Shape Reconstruction in a Planar Dynamic Environment

Mark Moll Michael A. Erdmann

May 15, 2001

CMU-CS-01-107

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This work was supported in part by the National Science Foundation under grant IIS-9820180

Keywords: Tactile sensing, shape reconstruction, nonprehensile manipulation, contact kinematics

Abstract

We present a new method to reconstruct the shape of an unknown object using tactile sensors, without requiring object immobilization. Instead, sensing and nonprehensile manipulation occur simultaneously. The robot infers the shape, motion and center of mass of the object based on the motion of the contact points as measured by the tactile sensors. We present analytic results and simulation results assuming quasistatic dynamics. We prove that the shape and motion are observable in both the quasistatic and the fully dynamic case.

Contents

1 Introduction

Robotic manipulators typically cannot deal very well with objects of partially unknown shape and weight. Humans, on the other hand, seem to have few problems with manipulating objects of unknown shape and weight. For example, Klatzky et al. (1985) showed that blindfolded human observers identified 100 common objects with over 96% accuracy, in only 1 to 2 seconds for most objects. Besides recognizing shape and size, humans also use touch to determine various features such as texture, hardness, thermal qualities, weight and movement (Lederman and Browse, 1988).

It seems unlikely that people mentally keep track of the *exact* position, shape and mass properties of the objects in their environment. So somehow during the manipulation of an unknown object the tactile sensors in the human hand give enough information to find the pose and shape of that object. At the same time some mass properties of the object are inferred to determine a good grasp. These observations are an important motivation for our research. To recover the shape of an unknown object with tactile sensors either the motion of the object needs to be known, or we need many sensors (in order to reconstruct the shape and motion of the object simultaneously). Typically parts are assumed to be in a fixture. But how can we put an object in a fixture if we do not know its shape?

Let a *palm* be defined as a *planar surface covered with tactile sensors.* Suppose we have an unknown smooth convex object resting in contact with a number of palms (two palms in two dimensions, three in three dimensions); the only forces acting on the object are gravity and the normal forces. For simplicity we assume that there is no friction. We can realize these assumptions by moving the palms slowly enough so that the object is always in a local potential energy minimum. Moving the palms slowly is not sufficient to guarantee that the object is in a potential energy minimum. We also let the palms vibrate in order to break the frictional forces. As we continuously move the palms, the object will move to a (piecewise) continuously changing minimum of a potential energy well. The motion of the object will be a function of the palms' motion, the object's shape and center of mass. In this report we present methods for deriving the shape and center of mass of unknown objects in terms of palm motions and sensor values. Currently, the results are limited to planar smooth convex objects. Numerical simulation with limited resolution (in time) shows the feasibility of the analytic results.

Figure 1.1 illustrates the basic idea. There are two palms that each have one rotational degree of freedom at the point where they connect. That allows us to change the angle between palm 1 and palm 2 and between the palms and the global frame. As we change the palm angles we keep track of the contact points through tactile elements on the palms. The palms can be mounted on an air table to simulate a planar world. We are using touchpads as tactile sensors on the palms. Some preliminary experiments have shown that we can track a contact point on a touchpad with approximately 0.1mm accuracy at 100Hz. Figure 1.2 shows a possible arrangement of three palms in three dimensions. Palm 2 and palm 3 are free to rotate around their line of intersection. This axis is connected to palm 1. Palm 1 can rotate around its bottom edge.

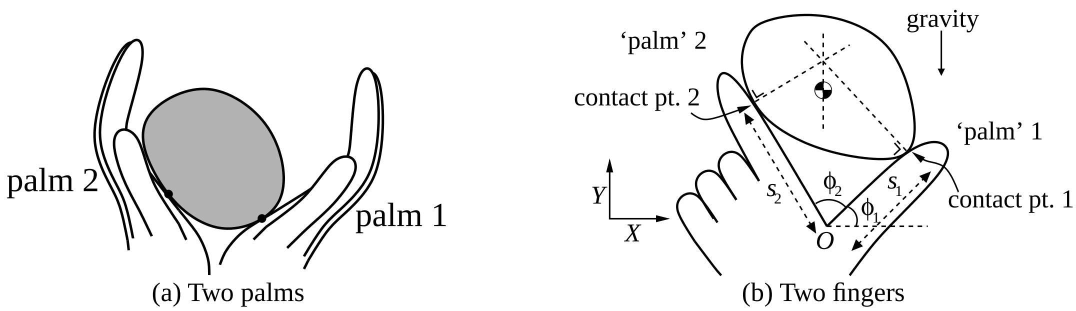


Figure 1.1: Two possible arrangements of a smooth convex object resting on palms that are covered with tactile sensors.

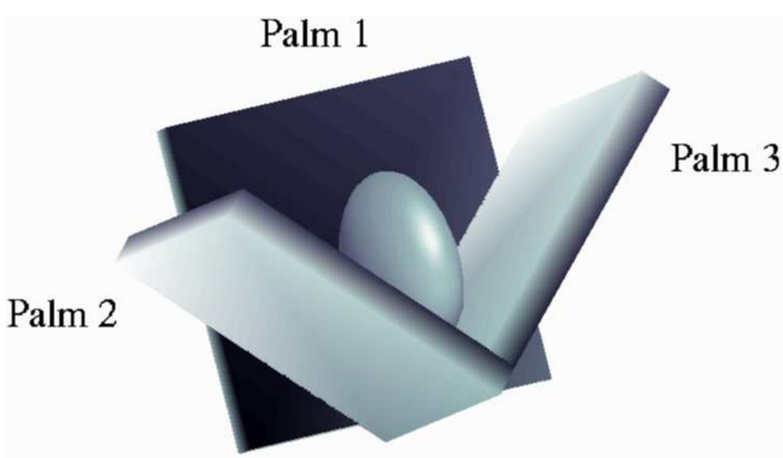


Figure 1.2: An object resting on three palms.

2 Related Work

Our research builds on many different areas in robotics. These areas can be roughly divided into four different categories: probing, nonprehensile manipulation, grasping, and tactile sensing. We can divide the related work in tactile sensing further into three subcategories: shape and pose recognition with tactile sensors, tactile exploration, and tactile sensor design. We now briefly discuss some of the research in these areas.

2.1 Probing

Shape sensing can be approached purely geometrically and algorithmically. Sensing is then often called probing. One can define different kinds of probes that correspond to abstractions of sensor devices. For instance, a *finger probe* corresponds to a robotic finger moving along a line until it contacts an object (or misses the object). The probe outcome is then the point where the probe contacted the object. Typical questions are:

- How many probes are sufficient to reconstruct the shape of an object?
- How many probes are sufficient to recognize the pose of a known object?

Often these problems are restricted to a class of shapes (such as polygons). We can relax the questions above by trying to solve for the number of probes needed for a bounded error approximation of the exact answers. Cole and Yap (1987) showed that the answer to the first question using finger probes is $3n$ for a convex n-sided polygon. Furthermore, they showed that $3n-1$ probes are necessary. If we assume that a finger probe outcome is never exactly a vertex of the polygon, then $3n$ probes are necessary. Shortly after (Cole and Yap, 1987) Dobkin et al. (1986) investigated the complexity of determining the shape and pose of convex polytopes for a variety of different probes, including probes with errors. Boissonnat and Yvinec (1992) extended the probe model of Cole and Yap: their probe outcome includes the normal at the contact point. With this probe model they showed that at most $3n-3$ probes are needed for simple *non-convex* polygons with no collinear edges. Their results can be extended to probe a set of polygons and to probe a set of polyhedra.

Li (1988) gave algorithms that reconstruct convex polygons with $3n+1$ *line probes* or with $3n-2$ *projection probes*. Line probes slide a straight line in a particular direction over the plane until it hits the object. Projection probes consist of two line probes that move in opposite directions towards each other. Lindenbaum and Bruckstein (1994) gave an approximation algorithm for arbitrary planar convex shapes using line probes. Kölzow et al. (1989) presented an approximation algorithm using projection probes, but their projection probes are defined as the length of the intersection of a line with the object. In (Lindenbaum and Bruckstein, 1991) bounds were given on the number of *parallel* probes that are necessary to recover the shape of a planar polygon. With parallel probes, k probes ($k > 1$) are performed at the same time. Skiena (1989) observed that the line probe can be generalized to a new kind of probe which is the dual of the finger probe, so that there is a one-to-one correspondence between algorithms that use finger probes and ones that use this generalized line probe.

Rao and Goldberg (1994) studied the problem of determining the shape of a convex polygon using diameter measurements from a parallel jaw gripper. They showed that there is an infinite set of polygonal shapes for a given set of diameter measurements. However, it is possible to recognize a shape from a known (finite) set of shapes. Rao and Goldberg presented sensing plans that require no more than n measurements, where n is the number of stable faces. Arkin et al. (1998) proved that finding a minimal length plan is *NP*-hard and gave a polynomial-time approximation algorithm with a good performance guarantee. Akella and Mason (1998) showed how to orient and distinguish (sets of) polygonal parts using diameter measurements.

Skiena (1989) described many different probes and many (open) problems in probing. An overview of research on probing can be found in (Romanik, 1995).

2.2 Nonprehensile Manipulation

The basic idea behind nonprehensile manipulation is that robots can manipulate objects even if the robots do not have full control over these objects. This idea was pioneered by Mason. In his Ph.D. thesis (Mason, 1982) and the companion paper (Mason, 1985) nonprehensile manipulation took the form of pushing an object in the plane to reduce uncertainty about the object's pose. Further work by Peshkin and colleagues (Peshkin and Sanderson, 1988; Wiegley et al., 1996) analyzed the pushing problem and showed how to design fences for a conveyor belt system. Lynch (1997) further built on Mason's work. In his Ph.D. thesis Lynch described a path planner for quasistatically pushing objects among obstacles. He also investigated controllability of dynamic nonprehensile manipulation such as throwing and catching a part. Lynch et al. (1998) showed how to make a robotic manipulator perform a certain juggling motion with a suitable parameterization of the shape and motion of the manipulator. Much research on juggling balls has been done in Koditschek's research group (see e.g. (Rizzi and Koditschek, 1993) and (Whitcomb et al., 1993)). Rizzi and Koditschek (1993) described a system consisting of a robot arm and a camera that can juggle two balls. In (Abell and Erdmann, 1995) nonprehensile manipulation took the (abstract) form of moving two frictionless contacts on a polygonal part in a planar gravitational field. Abell and Erdmann presented an algorithm to orient such a polygonal part by moving the contact points and performing hand-offs between two pairs of contact points.

Erdmann and Mason (1988) described sensorless manipulation within the formal framework of the pre-image methodology. In particular, Erdmann and Mason showed how to orient a planar object by a tray tilting device: first, the object is placed in a random pose in the tray and, second, the tray is tilted at a sequence of angles to bring the object in a unique pose. In (Erdmann et al., 1993) the tray tilting idea was extended to polyhedra.

One of the first papers in palmar manipulation is (Salisbury, 1987). Salisbury suggested a new approach to manipulation in which the whole robot arm is used as opposed to just the fingertips. Paljug et al. (1994) investigated the problem of multi-arm manipulation. Paljug et al. presented a nonlinear feedback scheme for simultaneous control of the trajectory of the object being manipulated as well as the contact conditions. Erdmann (1998a) showed how to manipulate a known object with two palms. He also presented methods for determining the contact modes of each palm: rolling, sliding and holding the object. Zumel (1997) described a palmar system like the one shown in figure 1.1(b), but without tactile sensors. Zumel derived sufficient conditions for orienting known polygonal parts with these palms. She also showed that an orienting plan for a polygon can be computed in $O(N^2)$ and that the length is $O(N)$, where N is the number of stable edges of the polygon.

2.3 Grasping

The problem of grasping has been widely studied. This section will not try to give a complete overview of the results in this area, but instead just mention some of the work that is most important to our problem. Much of the grasp research focuses on computing grasps that establish *force-closure* (the ability to resist external forces) and *form-closure* (a kinematic constraint condition that prevents all motion). Important work includes (Salisbury, 1982), (Cutkosky, 1985), (Fearing, 1984), (Kerr and Roth, 1986), (Mishra et al., 1987), (Montana, 1988), (Nguyen, 1988), (Trinkle et al., 1988), (Hong et al., 1990), (Markenscoff et al., 1990) and (Ponce et al., 1997). For an overview of grasp synthesis algorithms see e.g. (Shimoga, 1996).

In order to grasp an object we need to understand the kinematics of contact. Independently, Montana (1988) and Cai and Roth (1986, 1987) derived the relationship between the relative motion of two objects and the motion of their contact point. In (Montana, 1995) these results were extended to multi-fingered manipulation.

Sudsang et al. (2000) looked at the problem of manipulating three-dimensional objects with a reconfigurable gripper. The gripper consisted of two horizontal plates, of which the top one had a regular grid of actuated pins. They presented a planner that computed a sequence of pin configurations that brought an object from one configuration to another using so-called immobility regions. For each (intermediate) configuration only three pins were needed. Plans were restricted to ones where the object maintains the same set of contact points with the bottom plate. Rao et al. (1994, 1995) showed how to reorient a polyhedral object with *pivoting grasps*: the object was grasped with two hard finger contacts so that it pivoted under gravity when lifted. Often only one pivot grasp was sufficient to bring the object from one stable pose to another (provided the friction coefficient is large enough).

Trinkle and colleagues (Trinkle et al., 1993; Trinkle and Hunter, 1991; Trinkle and Paul, 1990; Trinkle et al., 1988) investigated the problem of dexterous manipulation with frictionless contact. They analyzed the problem of lifting and manipulating an object with enveloping grasps.

Yoshikawa et al. (1993) did not assume frictionless contacts and showed how to regrasp an object using quasistatic slip motion. Nagata et al. (1993) described a method of repeatedly regrasping an object to build up a model of its shape.

Teichmann and Mishra (2000) presented an algorithm that determines a good grasp for an unknown object using a parallel-jaw gripper equipped with light beam sensors. This paper presented a tight integration of sensing and manipulation. Interestingly, the object is not disturbed until good grasp points are found. Recently, Jia (2000) showed how to achieve an antipodal grasp of a curved planar object with two fingers. By rolling the fingers around the object the pose of the object is determined and then the fingers are rolled to two antipodal points.

2.4 Shape and Pose Recognition

The problem of shape and pose recognition can be stated as follows: suppose we have a known set of objects, how can we recognize one of the objects if it is in an unknown pose? For an infinite set of objects the problem is often phrased as: suppose we have a class of parametrized shapes, can we establish the parameters for an object from that class in an unknown pose? Schneiter and Sheridan (1990) developed a method for determining sensor paths to solve the first problem. In Siegel (1991) a different approach is taken: the pose of an object is determined by using an enveloping grasp. This method uses only joint angle and torque sensing.

Jia and Erdmann (1996) proposed a 'probing-style' solution: they determined possible poses for polygons from a finite set of possible poses. One can think of this finite set as the stable poses (for some sense of stable). One method determines the pose by bounding the polygon by supporting lines. The second method they propose is to sense by point sampling. They prove that solving this problem is *NP*-complete and present a polynomial time approximation algorithm.

Keren et al. (1998) proposed a method for recognizing three-dimensional objects using curve invariants. This idea was motivated by the fact that tactile sensor data often takes the form of a curve on the object. They apply their method to geometric primitives like spheres and cylinders.

Jia and Erdmann (1999) investigated the problem of determining not only the pose, but also the motion of a known object. The motion of the object is induced by having a robotic finger push the object. By tracking the contact point on the finger, they were able to recover the pose and motion using nonlinear observer theory.

2.5 Tactile Exploration

With tactile exploration the goal is to build up an accurate model of the shape of an unknown object. One early paper by Goldberg and Bajcsy (1984) described a system requiring very little information to reconstruct an unknown shape. The system consisted of a cylindrical finger covered with 133 tactile elements. The finger could translate and tap different parts of an object.

Often the unknown shape is assumed to be a member of a parametrized class of shapes, so one could argue that this is really just shape recognition. Nevertheless, with some parametrized shape models, a large variety of shapes can still be characterized. In (Fearing, 1990), for instance, results are given for recovering generalized cylinders. In (Chen et al., 1996) tactile data are fit to a general quadratic form. Finally, (Roberts, 1990) proposed a tactile exploration method for polyhedra.

Allen and Michelman (1990) presented methods for exploring shapes in three stages, from coarse to fine: grasping by containment, planar surface exploring and surface contour following. Montana (1988) described a method to estimate curvature based on a number of probes. Montana also presented a control law for contour following. Charlebois et al. (1996, 1997) introduced two different tactile exploration methods. The first method is based on rolling a finger around the object to estimate the curvature using Montana's contact equations. Charlebois et al. analyze the sensitivity of this method to noise. With the second method a B-spline surface is fitted to the contact points and normals obtained by sliding multiple fingers along an unknown object.

Marigo et al. (1997) showed how to manipulate a known polyhedral part by rolling it between the two palms of a parallel-jaw gripper. Recently, (Bicchi et al., 1999) extended these results to tactile exploration of unknown objects with a parallel-jaw gripper equipped with tactile sensors. The two palms of the gripper roll the object without slipping and track the contact points. Using tools from regularization theory they produce spline-like models that best fit the sensor data. A different approach is taken by Kaneko and Tsuji (2000), who try to recover the shape by pulling a finger over the surface. With this finger they can also probe concavities. In (Okamura and Cutkosky, 1999; Okamura et al., 1999, 2000) the emphasis is on detecting fine surface features such as bumps and ridges. Sensing is done by rolling a finger around the object. (Okamura et al., 1999) show how one can measure friction by dragging a block over a surface at different velocities, measure the forces and solve for the unknowns.

Much of our work builds forth on (Erdmann, 1998b). There, the shape of planar objects is recognized by three palms; two palms are at a fixed angle, the third palm can translate compliantly, ensuring that the object touches all three palms. Erdmann derives the shape of an unknown object with an unknown motion as a function of the sensor values. In our work we restrict the motion of the object: we assume quasistatic dynamics and we assume there is no friction. Only gravity and the contact forces are acting on the object. As a result we can recover the shape with fewer sensors. We can realize these assumptions by moving the palms slowly enough so that the object is always in a local potential energy minimum.

2.6 Tactile Sensor Design

Despite the large body of work in tactile sensing and haptics, making reliable and accurate tactile sensors has proven to be very hard. Many different designs have been proposed. This section will mention just a few. For an overview of sensing technologies, see e.g. (Howe and Cutkosky, 1992). Fearing and Binford (1988) describe a cylindrical tactile sensor to determine the curvature of convex unknown shapes. Speeter (1990) describes a tactile sensing system consisting of up to 16 arrays of 256 tactile elements that can be accessed in parallel. He discusses the implementation issues involved with using these sensors with the Utah/MIT Hand. The underlying tactile technology is based on force sensing resistors from Interlink Electronics. Choi et al. (1998) present a different design for tactile sensors for multifingered robots based on capacitive tactile elements. They compare their experimental results with Montana's contact equations (Montana, 1988).

In our own experiments we are relying on off-the-shelf components. The tactile sensors are touchpads, as found on many notebooks. Most touchpads use capacitive technology, but the ones we are using are based on force-sensing resistors. In section 3.5 we report on our findings so far.

3 Quasistatic Shape Reconstruction

In this chapter we will present a quasistatic method for reconstructing the shape of an unknown smooth convex object. The object is placed between the two palms, and we can vary the angles between the palms and the world frame. We say that the object is in *force/torque balance* if and only if all forces and torques acting on the object add up to 0. Below, we will show that if we assume that the object is always in force/torque balance and if there is no friction between the object and the palms, then we can reconstruct the shape with two palms.

Figure 3.1 shows the two inputs and the two sensor outputs. The inputs are ϕ_1, the angle between palm 1 and the X-axis of the global frame, and ϕ_2, the angle between palm 1 and 2. The tactile sensor elements return the contact points s_1 and s_2 on palm 1 and 2, respectively. Gravity acts in the negative Y direction. If the object is at rest, there is force/torque balance. In that case, since we assume there is no friction, the lines through the normal forces at the contact points and gravity acting on the center of mass intersect at a common point. In other words, the sensor values tell us where the X-coordinate of the center of mass is in the global frame. Below we will show that this constraint on the position of the center of mass and the constraints induced by the sensor values will allow us to derive an expression for the curvature at the contact points. However, this expression depends on the initial position of the center of mass. We can search for

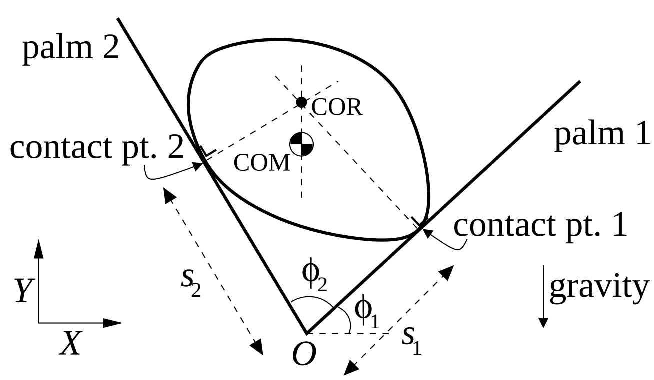


Figure 3.1: Input values are ϕ_1 and ϕ_2, output values are the contact point locations s_1 and s_2. The contact normals intersect at the center of rotation (COR), which lies on the vertical line through the center of mass (COM).

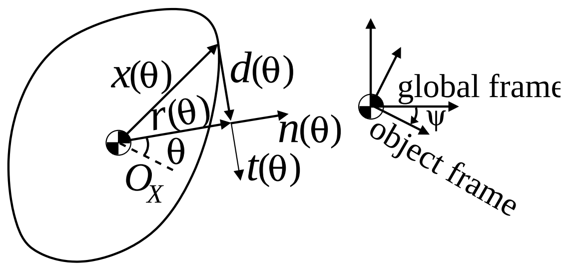


Figure 3.2: The contact support function $(r(\theta), d(\theta))$ and the object frame. O_X denotes the X-axis of the object frame.

this position with an initial pose observer that minimizes the error between what the curvature expression predicts and what the sensor values tell us.

3.1 Notation

Frames A useful tool for recovering the shape of the object will be the radius function (see e.g. (Santaló, 1976)). Figure 3.2 shows the basic idea. We assume that the object is smooth and convex. We also assume that the origin of the object frame is at the center of mass. For every angle θ there exists a point $\mathbf{x}(\theta)$ on the surface of the object such that the outward pointing normal $\mathbf{n}(\theta)$ at that point is $(\cos\theta, \sin\theta)^T$. Let the tangent $\mathbf{t}(\theta)$ be equal to $(\sin\theta, -\cos\theta)^T$ so that $[\mathbf{t}, \mathbf{n}]$ constitutes a right-handed frame. We can also define right-handed frames at the contact points with respect to the palms:

$$\begin{cases} \bar{\mathbf{n}}_1 = \big(-\sin\phi_1, \cos\phi_1\big)^T \\ \bar{\mathbf{t}}_1 = \big(\cos\phi_1, \sin\phi_1\big)^T \end{cases} \quad \text{and} \quad \begin{cases} \bar{\mathbf{n}}_2 = \big(\sin(\phi_1+\phi_2), -\cos(\phi_1+\phi_2)\big)^T \\ \bar{\mathbf{t}}_2 = \big(-\cos(\phi_1+\phi_2), -\sin(\phi_1+\phi_2)\big)^T \end{cases}$$

Note that $\bar{\mathbf{n}}_1$ and $\bar{\mathbf{n}}_2$ point into the free space between the palms. Let ψ be the angle between the object frame and the global frame, such that a rotation matrix $\mathbf{R}(\psi)$ maps a point from the object frame to the global frame:

$$\mathbf{R}(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix}$$

The object and palm frames are then related in the following way:

$$\begin{pmatrix} \bar{\mathbf{n}}_1 & \bar{\mathbf{t}}_1 \end{pmatrix} = -\mathbf{R}(\psi)\begin{pmatrix} \mathbf{n}(\theta) & \mathbf{t}(\theta) \end{pmatrix}$$
$$\begin{pmatrix} \bar{\mathbf{n}}_2 & \bar{\mathbf{t}}_2 \end{pmatrix} = -\mathbf{R}(\psi)\begin{pmatrix} \mathbf{n}(\theta+\phi_2-\pi) & \mathbf{t}(\theta+\phi_2-\pi) \end{pmatrix}$$

The different frames are shown in figure 3.3. From these relationships it follows that

$$\theta = \phi_1 - \psi - \tfrac{\pi}{2} \tag{3.1}$$

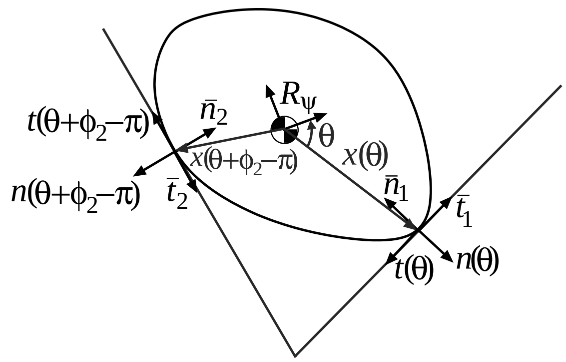


Figure 3.3: The different coordinate frames

Differentiation We will use '$\dot{}$' to represent differentiation with respect to time t and '$'$' to represent differentiation with respect to a function's parameter. So, for instance, $\dot{\mathbf{x}}(\theta) = \mathbf{x}'(\theta)\dot{\theta}$. From the Frenet formulas it follows that the parameterization velocity $v(\theta) = \|\mathbf{x}'(\theta)\|$ is the radius of curvature of the shape at the point $\mathbf{x}(\theta)$. We can write $v(\theta)$ as $-\mathbf{x}'(\theta) \cdot \mathbf{t}(\theta)$ and $\mathbf{x}'(\theta)$ as $-v(\theta)\mathbf{t}(\theta)$.

Support Functions We now define $r(\theta)$ to be the projection of the contact point $\mathbf{x}(\theta)$ onto the normal $\mathbf{n}(\theta)$:

$$r(\theta) = \mathbf{x}(\theta) \cdot \mathbf{n}(\theta)$$

This function is called a *radius function* or *support function.* For our shape recovery analysis it will be useful to define another function, $d(\theta)$, to be the projection of the contact point $\mathbf{x}(\theta)$ onto the tangent $\mathbf{t}(\theta)$:

$$d(\theta) = \mathbf{x}(\theta) \cdot \mathbf{t}(\theta)$$

We will refer to the pair $(r(\theta), d(\theta))$ as a *contact support function.* The goal is now to derive a solution for $\mathbf{x}(\theta)$ as we change the palm angles ϕ_1 and ϕ_2.

One final bit of notation we need is a generalization of the contact support function, which we will define as a projection of the vector between the two contact points. We define *the generalized contact support function relative to contact point 1* as:

$$\tilde{r}_1(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta + \phi_2 - \pi)\big) \cdot \mathbf{n}(\theta) \tag{3.2}$$

$$\tilde{d}_1(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta + \phi_2 - \pi)\big) \cdot \mathbf{t}(\theta) \tag{3.3}$$

Similarly, we can define *the generalized contact support function relative to contact point 2* as:

$$\tilde{r}_2(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta + \phi_2 - \pi)\big) \cdot \mathbf{n}(\theta + \phi_2 - \pi) \tag{3.4}$$

$$\tilde{d}_2(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta + \phi_2 - \pi)\big) \cdot \mathbf{t}(\theta + \phi_2 - \pi) \tag{3.5}$$

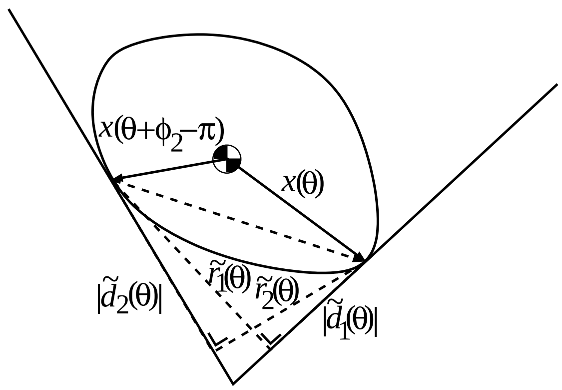


Figure 3.4: The generalized contact support functions.

Below we drop the function arguments where it does not lead to confusion, and instead use subscripts '1' and '2' to denote the contact point on palm 1 and 2. So we will write e.g. $\mathbf{Rn}_2$ for $\mathbf{R}(\psi)n(\theta + \phi_2 - \pi)$.

The generalized contact support functions have the property that they can be expressed directly in terms of the palm angles and sensor values (assuming the object is in two-point contact):

$$\begin{cases} \tilde{r}_1 = s_2 \sin\phi_2 \\ \tilde{d}_1 = s_2 \cos\phi_2 - s_1 \end{cases} \quad \text{or} \quad \begin{cases} \tilde{r}_2 = -s_1 \sin\phi_2 \\ \tilde{d}_2 = s_1 \cos\phi_2 - s_2 \end{cases} \tag{3.6}$$

These equalities can be obtained by inspection from figures 3.1 and 3.4. We can also obtain these equalities analytically. First, we write the constraints that two-point contact induces as

$$s_1\bar{\mathbf{t}}_1 = \mathbf{c}_m + \mathbf{R}\mathbf{x}_1 \tag{3.7}$$

$$-s_2\bar{\mathbf{t}}_2 = \mathbf{c}_m + \mathbf{R}\mathbf{x}_2, \tag{3.8}$$

where $\mathbf{c}_m$ is the position of the center of mass. Next, we can eliminate $\mathbf{c}_m$ from these equations and write

$$\mathbf{R}\,(\mathbf{x}_1 - \mathbf{x}_2) = s_1\bar{\mathbf{t}}_1 + s_2\bar{\mathbf{t}}_2 \tag{3.9}$$

The expressions in 3.6 then follow by computing the dot product on both sides of expression 3.9 with the palm normals and tangents.

Above we have shown that although the generalized contact support functions were defined in the object frame, we can also express them directly in terms of sensor values and palm angles. This is useful because it can be shown (Erdmann, 1998b) that the radii of curvature at the contact points can be written in terms of the generalized contact support functions as

$$v_1 = -\frac{\tilde{r}_2' + \tilde{d}_2}{\sin\phi_2} \tag{3.10}$$

$$v_2 = -\frac{\tilde{r}_1' + \tilde{d}_1}{\sin\phi_2} \tag{3.11}$$

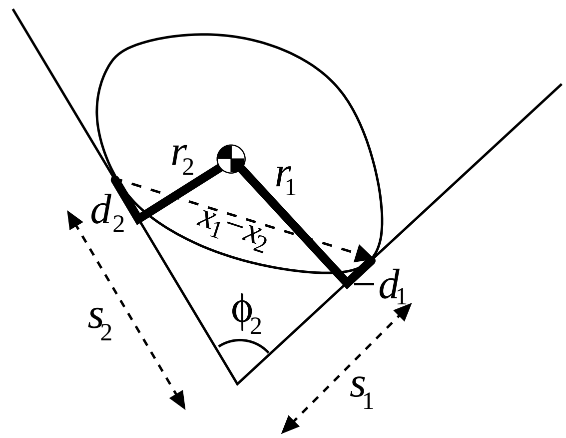


Figure 3.5: The dependencies between sensor values, the support function and the angle between the palms when the object makes two-point contact

The derivation of these expressions is included in the appendix. Note that these expressions are *not* sufficient to observe the local shape, even though the generalized support functions are directly observable. To observe the shape we will also need an expression for the *time* derivative of the function parameters. This is the topic of section 3.3.

Equation 3.9 can also be rewritten in terms of the contact support function:

$$-(r_1\bar{\mathbf{n}}_1 + d_1\bar{\mathbf{t}}_1) + (r_2\bar{\mathbf{n}}_2 + d_2\bar{\mathbf{t}}_2) = s_1\bar{\mathbf{t}}_1 + s_2\bar{\mathbf{t}}_2 \tag{3.12}$$

Solving this constraint for d_1 and d_2 we get:

$$d_1 = \frac{r_1 \cos\phi_2 + r_2}{\sin\phi_2} - s_1 \tag{3.13}$$

$$d_2 = -\frac{r_2 \cos\phi_2 + r_1}{\sin\phi_2} + s_2 \tag{3.14}$$

See also figure 3.5. Note that by construction $r'(\theta) = -d(\theta)$. So a solution for $r(\theta)$ can be used in two ways to arrive at a solution for $d(\theta)$: (1) using the property $d(\theta) = -r'(\theta)$ of the radius function, or (2) using expressions 3.13 and 3.14. In other words, to recover the shape it is sufficient to reconstruct the radius function.

3.2 Geometric Interpretation of Force/Torque Balance and Shape

Before we will derive the equations for the local shape of the object as a function of the palm angles and sensor values, we will take a step back and analyze how stable poses relate to palm angles and orientation. This is important for deciding whether we can *globally* reconstruct the shape. In other words, we would like to answer the question: is it always possible to reconstruct the entire shape? To answer this question it will be useful to consider the motion of the center of mass and the center of rotation relative to each other.

For force/torque balance the lines through the normals at the contact points and the gravity force acting on the center of mass intersect at one point called the *center of rotation* (see figure 3.1). The lines through the normals can be described by:

$$\ell_1 : q_1 \mapsto s_1\bar{\mathbf{t}}_1 + q_1\bar{\mathbf{n}}_1 \tag{3.15}$$

$$\ell_2 : q_2 \mapsto -s_2\bar{\mathbf{t}}_2 + q_2\bar{\mathbf{n}}_2 \tag{3.16}$$

These lines intersect if and only if

$$q_1 = \tfrac{s_2 - s_1 \cos\phi_2}{\sin\phi_2} \text{ and } q_2 = \tfrac{s_1 - s_2\cos\phi_2}{\sin\phi_2}.$$

Using the generalized contact support functions we can simplify this to $q_1 = -\tilde{d}_2/\sin\phi_2$ and $q_2 = -\tilde{d}_1/\sin\phi_2$. So we can write the following equations for the center of mass, $\mathbf{c}_m$, and the center or rotation, $\mathbf{c}_r$:

$$\begin{aligned}
\mathbf{c}_m(\psi, \phi_1, \phi_2) &= s_1\bar{\mathbf{t}}_1 - \mathbf{R}\mathbf{x}_1 \\
&= -\tilde{r}_2\bar{\mathbf{t}}_1/\sin\phi_2 - \mathbf{R}\mathbf{x}_1 && (3.17)\\
\mathbf{c}_r(\psi, \phi_1, \phi_2) &= s_1\bar{\mathbf{t}}_1 + q_1\bar{\mathbf{n}}_1 \\
&= s_1\bar{\mathbf{t}}_1 - \tilde{d}_2\bar{\mathbf{n}}_1/\sin\phi_2 \\
&= -(\tilde{r}_2\bar{\mathbf{t}}_1 + \tilde{d}_2\bar{\mathbf{n}}_1)/\sin\phi_2 && (3.18)
\end{aligned}$$

In the appendix it is shown that the partial derivatives of $\mathbf{c}_m$ and $\mathbf{c}_r$ can be written as

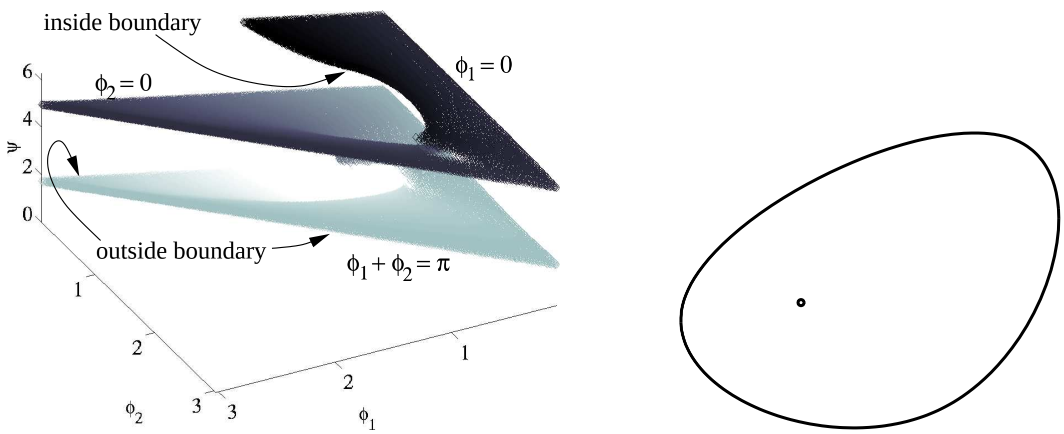
$$\frac{\partial \mathbf{c}_m}{\partial\psi} = -\frac{\tilde{d}_2\bar{\mathbf{t}}_1}{\sin\phi_2} - \Big(\frac{\partial}{\partial\psi}\mathbf{R}\Big)\mathbf{x}_1, \tag{3.19}$$

$$\frac{\partial \mathbf{c}_r}{\partial\psi} = -\big(v_1\bar{\mathbf{n}}_2 - v_2\bar{\mathbf{n}}_1 - \tilde{r}_2\bar{\mathbf{n}}_1 + \tilde{d}_2\bar{\mathbf{t}}_1\big)/\sin\phi_2. \tag{3.20}$$

and that we can rewrite equation 3.19 in terms of the relative distance between the center of mass and the center of rotation:

$$\frac{\partial \mathbf{c}_m}{\partial\psi} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}(\mathbf{c}_m - \mathbf{c}_r)\,. \tag{3.21}$$

With the results above we can easily describe all the stable poses of an object. We define a *stable pose* as a local minimum of the potential energy function with respect to ψ. The potential energy of an object in two-point contact with the palms is simply the Y coordinate of $\mathbf{c}_m$, which can be written as $\mathbf{c}_m \cdot \binom{0}{1}$. At a local minimum the first derivative with respect to ψ of this expression will be equal to 0. We can write this condition using equation 3.21 as $(\mathbf{c}_m - \mathbf{c}_r)\cdot\binom{1}{0}$. In other words, at the minima of the potential energy function the X coordinates of $\mathbf{c}_m$ and $\mathbf{c}_r$



(a) The stable pose surface in configuration space (b) The corresponding object shape

Figure 3.6: Stable poses for a particular shape. Note that only one 2π period of ψ is shown in (a) and the surface extends from $-\infty$ to ∞ in the ψ direction, i.e., the inside and outside boundary do not meet.

have to be equal. Since we assume that the object is always in force/torque balance and, hence, at a minimum of the potential energy function, we can directly observe the X coordinate of the center of mass. Or, equivalently, we can directly observe the projection onto the X-axis of the vector from the center of mass to contact point 1 by using expressions 3.17 and 3.18:

$$(\mathbf{c}_m - \mathbf{c}_r) \cdot \binom{1}{0} = 0 \qquad \Rightarrow \qquad (\mathbf{R}\mathbf{x}_1) \cdot \binom{1}{0} = -\tilde{d}_2 \tfrac{\sin\phi_1}{\sin\phi_2} \tag{3.22}$$

For a local minimum of the potential energy function the Y coordinate of the second partial derivative of $\mathbf{c}_m$ with respect to ψ has to be greater than 0, i.e., $\frac{\partial}{\partial\psi}\big((\mathbf{c}_m - \mathbf{c}_r) \cdot \binom{1}{0}\big) > 0$. The stable poses induce a two-dimensional subset of the (ϕ_1, ϕ_2, ψ)-configuration space. Figure 3.6 shows all the stable poses for a given shape. These stable configurations form a spiraling surface. From figure 3.6 it follows that for this particular shape it is indeed possible to reconstruct the entire shape, because there exists a path on the surface of stable configurations between any two stable configurations. Below we will show that this is true in general, i.e., we will prove that:

> For any smooth convex shape there exists a surface of stable configurations such that we can bring the object from any orientation to any other orientation by moving along the surface.

We prove this statement by considering the boundaries of the stable configuration surface. Let us define the *inside boundary* as those configurations where both the first and second derivative with respect to ψ of the potential energy function vanish. Using expressions 3.17– 3.22 we can write these two constraints as:

$$(\mathbf{c}_m - \mathbf{c}_r) \cdot \binom{1}{0} = -\tilde{d}_2 \sin\phi_1 / \sin\phi_2 - \binom{\cos\psi}{-\sin\psi} \cdot \mathbf{x}_1 = 0, \tag{3.23}$$

$$\frac{\partial}{\partial\psi}\left((\mathbf{c}_m - \mathbf{c}_r) \cdot \binom{1}{0}\right) = \left(v_1 \sin(\phi_1 + \phi_2) + (v_2 + \tilde{r}_2) \sin\phi_1\right) / \sin\phi_2 + \binom{\sin\psi}{\cos\psi} \cdot \mathbf{x}_1 = 0. \tag{3.24}$$

The *outside boundary* of the stable configuration surface is determined by limits on the palm angles: $\phi_1, \phi_2 > 0$ and $\phi_1 + \phi_2 < \pi$. These limits can be geometrically interpreted as follows:

$\boldsymbol{\phi_1 = 0^+,\ 0 < \phi_2 < \pi}$**:** When $\phi_2 = \pi^-$, both palms are nearly horizontal, pointing in nearly opposite directions. In the limit, as ϕ_2 approaches π, $s_1 = s_2 = 0$, and the contact point is a minimum of the radius function (since the center of mass is right above the contact point and therefore $r'(\theta) = -d(\theta) = 0$). As ϕ_2 decreases, contact point 2 covers nearly half the shape. As ϕ_2 approaches 0, the palms form an antipodal grasp. The contact points are then at a minimum of the diameter function $D(\theta) \stackrel{\text{def}}{\equiv} r(\theta) + r(\theta - \pi)$.

$\boldsymbol{\phi_2 = 0^+,\ 0 < \phi_1 < \pi}$**:** When $\phi_1 = 0^+$, this boundary connects to the previous one. As ϕ_1 increases, the palms maintain an antipodal grasp, so the contact points do not change. As ϕ_1 approaches π, palm 1 and 2 both point to the left.

$\boldsymbol{0 < \phi_1 < \pi,\ \phi_2 = \pi^- - \phi_1}$**:** This case is symmetrical to the first one. Now contact point 1 covers nearly half the shape.

From this geometric interpretation it is clear that we can bring the object to any orientation by moving along these outside boundaries. More importantly, by following these boundaries we can reconstruct the entire shape. However, the boundary itself is not part of the surface, so the question is whether there always exist stable configurations arbitrarily close to the outside boundary. The answer is "yes", *provided the inside and outside boundary do not meet*. Let a *generic smooth convex object* be defined as a smooth convex object in general position such that none of the singular cases described below apply. Below we will prove that for a generic smooth convex shape the inside boundary *never* (with probability 1) meets the outside boundary and, hence, there exists a path on the surface connecting any two orientations of the object. For each outside boundary condition we can analyze what conditions must hold for the inside boundary to meet the outside boundary:

$\boldsymbol{\phi_1 = 0^+}$**:** Without loss of generality, we can assume that $\psi = 0$, in this case and below. If $\phi_1 = 0^+$, then equations 3.23 and 3.24 simplify to $\mathbf{x}_1 \cdot \binom{1}{0} = 0$ and $\mathbf{x}_1 \cdot \binom{0}{1} = -v_1$. In other words, contact point 1 is right below the center of mass. Furthermore, if we draw a circle with radius equal to v_1 and tangent to the contact point, its center coincides with the center of mass. For each point on the shape we can determine the center of the circle with radius v_1 and tangent to that point. The locus of these circle centers forms a curve. For a generic smooth object the center of mass is not on this curve.

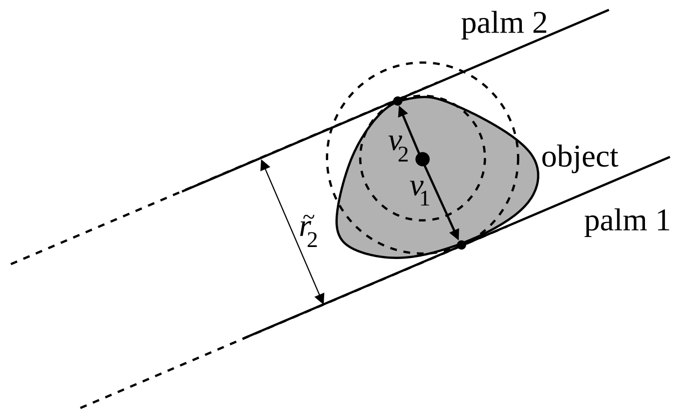


Figure 3.7: For $\phi_2 = 0$ the inside and outside boundary meet if the sum of the radii of curvature of two antipodal points is equal to the distance between these points.

$\boldsymbol{\phi_2 = 0^+}$**:** Since the palms make an antipodal grasp, the possible contact points on the object are restricted to a finite set. Now for the inside boundary to meet the outside boundary we have the following condition:

$$\lim_{\phi_2 \downarrow 0} \Big((v_1 \sin(\phi_1 + \phi_2) + (v_2 + \tilde{r}_2) \sin \phi_1) / \sin \phi_2 + \mathbf{x}_1 \cdot \tbinom{0}{1} \Big) = 0. \tag{3.25}$$

This limit only converges if

$$\lim_{\phi_2 \downarrow 0} \big(v_1 \sin(\phi_1 + \phi_2) + (v_2 + \tilde{r}_2) \sin \phi_1 \big) = 0. \tag{3.26}$$

$-\tilde{r}_2$ will converge to the distance between the contact points. So expression 3.26 converges if the sum of the radii of curvature at the contact points is equal to the distance between the contact points. A geometric interpretation of this constraint is shown in figure 3.7. A generic smooth object does not have such a pair of antipodal points.

$\boldsymbol{\phi_1 + \phi_2 = \pi}$**:** This case is, as before, symmetrical to the first one.

For practical reasons it is undesirable to plan paths on the surface that are close to the boundary. First, we would need palms of infinite length for an antipodal grasp. We can get around this by removing the joint between the palms, thereby allowing the palms to move freely. For the analysis it is not essential that the palms are connected; the analysis just depends on relative orientations of the palms. The second reason that moving on the boundary of the stable configuration surface is undesirable is that for a wide angle between the palms, we are relying heavily on the assumption that there is no friction. Even the slightest amount of friction will throw off our estimate of the X-coordinate of the center of mass.

Figure 3.6 also shows that for almost all combinations or ϕ_1 and ϕ_2 there exist exactly two stable poses. However, it is possible that for a given ϕ_1 and ϕ_2 there are many stable poses. We

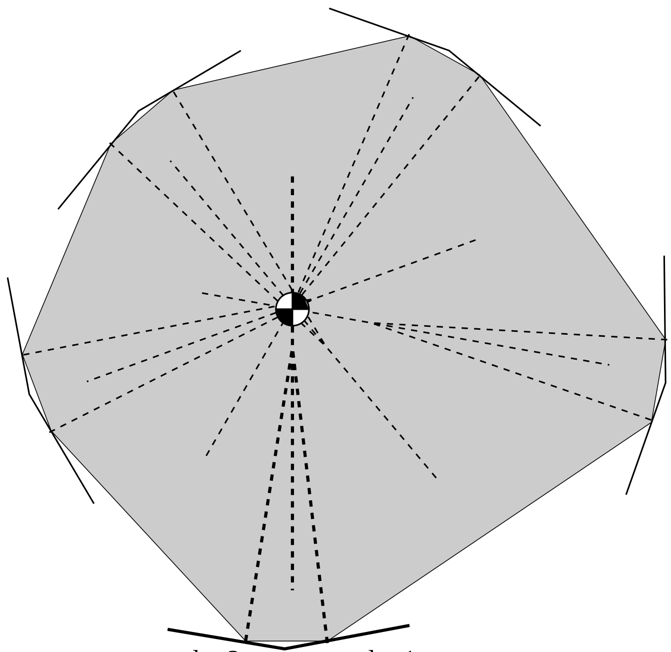


Figure 3.8: Many stable poses are possible for a given palm configuration that produce the same sensor readings.

can construct a shape with an arbitrary number of stable poses for a given palm configuration in the following way. Consider the arrangement of lines through the contact normals and the line through the center of mass (along the direction of gravity). We can rotate this arrangement around the center of mass and then translate along the line through the center of mass to create a new stable configuration. We pick the new contact points to be at the same distance from the intersection of the lines as in the original arrangement. This means that this new configuration produces the same sensor readings as well. We can repeat this process, picking a different amount of translation at each step to create an arbitrary number of stable configurations. We can create a smooth convex shape that has these stable poses in the following way. Consider the convex polygon that has the contact points of all these stable poses as vertices. If such a polygon does not exist we remove the poses that cause the concavities. The arrangement of lines described above corresponds to critical points of the potential energy function. To make sure that all the critical points are local minima we need to consider the second derivative of the potential energy function (see equation 3.24). For each contact point we can pick the radius of curvature to be arbitrarily large such that the second derivative is greater than 0. We can locally deform the polygon around each vertex such that at the contact point the radius of curvature is as desired and the shape remains convex. Figure 3.8 illustrates this geometric construction. Since the polygon induced by the contact points is convex, there exists a smooth convex shape with these stable poses.

3.3 Recovering Shape

We can write the derivative $\dot{\mathbf{x}}$ of the function $\mathbf{x}(\theta)$ that describes the shape as $\dot{\theta} v(\theta)\mathbf{t}(\theta)$. So if we can solve for $\dot{\theta}$, $v(\theta)$ and the initial conditions, we can find the shape by integrating $\dot{\mathbf{x}}$. In other words, if we can observe the curvature at the contact points and the rotational velocity of the object, we can recover the shape of an unknown object. By differentiating the generalized support functions with respect to time, we can rewrite expressions 3.10 and 3.11 as

$$v_1 = -\frac{\dot{\tilde{r}}_2 + (\dot{\theta} + \dot{\phi}_2)\tilde{d}_2}{\dot{\theta} \sin\phi_2} \tag{3.27}$$

$$v_2 = -\frac{\dot{\tilde{r}}_1 + \dot{\theta}\tilde{d}_1}{(\dot{\theta} + \dot{\phi}_2)\sin\phi_2} \tag{3.28}$$

See the appendix for details. So in order to observe the curvature at the contact points, we need to derive an expression for the rotational velocity of the object that depends only on palm angles, sensor values and their derivatives. Note that we can not observe the curvature at the two contact points if $\dot{\theta} = 0$ or $\dot{\theta} + \dot{\phi}_2 = 0$, respectively.

We can recover the rotational velocity by looking at the constraint the force/torque balance imposes on the motion of the object. Recall equation 3.22:

$$(\mathbf{R}\mathbf{x}_1) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -\tilde{d}_2 \tfrac{\sin\phi_1}{\sin\phi_2} \tag{3.29}$$

The left-hand side of this equation can be rewritten as

$$(\mathbf{R}\mathbf{x}_1) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = (\mathbf{R}(r_1\mathbf{n}_1 + d_1\mathbf{t}_1)) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{3.30}$$

$$= r_1 \sin\phi_1 - d_1 \cos\phi_1 \tag{3.31}$$

This expression (implicitly) depends on the orientation of the object. In the appendix it is shown how by differentiating this expression and the right-hand side of equation 3.29 we can obtain the following expression for the rotational velocity of the object:

$$\dot{\psi} = \frac{\dot{\tilde{r}}_2 \cos\phi_1 - \dot{\tilde{d}}_2 \sin\phi_1 + \tilde{d}_2\dot{\phi}_2 \frac{\sin\phi_{12}}{\sin\phi_2}}{r_1 \sin\phi_{12} + (r_2 + \tilde{r}_2)\sin\phi_1 + \tilde{d}_2\cos\phi_1}, \tag{3.32}$$

where $\phi_{12} = \phi_1 + \phi_2$. This expression for $\dot{\psi}$ depends on the control inputs, the sensor values, their derivatives and the current values of radius function at the contact points. The system of differential equations describing the (sensed) shape and motion can be summarized as follows:

$$\dot{r}_1 = -d_1(\dot{\phi}_1 - \dot{\psi}) \tag{3.33}$$

$$\dot{r}_2 = -d_2(\dot{\phi}_{12} - \dot{\psi}) \tag{3.34}$$

$$\dot{\psi} = \frac{\dot{\tilde{r}}_2 \cos\phi_1 - \dot{\tilde{d}}_2 \sin\phi_1 + \tilde{d}_2\dot{\phi}_2 \frac{\sin\phi_{12}}{\sin\phi_2}}{r_1 \sin\phi_{12} + (r_2 + \tilde{r}_2)\sin\phi_1 + \tilde{d}_2\cos\phi_1} \tag{3.35}$$

So far we have assumed that we have sensor data that is continuous and without any error. In practice sensors will be discrete, both in time and space, and there will also be errors. We would like to recover the shape of an unknown object in such a setting as well. There are two main directly observable error terms at each time step. First, one can check the error in the force/torque balance constraint (equation 3.29). Let that error be denoted by e_f. So at $t = t_i$, $i = 1, 2, \ldots$, $e_f(t_i)$ is equal to

$$e_f(t_i) = \left[(\mathbf{R}(\hat{\psi})\hat{\mathbf{x}}_1) \cdot \binom{1}{0} + \tilde{d}_2 \tfrac{\sin\phi_1}{\sin\phi_2}\right]\Big|_{t=t_i}, \tag{3.36}$$

where '$\hat{\ }$' denotes the estimated value of a variable. The second observable error is the error in the two-point contact constraint (equation 3.9). Let this error be denoted by $\mathbf{e}_c$. In other words,

$$\mathbf{e}_c(t_i) = \left[\mathbf{R}(\hat{\psi})(\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2) - s_1\bar{\mathbf{t}}_1 - s_2\bar{\mathbf{t}}_2\right]\Big|_{t=t_i} \tag{3.37}$$

Our program searches for the initial conditions of our system by minimizing the sum of all locally observable errors.

In our current implementation we use a fourth-order Adams-Bashforth-Moulton predictor-corrector method to integrate equations 3.33–3.35. This high-order method tends to filter out most of the noise and numerical errors. In our simulation results hardly any error accumulates during integration (see section 3.4).

3.4 Simulation Results

Figure 3.9 shows an example of the shape reconstruction process. The results are based on numerical simulation. 270 measurements were used to reconstruct the shape. In each frame the part of the shape that has been observed up to that point in time is shown. Also drawn are the contact points, the center of mass, and the palms. Notice how the (observed) shape sometimes intersects the palms. This means that there is a conflict between the currently observed shape and the previously observed shape, which could potentially be used to guide the search for initial conditions. The motion of the palms is open-loop. Initially palm 1 and palm 2 are nearly horizontal; the object is squeezed (but without friction!) between the palms. The motion of the palms can roughly be described as a sequence of squeeze-and-rotate motions and motions where one of the palms stays put and the other palm opens up. Notice how in the penultimate frame the simulator misgauges the shape, but has recovered in the last frame.

In figure 3.10 the differences are shown between the reconstructed and actual shape and motion of the object. One can not directly observe the errors in $\dot{\psi}$ and ψ, but one *can* observe the error in the X-coordinate of the center of mass and the error in the two-point contact constraint. These errors are shown in figure 3.11. Note that the big errors in error plot 3.11(d) occur at the same times as when the rotational speed of the object was misgauged. This suggests that our system could at least detect where the observed shape will be wrong. It is possible that the system could even detect that such a situation is approaching and maybe even prevent it by changing the motion of the palms. Also, the error in the norm of the contact point vector is very small, but does not appear to be completely random, suggesting that there is still room for improvement in the integration step.

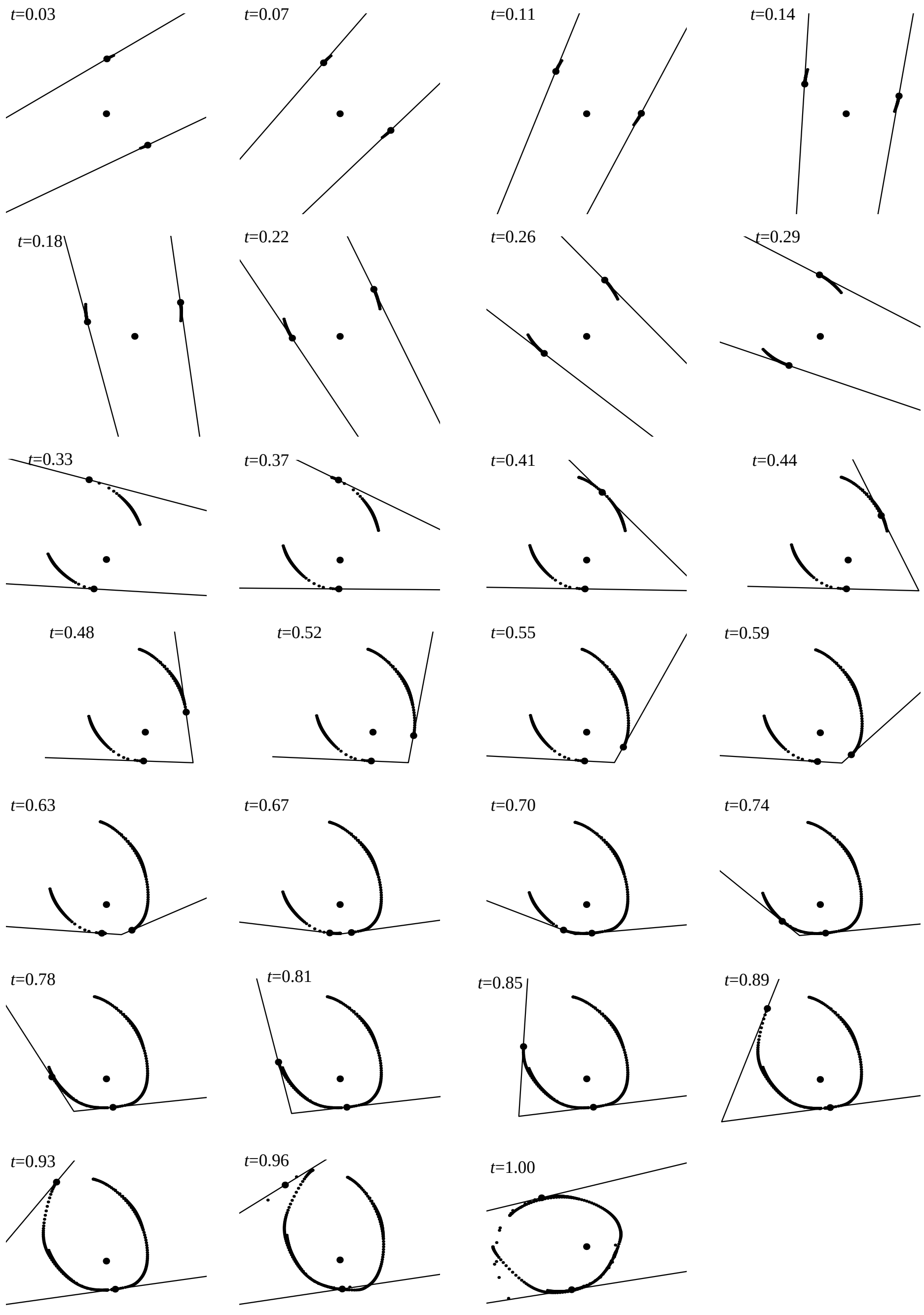
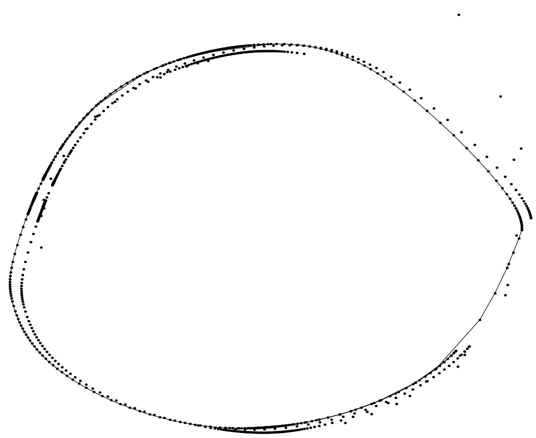
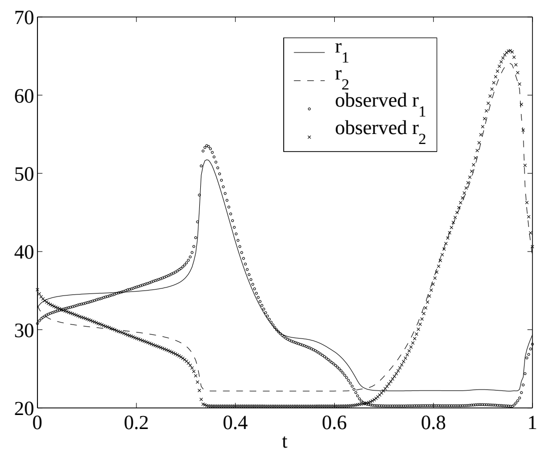


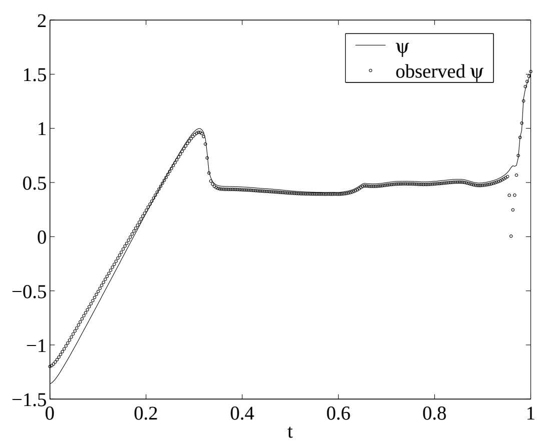
Figure 3.9: The frames show the reconstructed shape after 10, 20,…,270 measurements. The three large dots indicate the center of mass and the contact points at each time, the smaller dots show the part of the shape that has been reconstructed at that time.



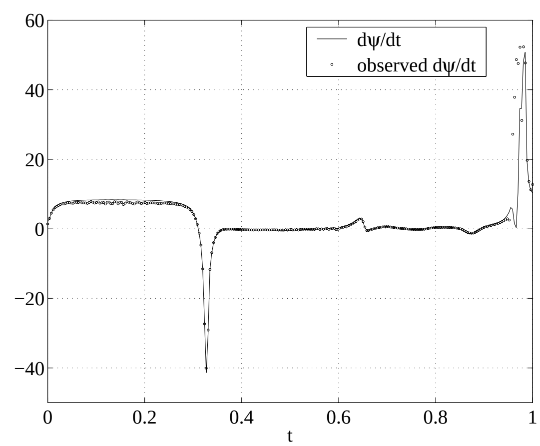
(a) The actual shape and the observed shape.



(b) The actual and observed values of the radius function.

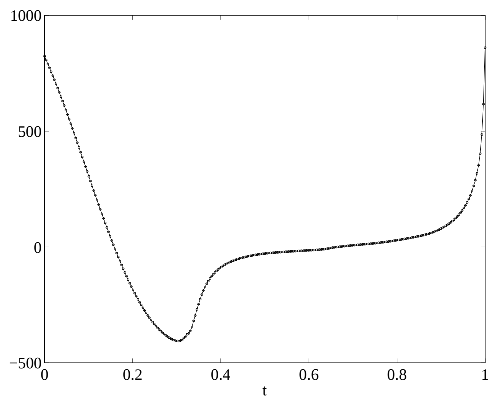


(c) The actual and observed orientation of the object.

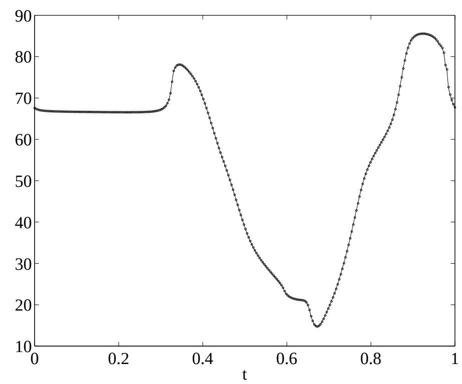


(d) The actual and observed rotational velocity of the object.

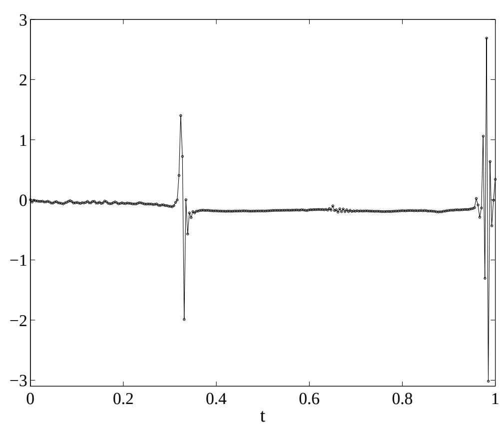
Figure 3.10: The differences between the actual and observed shape.



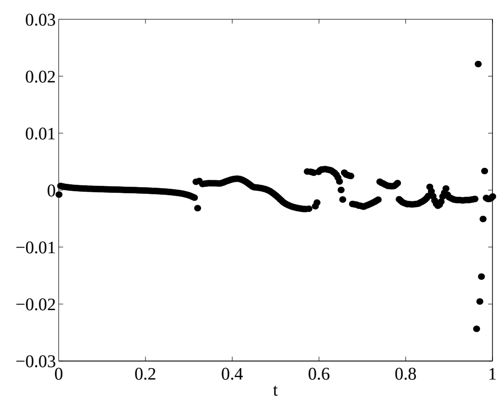
(a) The real and observed X-coordinate of the center of mass



(b) The real and observed norm of the vector between the contact points

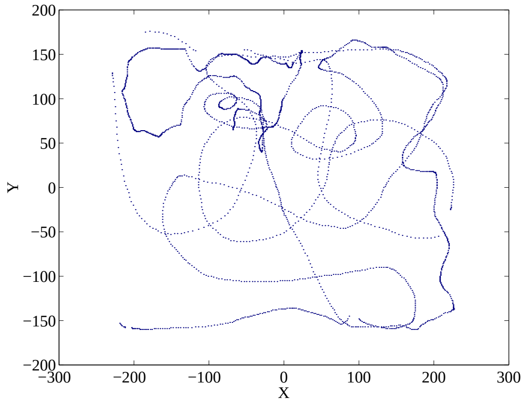


(c) The error in the X-coordinate of the center of mass

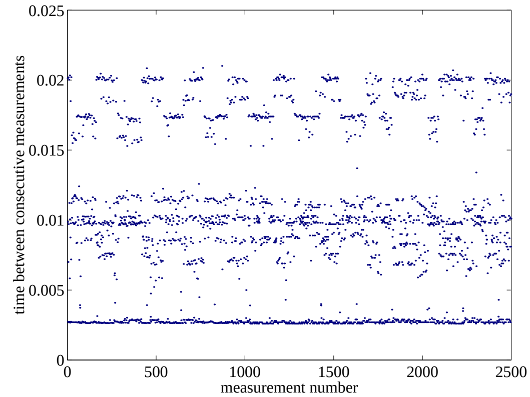


(d) The error in the norm of the vector between the contact points

Figure 3.11: The observable error for the reconstructed shape.



(a) The contact point of a marble being rolled on a touchpad. X and Y are measured in 'tactels.'



(b) The speed at which measurements are reported. The average time between measurements is 0.010 seconds, the standard deviation is 0.0050.

Figure 3.12: Resolution and sensing frequency of the VersaPad

3.5 Experimental Results

The tactile sensors we are using are touchpads made by Interlink Electronics (http://www.interlinkelec.com). These touchpads are most commonly used in notebook computers. They use so-called force sensing resistors to measure the location and the applied pressure at the contact point. One of the advantages of this technology, according to Interlink, is that it does not suffer as much from electrostatic contamination as capacitance-based touchpads. If there is more than one contact point, the pad returns the centroid. The physical pad has a resolution of 1000 counts per inch (CPI) in the X and Y direction, but the firmware limits the resolution to 200 CPI. It can report 128 pressure levels. The pad measures $55.5 \times 39.5\text{mm}^2$. Sensor data can be read out through a RS232 serial port connection.

Figure 3.12 shows the results of a simple test to establish the feasibility of the touchpad. The test consisted of rolling a marble around on the touchpad and tracking the contact point. Figure 3.12(a) shows the 'curve' traced out by the contact point. Figure 3.12(b) shows how fast we can get sensor readings from the touchpad. Notice how the times between measurements are roughly centered around 3 bands. This could be related to the way our driver polls the touchpad for data; further tweaking might increase the speed at which measurements are reported.

For the actual palms, we are using an Adept SCARA-type arm to control two planar surfaces connected with hinges. The Adept robot arm holds the endpoint of one palm. The endpoint of the other palm is attached to a fixed base. Figure 3.13 shows the basic idea (not to scale). The touchpads are mounted on the two surfaces and connected to a PC. It is important to realize that this experimental setup is just meant to be a proof of concept. Mechanically the sensing mechanism can be much simpler. More importantly, the palms do not need to be connected at all: the analysis only depends on the relative angle between the palms and the world. So in theory,

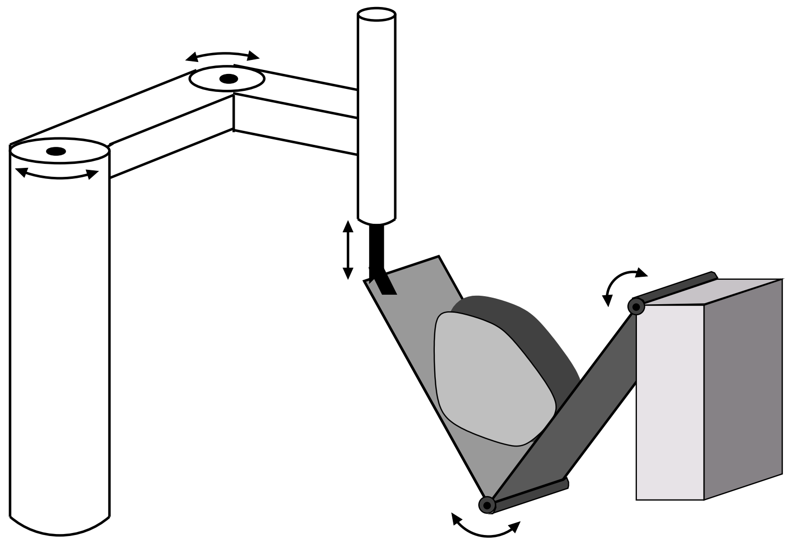


Figure 3.13: Setup of the palms

our proposed sensing strategies can also be applied to a robot hand equipped with tactile sensors. Our actual experimental setup is shown in figure 3.14. The white arrows on the object and the palms are tracked by an Adept vision system in order to establish ‘ground truth’, which can be compared with the shape and motion inferred from the tactile data.

Figures 3.15(a) and 3.15(b) show the reconstructed shape and motion, respectively. The observed motion is far from perfect, but the observed shape comes close to the actual shape. This seems to suggest that the system of differential equations 3.33–3.35 is fairly stable in this case, i.e., the errors in the motion did not cause the radius function to shoot off to infinity. The palms made back-and-forth motions in order to cover the shape several times. This means that we can prune those parts of the reconstructed shape that have been touched only once. For instance, in figure 3.15(a) we can eliminate the sparse point distribution in the top right and bottom middle. To determine which parts to eliminate one can draw a curve interpolating the points $(t_i, r(\theta(t_i)))$, $i = 1, \ldots$ The points we can eliminate are those for which $(t_i, r(\theta(t_i)))$ is the only intersection with the line $t = t_i$.

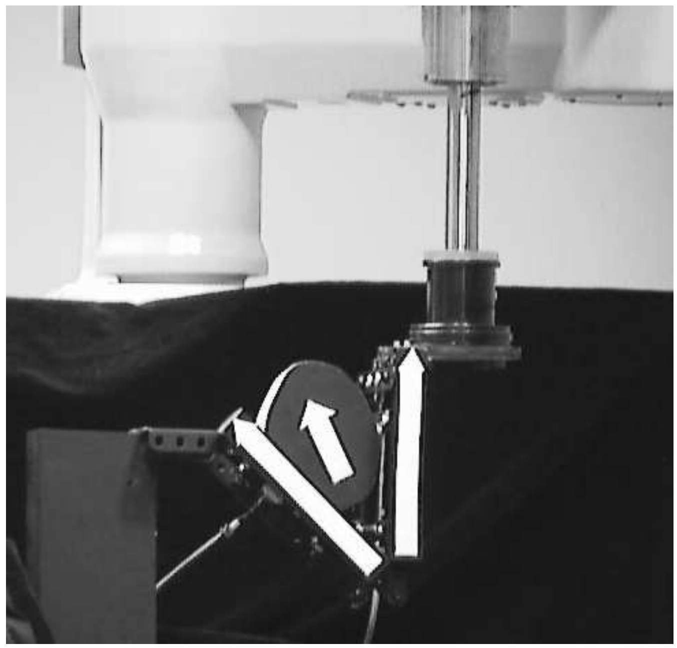
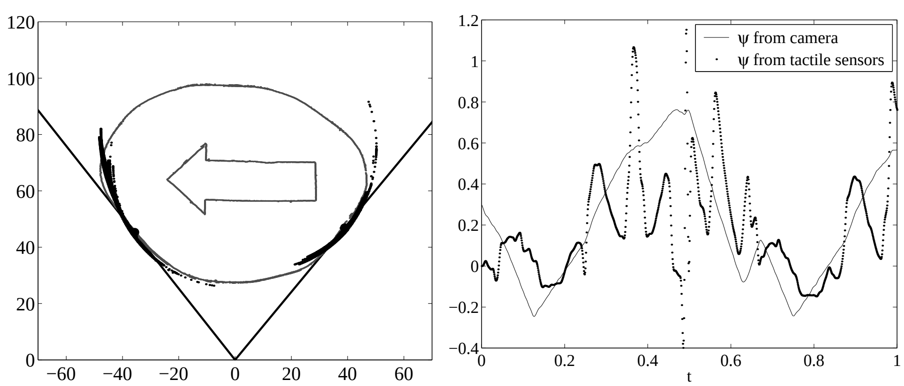


Figure 3.14: Experimental setup



(a) Partially reconstructed shape

(b) Orientation measured by the vision system and the observed orientation

Figure 3.15: Experimental Results

4 Dynamic Shape Reconstruction

In this chapter we drop the assumption that force/torque balance is maintained at all time. It turns out that it is still possible to observe the shape, but now we will need to consider second-order effects. Our approach is to construct an *observer* for our system. The first step is to write our system in the following form:

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}) + \tau_1 \mathbf{g}_1(\mathbf{q}) + \tau_2 \mathbf{g}_2(\mathbf{q}), \tag{4.1}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{q}) \tag{4.2}$$

where $\mathbf{q}$ is a state vector, $\mathbf{f}$, $\mathbf{g}_1$ and $\mathbf{g}_2$ are vector fields, and $\mathbf{h}$ is called the output function. In our case, the state is a vector of sensor readings and the configuration of the robot and the output function returns (a function of) the sensor readings. The vector fields $\mathbf{g}_1$ and $\mathbf{g}_2$ are called the *input vector fields* and describe the rate of change of our system as torques are being applied on palm 1 and palm 2, respectively, at their point of intersection. The vector field $\mathbf{f}$ is called the *drift vector field*. It includes the effects of gravity.

The second step is to find out whether the system described by equations 4.1 and 4.2 is *observable*. Informally, this notion can be defined as: for any two states there exists a control strategy such that the output function will return a different value after some time.

The final step is then to construct the actual observer, which is basically a control law. We can estimate the initial state and if our estimate is not too far from the true initial state, the observer will rapidly converge to the actual state. For more on nonlinear control and nonlinear observers see, e.g., (Isidori, 1995) and (Nijmeijer and van der Schaft, 1990).

4.1 Equations of Motion

The dynamics of our simple model are very straightforward. We assume the effect of gravity on the palms is negligible. As in the previous chapter we assume there is no friction. The contact forces exert a pure torque on the palms. Let $\mathbf{F}_{c_1} = f_{c_1}\bar{\mathbf{n}}_1$ and $\mathbf{F}_{c_2} = f_{c_2}\bar{\mathbf{n}}_2$ be equal to the contact forces acting on the object. The torques generated by the two contact forces on the object are then

$$\tau_{c_1} = (\mathbf{R}\mathbf{x}_1) \times \mathbf{F}_{c_1} = -f_{c_1} d_1 \tag{4.3}$$

$$\tau_{c_2} = (\mathbf{R}\mathbf{x}_2) \times \mathbf{F}_{c_2} = -f_{c_2} d_2 \tag{4.4}$$

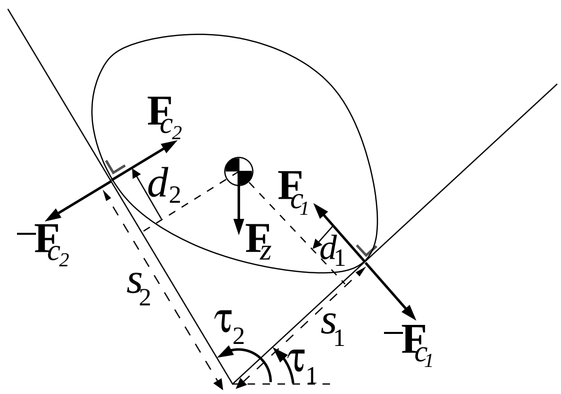


Figure 4.1: Forces acting on the palms and the object.

The dynamics of the system are described by the following equations (see also figure 4.1):

$$m\mathbf{a}_0 = \mathbf{F}_z + \mathbf{F}_{c_1} + \mathbf{F}_{c_2} \tag{4.5}$$

$$I_0\alpha_0 = \tau_{c_1} + \tau_{c_2} = -f_{c_1}d_1 - f_{c_2}d_2, \qquad f_{c_1},\ f_{c_2} \geq 0 \tag{4.6}$$

$$I_1\alpha_1 = \tau_1 - f_{c_1}s_1 \tag{4.7}$$

$$I_2(\alpha_1 + \alpha_2) = \tau_2 + f_{c_2}s_2 \tag{4.8}$$

Here the subscript i, $(i = 0, 1, 2)$ refers to the object, palm 1 and palm 2, respectively. $\mathbf{F}_z = m\mathbf{g}$ is the gravitational force on the object. Note that α_2 is the angular acceleration of palm 2 measured with respect to palm 1, so that $\ddot{\phi}_2 = \alpha_2$. Solving for $\mathbf{a}_0$ and α_0, we get

$$\mathbf{a}_0 = -\frac{I_1\alpha_1 - \tau_1}{ms_1}\bar{\mathbf{n}}_1 + \frac{I_2(\alpha_1 + \alpha_2) - \tau_2}{ms_2}\bar{\mathbf{n}}_2 + \mathbf{g} \tag{4.9}$$

$$\alpha_0 = \frac{I_1\alpha_1 - \tau_1}{m\rho^2 s_1}d_1 - \frac{I_2(\alpha_1 + \alpha_2) - \tau_2}{m\rho^2 s_2}d_2 \tag{4.10}$$

where $\rho = \sqrt{I_0/m}$ is the radius of gyration of the object.

We can measure the mass m by letting the object come to rest. In that case $\mathbf{a}_0 = \mathbf{0}$ and we can solve for m by using $m = -(\mathbf{F}_{c_1} + \mathbf{F}_{c_2})/g$. We have to solve for the radius of gyration by other means, shown in the next section. The mass properties of the palms are assumed to be known.

4.2 General Case

We will now rewrite the constraints on the shape and motion of the object in the form of equation 4.1. We will introduce the state variables ω_0, ω_1 and ω_2 to denote $\dot{\psi}$, $\dot{\phi}_1$ and $\dot{\phi}_2$, respectively. Recall the position constraint on contact point 1 (equation 3.7):

$$s_1\bar{\mathbf{t}}_1 = \mathbf{c}_m + \mathbf{R}\mathbf{x}_1$$

We can differentiate this constraint twice to get a constraint on the acceleration of contact point 1. The right-hand side will contain a term with the curvature at contact point 1. In the appendix it is shown how the acceleration constraint can be turned into the following constraint on the curvature at contact point 1:

$$v_1 = \frac{2\dot{s}_1\omega_1 + s_1\alpha_1 - \omega_0^2 r_1 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1}{\omega_1^2 - \omega_0^2} \tag{4.11}$$

From before (equations 3.6 and 3.27) we had:

$$v_1 = -\frac{\dot{\tilde{r}}_2 + (\dot{\theta} + \dot{\phi}_2)\tilde{d}_2}{\dot{\theta}\sin\phi_2} = -\frac{(-\dot{s}_1 \sin\phi_2 - s_1\omega_2\cos\phi_2) + (\omega_{12} - \omega_0)(s_1\cos\phi_2 - s_2)}{(\omega_1 - \omega_0)\sin\phi_2}, \tag{4.12}$$

where ω_{12} is equal to $\omega_1 + \omega_2$. We can equate these two expressions for v_1 and solve for $\dot{s}_1$:

$$\dot{s}_1 = -\frac{s_1\alpha_1 - \omega_0^2 r_1 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1}{\omega_1 - \omega_0} - \frac{\omega_1 + \omega_0}{\tan\phi_2}s_1 + \frac{(\omega_1 + \omega_0)(\omega_{12} - \omega_0)}{(\omega_1 - \omega_0)\sin\phi_2}s_2$$

Similarly we can derive an expression for $\dot{s}_2$. The differential equations describing our system can be summarized as follows:

$$\dot{r}_1 = -d_1(\omega_1 - \omega_0) \tag{4.13}$$
$$\dot{r}_2 = -d_2(\omega_{12} - \omega_0) \tag{4.14}$$
$$\dot{\omega}_0 = \alpha_0 \tag{4.15}$$
$$\dot{s}_1 = -\frac{s_1\alpha_1 - \omega_0^2 r_1 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1}{\omega_1 - \omega_0} - \frac{\omega_1 + \omega_0}{\tan\phi_2}s_1 + \frac{(\omega_1 + \omega_0)(\omega_{12} - \omega_0)}{(\omega_1 - \omega_0)\sin\phi_2}s_2 \tag{4.16}$$
$$\dot{s}_2 = \frac{-s_2\alpha_{12} - \omega_0^2 r_2 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_2 + \alpha_0 d_2}{\omega_{12} - \omega_0} + \frac{\omega_{12} + \omega_0}{\tan\phi_2}s_2 - \frac{(\omega_{12} + \omega_0)(\omega_1 - \omega_0)}{(\omega_{12} - \omega_0)\sin\phi_2}s_1 \tag{4.17}$$
$$\dot{\phi}_1 = \omega_1 \tag{4.18}$$
$$\dot{\omega}_1 = \alpha_1 \tag{4.19}$$
$$\dot{\phi}_2 = \omega_2 \tag{4.20}$$
$$\dot{\omega}_2 = \alpha_2 \tag{4.21}$$
$$\dot{\rho} = 0 \tag{4.22}$$

Equation 4.13 and 4.14 follow from the properties of the radius function. Recall from section 3.1 that d_1 and d_2 can be written in terms of s_1, s_2, r_1, r_2 and ϕ_2. Therefore d_1 and d_2 do not need to be part of the state of our system. Leaving redundancies in the state would also make it hard, if not impossible, to prove observability of the system. Note also that the control inputs τ_1 and τ_2 are 'hidden' inside $\mathbf{a}_0$ and α_0. The expressions $-\mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1$ and $-\mathbf{a}_0 \cdot \bar{\mathbf{n}}_2 + \alpha_0 d_2$ can be rewritten using equations 4.9 and 4.10 as

$$-\mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1 = \frac{(I_1\alpha_1 - \tau_1)(\rho^2 + d_1^2)}{m\rho^2 s_1} + \frac{(I_2\alpha_{12} - \tau_2)(\rho^2 \cos\phi_2 - d_1 d_2)}{m\rho^2 s_2} - g\cos\phi_1, \tag{4.23}$$

$$-\mathbf{a}_0 \cdot \bar{\mathbf{n}}_2 + \alpha_0 d_2 = -\frac{(I_1\alpha_1 - \tau_1)(\rho^2 \cos\phi_2 - d_1 d_2)}{m\rho^2 s_1} - \frac{(I_2\alpha_{12} - \tau_2)(\rho^2 + d_2^2)}{m\rho^2 s_2} + g\cos\phi_{12}, \tag{4.24}$$

where $\alpha_{12} = \alpha_1 + \alpha_2$ and $\phi_{12} = \phi_1 + \phi_2$.

Let $\mathbf{q} = (r_1, r_2, \omega_0, s_1, s_2, \phi_1, \omega_1, \phi_2, \omega_2, \rho)^T$ be our state vector. Since τ_1 and τ_2 appear linearly in equations 4.13–4.22, our system fits the format of equation 4.1. The drift vector field is

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} -d_1(\omega_1 - \omega_0) \\ -d_2(\omega_{12} - \omega_0) \\ \frac{I_1\alpha_1 d_1}{m\rho^2 s_1} - \frac{I_2\alpha_{12} d_2}{m\rho^2 s_2} \\ \frac{-s_1\alpha_1 + \omega_0^2 r_1 + g\cos\phi_1}{\omega_1 - \omega_0} - \frac{I_1\alpha_1(\rho^2 + d_1^2)}{m\rho^2 s_1(\omega_1 - \omega_0)} - \frac{I_2\alpha_{12}(\rho^2\cos\phi_2 - d_1 d_2)}{m\rho^2 s_2(\omega_1 - \omega_0)} - \frac{\omega_1 + \omega_0}{\tan\phi_2} s_1 + \frac{(\omega_1 + \omega_0)(\omega_{12} - \omega_0)}{(\omega_1 - \omega_0)\sin\phi_2} s_2 \\ \frac{-s_2\alpha_{12} - \omega_0^2 r_2 + g\cos\phi_{12}}{\omega_{12} - \omega_0} - \frac{I_1\alpha_1(\rho^2\cos\phi_2 - d_1 d_2)}{m\rho^2 s_1(\omega_{12} - \omega_0)} - \frac{I_2\alpha_{12}(\rho^2 + d_2^2)}{m\rho^2 s_2(\omega_{12} - \omega_0)} + \frac{\omega_{12} + \omega_0}{\tan\phi_2} s_2 - \frac{(\omega_{12} + \omega_0)(\omega_1 - \omega_0)}{(\omega_{12} - \omega_0)\sin\phi_2} s_1 \\ \omega_1 \\ \alpha_1 \\ \omega_2 \\ \alpha_2 \\ 0 \end{pmatrix}, \tag{4.25}$$

The input vector fields are

$$\mathbf{g}_1(\mathbf{q}) = \begin{pmatrix} 0 \\ 0 \\ -\frac{d_1}{m\rho^2 s_1} \\ \frac{\rho^2 + d_1^2}{m\rho^2 s_1(\omega_1 - \omega_0)} \\ \frac{\rho^2\cos\phi_2 - d_1 d_2}{m\rho^2 s_1(\omega_{12} - \omega_0)} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{g}_2(\mathbf{q}) = \begin{pmatrix} 0 \\ 0 \\ \frac{d_2}{m\rho^2 s_2} \\ \frac{\rho^2\cos\phi_2 - d_1 d_2}{m\rho^2 s_2(\omega_1 - \omega_0)} \\ \frac{\rho^2 + d_2^2}{m\rho^2 s_2(\omega_{12} - \omega_0)} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{4.26}$$

Finally, our output function $\mathbf{h}(\mathbf{q}) = \big(h_1(\mathbf{q}), \ldots, h_k(\mathbf{q})\big)^T$ is

$$\mathbf{h}(\mathbf{q}) = (s_1, s_2, \phi_1, \omega_1, \phi_2, \omega_2)^T. \tag{4.27}$$

Before we can determine the observability of this system we need to introduce some more notation. We define the *differential* $d\phi$ of a function ϕ defined on a subset of $\mathbb{R}^n$ as

$$d\phi(\mathbf{x}) = \left(\tfrac{\partial\phi}{\partial x_1}, \ldots, \tfrac{\partial\phi}{\partial x_n}\right)$$

The Lie derivative of a function ϕ along a vector field X, denoted $L_X\phi$, is defined as

$$L_X\phi = X \cdot d\phi$$

To determine whether the system above is observable we have to consider the *observation space* $\mathcal{O}$. The observation space is defined as the linear space of functions that includes $h_1, \ldots, h_k$, and all repeated Lie derivatives

$$L_{X_1}L_{X_2}\cdots L_{X_l}h_j, \qquad j = 1, \ldots, k, \quad l = 1, 2, \ldots \tag{4.28}$$

where $X_i \in \{\mathbf{f}, \mathbf{g}_1, \mathbf{g}_2\},\ 1 \le i \le l$. Let the *observability codistribution* at a state $\mathbf{q}$ be defined as

$$d\mathcal{O} = \text{span}\{dH(\mathbf{q})|H \in \mathcal{O}\}. \tag{4.29}$$

Then the system described by equation 4.1 is locally observable at state $\mathbf{q}$ if $\dim d\mathcal{O}(\mathbf{q}) = n$, where n is the dimensionality of the state space (Hermann and Krener, 1977). For the system described by equations 4.13–4.22 this condition is too complicated to verify analytically, but one can still do this numerically.

The differentials of the components of the output function are

$$ds_1 = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0)^T \tag{4.30}$$
$$ds_2 = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0)^T \tag{4.31}$$
$$d\phi_1 = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)^T \tag{4.32}$$
$$d\omega_1 = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0)^T \tag{4.33}$$
$$d\phi_2 = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0)^T \tag{4.34}$$
$$d\omega_2 = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0). \tag{4.35}$$

To determine whether the system is observable we need to compute (numerically) the differentials of at least four Lie derivatives. In general $dL_{\mathbf{g}_1}s_1$, $dL_{\mathbf{g}_2}s_2$, $dL_{\mathbf{g}_1}L_{\mathbf{g}_1}s_1$ and $dL_{\mathbf{g}_2}L_{\mathbf{g}_2}s_2$ and the differentials above will span the observability codistribution $d\mathcal{O}$. Note that we can not use the vector field $\mathbf{f}$, because we do not have expressions for α_1 and α_2 in terms of the state variables and, hence, we can not compute the differentials of Lie derivatives along $\mathbf{f}$.

The results above show that in general we will be able to observe the shape of an unkown object. Moreover, the output function contains enough information to recover a constant like the radius of gyration. This leads us to suspect that it may be possible to recover another constant as well: the coefficient of friction. Currently friction is not modeled, but we plan to address this in future work.

4.3 Moving the Palms at a Constant Rate

Although we have shown that the system in the previous section is observable, this does not directly translate to an actual observer. The observability tells us that an observer exists, but constructing a well-behaved observer for a nonlinear system is nontrivial and is still an active area of research. Many observers (such as those proposed by Gauthier et al. (1992) and Zimmer (1994)) rely on Lie derivatives of the drift field. If we want to use such an observer we have to constrain the motion of the palms by restricting them to move at a constant rate, i.e., $\alpha_1 = \alpha_2 = 0$. With this constraint the angular acceleration of the palms vanishes and we do not have to compute derivatives of the accelerations with respect to the state variables. Provided the palms are sufficiently stiff compared to the object, we can easily realize this. Note that this is an *assumption* and that in general a torque-based control system does not automatically translate to a velocity-based or position-based control system. For simplicity we will also assume that we already have recovered the radius of gyration.

Suppose we move palm 1 and 2 at the same rate. Then $\omega_2 = 0$, since it measures the relative rate of palm 2 to palm 1. Our state vector then reduces to $\mathbf{q} = (r_1, r_2, \omega_0, s_1, s_2, \phi_1)^T$. The output function is now $\mathbf{h}(\mathbf{q}) = (s_1, s_2, \phi_1)^T$ and the drift vector field simplifies to

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} -d_1(\omega_1 - \omega_0) \\ -d_2(\omega_1 - \omega_0) \\ 0 \\ \frac{\omega_0^2 r_1 + g\cos\phi_1}{\omega_1 - \omega_0} + (\omega_1 + \omega_0)(s_2 - s_1\cos\phi_2)/\sin\phi_2 \\ \frac{-\omega_0^2 r_2 + g\cos\phi_{12}}{\omega_1 - \omega_0} + (\omega_1 + \omega_0)(s_2\cos\phi_2 - s_1)/\sin\phi_2 \\ \omega_1 \end{pmatrix}. \tag{4.36}$$

We can compute the differentials and Lie derivatives that are necessary to prove the observability:

$$ds_1 = (0, 0, 0, 1, 0, 0)^T \tag{4.37}$$

$$ds_2 = (0, 0, 0, 0, 1, 0)^T \tag{4.38}$$

$$d\phi_1 = (0, 0, 0, 0, 0, 1)^T \tag{4.39}$$

$$L_{\mathbf{f}} s_1 = \mathbf{f} \cdot ds_1 = \frac{\omega_0^2 r_1 + g\cos\phi_1}{\omega_1 - \omega_0} + (\omega_1 + \omega_0)(s_2 - s_1\cos\phi_2)/\sin\phi_2 \tag{4.40}$$

$$dL_{\mathbf{f}} s_1 = \left(\frac{\omega_0^2}{\omega_1 - \omega_0},\ 0,\ \frac{r_1\omega_0(2\omega_1 - \omega_0) + g\cos\phi_1}{(\omega_1 - \omega_0)^2} + \frac{s_2 - s_1\cos\phi_2}{\sin\phi_2},\ -\frac{\omega_1 + \omega_0}{\tan\phi_2},\ \frac{\omega_1 + \omega_0}{\sin\phi_2},\ -\frac{g\sin\phi_1}{\omega_1 - \omega_0}\right)^T \tag{4.41}$$

$$L_{\mathbf{f}} s_2 = \frac{-\omega_0^2 r_2 + g\cos\phi_{12}}{\omega_1 - \omega_0} + (\omega_1 + \omega_0)(s_2\cos\phi_2 - s_1)/\sin\phi_2 \tag{4.42}$$

$$dL_{\mathbf{f}} s_2 = \left(0,\ \frac{-\omega_0^2}{\omega_1 - \omega_0},\ \frac{-r_2\omega_0(2\omega_1 - \omega_0) + g\cos\phi_{12}}{(\omega_1 - \omega_0)^2} + \frac{s_2\cos\phi_2 - s_1}{\sin\phi_2},\ -\frac{\omega_1 + \omega_0}{\sin\phi_2},\ \frac{\omega_1 + \omega_0}{\tan\phi_2},\ -\frac{g\sin\phi_{12}}{\omega_1 - \omega_0}\right)^T \tag{4.43}$$

$$L_{\mathbf{f}} L_{\mathbf{f}} s_1 = -\frac{2\omega_0^2\omega_1(r_1\cos\phi_2 + r_2)}{(\omega_1 - \omega_0)\sin\phi_2} - s_1\omega_1(2\omega_0 + \omega_1) - \frac{g\sin\phi_1(\omega_0 + 2\omega_1)}{\omega_1 - \omega_0} \tag{4.44}$$

$$\begin{aligned} dL_{\mathbf{f}} L_{\mathbf{f}} s_1 = \Big(&\frac{-2\omega_0^2\omega_1}{(\omega_1 - \omega_0)\tan\phi_2},\ \frac{-2\omega_0^2\omega_1}{(\omega_1 - \omega_0)\sin\phi_2},\ \frac{-2\omega_0\omega_1(2\omega_1 - \omega_0)(r_1\cos\phi_2 + r_2)}{(\omega_1 - \omega_0)^2\sin\phi_2} - 2s_1\omega_1 - \frac{3\omega_1 g\sin\phi_1}{(\omega_1 - \omega_0)^2}, \\ &- \omega_1(2\omega_0 + \omega_1),\ 0,\ \frac{-g\cos\phi_1(\omega_0 + 2\omega_1)}{\omega_1 - \omega_0}\Big)^T \end{aligned} \tag{4.45}$$

It can be shown that the determinant of the matrix that has colums ds_1, ds_2, $d\phi_1$, $dL_{\mathbf{f}}s_1$, $dL_{\mathbf{f}}s_2$ and $dL_{\mathbf{f}}L_{\mathbf{f}}s_1$ is equal to

$$\frac{g\omega_0^4\omega_1\sin\phi_1}{(\omega_1-\omega_0)^4} \tag{4.46}$$

In other words, the system is observable as long as ω_0 and ω_1 are nonzero and not equal to each other. Note that the system is observable solely in terms of the drift vector field.

Let us now consider the case $\omega_1 = 0$: palm 1 is fixed and palm 2 is moving at a constant rate. The state vector then reduces to $\mathbf{q} = (r_1, r_2, \omega_0, s_1, s_2, \phi_2)^T$. The output function is now $\mathbf{h}(\mathbf{q}) = (s_1, s_2, \phi_2)^T$ and the drift vector field simplifies to

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} d_1\omega_0 \\ -d_2(\omega_2-\omega_0) \\ 0 \\ -\frac{\omega_0^2 r_1 + g\cos\phi_1}{\omega_0} - \frac{\omega_0 s_1}{\tan\phi_2} - \frac{(\omega_2-\omega_0)s_2}{\sin\phi_2} \\ \frac{-\omega_0^2 r_2 + g\cos\phi_{12}}{\omega_2-\omega_0} + \frac{(\omega_2+\omega_0)s_2}{\tan\phi_2} + \frac{(\omega_2+\omega_0)\omega_0 s_1}{(\omega_2-\omega_0)\sin\phi_2} \\ \omega_2 \end{pmatrix} \tag{4.47}$$

As before, we need to compute the differentials and Lie derivatives to determine observability:

$$ds_1 = (0,0,0,1,0,0)^T \tag{4.48}$$

$$ds_2 = (0,0,0,0,1,0)^T \tag{4.49}$$

$$d\phi_2 = (0,0,0,0,0,1)^T \tag{4.50}$$

$$L_{\mathbf{f}}s_2 = -\frac{\omega_0^2 r_1 + g\cos\phi_1}{\omega_0} - \frac{\omega_0 s_1}{\tan\phi_2} - \frac{(\omega_2-\omega_0)s_2}{\sin\phi_2} \tag{4.51}$$

$$dL_{\mathbf{f}}s_1 = \Big(-\omega_0,\ 0,\ -r_1 + \frac{g\cos\phi_1}{\omega_0^2} - \frac{s_1}{\tan\phi_2} + \frac{s_2}{\sin\phi_2},\ -\frac{\omega_0}{\tan\phi_2},\ -\frac{\omega_2-\omega_0}{\sin\phi_2},\ \frac{\omega_0 s_1 + (\omega_2-\omega_0)s_2\cos\phi_2}{\sin^2\phi_2}\Big)^T \tag{4.52}$$

$$L_{\mathbf{f}}s_2 = \frac{-\omega_0^2 r_2 + g\cos\phi_{12}}{\omega_2-\omega_0} + \frac{(\omega_2+\omega_0)s_2}{\tan\phi_2} + \frac{(\omega_2+\omega_0)\omega_0 s_1}{(\omega_2-\omega_0)\sin\phi_2} \tag{4.53}$$

$$\begin{aligned} dL_{\mathbf{f}}s_2 = \Big(0,\ &\frac{-\omega_0^2}{\omega_2-\omega_0},\ \frac{-r_2\omega_0(2\omega_2-\omega_0)+g\cos\phi_{12}}{(\omega_2-\omega_0)^2} + \frac{s_2}{\tan\phi_2} - \frac{s_1(\omega_2^2+2\omega_0\omega_2-\omega_0^2)}{(\omega_2-\omega_0)^2\sin\phi_2},\ \frac{(\omega_2+\omega_0)\omega_0}{(\omega_2-\omega_0)\sin\phi_2}, \\ &\frac{\omega_2+\omega_0}{\tan\phi_2},\ \frac{-g\sin\phi_{12}}{\omega_2-\omega_0} - \frac{s_2(\omega_2+\omega_0)}{\sin^2\phi_2} - \frac{(\omega_2+\omega_0)\omega_0 s_1\cos\phi_2}{(\omega_2-\omega_0)\sin^2\phi_2}\Big)^T \end{aligned} \tag{4.54}$$

$$L_{\mathbf{f}}L_{\mathbf{f}}s_2 = -\frac{2\omega_0^2\omega_2(r_1+r_2\cos\phi_2)}{(\omega_2-\omega_0)\sin\phi_2} - s_2\omega_2(2\omega_0+\omega_2) - \frac{g(\omega_0+2\omega_2)\sin\phi_{12}}{\omega_2-\omega_0} \tag{4.55}$$

$$\begin{aligned} dL_{\mathbf{f}}L_{\mathbf{f}}s_2 = \Big(&\frac{-2\omega_0^2\omega_2}{(\omega_2-\omega_0)\sin\phi_2},\ \frac{-2\omega_0^2\omega_2}{(\omega_2-\omega_0)\tan\phi_2},\ \frac{-2\omega_0\omega_2(2\omega_2-\omega_0)(r_1+r_2\cos\phi_2)}{(\omega_2-\omega_0)^2\sin\phi_2} - 2s_2\omega_2 - \frac{3\omega_2 g\sin\phi_{12}}{(\omega_2-\omega_0)^2}, \\ &0, -\omega_2(2\omega_0+\omega_2),\ \frac{2\omega_0^2\omega_2(r_1\cos\phi_2+r_2)}{(\omega_2-\omega_0)\sin^2\phi_2} - \frac{-g\cos\phi_{12}(\omega_0+2\omega_2)}{\omega_2-\omega_0}\Big)^T \end{aligned} \tag{4.56}$$

It can be shown that the determinant of the matrix that has colums ds_1, ds_2, $d\phi_1$, $dL_{\mathbf{f}}s_1$, $dL_{\mathbf{f}}s_2$ and $dL_{\mathbf{f}}L_{\mathbf{f}}s_2$ is equal to

$$-\frac{2\omega_0^3\omega_2^2}{(\omega_2-\omega_0)^3\sin\phi_2}\Big[\Big(\frac{\cos\phi_1}{\omega_0} + \frac{\sin\phi_1\sin\phi_{12}}{2\omega_2}\Big)g + \omega_0 r_1 + \frac{s_1(\omega_2+\omega_0)}{\tan\phi_2} + \frac{s_2(\omega_2-\omega_0)}{\sin\phi_2}\Big]. \tag{4.57}$$

This determinant is generally nonzero if ω_0 and ω_2 are nonzero and not equal to each other. So even if we move only palm 2 at a constant rate and hold palm 1 fixed, the system is still locally observable.

4.4 Fixed Palms

For illustrative purposes we will now consider one more special case, where it is possible to analytically determine observability. Suppose our control strategy consists of keeping the palms in the same position. In other words, $\omega_1 = \omega_2 = \alpha_1 = \alpha_2 = 0$. Note that this is not the same as clamping the palms, since we still assume that the palms are actively controlled. We can then reduce our state even further to $\mathbf{q} = (r_1, r_2, \omega_0, s_1, s_2)^T$. The input and control vector fields simplify to

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} d_1\omega_0 \\ d_2\omega_0 \\ 0 \\ -\frac{g\cos\phi_1}{\omega_0} - \omega_0\big(r_1 + \frac{s_1}{\tan\phi_2} - \frac{s_2}{\sin\phi_2}\big) \\ -\frac{g\cos\phi_{12}}{\omega_0} + \omega_0\big(r_2 + \frac{s_2}{\tan\phi_2} - \frac{s_1}{\sin\phi_2}\big) \end{pmatrix},$$

$$\mathbf{g}_1(\mathbf{q}) = \begin{pmatrix} 0 \\ 0 \\ -\frac{d_1}{m\rho^2 s_1} \\ -\frac{\rho^2+d_1^2}{m\rho^2 s_1\omega_0} \\ -\frac{\rho^2\cos\phi_2 - d_1 d_2}{m\rho^2 s_1\omega_0} \end{pmatrix}, \quad \text{and} \quad \mathbf{g}_2(\mathbf{q}) = \begin{pmatrix} 0 \\ 0 \\ \frac{d_2}{m\rho^2 s_2} \\ -\frac{\rho^2\cos\phi_2 - d_1 d_2}{m\rho^2 s_2\omega_0} \\ -\frac{\rho^2+d_2^2}{m\rho^2 s_2\omega_0} \end{pmatrix}. \tag{4.58}$$

The output function is now simply $\mathbf{h}(\mathbf{q}) = (s_1, s_2)^T$. Since the output function has two components and our state space is five-dimensional, we need to take at least three Lie derivatives. Consider the following differentials and Lie derivatives:

$$ds_1 = (0, 0, 0, 1, 0)^T \tag{4.59}$$

$$ds_2 = (0, 0, 0, 0, 1)^T \tag{4.60}$$

$$L_{\mathbf{f}}s_1 = -\frac{g\cos\phi_1}{\omega_0} - \omega_0\big(r_1 + \frac{s_1}{\tan\phi_2} - \frac{s_2}{\sin\phi_2}\big) \tag{4.61}$$

$$dL_{\mathbf{f}}s_1 = \Big(-\omega_0, 0, \frac{g\cos\phi_1}{\omega_0^2} - \big(r_1 + \frac{s_1}{\tan\phi_2} - \frac{s_2}{\sin\phi_2}\big), -\frac{\omega_0}{\tan\phi_2}, \frac{\omega_0}{\sin\phi_2}\Big)^T \tag{4.62}$$

$$L_{\mathbf{f}}s_2 = -\frac{g\cos\phi_{12}}{\omega_0} - \omega_0\big(r_2 + \frac{s_2}{\tan\phi_2} - \frac{s_1}{\sin\phi_2}\big) \tag{4.63}$$

$$dL_{\mathbf{f}}s_2 = \Big(0, -\omega_0, -\frac{g\cos\phi_{12}}{\omega_0^2} - \big(r_2 + \frac{s_2}{\tan\phi_2} - \frac{s_1}{\sin\phi_2}\big), \frac{\omega_0}{\sin\phi_2}, -\frac{\omega_0}{\tan\phi_2}\Big)^T \tag{4.64}$$

$$L_{\mathbf{f}}L_{\mathbf{f}}s_1 = \omega_0^2(-d_1 + \frac{r_1}{\tan\phi_2} + \frac{r_2}{\sin\phi_2} - s_1) + g\sin\phi_1 = g\sin\phi_1 \tag{4.65}$$

$$dL_{\mathbf{f}}L_{\mathbf{f}}s_1 = (0, 0, 0, 0, 0)^T \tag{4.66}$$

The step in equation 4.65 follows from expression 3.13. Because of symmetry $dL_{\mathbf{f}}L_{\mathbf{f}}s_2$ is equal to the $\mathbf{0}$ vector as well. This means that with fixed palms the system is *not* observable in terms of *just* the drift field $\mathbf{f}$. Now suppose we compute the Lie derivative of s_1 along $\mathbf{g}_1$ and its differential:

$$L_{\mathbf{g}_1}s_1 = -\frac{\rho^2 + d_1^2}{m\rho^2 s_1\omega_0} \tag{4.67}$$

$$dL_{\mathbf{g}_1}s_1 = \tfrac{1}{m\rho^2 s_1\omega_0}\Big(\tfrac{-2d_1}{\tan\phi_2},\ \tfrac{-2d_1}{\sin\phi_2},\ \tfrac{\rho^2+d_1^2}{\omega_0},\ 2d_1 + \tfrac{\rho^2+d_1^2}{s_1},\ 0\Big)^T \tag{4.68}$$

The differentials $ds_1, ds_2, dL_{\mathbf{f}}s_1, dL_{\mathbf{f}}s_2$ and $dL_{\mathbf{g}_1}s_1$ generally span the observability codistribution and, hence, the system is observable. It is important to remember that the palms are actively controlled, i.e., the palms are not clamped. Otherwise we would not know the torques exerted by the palms. We need the torques in order to integrate (by using an observer) the differential equation 4.1. As mentioned before, the construction of an observer that relies on the control vector fields is nontrivial. Since the motion of the palms is so constrained, the system is likely to observe only a small fraction of an unknown shape. Therefore we suspect that if one were to construct an observer it would have very limited practical value.

5 Discussion

In this report we have shown how to reconstruct the shape of an unknown smooth convex planar shape using two tactile sensors. We presented the analysis, simulation results and experimental results for the quasistatic case. Firstly, we showed that any generic smooth shape is globally observable. Secondly, we derived expressions for the curvature at the contact points and the rotational speed of the object. This completely describes the shape and motion of an unknown object as a system of differential equations. The simulation results showed that our approach works reasonably well, but further research is needed on finding the initial conditions for this system of differential equations. Our experimental results show that our approach also works in practice. Athough there were large errors in the reconstructed motion of our experimental object, the reconstructed shape was close to the actual shape.

In chapter 4 we addressed the dynamic case, where force/torque balance is no longer assumed. We established that it is possible to reconstruct the shape locally in this case too, as long as at least one of the palms is moving. By moving the palms slowly enough, we can approximate the quasistatic case. This seems to suggest that we may be able to achieve global observability in the dynamic case as well. Further research is needed to confirm this intuition. If both palms are motionless, the shape is still observable. However, the construction of an observer for this case is very nontrivial and of limited practical value.

In future work we hope to extend our analytic results in the following ways. Firstly, we are planning to model friction. We hope to reconstruct the value of the friction coefficient using a nonlinear observer (cf. the radius of gyration in chapter 4). Secondly, we will analyze the three-dimensional case. In 3D we cannot expect to reconstruct the entire shape, since the contact points trace out only curves on the surface of the object. Nevertheless, by constructing a sufficiently fine mesh with these curves, we can create a good approximation. The quasistatic approach will most likely not work in 3D, because in 3D the rotation velocity has three degrees of freedom and force/torque balance only gives us two constraints.

Other Sensing Strategies

Besides the model of two planar palms, there are many other ways to reconstruct the shape of an unknown moving object with tactile sensors. The simplest extension is to change the shape of the palms. Figure 5.1 shows palms shaped as arcs of a circle. One advantage is that with less

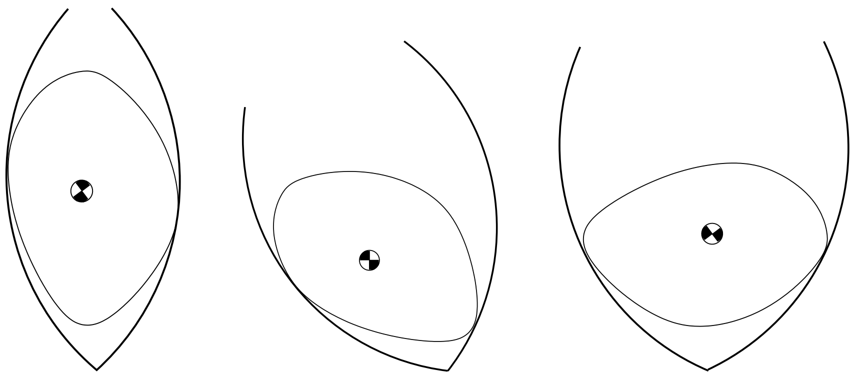


Figure 5.1: Curved palms

manipulation we can cover more of the shape, since a grasp with these palms comes closer to an enveloping grasp. Further research is needed to quantify this claim. The radius of curvature of the palms should be chosen such that there is always just one contact point on each palm. However, if we can detect multiple contacts, it makes more sense to maximize the number of contact points (e.g., by using deformable palms).

Figure 5.2 shows an entirely different sensing strategy. An object is grasped by a robot arm using a pivoting gripper (Rao et al., 1994, 1995). With such a gripper the object is free to rotate around the line through the grasp contact points. The sensing strategy consists of dragging or pushing the object over a surface coated with tactile sensors. We think it would be interesting to determine whether this system is observable as well.

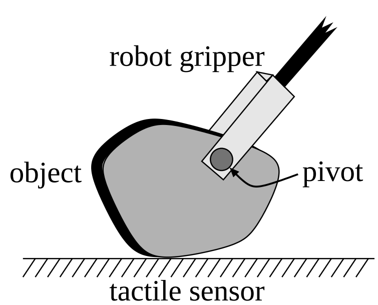


Figure 5.2: Dragging an object over a tactile sensor with a pivoting grasp

A Derivations

A.1 Quasistatic Shape Reconstruction

A.1.1 Curvature at the Contact Points

The expressions for the curvature at the contact points can be found by differentiating the generalized contact support functions:

$$\begin{aligned}
\tilde{r}_1' &= (x_1' - x_2') \cdot n_1 + (x_1 - x_2) \cdot n_1' \\
&= v_2 t_2 \cdot n_1 - (x_1 - x_2) \cdot t_1 \\
&= -v_2 \sin\phi_2 - d_1
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\dot{\tilde{r}}_1 &= (\dot{x}_1 - \dot{x}_2) \cdot n_1 + (x_1 - x_2) \cdot \dot{n}_1 \\
&= (\dot\theta x_1' - (\dot\theta + \dot\phi_2) x_2') \cdot n_1 + (x_1 - x_2) \cdot (\dot\theta n_1') \\
&= -v_2 (\dot\theta + \dot\phi_2) \sin\phi_2 - \dot\theta d_1
\end{aligned} \tag{A.2}$$

$$\begin{aligned}
\tilde{r}_2' &= (x_1' - x_2') \cdot n_2 + (x_1 - x_2) \cdot n_2' \\
&= -v_1 t_1 \cdot n_2 - (x_1 - x_2) \cdot t_2 \\
&= -v_1 \sin\phi_2 - d_2
\end{aligned} \tag{A.3}$$

$$\begin{aligned}
\dot{\tilde{r}}_2 &= (\dot{x}_1 - \dot{x}_2) \cdot n_2 + (x_1 - x_2) \cdot \dot{n}_2 \\
&= (\dot\theta x_1' - (\dot\theta + \dot\phi_2) x_2') \cdot n_2 + (x_1 - x_2) \cdot ((\dot\theta + \dot\phi_2) n_2') \\
&= -v_1 \dot\theta \sin\phi_2 - (\dot\theta + \dot\phi_2) d_2
\end{aligned} \tag{A.4}$$

From these expressions it follows that the curvature at the contact points can be written as

$$v_1 = -\frac{\tilde{r}_2' + \tilde{d}_2}{\sin\phi_2} = -\frac{\dot{\tilde{r}}_2 + (\dot\theta + \dot\phi_2)\tilde{d}_2}{\dot\theta \sin\phi_2} \tag{A.5}$$

$$v_2 = -\frac{\tilde{r}_1' + \tilde{d}_1}{\sin\phi_2} = -\frac{\dot{\tilde{r}}_1 + \dot\theta \tilde{d}_1}{(\dot\theta + \dot\phi_2)\sin\phi_2} \tag{A.6}$$

A.1.2 Derivatives of Center of Mass and Center of Rotation

Recall the following expressions for the center of mass and the center of rotation:

$$\mathbf{c}_m(\psi,\phi_1,\phi_2) = -\tilde{r}_2\bar{\mathbf{t}}_1/\sin\phi_2 - \mathbf{R}\mathbf{x}_1 \tag{A.7}$$

$$\mathbf{c}_r(\psi,\phi_1,\phi_2) = -(\tilde{r}_2\bar{\mathbf{t}}_1 + \tilde{d}_2\bar{\mathbf{n}}_1)/\sin\phi_2 \tag{A.8}$$

We are interested in the partial derivatives of these expressions with respect to ψ, because they tell us something about the stable poses of the object. In the previous section we computed derivatives with respect to curve parameters. The partial derivative of the curve parameter θ with respect to ψ is equal to -1. This follows from $\theta = \phi_1 - \psi - \pi/2$ (equation 3.1, p. 14). The partial derivatives with respect to ψ of equations A.7 and A.8 are therefore

$$\begin{aligned}
\frac{\partial\mathbf{c}_m}{\partial\psi} &= \tilde{r}_2'\bar{\mathbf{t}}_1/\sin\phi_2 - \left(\tfrac{\partial}{\partial\psi}\mathbf{R}\right)\mathbf{x}_1 + v_1\bar{\mathbf{t}}_1 && \text{(A.9)}\\
&= -(v_1\sin\phi_2 + d_2)\bar{\mathbf{t}}_1/\sin\phi_2 - \left(\tfrac{\partial}{\partial\psi}\mathbf{R}\right)\mathbf{x}_1 + v_1\bar{\mathbf{t}}_1 && \text{(A.10)}\\
&= -\frac{\tilde{d}_2\bar{\mathbf{t}}_1}{\sin\phi_2} - \left(\tfrac{\partial}{\partial\psi}\mathbf{R}\right)\mathbf{x}_1, && \text{(A.11)}\\
\frac{\partial\mathbf{c}_r}{\partial\psi} &= (\tilde{r}_2'\bar{\mathbf{t}}_1 + \tilde{d}_2'\bar{\mathbf{n}}_1)/\sin\phi_2 && \text{(A.12)}\\
&= (-(v_1\sin\phi_2 + d_2)\bar{\mathbf{t}}_1 + (v_1\cos\phi_2 + v_2 + r_2)\bar{\mathbf{n}}_1)/\sin\phi_2 && \text{(A.13)}\\
&= \big(-v_1\bar{\mathbf{n}}_2 + v_2\bar{\mathbf{n}}_1 + \tilde{r}_2\bar{\mathbf{n}}_1 - \tilde{d}_2\bar{\mathbf{t}}_1\big)/\sin\phi_2. && \text{(A.14)}
\end{aligned}$$

The derivative of $\tilde{d}_2$ can be obtained in a similar fashion as the derivatives of $\tilde{r}_1$ and $\tilde{r}_2$ in section A.1.1. Notice that equation A.11 is very similar to $\mathbf{c}_m - \mathbf{c}_r$:

$$\mathbf{c}_m - \mathbf{c}_r = \tilde{d}_2\bar{\mathbf{n}}_1/\sin\phi_2 - \mathbf{R}\mathbf{x}_1 \tag{A.15}$$

In fact, upon careful inspection we see that

$$\frac{\partial\mathbf{c}_m}{\partial\psi} = \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}(\mathbf{c}_m - \mathbf{c}_r)\,. \tag{A.16}$$

A.1.3 Rotational Velocity

In section 3.3 it was shown that the force/torque balance constraint can be written as

$$r_1\sin\phi_1 - d_1\cos\phi_1 = -\tilde{d}_2\tfrac{\sin\phi_1}{\sin\phi_2}. \tag{A.17}$$

Differentiating the left-hand side of this equation we get:

$$\begin{aligned}
\tfrac{d}{dt}(r_1\sin\phi_1 - d_1\cos\phi_1) &= (\dot{r}_1 + d_1\dot{\phi}_1)\sin\phi_1 + (r_1\dot{\phi}_1 - \dot{d}_1)\cos\phi_1 && \text{(A.18)}\\
&= d_1(\dot{\phi}_1 - \dot{\theta})\sin\phi_1 + r_1(\dot{\phi}_1 - \dot{\theta})\cos\phi_1 - \tfrac{\dot{\tilde{r}}_2 + (\dot{\theta}+\dot{\phi}_2)\tilde{d}_2}{\sin\phi_2}\cos\phi_1 && \text{(A.19)}\\
&= d_1\dot{\psi}\sin\phi_1 + r_1\dot{\psi}\cos\phi_1 - \tfrac{\dot{\tilde{r}}_2 + (\dot{\phi}_1+\dot{\phi}_2-\dot{\psi})\tilde{d}_2}{\sin\phi_2}\cos\phi_1 && \text{(A.20)}\\
&= \dot{\psi}\Big(d_1\sin\phi_1 + r_1\cos\phi_1 + \tilde{d}_2\tfrac{\cos\phi_1}{\sin\phi_2}\Big) - \tfrac{\dot{\tilde{r}}_2 + (\dot{\phi}_1+\dot{\phi}_2)\tilde{d}_2}{\sin\phi_2}\cos\phi_1 && \text{(A.21)}
\end{aligned}$$

The step in equation A.19 follows from properties of the contact support function: $r'(\theta) = -d(\theta)$ and $d'(\theta) = r(\theta) - v(\theta)$. The derivative of the right-hand side of equation A.17 can be written as

$$\tfrac{d}{dt}(-\tilde{d}_2 \tfrac{\sin\phi_1}{\sin\phi_2}) = \big(- \dot{\tilde{d}}_2 \sin\phi_1 - \tilde{d}_2 \dot{\phi}_1 \cos\phi_1 + \tilde{d}_2 \dot{\phi}_2 \sin\phi_1 \cot\phi_2 \big) / \sin\phi_2 \tag{A.22}$$

Equating expressions A.21 and A.22, substituting expression 3.13 for d_1, and solving for $\dot{\psi}$ we arrive at the following expression for $\dot{\psi}$:

$$\dot{\psi} = \frac{\dot{\tilde{r}}_2 \cos\phi_1 - \dot{\tilde{d}}_2 \sin\phi_1 + \tilde{d}_2 \dot{\phi}_2 \frac{\sin\phi_{12}}{\sin\phi_2}}{r_1 \sin\phi_{12} + (r_2 + \tilde{r}_2)\sin\phi_1 + \tilde{d}_2 \cos\phi_1}, \tag{A.23}$$

where $\phi_{12} = \phi_1 + \phi_2$.

A.2 Dynamic Shape Reconstruction: Velocity of the Contact Points

Recall the position constraint on contact point 1 (equation 3.7, p. 16):

$$s_1 \bar{\mathbf{t}}_1 = \mathbf{c}_m + \mathbf{R}\mathbf{x}_1$$

We can differentiate this constraint to get a constraint on the velocity of contact point 1:

$$\begin{aligned} \dot{s}_1 \bar{\mathbf{t}}_1 + s_1 \omega_1 \bar{\mathbf{n}}_1 &= \dot{\mathbf{c}}_m + \omega_0 \times (\mathbf{R}\mathbf{x}_1) + \mathbf{R}\dot{\mathbf{x}}_1 \\ &= \dot{\mathbf{c}}_m + \omega_0 \times (\mathbf{R}\mathbf{x}_1) + (\omega_1 - \omega_0) v_1 \bar{\mathbf{t}}_1 \end{aligned}$$

This follows from $\theta = \phi_1 - \psi - \frac{\pi}{2}$ and from our parametrization of the shape of the object. Differentiating again results in the following constraint on the acceleration:

$$\begin{aligned} \ddot{s}_1 \bar{\mathbf{t}}_1 + 2\dot{s}_1 \omega_1 \bar{\mathbf{n}}_1 + s_1 \alpha_1 \bar{\mathbf{n}}_1 - s_1 \omega_1^2 \bar{\mathbf{t}}_1 &= \mathbf{a}_0 + \alpha_0 \times (\mathbf{R}\mathbf{x}_1) + \omega_0 \times \big(\omega_0 \times (\mathbf{R}\mathbf{x}_1) + (\omega_1 - \omega_0) v_1 \bar{\mathbf{t}}_1\big) \\ &\quad + (\alpha_1 - \alpha_0) v_1 \bar{\mathbf{t}}_1 + (\omega_1 - \omega_0)(\dot{v}_1 \bar{\mathbf{t}}_1 + \omega_1 v_1 \bar{\mathbf{n}}_1) \\ &= \mathbf{a}_0 + \alpha_0 \times (\mathbf{R}\mathbf{x}_1) - \omega_0^2 \mathbf{R}\mathbf{x}_1 + (\omega_1^2 - \omega_0^2) v_1 \bar{\mathbf{n}}_1 \\ &\quad + (\alpha_1 - \alpha_0) v_1 \bar{\mathbf{t}}_1 + (\omega_1 - \omega_0) \dot{v}_1 \bar{\mathbf{t}}_1 \end{aligned}$$

The acceleration constraint in the $\bar{\mathbf{n}}_1$ direction is therefore:

$$2\dot{s}_1 \omega_1 + s_1 \alpha_1 = \mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \omega_0^2 r_1 - \alpha_0 d_1 + (\omega_1^2 - \omega_0^2) v_1.$$

We can solve this constraint for v_1:

$$v_1 = \frac{2\dot{s}_1 \omega_1 + s_1 \alpha_1 - \omega_0^2 r_1 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1}{\omega_1^2 - \omega_0^2} \tag{A.24}$$

From before (equations 3.6 and 3.27) we had:

$$v_1 = -\frac{\dot{\tilde{r}}_2 + (\dot{\theta} + \dot{\phi}_2)\tilde{d}_2}{\dot{\theta} \sin\phi_2} = -\frac{(-\dot{s}_1 \sin\phi_2 - s_1 \omega_2 \cos\phi_2) + (\omega_{12} - \omega_0)(s_1 \cos\phi_2 - s_2)}{(\omega_1 - \omega_0)\sin\phi_2}$$

(A.25)

We can equate these two expressions for v_1 and solve for $\dot{s}_1$:

$$\dot{s}_1 = -\frac{s_1\alpha_1 - \omega_0^2 r_1 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_1 + \alpha_0 d_1}{\omega_1 - \omega_0} - \frac{\omega_1 + \omega_0}{\tan\phi_2} s_1 + \frac{(\omega_1 + \omega_0)(\omega_{12} - \omega_0)}{(\omega_1 - \omega_0)\sin\phi_2} s_2$$

Similarly we can derive an expression for $\dot{s}_2$:

$$\dot{s}_2 = \frac{-s_2\alpha_{12} - \omega_0^2 r_2 - \mathbf{a}_0 \cdot \bar{\mathbf{n}}_2 + \alpha_0 d_2}{\omega_{12} - \omega_0} + \frac{\omega_{12} + \omega_0}{\tan\phi_2} s_2 - \frac{(\omega_{12} + \omega_0)(\omega_1 - \omega_0)}{(\omega_{12} - \omega_0)\sin\phi_2} s_1$$

References

Abell, T. and Erdmann, M. A. (1995). Stably supported rotations of a planar polygon with two frictionless contacts. In *Proc. 1995 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 411–418, Pittsburgh, Pennsylvania.

Akella, S. and Mason, M. T. (1998). Parts orienting with partial sensor information. In *Proc. 1998 IEEE Intl. Conf. on Robotics and Automation*, pages 557–564, Leuven, Belgium.

Allen, P. K. and Michelman, P. (1990). Acquisition and interpretation of 3-D sensor data from touch. *IEEE Trans. on Robotics and Automation*, 6(4):397–404.

Arkin, E., Held, M., Mitchell, J., and Skiena, S. (1998). Recognizing polygonal parts from width measurements. *Computational Geometry: Theory and Applications*, 9(4):237–246.

Bicchi, A., Marigo, A., and Prattichizzo, D. (1999). Dexterity through rolling: Manipulation of unknown objects. In *Proc. 1999 IEEE Intl. Conf. on Robotics and Automation*, pages 1583–1588, Detroit, Michigan.

Boissonnat, J. D. and Yvinec, M. (1992). Probing a scene of nonconvex polyhedra. *Algorithmica*, 8:321–342.

Cai, C. S. and Roth, B. (1986). On the planar motion of rigid bodies with point contact. *Mechanism and Machine Theory*, 21(6):453–466.

Cai, C. S. and Roth, B. (1987). On the spatial motion of a rigid body with point contact. In *Proc. 1987 IEEE Intl. Conf. on Robotics and Automation*, pages 686–695.

Charlebois, M., Gupta, K., and Payandeh, S. (1996). Curvature based shape estimation using tactile sensing. In *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*, pages 3502–3507.

Charlebois, M., Gupta, K., and Payandeh, S. (1997). Shape description of general, curved surfaces using tactile sensing and surface normal information. In *Proc. 1997 IEEE Intl. Conf. on Robotics and Automation*, pages 2819–2824.

Chen, N., Rink, R., and Zhang, H. (1996). Local object shape from tactile sensing. In *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*, pages 3496–3501.

Choi, K. K., Jiang, S. L., and Li, Z. (1998). Multifingered robotic hands: Contact experiments using tactile sensors. In *Proc. 1998 IEEE Intl. Conf. on Robotics and Automation*, pages 2268–2273, Leuven, Belgium.

Cole, R. and Yap, C. K. (1987). Shape from probing. *Journal of Algorithms*, 8(1):19–38.

Cutkosky, M. R. (1985). *Robotic Grasping and Fine Manipulation*. Kluwer, Dordrecht; Boston.

Dobkin, D., Edelsbrunner, H., and Yap, C. K. (1986). Probing convex polytopes. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 424–432, Berkeley, California.

Erdmann, M. A. (1998a). An exploration of nonprehensile two-palm manipulation: Planning and execution. *Intl. J. of Robotics Research*, 17(5).

Erdmann, M. A. (1998b). Shape recovery from passive locally dense tactile data. In *Workshop on the Algorithmic Foundations of Robotics*.

Erdmann, M. A. and Mason, M. T. (1988). An exploration of sensorless manipulation. *IEEE J. of Robotics and Automation*, 4(4):369–379.

Erdmann, M. A., Mason, M. T., and Vaněček, Jr., G. (1993). Mechanical parts orienting: The case of a polyhedron on a table. *Algorithmica*, 10:226–247.

Fearing, R. S. (1984). Simplified grasping and manipulation with dextrous robot hands. In *Proceedings of the 1984 American Control Conference*, pages 32–38, San Diego, California.

Fearing, R. S. (1990). Tactile sensing for shape interpretation. In Venkataraman, S. T. and Iberall, T., editors, *Dexterous Robot Hands*, chapter 10, pages 209–238. Springer Verlag, Berlin; Heidelberg; New York.

Fearing, R. S. and Binford, T. O. (1988). Using a cylindrical tactile sensor for determining curvature. In *Proc. 1988 IEEE Intl. Conf. on Robotics and Automation*, pages 765–771.

Gauthier, J. P., Hammouri, H., and Othman, S. (1992). A simple observer for nonlinear systems applications to bioreactors. *IEEE Trans. on Automatic Control*, 37(6):875–880.

Goldberg, K. Y. and Bajcsy, R. (1984). Active touch and robot perception. *Cognition and Brain Theory*, 7(2):199–214.

Hermann, R. and Krener, A. J. (1977). Nonlinear controllability and observability. *IEEE Trans. on Automatic Control*, AC-22(5):728–740.

Hong, J., Lafferriere, G., Mishra, B., and Tan, X. (1990). Fine manipulation with multifinger hands. In *Proc. 1990 IEEE Intl. Conf. on Robotics and Automation*, pages 1568–1573, Cincinnati, Ohio.

Howe, R. D. and Cutkosky, M. R. (1992). Touch sensing for robotic manipulation and recognition. In Khatib, O. and Lozano-Pérez, J. J. C. T., editors, *The Robotics Review 2*. MIT Press, Cambridge, MA.

Isidori, A. (1995). *Nonlinear Control Systems*. Springer Verlag, Berlin; Heidelberg; New York, third edition.

Jia, Y.-B. (2000). Grasping curved objects through rolling. In *Proc. 2000 IEEE Intl. Conf. on Robotics and Automation*, pages 377–382, San Fransisco, California.

Jia, Y.-B. and Erdmann, M. (1996). Geometric sensing of known planar shapes. *Intl. J. of Robotics Research*, 15(4):365–392.

Jia, Y.-B. and Erdmann, M. A. (1999). Pose and motion from contact. *Intl. J. of Robotics Research*, 18(5).

Kaneko, M. and Tsuji, T. (2000). Pulling motion based tactile sensing. In *Workshop on the Algorithmic Foundations of Robotics*, Hanover, New Hampshire.

Keren, D., Rivlin, E., Shimsoni, I., and Weiss, I. (1998). Recognizing surfaces using curve invariants and differential properties of curves and surfaces. In *Proc. 1998 IEEE Intl. Conf. on Robotics and Automation*, pages 3375–3381, Leuven, Belgium.

Kerr, J. and Roth, B. (1986). Analysis of multifingered hands. *Intl. J. of Robotics Research*, 4(4):3–17.

Klatzky, R. L., Lederman, S. J., and Metzger, V. A. (1985). Identifying objects by touch: An “expert system”. *Perception and Psychophysics*, 37:299–302.

Kölzow, D., Kuba, A., and Volčič, A. (1989). An algorithm for reconstructing convex bodies from their projections. *Discrete and Computation Geometry*, 4:205–237.

Lederman, S. J. and Browse, R. A. (1988). The physiology and psychophysics of touch. In Dario, P., editor, *Sensors and Sensory Systems for Advanced Robots*, pages 71–91, Berlin; Heidelberg; New York. Springer Verlag.

Li, S.-Y. R. (1988). Reconstruction of polygons from projections. *Information Processing Letters*, 28:235–240.

Lindenbaum, M. and Bruckstein, A. (1991). Parallel strategies for geometric probing. In *Proc. 1991 IEEE Intl. Conf. on Robotics and Automation*, pages 392–397.

Lindenbaum, M. and Bruckstein, A. M. (1994). Blind approximation of planar convex sets. *IEEE Trans. on Robotics and Automation*, 10(4):517–529.

Lynch, K. M. (1997). *Nonprehensile Robotic Manipulation: Controllability and Planning*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Lynch, K. M., Shiroma, N., Arai, H., and Tanie, K. (1998). The roles of shape and motion in dynamic manipulation: The butterfly example. In *Proc. 1998 IEEE Intl. Conf. on Robotics and Automation*.

Marigo, A., Chitour, Y., and Bicchi, A. (1997). Manipulation of polyhedral parts by rolling. In *Proc. 1997 IEEE Intl. Conf. on Robotics and Automation*, pages 2992–2997.

Markenscoff, X., Ni, L., and Papadimitriou, C. (1990). The geometry of grasping. *Intl. J. of Robotics Research*, 9(1):61–74.

Mason, M. T. (1982). *Manipulator Grasping and Pushing Operations*. PhD thesis, AI-TR-690, Artificial Intelligence Laboratory, MIT.

Mason, M. T. (1985). The mechanics of manipulation. In *Proc. 1985 IEEE Intl. Conf. on Robotics and Automation*, pages 544–548, St. Louis.

Mishra, B., Schwartz, J. T., and Sharir, M. (1987). On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558.

Montana, D. J. (1988). The kinematics of contact and grasp. *Intl. J. of Robotics Research*, 7(3):17–32.

Montana, D. J. (1995). The kinematics of multi-fingered manipulation. *IEEE Trans. on Robotics and Automation*, 11(4):491–503.

Nagata, K., Keino, T., and Omata, T. (1993). Acquisition of an object model by manipulation with a multifingered hand. In *Proc. 1993 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, volume 3, pages 1045–1051, Osaka, Japan.

Nguyen, V.-D. (1988). Constructing force-closure grasps. *Intl. J. of Robotics Research*, 7(3):3–16.

Nijmeijer, H. and van der Schaft, A. (1990). *Nonlinear Dynamical Control Systems*. Springer Verlag, Berlin; Heidelberg; New York.

Okamura, A. M., Costa, M. A., Turner, M. L., Richard, C., and Cutkosky, M. R. (1999). Haptic surface exploration. In Corke, P. and Trevelyan, J., editors, *Experimental Robotics VI*, Sydney, Australia. Springer Verlag.

Okamura, A. M. and Cutkosky, M. R. (1999). Haptic exploration of fine surface features. In *Proc. 1999 IEEE Intl. Conf. on Robotics and Automation*, pages 2930–2936, Detroit, Michigan.

Okamura, A. M., Smaby, N., and Cutkosky, M. R. (2000). An overview of dexterous manipulation. In *Proc. 2000 IEEE Intl. Conf. on Robotics and Automation*, pages 255–262, San Fransisco, California.

Paljug, E., Yun, X., and Kumar, V. (1994). Control of rolling contacts in multi-arm manipulation. *IEEE Trans. on Robotics and Automation*, 10(4):441–452.

Peshkin, M. A. and Sanderson, A. C. (1988). Planning robotic manipulation strategies for sliding objects. *IEEE J. of Robotics and Automation*, 4(5).

Ponce, J., Sullivan, S., Sudsang, A., Boissonnat, J.-D., and Merlet, J.-P. (1997). On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Intl. J. of Robotics Research*, 16(1):11–35.

Rao, A., Kriegman, D., and Goldberg, K. (1994). Complete algorithms for feeding polyhedral parts using pivot grasps. Technical Report UU-CS-1994-49, Utrecht University. ftp://ftp.cs.ruu.nl/pub/RUU/CS/techreps/CS-1994/1994-49.ps.gz.

Rao, A., Kriegman, D., and Goldberg, K. (1995). Complete algorithms for reorienting polyhedral parts using a pivoting gripper. In *Proc. 1995 IEEE Intl. Conf. on Robotics and Automation*.

Rao, A. S. and Goldberg, K. Y. (1994). Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *Intl. J. of Robotics Research*, 13(1):16–37.

Rizzi, A. A. and Koditschek, D. E. (1993). Further progress in robot juggling: The spatial two juggle. In *Proc. 1993 IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 919–924, Atlanta, Georgia.

Roberts, K. S. (1990). Robot active touch exploration: Constraints and strategies. In *Proc. 1990 IEEE Intl. Conf. on Robotics and Automation*, pages 980–985.

Romanik, K. (1995). Geometric probing and testing – a survey. Technical Report 95-42, DIMACS Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University.

Salisbury, J. K. (1982). *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Department of Mechanical Engineering, Stanford University.

Salisbury, K. (1987). Whole arm manipulation. In *Proc. Fourth Intl. Symp. on Robotics Research*, pages 183–189, Santa Cruz, California.

Santaló, L. A. (1976). *Integral Geometry and Geometric Probability*, volume 1 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, MA.

Schneiter, J. L. and Sheridan, T. B. (1990). An automated tactile sensing strategy for planar object recognition and localization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(8):775–786.

Shimoga, K. B. (1996). Robot grasp synthesis algorithms: A survey. *Intl. J. of Robotics Research*, 15(3):230–266.

Siegel, D. M. (1991). Finding the pose of an object in a hand. In *Proc. 1991 IEEE Intl. Conf. on Robotics and Automation*, pages 406–411.

Skiena, S. S. (1989). Problems in geometric probing. *Algorithmica*, 4(4):599–605.

Speeter, T. H. (1990). A tactile sensing system for robotic manipulation. *Intl. J. of Robotics Research*, 9(6):25–36.

Sudsang, A., Ponce, J., and Srinivasa, N. (2000). In-hand manipulation: Geometry and algorithms. *Algorithmica*, 26(4):466–493.

Teichmann, M. and Mishra, B. (2000). Reactive robotics I: Reactive grasping with a modified gripper and multi-fingered hands. *Intl. J. of Robotics Research*, 19(7):697–708.

Trinkle, J. C., Abel, J. M., and Paul, R. P. (1988). An investigation of frictionless enveloping grasping in the plane. *Intl. J. of Robotics Research*, 7(3):33–51.

Trinkle, J. C. and Hunter, J. J. (1991). A framework for planning dexterous manipulation. In *Proc. 1991 IEEE Intl. Conf. on Robotics and Automation*, pages 1245–1251.

Trinkle, J. C. and Paul, R. P. (1990). Planning for dexterous manipulation with sliding contacts. *Intl. J. of Robotics Research*, 9(3):24–48.

Trinkle, J. C., Ram, R. C., Farahat, A. O., and Stiller, P. F. (1993). Dexterous manipulation planning and execution of an enveloped slippery workpiece. In *Proc. 1993 IEEE Intl. Conf. on Robotics and Automation*, pages 442–448.

Whitcomb, L. L., Rizzi, A. A., and Koditschek, D. E. (1993). Comparative experiments with a new adaptive controller for robot arms. *IEEE Trans. on Robotics and Automation*, 9(1):59–70.

Wiegley, J., Goldberg, K., Peshkin, M., and Brokowski, M. (1996). A complete algorithm for designing passive fences to orient parts. In *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*.

Yoshikawa, T., Yokokohji, Y., and Nagayama, A. (1993). Object handling by three-fingered hands using slip motion. In *Proc. 1993 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 99–105, Yokohama, Japan.

Zimmer, G. (1994). State observation by on-line minimization. *Intl. J. of Control*, 60(4):595–606.

Zumel, N. B. (1997). *A Nonprehensile Method for Reliable Parts Orienting*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.