Dynamic Shape Reconstruction Using Tactile Sensors

Mark Moll mmoll@cs.cmu.edu
Michael A. Erdmann me@cs.cmu.edu
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract—**We present new results on reconstruction of the shape and motion of an unknown object using tactile sensors without requiring object immobilization. A robot manipulates the object with two flat palms covered with tactile sensors. We model the full dynamics and prove local observability of the shape, motion and center of mass of the object based on the motion of the contact points as measured by the tactile sensors.**

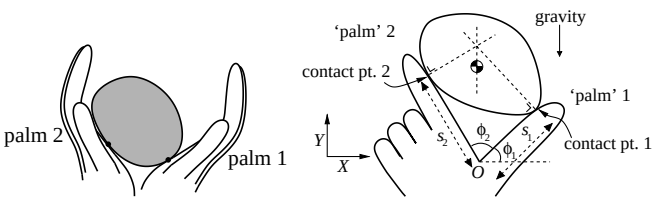
Keywords— **tactile sensing, manipulation, shape reconstruction, observability**

I. Introduction

Robotic manipulation of objects of unknown shape and weight is very difficult. To manipulate an object reliably a robot typically requires precise information about the object's shape and mass properties. Humans, on the other hand, seem to have few problems with manipulating objects of unknown shape and weight. For example, Klatzky et al. [24] showed that blindfolded human observers identified 100 common objects with over 96% accuracy, in only 1 to 2 seconds for most objects. 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. In this paper we present a model that integrates manipulation and tactile sensing. We derive equations for the shape and motion of an unknown object as a function of the motion of the manipulators and the sensor readings.

Figure 1 illustrates the basic idea. There are two palms that each have one rotational degree of freedom at the point where they connect, allowing the robot to change the angle between palm 1 and palm 2 and between the palms and the global frame. As the robot changes the palm angles it keeps track of the contact points through tactile elements on the palms.

This work was supported in part by the NSF under grant IIS-9820180.



(a) Two palms (b) Two fingers

Fig. 1. Two possible arrangements of a smooth convex object resting on palms that are covered with tactile sensors.

In previous work [29] we showed that if we assume quasistatic dynamics we can reconstruct the shape and motion of unknown objects. Our experimental results established the feasibility of this approach. Our long-term goal is to smoothly interlace tactile sensing and manipulation. We wish our robots to manipulate objects dynamically without requiring prehension, immobilization or artificially constrained motions. To this end we have been investigating the observability of shape and motion with full dynamics. In this paper we will prove that the shape and motion are indeed observable.

II. 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.

Probing. Detecting information about an object with sensors can be phrased in a purely geometric way. Sensing is then often called *probing*. Most of the research in this area concerns the reconstruction of polygons or polyhedra using different probe models, but Lindenbaum and Bruckstein [25] gave an approximation algorithm for *arbitrary* planar convex shapes using *line probes*. With a line probe a line is moved from infinity until it touches the object. Each probe reveals a tangent line to the object. Lindenbaum and Bruckstein showed that for an object with perimeter L no more than $O(\sqrt{L/\varepsilon}\log\frac{L}{\varepsilon})$ probes are needed to get an approximation error of ε.

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 [27, 28] nonprehensile manipulation takes the form of pushing an object in the plane to reduce uncertainty about the object's pose.

One of the first papers in palmar manipulation is [39]. Paljug et al. [36] investigated the problem of multi-arm manipulation. Erdmann [9] showed how to manipulate a known object with two palms. Zumel [49] described a palmar system like the one shown in figure 1(b) (but without tactile sensors) that can orient known polygonal parts.

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. In order to grasp an object we need to understand the kinematics of contact. Independently, Montana [31] and Cai and Roth [3, 4] derived the relationship between the relative motion of two objects and the motion of their contact point. In [32] these results are extended to multi-fingered manipulation. Kao and Cutkosky [22] presented a method for dexterous manipulation with sliding fingers.

Trinkle and colleagues [44–46] have 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. [47] do not assume frictionless contacts and show how to regrasp an object using quasistatic slip motion. Nagata et al. [33] describe a method of repeatedly regrasping an object to build up a model of its shape.

In [43] an algorithm is presented that determines a good grasp for an unknown object using a parallel-jaw gripper equipped with some light beam sensors. This paper presents a tight integration of sensing and manipulation. Recently, Jia [18] showed how to achieve an antipodal grasp of a curved planar object with two fingers.

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 [41] and Ellis [8] developed methods for determining sensor paths to solve the first problem. In Siegel [42] a different approach is taken: the pose of an object is determined by using an enveloping grasp. Jia and Erdmann [19] proposed a 'probing-style' solution: they determined possible poses for polygons from a finite set of possible poses by point sampling. Keren et al. [23] proposed a method for recognizing three-dimensional objects using curve invariants. Jia and Erdmann [20] investigated the problem of determining not only the pose, but also the motion of a known object. The pose and motion of the object are inferred simultaneously while a robotic finger pushes the object.

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 [13] described a system that showed that very little information is necessary to reconstruct an unknown shape. With some parametrized shape models, a large variety of shapes can still be characterized. In [11], for instance, results are given for recovering generalized cylinders. In [7] tactile data are fit to a general quadratic form. Finally, [37] proposed a tactile exploration method for polyhedra.

Allen and Michelman [1] presented methods for exploring shapes in three stages, from coarse to fine: grasping by containment, planar surface exploring and surface contour following. Montana [31] describes a method to estimate curvature based on a number of probes. Montana also presents a control law for contour following. Charlebois et al. [5, 6] introduced two different tactile exploration methods: one uses Montana's contact equations and one fits a B-spline surface to the contact points and normals obtained by sliding multiple fingers along a surface.

Marigo et al. [26] showed how to manipulate a known polyhedral part by rolling between the two palms of a parallel-jaw gripper. Recently, Bicchi et al. [2] extended these results to tactile exploration of unknown objects with a parallel-jaw gripper equipped with tactile sensors. A different approach is taken by Kaneko and Tsuji [21], who try to recover the shape by pulling a finger over the surface. This idea has also been explored by Russell [38]. In [35] the emphasis is on detecting fine surface features such as bumps and ridges.

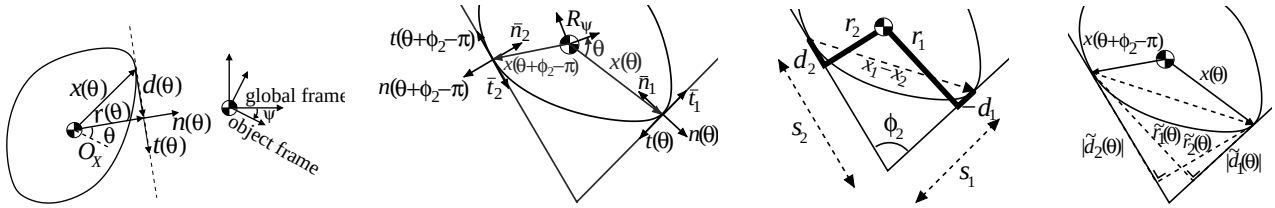
Much of our work builds forth on [10] and [29]. Erdmann derived the shape of an unknown object with an unknown motion as a function of the sensor values. In [29] we restricted the motion of the object: we assumed quasistatic dynamics and we assumed there was no friction. Only gravity and the contact forces were acting on the object. As a result the shape could be recovered with fewer sensors than if the motion of the object had been unconstrained. In this paper we remove the quasistatic dynamics assumption and show that the shape and motion of an unknown planar object are still observable with just two palms.

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. For an overview of sensing technologies, see e.g. [16] and [14]. In our own experiments [29] we relied on off-the-shelf components. The actual tactile sensors were touchpads as found on many notebooks. Most touchpads use capacitive technology, but the ones we used were based on force-sensing resistors, which are less sensitive to electrostatic contamination.

III. Notation

We will use the same notation as in [29]. Figure 1(b) 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.

Frames. A useful tool for recovering the shape of the object will be the radius function (see e.g. [40]). Figure 2(a) 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



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

(b) The different coordinate frames

(c) Dependencies between sensor values, support functions and angle between the palms

(d) The generalized contact support functions.

Fig. 2. The notation illustrated.

$\mathbf{t}(\theta)$ 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:

$$\bar{\mathbf{n}}_1 = (-\sin\phi_1, \cos\phi_1)^T \text{ and } \bar{\mathbf{n}}_2 = (\sin\phi_{12}, -\cos\phi_{12})^T$$
$$\bar{\mathbf{t}}_1 = (\cos\phi_1, \sin\phi_1)^T \qquad \bar{\mathbf{t}}_2 = -(\cos\phi_{12}, \sin\phi_{12})^T$$

Here, $\phi_{12} = \phi_1 + \phi_2$. 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. 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 2(b). From these relationships it follows that

$$\theta = \phi_1 - \psi - \tfrac{\pi}{2} \tag{1}$$

Differentiation. We will use '˙' 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 distance between $\ell(\theta)$ and the object origin:

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

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 signed distance of the contact point $x(\theta)$ to the foot of the supporting line $\ell(\theta)$:

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

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. Below we drop the function arguments where it doesn't 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. $r_2\mathbf{n}_2$ for $r(\theta+\phi_2-\pi)\mathbf{n}(\theta+\phi_2-\pi)$. Note that $r'(\theta) = -d(\theta)$, so it is sufficient to reconstruct the radius function. If the object is in two-point contact, $d(\theta)$ is redundant in another way as well. We can write the two-point contact constraint in terms of the contact support function:

$$(s_1 + d_1)\bar{\mathbf{t}}_1 + r_1\bar{\mathbf{n}}_1 = (-s_2 + d_2)\bar{\mathbf{t}}_2 + r_2\bar{\mathbf{n}}_2 \tag{2}$$

Solving this constraint for d_1 and d_2 we get:

$$d_1 = \tfrac{r_1\cos\phi_2 + r_2}{\sin\phi_2} - s_1 \quad \text{and} \quad d_2 = -\tfrac{r_2\cos\phi_2 + r_1}{\sin\phi_2} + s_2 \tag{3}$$

See also figure 2(c). 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 the expressions above.

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:

$$\tilde{r}_1(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta+\phi_2-\pi)\big)\cdot\mathbf{n}(\theta) \tag{4}$$
$$\tilde{d}_1(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta+\phi_2-\pi)\big)\cdot\mathbf{t}(\theta) \tag{5}$$

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

$$\tilde{r}_2(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta+\phi_2-\pi)\big)\cdot\mathbf{n}(\theta+\phi_2-\pi) \tag{6}$$
$$\tilde{d}_2(\theta) = \big(\mathbf{x}(\theta) - \mathbf{x}(\theta+\phi_2-\pi)\big)\cdot\mathbf{t}(\theta+\phi_2-\pi) \tag{7}$$

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

$$\begin{matrix}\tilde{r}_1 = s_2\sin\phi_2 \\ \tilde{d}_1 = s_2\cos\phi_2 - s_1\end{matrix} \text{ and } \begin{matrix}\tilde{r}_2 = -s_1\sin\phi_2 \\ \tilde{d}_2 = s_1\cos\phi_2 - s_2\end{matrix} \tag{8}$$

These equalities can be obtained by inspection from figures 1(b) and 2(d). By differentiating the generalized contact support functions with respect to time we can obtain the following two expressions for the radii of curvature [30]:

$$v_1 = -\tfrac{\dot{\tilde{r}}_2 + (\dot{\theta}+\dot{\phi}_2)\tilde{d}_2}{\dot{\theta}\sin\phi_2} \quad \text{and} \quad v_2 = -\tfrac{\dot{\tilde{r}}_1 + \dot{\theta}\tilde{d}_1}{(\dot{\theta}+\dot{\phi}_2)\sin\phi_2} \tag{9}$$

So we can observe the curvature at the contact points if we can derive an expression for $\dot{\theta}$ as a function of sensor values and palm angles. Equivalently, we can derive an expression for $\dot{\psi}$, since it follows from equation 1 that $\dot{\theta} = \dot{\phi}_1 - \dot{\psi}$.

IV. Dynamic Shape Reconstruction

Below we will show that it is possible to simultaneously observe the shape and motion. In order to do that 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:

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}) + \tau_1\mathbf{g}_1(\mathbf{q}) + \tau_2\mathbf{g}_2(\mathbf{q}), \tag{10}$$
$$\mathbf{y} = \mathbf{h}(\mathbf{q}) \tag{11}$$

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. The output function returns (a function of) the sensor readings. The vector

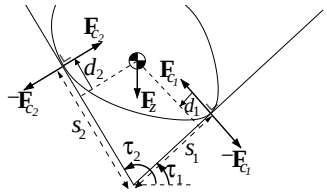


Fig. 3. Forces acting on the palms and the object.

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 10 and 11 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. Moreover, an observer should in general be able to handle noise in the output as well. For more on nonlinear control and nonlinear observers see, e.g., [17] and [34].

V. Equations of Motion

The dynamics of our simple model are very straightforward. We assume the effect of gravity on the palms is negligible and that 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

$$\tau_{c_1} = (\mathbf{R}\mathbf{x}_1) \times \mathbf{F}_{c_1} = -f_{c_1} d_1 \tag{12}$$

$$\tau_{c_2} = (\mathbf{R}\mathbf{x}_2) \times \mathbf{F}_{c_2} = -f_{c_2} d_2 \tag{13}$$

Although we can show that the system is locally observable for arbitrary motions of the palms [30], we restrict ourselves to the case where the palms move at a constant rate. This will simplify not only the analysis, but also the construction of 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. [12] and Zimmer [48]) rely on Lie derivatives of the drift field. This means that the drift field needs to be differentiable with respect to the state variables. If we want to use such an observer for our system we have to constrain the motion of the palms to make the drift field differentiable by restricting them to move at a constant rate, i.e., $\alpha_1 = \alpha_2 = 0$. 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 moment of inertia of the object. Under these assumptions the dynamics of the system are then described by the following equations (see also figure 3):

$$m\mathbf{a}_0 = \mathbf{F}_z + \mathbf{F}_{c_1} + \mathbf{F}_{c_2} \tag{14}$$

$$I_0\alpha_0 = \tau_{c_1} + \tau_{c_2} = -f_{c_1} d_1 - f_{c_2} d_2,\ \ f_{c_1},\ f_{c_2} \geq 0 \tag{15}$$

$$0 = \tau_1 - f_{c_1} s_1 \tag{16}$$

$$0 = \tau_2 + f_{c_2} s_2 \tag{17}$$

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. Solving for $\mathbf{a}_0$ and α_0, we get

$$\mathbf{a}_0 = \tfrac{\tau_1}{m s_1}\bar{\mathbf{n}}_1 - \tfrac{\tau_2}{m s_2}\bar{\mathbf{n}}_2 + \mathbf{g} \tag{18}$$

$$\alpha_0 = -\tfrac{\tau_1}{m\rho^2 s_1} d_1 + \tfrac{\tau_2}{m\rho^2 s_2} d_2 \tag{19}$$

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 can observe the radius of gyration by making it a state variable. This is described in [30]. The mass properties of the palms are assumed to be known.

VI. Local Observability

We will now rewrite the constraints on the shape and motion of the object in the form of equation 10. We will introduce the variables ω_0, ω_1 and ω_2 to denote $\dot{\psi}$, $\dot{\phi}_1$ and $\dot{\phi}_2$, respectively. We can write the position constraint on contact point 1 as

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

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. The acceleration constraint can be turned into the following constraint on the curvature at contact point 1:

$$v_1 = \tfrac{2\dot{s}_1\omega_1 - \omega_0^2 r_1 - \mathbf{a}_0\cdot\bar{\mathbf{n}}_1 + \alpha_0 d_1}{\omega_1^2 - \omega_0^2} \tag{20}$$

From before (equations 8 and 9) we had:

$$\begin{aligned} v_1 &= -\tfrac{\dot{\tilde{r}}_2 + (\dot{\theta}+\dot{\phi}_2)\tilde{d}_2}{\dot{\theta}\sin\phi_2} \\ &= -\tfrac{(-\dot{s}_1\sin\phi_2 - s_1\omega_2\cos\phi_2) + (\omega_{12}-\omega_0)(s_1\cos\phi_2 - s_2)}{(\omega_1-\omega_0)\sin\phi_2}, \end{aligned} \tag{21}$$

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

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

Similarly we can derive an expression for $\dot{s}_2$. Note that the control inputs τ_1 and τ_2 are 'hidden' inside $\mathbf{a}_0$ and α_0. The expression $\mathbf{a}_0\cdot\bar{\mathbf{n}}_1 - \alpha_0 d_1$ can be rewritten using equations 18 and 19 as

$$\mathbf{a}_0\cdot\bar{\mathbf{n}}_1 - \alpha_0 d_1 = \tfrac{(\rho^2 + d_1^2)\tau_1}{m\rho^2 s_1} + \tfrac{(\rho^2\cos\phi_2 - d_1 d_2)\tau_2}{m\rho^2 s_2} + g\cos\phi_1.$$

Let $\mathbf{q} = (r_1, r_2, \omega_0, s_1, s_2, \phi_1, \phi_2)^T$ be our state vector. Recall from section III that d_1 and d_2, can be written in terms of 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. Since τ_1 and τ_2 appear linearly in the previous equation, our system fits the format of equation 10. The drift

vector field is

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

and the input vector fields are

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

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

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

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

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

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

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

Then a system of the form described by equation 10 is locally observable at state $\mathbf{q}$ if $\dim d\mathcal{O}(\mathbf{q}) = n$, where n is the dimensionality of the state space [15].

The differentials of the components of the output function are

$$ds_1 = (0, 0, 0, 1, 0, 0, 0)$$
$$ds_2 = (0, 0, 0, 0, 1, 0, 0)$$
$$d\phi_1 = (0, 0, 0, 0, 0, 1, 0)$$
$$d\phi_2 = (0, 0, 0, 0, 0, 0, 1).$$

To determine whether the system is observable we need to compute the differentials of at least three Lie derivatives. In general $dL_{\mathbf{f}}s_1$, $dL_{\mathbf{f}}s_2$, $dL_{\mathbf{f}}L_{\mathbf{f}}s_1$ and the differentials above will span the observability codistribution $d\mathcal{O}$. Note that we only used the drift vector field to show local observability, since—as we mentioned before—this will facilitate observer design. The results above show that in general we will be able to observe the shape of an unknown object.

Let us now consider three special cases of the above system: (1) $\omega_1 = 0$, palm 1 is fixed, (2) $\omega_2 = 0$, both palms move at the same rate and (3) $\omega_1 = \omega_2 = 0$, both palms are fixed. These special cases are described in more detail in [30]. We will summarize the results below:

$\omega_1 = 0$: We can eliminate ϕ_1 from the state vector, because it is now a known constant. It can be shown that $L_{\mathbf{f}}L_{\mathbf{f}}s_1 = 0$. Fortunately, the Lie derivatives ds_1, ds_2, $d\phi_1$, $dL_{\mathbf{f}}s_1$, $dL_{\mathbf{f}}s_2$ and $dL_{\mathbf{f}}L_{\mathbf{f}}s_2$ still span the observability codistribution.

$\omega_2 = 0$: Now we can eliminate ϕ_2 from the state vector. The same Lie derivatives as in the general case can be used to prove observability.

$\omega_1 = \omega_2 = 0$: Both ϕ_1 and ϕ_2 are now eliminated as state variables. It can be shown that in this case the system is no longer locally observable by taking repeated Lie derivatives of the drift vector field alone; we need to consider the control vector fields as well. The differentials ds_1, ds_2, $d\phi_1$, $dL_{\mathbf{f}}s_1$, $dL_{\mathbf{f}}s_2$ and $dL_{\mathbf{g}_1}s_1$ generally span the observability codistribution.

In this last case 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 10. 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 for this case it would have very limited practical value.

VII. Future Work

In this paper we have set up a framework for reconstructing the shape of an unknown smooth convex planar shape using two tactile sensors. We have shown that the shape and motion of such a shape are locally observable. We are currently constructing an observer to be used in the experimental setup described in [29]. This setup proved to be very useful in analyzing the quasistatic case. In this paper we take into account second order effects. We therefore expect that the performance of our experimental system will be greatly enhanced by using an observer.

To make the model more realistic we plan to model friction as well. This may not fundamentally change the system: as long as the contact velocities are non-zero the contact force vectors are just rotated proportional to the friction angle. We hope to be able to reconstruct the value of the friction coefficient using a nonlinear observer.

Finally, 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 come up with a good approximation. We can compute a lower bound on the shape by computing the convex hull of the curves. An upper bound can be computed by computing the largest shape that fits inside the tangent

planes without intersecting them. 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. Instead, we intend to generalize the dynamics results presented in this paper.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Ellis, R. E. (1992). Planning tactile recognition in two and three dimensions. *Intl. J. of Robotics Research*, 11(2):87–111.

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

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

[11] 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.

[12] 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.

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

[14] Grupen, R. A., Henderson, T. C., and McCammon, I. D. (1989). A survey of general-purpose manipulation. *Intl. J. of Robotics Research*, 8(1):38–62.

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

[16] 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.

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

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

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

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

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

[22] Kao, I. and Cutkosky, M. R. (1992). Quasistatic manipulation with compliance and sliding. *Intl. J. of Robotics Research*, 11(1):20–40.

[23] 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.

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

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

[26] 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.

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

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

[29] Moll, M. and Erdmann, M. A. (2001a). Reconstructing shape from motion using tactile sensors. In *Proc. 2001 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Maui, HI.

[30] Moll, M. and Erdmann, M. A. (2001b). Shape reconstruction in a planar dynamic environment. Technical Report CMU-CS-01-107, Dept. of Computer Science, Carnegie Mellon University.

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

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

[33] 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.

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

[35] 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.

[36] 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.

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

[38] Russell, R. A. (1992). Using tactile whiskers to measure surface contours. In *Proc. 1992 IEEE Intl. Conf. on Robotics and Automation*, volume 2, pages 1295–1299, Nice, France.

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

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

[41] 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.

[42] 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.

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

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

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