingered Grasping: Grasp Reflexes and Control Context*. PhD thesis, University of Massachusetts, 2001.
- [13] Coelho, J., and Grupen, R. A control basis for learning multifingered grasps. *Journal of Robotic Systems* (1997).

- [14] Coelho, J, Piater, J., and Grupen, R. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. *Robotics and Autonomous Systems Journal, special issue on Humanoid Robots 37*, 2-3 (November 2001).
- [15] Connolly, C., and Grupen, R. Nonholonomic path planning using harmonic functions. Tech. rep., University of Massachusetts, 1994.
- [16] Cutkosky, M., and Howe, R. *Dextrous robot hands*. NY: Springer-Verlag, 1990, ch. Human grasp choice and robotic grasp analysis, pp. 5–31.
- [17] Cutkosky, M., and Wright, P. Modeling manufacturing grips and correlations with the design of robotic hands. In *IEEE Int’l Conf. Robotics Automation* (April 1986), vol. 3, pp. 1533–1539.
- [18] Dean, T., and Givan, R. Model minimization in markov decision processes. In *AAAI* (1997), pp. 106–111.
- [19] Dearden, R., Frieman, N., and Andre, D. Model based bayesian exploration. In *Proceedings of Fifteenth Conference on Uncertainty in Artificial Intelligence* (1999).
- [20] Drescher, G. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press, 1991.
- [21] Farooqi, M., Tanaka, T., Ikezawa, Y., and Omata, T. Sensor based control for the execution of regrasping primitives on a multifingered robot hand. In *IEEE Int’l Conf. Robotics Automation* (May 1999).
- [22] Faverjon, B., and Ponce, J. On computing two-finger force-closure grasps of curved 2d objects. In *IEEE Int’l Conf. Robotics Automation* (1991).
- [23] Fearing, R.S. Simplified grasping and manipulation with dextrous robot hands. *IEEE Journal of Robotics and Automation* 2, 4 (December 1986).
- [24] Ferrari, C., and Canny, J. Planning optimal grasps. In *IEEE Int’l Conf. Robotics Automation* (May 1992).
- [25] Fuentes, O., and Nelson, R. Learning dextrous manipulation skills for multifingered robot hands using the evolution strategy. *Machine Learning* (1998).
- [26] Grupen, Roderic. *Grasping and Manipulation with Multifingered Robot Hands*. PhD thesis, University of Utah, 1988.
- [27] Han, L., and Trinkle, J. Dextrous manipulation by rolling and finger gaiting. In *IEEE Int’l Conf. Robotics Automation* (May 1998), vol. 1, pp. 730 – 735.
- [28] Hong, J., Lafferriere, G., Mishra, B., and Tan, X. Fine manipulation with multifinger hands. In *IEEE Int’l Conf. Robotics Automation* (1990), pp. 1568–1573.

- [29] Huber, M. *A Hybrid Architecture for Adaptive Robot Control*. PhD thesis, U. Massachusetts, 2000.
- [30] Huber, M., and Grupen, R. Learning to coordinate controllers - reinforcement learning on a control basis. In *Proc. of the Fifteenth Int'l Joint Conference on Artificial Intelligence (1997)*, pp. 1366–1371.
- [31] Iberall, T. The nature of human prehension: Three dextrous hands in one. In *IEEE Int'l Conf. Robotics Automation (April 1987)*, pp. 396–401.
- [32] Ijspeert, J. A., Nakanishi, J., and Schaal, S. Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE Int'l Conf. Robotics Automation (2002)*.
- [33] Jameson, J., and Leifer, L. Automatic grasping: An optimization approach. *IEEE Transactions on Systems, Man, and Cybernetics smc-17*, 5 (September 1987), 806–813.
- [34] Kaplan, F., and Oudeyer, P. *Maximizing learning process: an internal reward system for development*. Springer-Verlag, 2004, pp. 259–270.
- [35] Karuppiah, D., Zhu, Z., Shenoy, P., and Riseman, E. A fault-tolerant distributed vision system architecture for object tracking in a smart room. In *International Workshop on Computer Vision Systems (July 2001)*.
- [36] Kerr, J., and Roth, B. Analysis of multifingered hands. *Int. Journal of Robotics Research* 4, 4 (1986), 3–17.
- [37] Kingdon, J. *Lowly Origin*. Princeton University Press, 2003.
- [38] Kirkpatrick, D., Mishra, B., and Yap, C. Quantitative steinitz's theorems with applications to multifingered grasping. In *20th ACM Symp. on Theory of Computing (May 1990)*, pp. 341–351.
- [39] Li, Z., and Sastry, S. Task-oriented optimal grasping by multifingered robot hands. In *IEEE Int'l Conf. Robotics Automation (March 1987)*, vol. 4, pp. 389–394.
- [40] MacKenzie, C., and Iberall, T. *The Grasping Hand*. North-Holland, 1994.
- [41] Maes, P., and Brooks, R. Learning to coordinate behaviors. In *AAAI (August 1990)*, pp. 796–802.
- [42] Mahadevan, S., and Connell, J. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence* 55, 2-3 (June 1992), 311–365.
- [43] Martin, T. B., Ambrose, R. O., Diftler, M. A., Jr., R. Platt, and Butzer, M. J. Tactile gloves for autonomous grasping with the nasa/darpa robonaut. In *IEEE Conference on Robotics and Automation (April 2004)*.

- [44] Marzke, M. *Evolution*. Amsterdam: Elsevier Science B.V., 1994, ch. 2.
- [45] Mason, M., and Salisbury, J. *Robot hands and the mechanics of manipulation*. MIT Press, 1985.
- [46] Michelman, P., and Allen, P. Forming complex dextrous manipulations from task primitives. In *IEEE Int'l Conf. Robotics Automation* (1994).
- [47] Mirtich, B., and Canny, J. Easily computable optimum grasps in 2-d and 3-d. In *IEEE Int'l Conf. Robotics Automation* (1994), pp. 739–747.
- [48] Moore, Andrew. Knowledge of knowledge and intelligent experimentation for learning control. In *Proceedings of the 1991 Seattle International Joint Conference on Neural Networks* (July 1991).
- [49] Murray, R., Li, Z., and Sastry, S. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [50] Nakamura, Y. *Advanced Robotics Redundancy and Optimization*. Addison-Wesley, 1991.
- [51] Napier, J. The prehensile movements of the human hand. *Journal Bone Joint Surgery* 38b, 4 (November 1956), 902–913.
- [52] Nguyen, V. Constructing force-closure grasps. In *IEEE Int'l Conf. Robotics Automation* (April 1986), vol. 3, pp. 1368–1373.
- [53] Nguyen, V. Constructing stable grasps in 3d. In *IEEE Int'l Conf. Robotics Automation* (March 1987), vol. 4, pp. 234–239.
- [54] Nicolescu, M., and Mataric, M. A hierarchical architecture for behavior-based robots. In *Proc. of the First Int'l Joint Conf. on Autonomous Agents and Multi-Agent Systems* (July 2002), pp. 227–233.
- [55] Piaget, J. *The Origins of Intelligence in Children*. Norton, NY, 1952.
- [56] Platt, R., Brock, O., Fagg, A. H., Karupiah, D., Rosenstein, M., Coelho, J., Huber, M., Piater, J., Wheeler, D., and Grupen, R. A framework for humanoid control and intelligence. In *Proceedings of the 2003 IEEE International Conference on Humanoid Robots* (October 2003).
- [57] Platt, R., Fagg, A. H., and Grupen, R. Extending fingertip grasping to whole body grasping. In *IEEE Int'l Conference on Robotics and Automation* (2003).
- [58] Platt, R., Fagg, A. H., and Grupen, R. Reusing schematic grasping policies. In *IEEE-RAS Int'l Conf. on Humanoid Robots* (December 2005).
- [59] Platt, R., Fagg, A. H., and Grupen, R. Improving grasp skills using schema structured learning. In *Fifth Int'l Conf. on Development and Learning* (May 2006).

- [60] Platt, R., Fagg, A. H., and Grupen, R. A. Nullspace composition of control laws for grasping. In *IEEE Int'l Conf. on Intelligent Robots and Systems* (2002).
- [61] Platt, R., Fagg, A. H., and Grupen, R. A. Manipulation gaits: Sequences of grasp control tasks. In *IEEE Int'l Conf. Robotics Automation* (2004).
- [62] Pollard, N. Synthesizing grasps from generalized prototypes. In *IEEE Int'l Conf. Robotics Automation* (1996).
- [63] Pollard, N. Closure and quality equivalence for efficient synthesis of grasps from examples. *International Journal of Robotics Research* 23, 6 (June 2004), 595–614.
- [64] Ponce, J., Sullivan, S., Sudsang, A., Boissonnat, J., and Merlet, J. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Int. J. Rob. Res.* (1996).
- [65] Popplestone, R., and Grupen, R. Symmetries in world geometry and adaptive system behaviour. In *2nd International Workshop on Algebraic Frames for the Perception-Action Cycle* (September 2000).
- [66] Puterman, M. L. *Markov Decision Processes*. John Wiley and Sons, New-York, 1994.
- [67] Ravindran, B. *An Algebraic Approach to Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts, 2004.
- [68] Ridley, M. *The Red Queen: Sex and the Evolution of Human Nature*. Penguin, 1993.
- [69] Rosenstein, M., and Barto, A. Robot weightlifting by direct policy search. In *Proc. of the Seventeenth Int'l Joint Conference on Artificial Intelligence* (2001), vol. 2, pp. 839–844.
- [70] Rus, D. In-hand dexterous manipulation of 3d piecewise-smooth objects. *International Journal of Robotics Research* (1997).
- [71] Schaal, S., and Atkeson, C. G. Memory-based robot learning. In *IEEE Int'l Conf. Robotics Automation* (1994), pp. 2928–2933.
- [72] Schlesinger, G. *Ersatzglieder und Arbeitshilfen für Kriegsbeschädigte und Unfalverletzte*. Berlin: Springer, 1919, ch. The mechanical structure of artificial limbs, pp. 21–600.
- [73] Schmidhuber, J. Curious model-building control systems. In *Int. Joint Conf. on Neural Networks* (1991), vol. 2, pp. 1458–1463.
- [74] Son, J., Howe, R., Wang, J., and Hager, G. Preliminary results on grasping with vision and touch. In *IEEE Int'l Conf. on Intelligent Robots and Systems* (November 1996), vol. 3.

- [75] Son, J. S., Cutkosky, M. R., and Howe, R. D. Comparison of contact sensor localization abilities during manipulation. In *IEEE Int'l Conf. on Intelligent Robots and Systems* (August 1995), vol. 2, pp. 96–101.
- [76] Sudsang, A., and Phoka, T. Regrasp planning for a 4-fingered hand manipulating a polygon. In *IEEE Int'l Conf. Robotics Automation* (September 2003).
- [77] Sudsang, A., and Ponce, J. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *IEEE Int'l Conf. Robotics Automation* (May 1995), vol. 2, pp. 1355–1360.
- [78] Sutton, R., and Barto, A. *Reinforcement Learning, An Introduction*. MIT Press, 1998.
- [79] Tedrake, R., Zhang, R., and Seung, S. Learning to walk in 20 minutes. In *Proc. of the Fourteenth Yale Workshop on Adaptive and Learning Systems* (2005).
- [80] Teichmann, M., and Mishra, B. Reactive algorithms for 2 and 3 finger grasping. In *International Symposium on Intelligent Robotic Systems* (July 1994).
- [81] Teichmann, M., and Mishra, B. Reactive algorithms for grasping using a modified parallel jaw gripper. In *IEEE Int'l Conf. Robotics Automation* (May 1994), vol. 3, pp. 1931–1936.
- [82] Trinkle, J., Ram, R., Farahat, A., and Stiller, P. Dexterous manipulation planning and execution of an enveloped slippery workpiece. In *IEEE Int'l Conf. Robotics Automation* (May 1993), vol. 2, pp. 442–448.
- [83] Ulam, P., and Balch, T. Niche selection for foraging tasks in multi-robot teams using reinforcement learning. In *Proc. of the 2nd Int'l Workshop on the Mathematics and Algorithms of Social Insects* (2003).
- [84] Wilson, F. *The Hand*. Random House, NY, 1998.
- [85] Yoshikawa, T. Analysis and control of robot manipulators with redundancy. In *Robotics Research*, M. Brady and R. Paul, Eds. MIT Press, 1984, pp. 735–747.
- [86] Yoshimi, B., and Allen, P. Integrating real-time vision and manipulation. In *Proc. of 13th Hawaii Int'l Conf. on System Sciences* (January 1997), vol. 5, pp. 178–187.