

- [27] R. D. Howe, I. Kao, and M. R. Cutkosky, "The sliding of robot fingers under combined torsion and shear loading," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, PA, Apr. 1988, pp. 103–105.
- [28] B.-R. Zuo and W.-H. Qian, "A force-closure test for soft multi-fingered grasps," *Sci. China*, ser. E, vol. 41, no. 1, pp. 62–69, Feb. 1998.
- [29] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Nice, France, May 1992, pp. 2290–2295.
- [30] Z. X. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE J. Robot. Autom.*, vol. 4, no. 1, pp. 32–44, Feb. 1988.

Localization of Curved Parts Through Continual Touch

Yan-Bin Jia

Abstract—We describe a simple system that localizes two-dimensional curved shapes through touch sensing, offering computational and experimental studies. The idea lies in determining the placement of a manipulator on a curved object during some special motion—rolling. A geometric algorithm is introduced to locate the boundary segment traced out by their contact using tactile data. Both completeness and local convergence have been established. The algorithm is asymptotically as efficient as evaluating the object's perimeter through numerical integration. For implementation, a two-axis force/torque sensor has been designed to realize contact sensing. Functioning like a "wrist," the sensor is calibrated over the ratio between the bending and twisting moments, eliminating the need for known weights. A simple geometry-based control strategy is devised to implement the rolling motion. Experiments have been conducted with an Adept Cobra 600 manipulator.

Index Terms—Curves, kinematics of rolling, parts localization, solid mechanics, touch sensing.

I. INTRODUCTION

Parts sensing and orienting involve determining the position and orientation of a part whose shape is often known. Not to disturb the part, parts sensing [27], [38], [44] invokes geometric algorithms to process sensor data which serve as constraints on the part. A practical drawback is that the robot is hardly reactive to sensing errors or minor disturbances on the part. A vision system, meanwhile, is unable to handle occlusions. Since a part is often machined according to some computer-aided-design (CAD) model, an image also contains redundant information that could become a source of errors and inefficiency in the process.

With performance guaranteed by mechanical analysis, parts orienting carries out operations, such as vibration [17], tray-tilting [11], parallel-jaw gripping [7], [14], pushing [2], [32], microelectromechanical systems (MEMS) actuation [5], or fixturing [8]. These methods trade sensing (and, thus, all sensor errors) for gained task robustness. Nevertheless, the tradeoff often requires special engineering of the task environment, which increases the cost while decreasing modularity and efficiency.

Manuscript received January 18, 2004; revised July 22, 2004. This work was supported in part by Iowa State University, and in part by the National Science Foundation under CAREER Award IIS-0133681. This paper was recommended for publication by Associate Editor Z. Li and Editor H. Arai upon evaluation of the reviewers' comments.

The author is with the Department of Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: jia@cs.iastate.edu).

Digital Object Identifier 10.1109/TRO.2005.844675

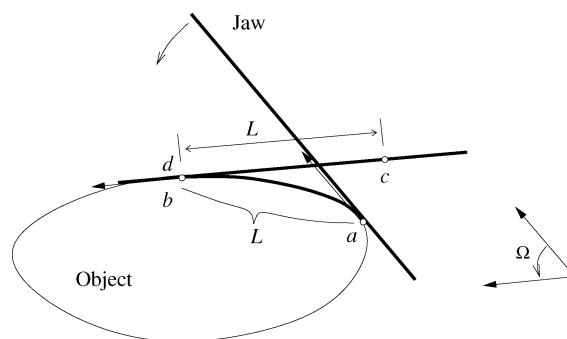


Fig. 1. Localizing a jaw on a motionless object through rolling. During the same period of rolling, the contact point moves from a to b on the object and from c to d on the jaw.

Knowledge about the geometry of a part facilitates its localization through the exploration of tactile information. In industrial automation, a workpiece is typically localized by finding the optimal registration of some measured points onto a given CAD model [19], [31], [33]. The general scheme iteratively improves on a transformation in order to minimize some least-squares error function and also on the registration of the measured data points. Though involved numerical routines have been developed, the local nature of nonlinear optimization guarantees neither completeness nor efficiency.

In most tasks, parts only need to be localized relative to the robot. In grasping, for instance, if the hand is already in contact with an object, then it needs to only know where the fingers are placed on the object rather than where the object is located (in the world coordinates). The human hand often calibrates itself by moving its fingers on the object's surface so as to "feel" the change of geometry.

To emulate such ability of "feeling," the robotic hand needs to be equipped with a force/torque or tactile array sensor. Such a sensor plays an important role in the dynamic integration of sensing into manipulation.

While a combination of tactile, force, and position sensing carries the promise of enhancing the flexibility and robustness of robotic manipulation [22], the integration of different control strategies for multiple sensor modalities can become very sophisticated and unreliable. From a minimalist point of view, one sensor modality should be preferred if it yields sufficient information needed for task execution.

The aim of this paper is to demonstrate that parts can be localized with very limited touch sensing plus a little action. By doing this, we hope to gain some insight into enabling the robot to "feel" the geometry and pose of an object. We believe that the retrieval and engineering of such knowledge will become important for skillful task execution in the long term.

Our investigation focuses on parts in curved shapes. A substantial amount of research has dealt with polygonal and polyhedral objects so far. These objects do not have local geometry (except at vertices). Nevertheless, actions and mechanics are inherently differential and subject to local geometric properties of bodies interacting with each other.

The specific problem studied in this paper is posed as follows. A jaw (as shown in Fig. 1) rolls from one location (a) to another (b) on an object. We would like to determine these two locations, thus locating the jaw on the object (i.e., determining the object's relative pose to the jaw). The idea is to measure the angle of rotation (Ω) by the jaw as well as the distance (L) of contact movement on the object boundary. We offer an algorithm that finds the segment traced out by the contact point in Section II.

To determine the rotation, we mount the jaw on an Adept robot which provides angle readings. To detect the location of contact on the jaw, we implement a two-axis force/torque sensor. The sensor is low cost [less than U.S.\$ 300 excluding the data-acquisition (DAQ) board]. Industrial sensors are somewhat expensive for lab experiments, particularly those on small scales. A tactile array sensor could easily cost more than U.S.\$ 10 000. Less expensive force/torque sensors, such as the ATI F/T-16, may be easily broken due to their narrow force and torque ranges.

Section III describes the sensor design and underlying principles, calibration, as well as a geometric strategy to implement the rolling motion. Some experimental results are presented in Section IV. Finally, Section V discusses a number of issues related to the presented work and outlines some future directions.

A. Related Work

The main theme of this work lies in the exploration of shape geometry, touch sensing, and the kinematics of rolling for localization of the manipulator.

Grimson and Lozano-Pérez [16] used tactile measurements to locate and identify a three-dimensional (3-D) polyhedron from a set of known polyhedra. Fearing [12] designed a cylindrical tactile fingertip for contact localization and regrasping. Skekhar *et al.* [40] also employed multiple tactile and pressure sensors to perform feature-based localization. In [21], Howe and Cutkosky carried out dynamic tactile sensing to capture fine surface features.

Kriegman and Ponce [29] applied elimination theory in matching curved 3-D objects with image contours. Allen [3] fit over a large amount of surface data to reconstruct an object's shape. Moll and Erdmann [34] used quasi-dynamic analysis in shape reconstruction and motion estimation from tactile readings.

Salisbury [39] investigated how to determine contact location and orientation from force and moment measurements. This work was extended by Bicchi [4] to infer some global qualities of contact. Principles on measuring contact were also derived by Tsujimura and Yabuta [42] for recognition of various shapes. Brock and Chiu [6] designed a fingertip sensor consisting of four strain gauge half-bridges to measure surface contact location and orientation as well as the center of mass. Zhou *et al.* [45] re-examined fingertip contact sensing with a focus on the elimination of gravitational biasing and error analysis.

Contact forces and locations on a multilink manipulator can be calculated from joint torque readings only. This type of estimation either exploits kinematic constraints only [28], or combines such constraints with linear and nonlinear observers [18], [20]. Contact estimates of this kind may be considerably improved with the integration of tactile information [15].

The two-axis force/torque sensor introduced in Section IV measures contact from the ratio of moments about two axes. It alleviates the need for calibration loads [41], [43], which are a source of errors. Its design is influenced by the work of Abe *et al.* [1] on implementing a three-axis force/torque sensor sensitive to chosen directions only.

Kinematics of point contact between 3-D rigid bodies were derived by Montana [35] and by Cai and Roth [9] as differential equations in response to their relative motion. The special case of rolling was considered by Li and Canny [30] in view of path planning.

In this paper, we use a simple geometry-based strategy to generate the "effect of rolling" on which localization is based. Raibert and Craig [37] described a hybrid position/force control strategy using an internal wrist sensor built on strain gauges to meet manipulator trajectory constraints. Dynamic control of rolling contacts with tactile information was studied by Paljug *et al.* [36] in the context of dexterous multiarm manipulation.

II. GEOMETRIC LOCALIZATION

The manipulator considered by us is a jaw with a straight edge (Fig. 1). It is described as a line segment by the local coordinate u . The jaw is rolling with angular velocity, say, ω , on a stationary object bounded by a curve $\alpha(s)$ with curvature κ . The parameters u and s determine the locations of contact on the jaw and the object, respectively. Their derivatives \dot{u} and \dot{s} with respect to time satisfy the following equations [26]:

$$\dot{s} \|\alpha'(s)\| = -\dot{u} \quad (1)$$

$$\dot{s} \|\alpha'(s)\| \kappa(s) = \omega. \quad (2)$$

Suppose in a time period the contact has moved from a to b on the object and from c to d on the jaw. Integrate (1) and (2) over the same time period

$$\int_a^b \|\alpha'(s)\| ds = -\int_c^d du = L \quad (3)$$

$$\int_a^b \kappa(s) \|\alpha'(s)\| ds = \Omega. \quad (4)$$

Here, L is the arc length traced out by the contact from a to b and Ω is the amount of jaw rotation known to the robot controller. Equation (3) says that the contact travels the same distance on the object and on the jaw. It allows us to measure the arc length L using a touch sensor to be described in Section III. Equation (4) states that the jaw rotation Ω is also the *total curvature* of the boundary segment from a and b .

If we find the starting location a (or the present location b) of contact on the object, its relative pose to the jaw is completely determined. Localization has thus become the geometric problem of *locating a curve segment(s) on α with length L and total curvature Ω* . Such a segment will be referred to as a *feasible segment* and its location as a *feasible location*.

Denote by $\ell(s, t) = \int_s^t \|\alpha'(\xi)\| d\xi$ the arc length of a segment from s to t on α and $\Phi(s, t) = \int_s^t \kappa(\xi) \|\alpha'(\xi)\| d\xi$ the total curvature of the segment. Thus, we would like to simultaneously solve the following two nonlinear equations:

$$\ell(s, t) = L \quad (5)$$

$$\Phi(s, t) = \Omega. \quad (6)$$

For most curves, the integral ℓ has no closed form and needs to be evaluated through numerical integration. Any existing root-finding technique relying on multiple evaluations of ℓ would be very inefficient. Meanwhile, the integral $\Phi(s, t)$ does not have a closed form either, unless it is within $[0, 2\pi)$ (in which case, it is determined by the curve tangents at s and t).

A. Convex Case

We here study the case that the boundary curve α is convex everywhere. Without loss of generality, we assume that the jaw is rolling counterclockwise on the object so the contact moves in the same direction. Hence, $L > 0$ and $\Omega > 0$.

Our idea for solving (5) and (6) is to slide a hypothesized segment $\alpha[s, t]$ with endpoints s and t until it reaches a feasible location. Because the curve speed $\|\alpha'\|$ is rarely constant, we *cannot* march both endpoints of the segment by the same step size h in the curve domain and expect its length to be maintained at L . Instead, we move the two endpoints separately while ensuring that they do not pass the closest location of a feasible segment. The algorithm starts at location $s_0 = 0$

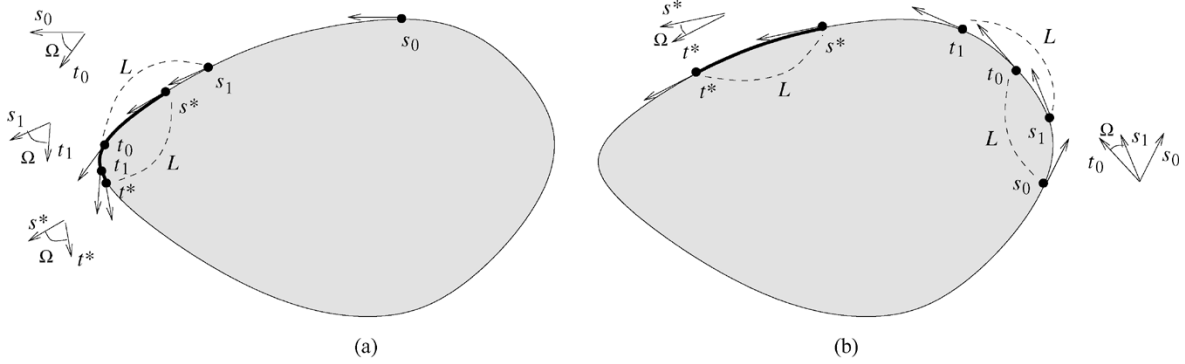


Fig. 2. Two cases of marching s and t : (a) $\Phi(s_i, t_i) = \Omega$ but $\ell(s_i, t_i) > L$; (b) $\ell(s_i, t_i) = L$ but $\Phi(s_i, t_i) > \Omega$. The sequences $\{s_i\}$ and $\{t_i\}$ converge to s^* and t^* , respectively, where $\Phi(s^*, t^*) = \Omega$ and $\ell(s^*, t^*) = L$.

and finds the first point u with $\Phi(s_0, u) = \Omega$ through numerical integration. Then, it generates $t_0, s_1, t_1, s_2, t_2, \dots$ as the following:

$$\begin{aligned} \ell(s_0, u) > L: & \quad \Phi(s_i, t_i) = \Omega \\ \ell(s_{i+1}, t_i) &= L \\ \ell(s_0, u) < L: & \quad \ell(s_i, t_i) = L \\ \Phi(s_{i+1}, t_i) &= \Omega. \end{aligned} \quad (7)$$

Except for a constant-speed curve, t_i and s_{i+1} have to be obtained from t_{i-1} and s_i through numerical integration. As illustrated in Fig. 2, the algorithm repeatedly elongates and contracts the curve segment $\alpha[s, t]$.

Proposition 1: The iteration (7) converges to the first curve segment $\alpha[s^*, t^*]$ with length L and total curvature Ω in the marching direction. The convergence rate is linear and given by the ratio between the curvatures $\kappa(s^*)$ and $\kappa(t^*)$.

A proof of the above proposition can be found in [26]. The procedure is extended in Appendix I to a nonconvex curve for which a possible implementation is discussed in Section V.

B. Completeness and Ambiguities

Almost always, more than one curve segment with length L and total curvature Ω exists on the boundary.¹ To find the next segment satisfying (5) and (6), we step over the found segment $\alpha[s^*, t^*]$ by resetting $s_0 \leftarrow s_i + \epsilon$ for large enough i and small constant $\epsilon > 0$ and repeat the same procedure.

Theorem 2: Let α over the domain $[0, \tau]$ be a simple and closed curve on which only a finite number of segments have length L and total curvature Ω . Suppose any two such feasible segments start more than $\epsilon \cdot \max \|\alpha'\|$ apart. The marching algorithm locates all feasible curve segments up to numerical resolution in $6\tau/h$ steps, where h is the step size.

We refer to [26] for a proof of the above theorem. The overall computation is asymptotically optimal since acquiring the perimeter of α by integration requires τ/h numerical steps.

In our implementation, the step size h is chosen to be 0.000 01. The numerical tolerances on arc length L and total curvature Ω are set to be ± 0.004 and ± 0.0004 , respectively. The minimum separation α between the starting points of feasible segments is set to be 0.5.

In case multiple feasible segments are found, we let the jaw continue rolling for another period of time, tracing out a curve segment immediately following the first one. The length and total curvature of this second segment is often enough to eliminate all of the ambigu-

¹To see this, consider the function of variable s with period τ that is the total curvature of the segment starting at s and having length L . Since the function has the same value at $s = 0$ and $s = \tau$, any total curvature between the maximum and minimum is attained at more than one s value.

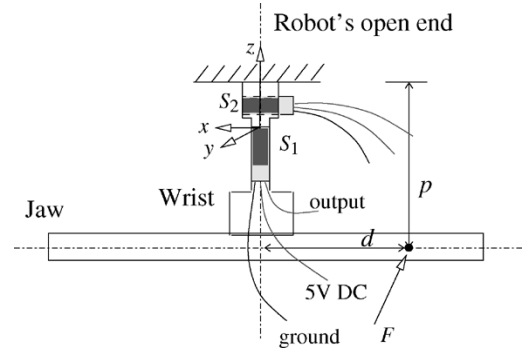


Fig. 3. Two-axis force/torque sensor for contact detection with a rectangular plastic jaw attached. Only the darkly shaded rectangular area at the tip of each chip sensor is sensitive to an applied force.

ties (unless α has some symmetry), as demonstrated by experiments in Section IV.

III. CONTACT SENSING

To implement the localization algorithm in Section II, we have designed a low-cost two-axis force/torque sensor shown in Fig. 3. The main structure of the sensor is an aluminum piece serving the purpose of a wrist which can bend inward/outward and twist about its axis of symmetry. Two chip sensors² S_1 and S_2 are glued to the wrist, one vertically and the other horizontally. Each chip sensor is a half-bridge of two strain gauges with electrical resistance of $900 \pm 150 \Omega$. The two-axis sensor is mounted on an Adept Cobra 600 robot.

Following the strain gauge principle, the voltage variation on each chip sensor is proportional to the variation of the total strain gauge length, which is, in turn, proportional to the stress. Below, we apply solid mechanics to examine how these relationships can be used to locate contact on the jaw.

A. Principle of Sensing

The chip sensor S_1 is sensitive to a bending moment about the horizontal axis of the jaw but not to a twisting moment about its vertical axis. The chip sensor S_2 is sensitive to a twisting moment but not to a bending moment. The coupling effects are small and assumed to be negligible in our analysis.

In Fig. 4(a), the stress at a point in the cross section in the x - y plane under a bending moment M_b on the sensor S_1 is

$$\sigma_1 = -\frac{M_b y}{I_{xx}} \quad (8)$$

²From Bokam Engineering Inc.

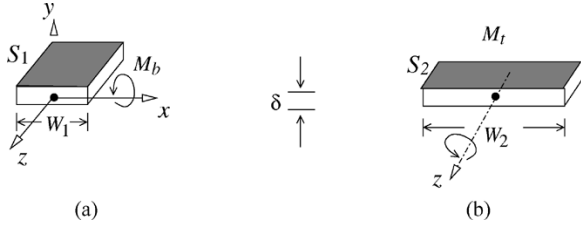


Fig. 4. Top-down view of the two sections of the aluminum wrist where the chip sensors are mounted. The coordinate system is consistent with that in Fig. 3. The dimensions are $W_1 = 0.3175$ cm, $W_2 = 0.635$ cm, and $\delta = 0.0794$ cm.

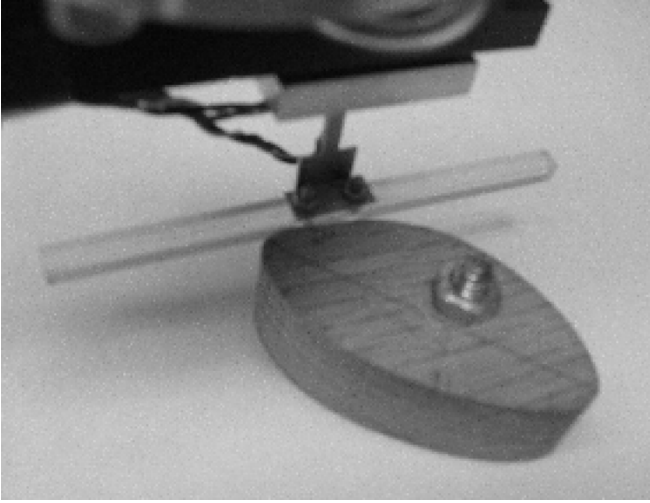


Fig. 5. Setup for calibration (and localization later on).

where I_{xx} is the angular inertia of the cross section about the x -axis [10]. We have

$$I_{xx} = 2 \int_0^{\delta/2} y^2 \cdot W_1 dy = \frac{\delta^3}{12} W_1. \quad (9)$$

Since the chip sensor S_1 is mounted at the boundary of the cross section, $y = \delta/2$ in (8). Substitute this and (9) into that equation

$$\sigma_1 = -\frac{M_b \cdot \frac{\delta}{2}}{W_1 \cdot \frac{\delta^3}{12}} = -\frac{6M_b}{W_1 \delta^2}. \quad (10)$$

In Fig. 4(b), let σ_r be the stress at a point in the cross section of S_2 at the horizontal distance r from the center. Then, we derive the twisting moment

$$M_t = \int_{-W_2/2}^{W_2/2} |r| \cdot \sigma_r \cdot \delta dr = 2 \int_0^{W_2/2} r \sigma_r \cdot \delta dr \quad (11)$$

which has solution $\sigma_r = 12M_t|r|/W_2^3\delta$. The average twisting stress at all points inside S_2 is

$$\sigma_2 = \frac{12M_t \cdot \left(\frac{W_2}{4}\right)}{W_2^3\delta} = \frac{3M_t}{W_2^2\delta}. \quad (12)$$

In our design, $W_2 = 2W_1 = 8\delta$. From (10) and (12), we obtain the ratio between the average twisting stress and the bending stress $\sigma_2/\sigma_1 = -M_t/32M_b$.

Suppose a contact force (of magnitude) F is applied normal to the jaw at distance d from its axis of symmetry and at distance p below the

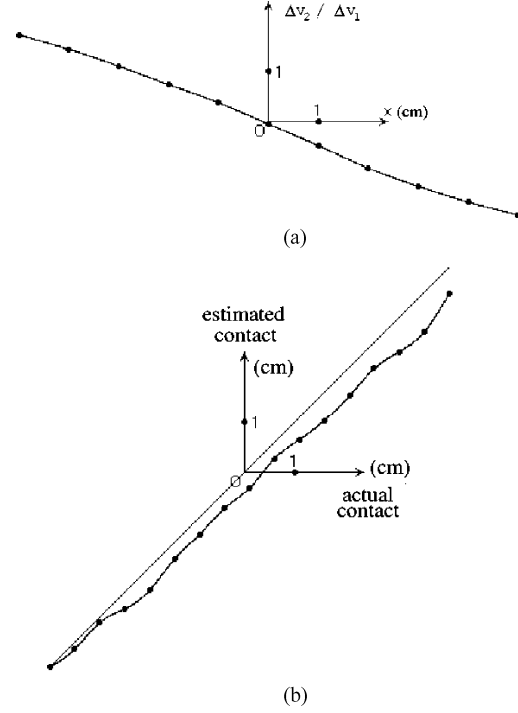


Fig. 6. (a) Spline interpolant for calibration; (b) actual contact movements versus estimated contact movements during one test.

robot's open end (Fig. 3). Then, $M_t = Fd$ and $M_b = Fp$. The ratio between the stresses on S_2 and S_1 is

$$\frac{\sigma_2}{\sigma_1} = -\frac{d}{32p}. \quad (13)$$

The stress ratio σ_2/σ_1 is proportional to the ratio between the variations Δv_2 and Δv_1 from the default readings on S_2 and S_1 . In our experiments, the height p is maintained the same. Thus, we can *measure the horizontal location d of contact on the jaw from the ratio $\Delta v_2/\Delta v_1$ after calibration.*

B. Calibration

Since contact location instead of force is to be measured, sensor calibration can be done *without* the use of weights, eliminating one source of error. To calibrate the two-axis sensor, the jaw in a fixed orientation repeatedly makes contact with an immobilized object as shown in Fig. 5. Before making the next contact, the jaw translates by a fixed distance along the tangential direction. As a result, the contact point on the object boundary does not change but the contact points on the jaw have uniform spacings. During a contact, voltage readings of the sensor S_1 are maintained at some constant level while readings of the sensor S_2 are taken. This is achieved by simple feedback control over the jaw translation in the normal direction.

Fig. 6(a) shows a cubic spline that interpolates 11 ratios between voltage variations on the two chip sensors when the contact distance x from the wrist sensor's axis of symmetry varies from -5 cm to 5 cm with 1-cm increments. The discrepancy between the spline interpolant and a straight line could be attributed to various factors, such as strain distribution over an area on the chip sensor, area contact, nonlinearity of the voltage output, wire strain on the chip sensors, imperfect sensor mounting, strain gauge noise, etc.

To determine the (horizontal) location of contact on the jaw, we need only intersect the horizontal line of an average reading $\Delta v_2/\Delta v_1$ with the calibration spline and read the x coordinate. A test on the accuracy of this spline-based calibration is presented in Fig. 6(b). Measurement

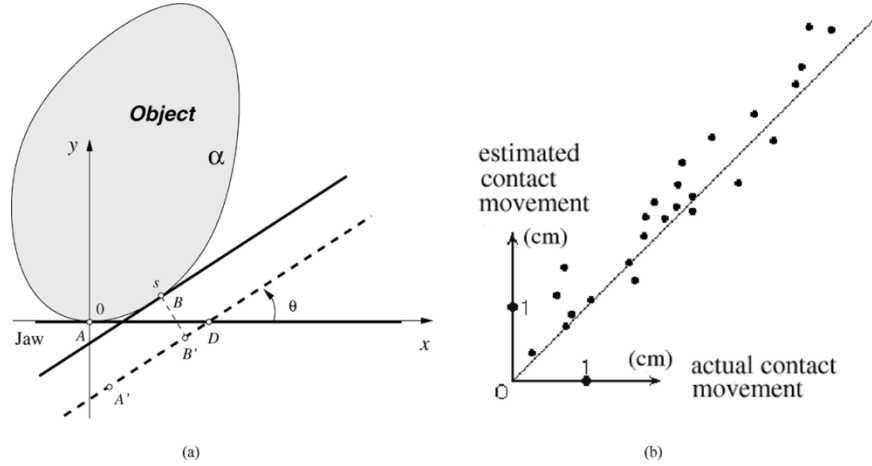


Fig. 7. (a) Rolling strategy; (b) 23 rolling instances with different starting points on one shape. The rotation angle θ per step is 0.2° .

drifts due to time and temperature were found to be relatively small and in one direction.

C. Strategy of Rolling

Rolling of the jaw is a precondition for measuring the distance traced out on the object's boundary through contact according to (3). A complicated force control strategy would not be reliable. Rather, we introduce a simple geometry-based control strategy exploiting the reading variation Δv_1 on the chip sensor S_1 , which reflects the contact force.

The jaw establishes contact with the object until $\Delta v_1 \approx 0.4$ V. It then repeatedly does the following. In each round, the jaw first rotates by a small angle. As it moves further "into" the object or away from it, the voltage readings of S_1 decrease or increase, respectively. It then backs up or translates forward accordingly until Δv_1 returns to around 0.4 V.

Let us justify that the above strategy simulates rolling. We choose the origin to be at the contact location A before a rotation, as shown in Fig. 7(a). There exists an arc-length parameterization $\alpha(u)$ of the object boundary such that $\alpha(0) = A$. The center of rotation on the jaw is at $D = (d, 0)$. Without loss of generality, we assume that $d > 0$. A counterclockwise rotation by an angle θ breaks the contact. The initial contact point A on the jaw has been rotated to the position A' . Next, the jaw translates along the normal direction to re-establish contact at the point $B = \alpha(s)$. The contact tangent has rotated by θ as it moves from A to B .

Because α is unit speed, the contact has "moved" a distance of s from A to B on the object. The contact "movement" on the jaw is

$$\begin{aligned} |A'B'| &= |DA'| - |DB'| \\ &= d + \left(\alpha(s) - \begin{pmatrix} d \\ 0 \end{pmatrix} \right) \cdot \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \\ &= 2d \sin^2 \frac{\theta}{2} + \alpha(s) \cdot \alpha'(s) \\ &= s + O(s^2) \quad (\text{Taylor expansion at } s = 0). \end{aligned}$$

The derivation above uses that $d\theta/ds = \kappa$, where κ is the curvature function, and $\alpha'(0) = (1, 0)^T$. When the rotation angle θ is very small, s is small. Hence, $|A'B'| \approx s$.

Fig. 7(b) compares expected contact movements and their estimates recorded during 23 rolling instances performed on an object in a known orientation.

The aluminum jaw and the object were in contact almost all of the time during the rolling.³ This is because the jaw's upper end had to tilt

³The presented analysis still carries over in this case.

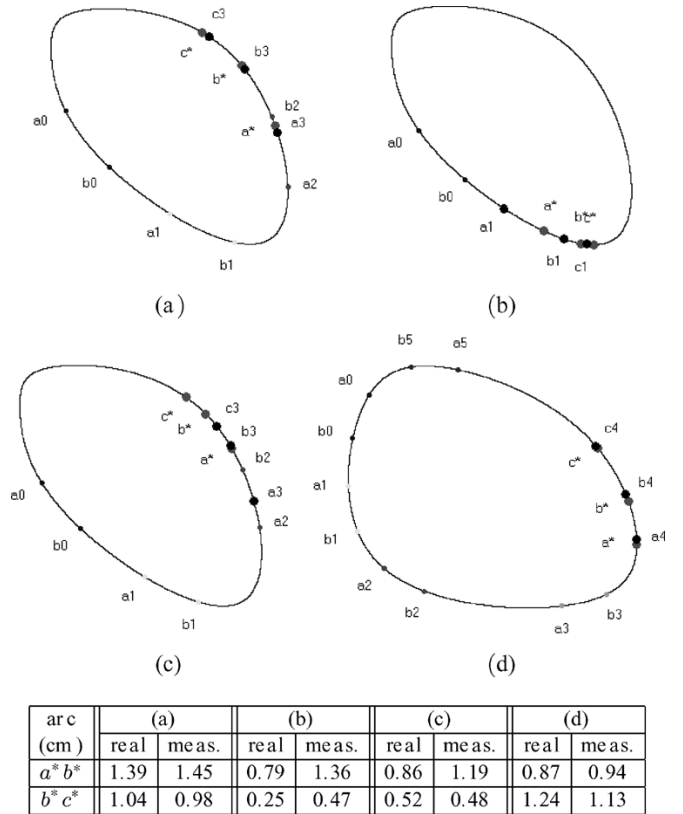


Fig. 8. Localizing contacts on two objects. The same scale in (a) applies to (b)–(d). The experimental setup is as in Fig. 5. The table below the figures lists the actual and measured lengths of each boundary segment traced out by the rolling jaw.

inward to maintain enough contact force to keep the reading variation $\Delta v_1 \approx 0.4$ V. Though the contact force could decrease with a jaw rotation, such decrease was rarely enough to break the contact.

Since the "rolling strategy" is an approximation to pure rolling, the contact point is not expected to move exactly the same distance on the jaw as on the object. Estimation errors were higher in instances where the jaw passed high-curvature portions of the object boundary. Sensor calibration assumes frictionless contact between the jaw and the object, which was not true in the experiments. This contributed to the measurement errors, which were likely also due to calibration inaccuracy, minor vibrations of the robot, etc.

IV. EXPERIMENTS

Experiments were conducted on localizing flat wooden parts in cubic spline shapes. Four instances of localization on two different parts are shown in Fig. 8. In each instance, the jaw made the initial contact at some point a^* on the part boundary, rolled along the boundary before stopping at another point b^* , then rolled again and finally stopped at a third point c^* . The boundary points a^* , b^* , and c^* were not known. The lengths of the two boundary segments $\alpha[a^*, b^*]$ and $\alpha[b^*, c^*]$ were estimated by the sensor under the rolling assumption. Their total curvatures were estimated as the corresponding jaw rotations read by the robot.

In each instance, $\alpha[a_i, b_i]$ represented all of the boundary segments found by the localization algorithm that agreed with the length and total curvature estimates of $\alpha[a^*, b^*]$. The ambiguities were then eliminated by examining the total curvatures of the following boundary segments $\alpha[b_i, c_i]$ whose lengths equaled the length estimate of $\alpha[b^*, c^*]$. The one whose total curvature best matched that of $\alpha[b^*, c^*]$ was then chosen. The initial and final contact locations on the object were then determined accordingly. They were a_3 and c_3 in the instances (a) and (c), a_1 and c_1 in (b), and a_4 and c_4 in (d).

In the instances (a) and (d), the estimates were very close to the actual locations a^* , b^* , c^* . In (b), although the estimates a_1 and b_1 had large errors, the estimate c_1 of the final location c^* was quite accurate. This is because the curvature increases considerably from a^* to c^* . Segments of the same total curvature starting quite apart from each other could end at points very close to c^* . In the instance (c), where the curvature does not vary much from a^* to c^* , estimation was not as robust to sensing errors.

Since only the final (or present) configuration of the jaw is of interest, we measure the success of localization by the arc length between the final contact c^* and its best estimate. We considered localization a success if this distance was less than 0.3 mm. So, Fig. 8(a)–(c) were successful cases.

The success rate of the conducted experiments stayed around 45%. A failure was related to estimation errors on the rotation angle and the arc length. The jaw's rotation angle is read from the Adept controller which has precision within $\pm 0.03^\circ$. But the jaw is not exactly aligned with the robot's open end, because the aluminum wrist twists slightly under a moment due to the contact force. Nevertheless, we observed that such error tended to be small and barely affected the localization outcome.

Errors in arc length estimation were mainly responsible for the failures in our experiments. This type of estimation errors was addressed earlier in Section III-C. As seen from the table in Fig. 8, two of the three successes occurred when arc length estimates had accuracies within 1 mm. Inaccuracies in machining could cause errors in the estimates of arc length as well as rotation angle because parts do not exactly match their models.

One way to increase robustness of localization is to use larger numerical tolerances on arc length and total curvature.⁴ More experiments are needed to determine bounds that are large enough to tolerate errors in machining and sensing but yet small enough to avoid generating many ambiguous poses. Another improvement is on the hardware—replacing the wrist with a commercial force/torque sensor which has a lower noise level.

V. DISCUSSION

This paper has addressed a number of issues: parts localization, contact sensing, and curve computation. The localization scheme is in-

spired by the ability of the human hand in locating and recognizing a familiar shape through touch.

The algorithm processes curves and has its own merit in geometric computing. It is *complete* in the sense of finding all consistent jaw placements up to numerical resolution, distinguishing itself from a local optimization method (which would also be very inefficient). It is easily extendible to objects bounded by piecewise twice continuously differentiable curves. Many two-dimensional (2-D) shapes are described by algebraic curves in implicit forms rather than parametric forms. Local parameterizations for these types of shapes always exist so the “marching” algorithm can be extended accordingly.

The force/torque sensor is capable of measuring contact location and is useful for simple force control. Improvement is needed regarding the noise in tactile data and the sensitivity of contact measurement to such noise.

The straight jaw is not able to make point contact with a concave boundary portion. A wheel-like finger would be more appropriate for the purpose. The axle of the wheel would be attached to the robot. Pure rolling could be implemented in a simpler and more efficient manner. The wheel (or multiple wheels) would be locked when manipulating a part after its configuration has been determined. Implementation, however, requires some engineering expertise not present in our lab but is promising future work.

An extension of the localization algorithm to a 3-D surface will need to rely on partitioning of the surface into “monotone patches.” For a part consisting of multiple surface patches, a contact point would be hypothesized on every patch and then verified.

For implementation in 3-D, we could precompute a hash table keyed by principal curvatures. The challenge would be to control the rolling of one finger on the object with no disturbance to its pose while estimating principal curvatures [13] at discrete points along the rolling path. Then, we could search the precomputed table for a rough pose estimate and refine it using, say, least squares.

Further down this line of research is reactive localization which conducts sensing and movement simultaneously instead of one after another. The ability to dynamically adjust (or even plan) the touch motion will not only improve the robustness to sensor errors but also emulate touch by the human hand.

APPENDIX COPING WITH CONCAVITY

We extend the localization procedure in Section II-A to any simple closed curve α defined over domain $[0, \tau)$. Here, τ is the period of α , that is, $\alpha(t) = \alpha(t + k\tau)$ for integer k . The correctness of that procedure relies on that the total curvature function $\Phi(s, t)$ has partial derivatives $\partial\Phi/\partial s < 0$ and $\partial\Phi/\partial t > 0$ for all $s < t$. This is no longer everywhere true when α has concavities. For example, if $\kappa(s) < 0$, then $\partial\Phi/\partial s > 0$.⁵

The extended algorithm still marches the two endpoints s and t of a hypothesized curve segment $\alpha[s, t]$ counterclockwise. It precomputes the following quantities on α :

- all points of inflection,⁶ say, z_1, \dots, z_n .
- arc lengths $\ell(z_1, z_i), \dots, \ell(z_1, z_n), \ell(z_n, z_1 + \tau)$.
- total curvatures $\Phi(z_1, z_i), \dots, \Phi(z_1, z_n), \Phi(z_n, z_1 + \tau)$.

We let $z_{j+n} = z_j + \tau$ cope with the situation where the endpoint t marches past τ . After the preprocessing, $\ell(z_i, z_j)$ and $\Phi(z_i, z_j)$, $i \leq j$, can be trivially evaluated.

⁵In case $\kappa(s) = 0$ (or $\kappa(t) = 0$, respectively), we need to look at the signs of $\kappa'(s)$ (or $\kappa'(t)$, respectively) to determine the sign of the $\partial\Phi/\partial s$ (or $\partial\Phi/\partial t$, respectively).

⁶A point s of *simple inflection* has curvature $\kappa(s) = 0$, but $\kappa'(s) \neq 0$. A higher order zero of κ is very unusual and, thus, is neglected here.

⁴They are set as ± 0.004 and ± 0.0004 , respectively.

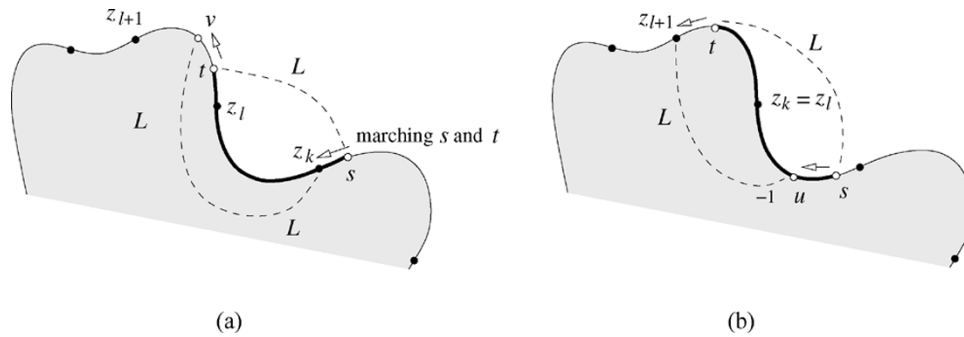


Fig. 9. Two of the four modes of the localization algorithm: (a) convex-convex, where $\kappa(s) > 0$ and $\kappa(t) > 0$, and its following mode (b) concave-convex, where $\kappa(s) < 0$ and $\kappa(t) > 0$. In both (a) and (b), z_k corresponds to the first point of inflection after s , and z_l the last point of inflection before t . In (a), t advances to v with $\ell(z_k, v) = L$ and s advances accordingly to z_k . In (b), s advances to u with $\ell(u, z_{l+1}) = L$ and t advances accordingly to z_{l+1} . The next mode will be concave-concave.

There are four basic modes of marching: *convex-convex* ($\kappa(s) \geq 0$ and $\kappa(t) \geq 0$), *concave-concave* ($\kappa(s) \leq 0$ and $\kappa(t) \leq 0$), *convex-concave* ($\kappa(s) \geq 0$ and $\kappa(t) \leq 0$), and *concave-convex* ($\kappa(s) \leq 0$ and $\kappa(t) \geq 0$). Transition from one mode to another happens when either s or t reaches a point of simple inflection.

Within the same mode, the algorithm increases only one of s and t while simultaneously tracking where the other should be in order to maintain $\ell(s, t) = L$. After one round of increase, the other variable is updated. To realize the above, the algorithm remembers the last inflection z_l passed by t and the first inflection z_k to be passed by s . It then determines, for instance, if s should be passing z_k due to an increase of t by checking if $\ell(z_k, z_l) + \ell(z_l, t) \geq L$.

Below, we describe the working of the algorithm in the modes convex-convex and concave-convex. The scenarios in the other two modes are, respectively, analogous.

In the mode convex-convex, we look at the case $\Phi(s, t) < \Omega$ only since the other case $\Phi(s, t) > \Omega$ is symmetric. Here, t increases until $\Phi(s, t) = \Omega$ or $t = z_{l+1}$ or $\ell(z_k, t) = L$.

- If $\Phi(s, t) = \Omega$ is satisfied first, then s increases until $\ell(s, t)$ decreases to L . If the first feasible segment starts before z_k and ends before z_{l+1} , the algorithm will converge to it in the same manner as described in Section II-A.
- If t reaches z_{l+1} first, the curve segment is turning concave at its ending point. Increase s until $\ell(s, z_{l+1}) = L$. The next mode will be convex-concave.
- If $\ell(z_k, t) = L$ is satisfied first, the curve segment of length L is turning concave at the starting point s . Increase s to z_k . The next mode will be concave-convex. This is illustrated in Fig. 9(a).

In the mode concave-convex, s increases to z_k or until $\ell(s, z_{l+1}) = L$.

- If s reaches z_k first, the next mode will be convex-convex.
- If $\ell(s, z_{l+1}) = L$ is satisfied, the next mode will be concave-concave. This is shown in Fig. 9(b).

Suppose $s = s_a$ and $t = t_a$ when entering the mode, and $s = s_b$ and $t = t_b$ when exiting the mode, where $\ell(s_a, t_a) = \ell(s_b, t_b) = L$. Since $\Phi(s, t)$ increases monotonically with s , there exists one (and exactly one) feasible location if and only if $\Phi(s_a, t_a) \leq \Omega$ and $\Phi(s_b, t_b) \geq \Omega$. When these two conditions hold, bisection is used to compute the feasible location.

ACKNOWLEDGMENT

The author would like to thank J. Mesterhazy for implementing a driver for data acquisition as well as a Visual C++ interface for the Adept Cobra 600, and D. Persky for his code that generated cubic spline shapes and parts. The author is grateful to B. Mishra for several valuable suggestions and to S. LaValle for his enthusiastic help. The author

would also like to acknowledge the following colleagues for generously sharing their expertise on force and tactile sensing: R. Fearing, K. Salisbury, R. Hollis, G. Luecke, R. Voyles, and R. Howe. Many thanks to the anonymous reviewers for their detailed, valuable, and constructive feedback. This paper combines the author's work reported earlier in [23]–[25]. The author would like to thank the anonymous reviewers of these conference papers for their feedback and comments.

REFERENCES

- [1] K. Abe, T. Miwa, and M. Uchiyama, "Development of a 3-axis planar force/torque sensor for very small force/torque measurement," *JSME Int. J., Series C: Mech. Syst., Mach. Elements Manufac.*, vol. 42, pp. 376–382, Jun. 1999.
- [2] S. Akella and M. T. Mason, "Posing polygonal objects in the plane by pushing," in *Proc. IEEE Int. Conf. Robotics Automation*, 1992, pp. 2255–2262.
- [3] P. K. Allen and P. Michelman, "Acquisition and interpretation of 3-D sensor data from touch," *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 397–404, Feb. 1990.
- [4] A. Bicchi, "Intrinsic contact sensing for soft fingers," in *Proc. IEEE Int. Conf. Robotics Automation*, 1990, pp. 968–973.
- [5] K.-F. Böhringer, B. R. Donald, R. Mihailovich, and N. C. MacDonald, "Sensorless manipulation using massively parallel microfabricated actuator arrays," in *Proc. IEEE Int. Conf. Robotics Automation*, 1994, pp. 826–833.
- [6] D. Brock and S. Chiu, "Environment perception of an articulated robot hand using contact sensors," in *Proc. IEEE Int. Conf. Robotics Automation*, 1987, pp. 89–96.
- [7] R. C. Brost, "Automatic grasp planning in the presence of uncertainty," *Int. J. Robot. Res.*, vol. 7, pp. 3–17, Feb. 1988.
- [8] R. C. Brost and K. Y. Goldberg, "A complete algorithm for designing planar fixtures using modular components," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 31–46, Feb. 1996.
- [9] C. Cai and B. Roth, "On the spatial motion of a rigid body with point contact," in *Proc. IEEE Int. Conf. Robotics Automation*, 1987, pp. 686–695.
- [10] S. H. Crandall, N. C. Dahl, and T. J. Lardner, *An Introduction to the Mechanics of Solids*, 2nd ed. New York: McGraw-Hill, 1978, pp. 371–375.
- [11] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," *IEEE J. Robot. Autom.*, vol. 4, no. 4, pp. 369–379, Aug. 1988.
- [12] R. S. Fearing, "Some experiments with tactile sensing during grasping," in *Proc. IEEE Int. Conf. Robotics Automation*, 1987, pp. 1637–1643.
- [13] R. S. Fearing and T. O. Binford, "Using a cylindrical tactile sensor for determining curvature," in *Proc. IEEE Int. Conf. Robotics Automation*, 1988, pp. 765–771.
- [14] K. Y. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 8, pp. 201–225, Oct. 1993.
- [15] S. J. Gordon and W. T. Townsend, "Integration of tactile-force and joint-torque information in a whole-arm manipulator," in *Proc. IEEE Int. Conf. Robotics Automation*, 1989, pp. 464–469.
- [16] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robot. Res.*, vol. 3, pp. 3–35, Jun. 1984.
- [17] D. D. Grossman and M. W. Blasgen, "Orienting mechanical parts by computer-controlled manipulator," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-5, no. 5, pp. 561–565, Sep/Oct. 1975.

- [18] M. Huber and R. A. Gruben, "2-D contact detection and localization using proprioceptive information," *IEEE Trans. Robot. Autom.*, vol. 10, no. 1, pp. 23–33, Feb. 1994.
- [19] K. Gunnarsson and F. Prinz, "CAD model-based localization of parts in manufacturing," *Comput.*, vol. 20, pp. 66–74, Aug. 1987.
- [20] S. Haidacher and G. Hirzinger, "Contact point identification in multi-fingered grasps exploiting kinematic constraints," in *Proc. IEEE Int. Conf. Robotics Automation*, 2002, pp. 1597–1603.
- [21] R. D. Howe and M. R. Cutkosky, "Dynamic tactile sensing: Perception of fine surface features with stress rate sensing," *IEEE Trans. Robot. Autom.*, vol. 9, no. 2, pp. 140–151, Apr. 1993.
- [22] R. D. Howe *et al.*, "Grasping, manipulation, and control with tactile sensing," in *Proc. IEEE Int. Conf. Robotics Automation*, 1990, pp. 1258–1263.
- [23] Y.-B. Jia, "Grasping curved objects through rolling," in *Proc. IEEE Int. Conf. Robotics Automation*, 2000, pp. 377–382.
- [24] —, "Localization on curved objects using tactile information," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, pp. 701–706.
- [25] —, "Contact sensing for parts localization: Sensor design and experiments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 516–522.
- [26] —, (2003) Tactile localization and reconstruction of curved objects. [Online]. Available: <http://www.cs.iastate.edu/~jia/papers/localize.pdf>
- [27] Y.-B. Jia and M. Erdmann, "Geometric sensing of known planar shapes," *Int. J. Robot. Res.*, vol. 15, pp. 365–392, Aug. 1996.
- [28] M. Kaneko and K. Tanie, "Contact point detection for grasping an unknown object using self-posture changeability," *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, pp. 355–367, Jun. 1994.
- [29] D. J. Kriegman and J. Ponce, "On recognizing and positioning curved 3-D objects from image contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 12, pp. 1127–1137, Dec. 1990.
- [30] Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 62–72, Feb. 1990.
- [31] Z. Li, J. Gou, and Y. Chu, "Geometric algorithms for workpiece localization," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 864–878, Dec. 1998.
- [32] K. Lynch, H. Maekawa, and K. Tanie, "Manipulation and active sensing by pushing using tactile feedback," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1992, pp. 416–421.
- [33] C.-H. Menq, H.-T. Yau, and G.-Y. Lai, "Automated precision measurement of surface profile in CAD-directed inspection," *IEEE Trans. Robot. Autom.*, vol. 8, no. 2, pp. 268–278, Apr. 1992.
- [34] M. Moll and M. A. Erdmann, "Reconstructing shape from motion using tactile sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, pp. 692–700.
- [35] D. J. Montana, "The kinematics of contact and grasp," *Int. J. Robot. Res.*, vol. 7, pp. 17–32, Jun. 1988.
- [36] E. Paljug, X. Yun, and V. Kumar, "Control of rolling contacts in multi-arm manipulation," *IEEE Trans. Robot. Autom.*, vol. 10, no. 4, pp. 441–452, Aug. 1994.
- [37] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *J. Dynamic Syst., Measurement, Contr., Trans. ASME*, vol. 102, pp. 126–133, 1981.
- [38] A. Rao and K. Goldberg, "Placing registration marks," *IEEE Trans. Ind. Electron.*, vol. 41, no. 1, pp. 51–59, Feb. 1994.
- [39] K. Salisbury, "Interpretation of contact geometries from force measurements," in *Robot. Res.*, M. Brady and R. Paul, Eds. Cambridge, MA: MIT Press, 1984, pp. 565–577.
- [40] S. Shekhar, O. Khatib, and M. Shimojo, "Object localization with multiple sensors," *Int. J. Robot. Res.*, vol. 7, pp. 34–44, Dec. 1988.
- [41] B. Shimano and B. Roth, "On force sensing information and its use in controlling manipulators," in *Proc. Int. Feder. Automatic Control*, 1977, pp. 119–126.
- [42] T. Tsujimura and T. Yabuta, "Object detection by tactile sensing method employing force/torque information," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 444–450, Aug. 1989.
- [43] R. M. Voyles, J. D. Morrow, and P. K. Khosla, "The shape from motion approach to rapid and precise force/torque sensor calibration," *J. Dynamic Syst., Measurement, Contr.*, vol. 119, pp. 229–235, Jun. 1997.
- [44] A. S. Wallack, J. F. Canny, and D. Manocha, "Object localization using crossbeam sensing," in *Proc. IEEE Int. Conf. Robotics Automation*, 1993, pp. 692–699.
- [45] X. Zhou, Q. Shi, and Z. Li, "Contact localization using force/torque measurements," in *Proc. IEEE Int. Conf. Robotics Automation*, 1996, pp. 1339–1344.

A Simulation/Experimental Study of the Noisy Behavior of the Time-Domain Passivity Controller

Jee-Hwan Ryu, Blake Hannaford, Dong-Soo Kwon, and Jong-Hwan Kim

Abstract—A noisy behavior of the time-domain passivity controller during the period of low velocity is analyzed. Main reasons of the noisy behavior are investigated through a simulation with a one-DOF haptic interface model. It is shown that the PO/PC is ineffective in dissipating the produced energy when the sign of the velocity, which is numerically calculated from the measured position, is suddenly changed, and when this velocity is zero. These cases happen during the period of low velocity due to the limited resolution of the position sensor. New methods, ignoring the produced energy from the velocity sign change, and holding the control force while the velocity is zero, are proposed for removing the noisy behavior. The feasibility of the developed methods is proved with both a simulation and a real experiment.

Index Terms—Haptic interface, noisy behavior, passivity controller, passivity observer, time-domain passivity.

I. INTRODUCTION

A haptic interface is a kinesthetic link between a human operator and a virtual environment (VE). One of the most significant problems in haptic interface design is to create a control system which simultaneously is stable (i.e., does not exhibit vibration or divergent behavior) and gives high fidelity under any operating conditions and for any virtual environment parameters. There are several mechanisms by which a virtual environment or other part of the system might exhibit active behavior. These include quantization [4], interactions between the discrete time system and the continuous time device/human operator [5], and delays due to numerical integration schemes [14].

Initial efforts to solve this problem introduced the "virtual coupling" between the virtual environment and the haptic device [1], [4], [22]. The virtual coupling parameters can be set empirically, but several previous research projects have sought out a theoretical design procedure using control theory. However, interesting virtual environments are always nonlinear and the dynamic properties of a human operator are always involved. These factors make it difficult to analyze haptic systems in terms of system models with known parameters and linear control theory. Anderson and Spong [2] and Neimeyer and Slotine [15] have used passivity ideas in the related area of stable control of force-feedback teleoperation with time delay. Colgate and Schenkel [5] have used it to derive fixed parameter virtual couplings (i.e., haptic interface controllers). The major problem with using passivity for design of haptic

Manuscript received September 22, 2004; revised January 21, 2005. This paper was recommended for publication by Associate Editor C. Melchiorri and Editor I. Walker upon evaluation of the reviewers' comments. This work was supported in part by a grant from BK21 School of Information Technology of KAIST, and ITRC-Intelligent Robot Research Center of KAIST.

J.-H. Ryu is with the School of Mechanical Engineering, Korea University of Technology and Education, Cheonan-city 330-708, Korea (e-mail: jhryu@kut.ac.kr).

B. Hannaford is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195-2500, USA (e-mail: blake@u.washington.edu).

D.-S. Kwon is with the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Taejeon 305-701, Korea (e-mail: kwonds@kaist.ac.kr).

J.-H. Kim is with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Taejeon 305-701, Korea (e-mail: johkim@rit.kaist.ac.kr).

Digital Object Identifier 10.1109/TRO.2005.847611