Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

# High Precision Contour Tracking with a Joystick Sensor

Liangchuan Mi Yan-Bin Jia

Department of Computer Science
Iowa State University
Ames, IA 50011-1040, U.S.A.
lcmi,jia@cs.iastate.edu

## Abstract

High performance contour tracking is achievable with simple sensing and control. We describe a system that tracks unknown shapes with a joystick sensor mounted on an Adept SCARA robot. The joystick's limited force sensing is combined with the Adept's high positional accuracy to yield precise contact measurements. Force control is carried out with a simple feedback loop to avoid unnecessary oscillations. Due to contact friction and random sensor noise, prediction of the tangential motion from force measurements alone is unreliable. A position control strategy is devised to update the tracking direction based on a quadratic t to local turning of the contour. Fitting yields more accurate shape approximation while the use of local data keeps the computational cost low for constant updates. The result is fast tracking with hardly any loss of shape accuracy.

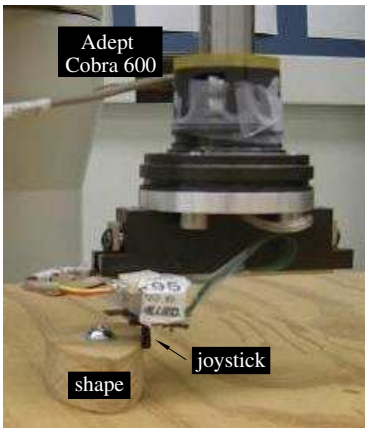



**Figure 1**: Shape tracking with a joystick sensor.

## 1 Introduction

Contour tracking is one of the basic tasks in robot manipulation. In the task, the robot is holding a tool to follow the contour of an object whose shape and pose are often unknown. Applications include part polishing, inspection, paint spraying, cleaning, modeling, etc. During the tracking, the tool is constrained on the surface to maintain contact force while moving along some tangential direction. This has made hybrid position/force control a favorable implementation strategy.

Dynamics-based hybrid control needs to deal with force oscillations, which adds to the complexity of implementation. For industrial robots, direct access to the joint torques is not allowed. This presents an obstacle to implementation of contour tracking through force control on such robots.

In this paper, we investigate contour tracking in the context of shape reconstruction. With a robot of high positional precision, we can control the tracking motion based on just contact force information. The generated contour data has a smaller error range (0.1mm) than acquired using a vision system.

A laser range nder can generate the global shape more ef - ciently. But it is susceptible to occlusions (which are sometimes caused by concavities of the shape). The quality of range data also deteriorates in case of motion. Meanwhile, tracking is natural in a task like dexterous manipulation when the robot hand needs to determine the local shape geometry before executing the next maneuver.

As shown in Figure 1, our tracking tool is a joystick sensor mounted on an Adept Cobra 600 robot (which has $\pm 0.02$mm positional error). The contact force between the sensor and the shape is controlled by a simple feedback loop which adjusts the robot movement along the normal direction.

Due to random noise and contact friction, force readings by the joystick have non-negligible errors in magnitude as well as direction. So we cannot rely on them entirely to carry out position control. Given the Adept robot's high precision, the contact can be located with a very small error range. Thus a polynomial t to a recent sequence of tracking data should approximate the local shape very well. We then compute the tangent to the polynomial at the current contact. Combining this tangent with the direction orthogonal to that of the current force reading will generate a more reliable direction for the tracking motion.

The above ability to estimate curvature while tracking also yields control over the tracking speed. Around a corner, tracking needs to slow down to capture the large variation in local geometry. In some early work, such guidance had to be provided by a vision system.

Section 2 describes the experimental platform and system control architecture. Section 3 focuses on the generation of

0-7803-8463-6/04/$20.00 ©2004 IEEE

tracking motion which makes use of tting to local geometry. Section 4 describes contour adjustment from force readings. Some tracking results are then presented in Section 5. Section 6 concludes with a discussion on future improvement.

### 1.1 Related Work

Hybrid control has been used for generating constrained motion. The scheme, proposed by Raibert and Craig [12], applies independent position control and force control along unconstrained and constrained directions, respectively. Their work was later extended by Khatib and Burdick [8] to incorporate dynamic coupling effects. Mason [10] synthesized control strategies for compliant motions by looking into the semantics of motion primitives.

In practice, contours (and motion constraints) are often unknown. Dynamics-based tracking causes force oscillations due to joint transmissions. Such effects were compensated by Jatta *et al.* [6] through the addition of a normal velocity feedback loop. Yoshikawa and Sudou [14] estimated the instantaneous contact frame (and thus local geometry) using the latest trajectory information and contact force measurements. Nevertheless, their frame estimation was not quite accurate since only two data points on the trajectory were used. In Section 3.1, we will describe more accurate contact geometry estimation through tting over local data.

Baeten and De Schutter [2] employed a vision system to guide sharp corner turns, while applying force control to perform the rest of the tracking. Xiao *et al.* [13] also used visual guidance fused with hybrid position/force control to follow a trajectory (speci ed in the image plane) on an unknown surface. Without resorting to direct force control, Lange and Herzinger [9] enhanced second round contour tracking by transforming readings from an external F/T sensor into joint torques for position control.

Fearing [5] described how a cylindrical tactile ngertip could recover the pose of a generalized convex cone from a small amount of data. Allen and Michelman [1] employed a Utah-MIT hand to obtain sparse contact points around an object and then t a superquadric surface to the data as the reconstructed shape. Ellis and Qin [4] studied shape recovery from the strain on a tactile sensor, formulating it as an optimization problem solvable by the Levenberg-Marquardt method. Moll and Erdmann [11] showed how to simultaneously estimate the shape and motion of an unknown convex object from tactile readings on multiple manipulating palms under frictionless contact.

In the coauthor's recent work [7], a 2-axis force/torque sensor was designed to sense contacts and localize a jaw on a 2D curved shape through rolling. The sensor is, however, not accurate enough for sensing the global shape.

## 2 System Architecture

Our tracking system is implemented with an Adept Cobra robot with 4-DOF and a position control interface. The robot communicates with a host computer via a TCP/IP port. A joystick sensor from Interlink Inc. is used. As shown in Figure 2, the

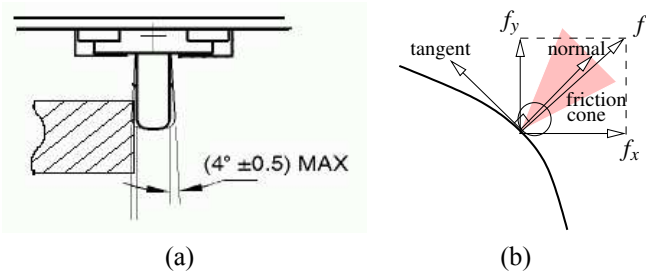



**Figure 2**: (a) The joystick sensor and (b) estimating the tangential direction from a force measurement.

sensor is mounted upside down on the Adept's end effector. It has a force range of 20–170g and a serial interface to the host computer for data transmission.

The sensor's $x$- and $y$-axes are aligned with the world coordinate frame so that force and position measurements have the same reference. Suppose the sensor yields a force measurement $f = (f_x, f_y)$. Then the outward normal and the tangent at the contact may be estimated as $(f_x/\|f\|, f_y/\|f\|)$ and $(-f_y/\|f\|, f_x/\|f\|)$, respectively. However, force measurements by the sensor have random errors up to 20% in both $x$- and $y$-directions. Also, due to friction, $f$ may lie anywhere inside the contact friction cone (see Figure 2(b)). Section 3.1 will describe a more accurate tangent estimation which combines the force measurement with recent tracking history.

Tangent estimation serves for a major component of the system: motion generation. The other major component is contour adjustment, as shown in Figure 3.

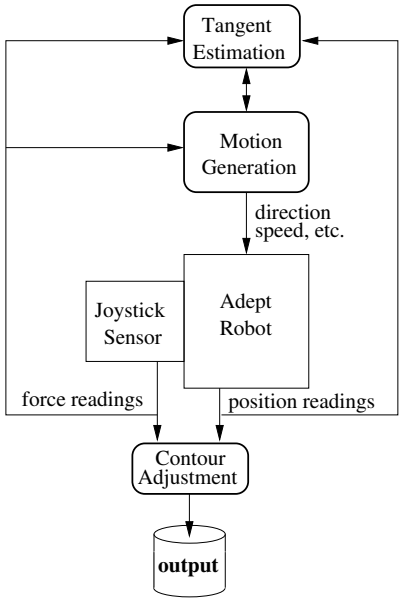



**Figure 3**: Architecture of the tracking system.

# 3 Motion Generation

The motion generation module commands the direction and speed for the next sensor movement in the middle of tracking. Based on the sensor force readings and the estimated tangent direction, one of the following three modes of motion is carried out:

- When the contact force lies within the normal range, the robot continues moving along the current tangent direction.
- When the contact force is below the range, the joystick is breaking the contact. A motion is generated to move the joystick (mostly) along the currently estimated normal direction toward the shape. After the contact force returns to the normal range, the tangent estimation module is trigged to compute a new tangent direction, in which the robot begins new movement.
- When the contact force is above the range, the joystick is "pushing into" the contour. This usually happens while tracking a concavity. The joystick first backs away from the shape until the contact force falls within the range. Then a new tangent direction is estimated for the next movement.

Contact measurements are taken at a rate of 10–100Hz depending on the tracking speed.

## 3.1 Tangent Estimation

During most time of tracking, the sensor is moving along the tangential direction. The tangent direction is updated every time the measured contact force drops below a certain threshold. After the update (and before the next one), the sensor will continue moving along the new tangent and sample shape data at a fixed rate.

Inaccurate estimation of the tangent could result in a significant number of movements in the normal direction in order to adjust the contact force, thus lowering the speed of tracking. As illustrated in Figure 2(b), due to friction between the sensor and the shape, the contact normal is rarely aligned with the contact force. This factor, together with the joystick sensor noise, prevents accurate estimation of the contact normal and tangent from single force measurement. Improvement on tangent estimation will reduce zigzagging of the sensor and thus increase the tracking speed. It will also yield more accurate shape approximation.

Although boundary point measurements have individual errors, the contour still approximates the original shape very well, due to the Adept's high positional accuracy. This prompts us to circumvent the inaccuracy of numerical differentiation (to obtain the tangent direction) by means of fitting and extrapolation.

The data set for fitting includes the points where the $k$ most recent tangent updates took place.[1] These points are stored in a queue and have arc lengths $s_0 = 0, s_1, \ldots, s_{k-1}$ from the first of these points. And $s_k$ is the arc length of the current contact. The arc length between two adjacent points is approximated by their Euclidean distance, which is read by the robot.

The function we choose to fit describes the *tangential angle $\theta$ in terms of arc length $s$*:

$$\theta(s) = as^2 + bs + c. \tag{1}$$

The tangent and curvature are estimated as $(\cos(\theta), \sin(\theta))$ and

$$\kappa(s) = \theta'(s) = 2as + b, \tag{2}$$

respectively. Since a line can approximate the curvature function well enough locally, a quadratic polynomial is chosen as the approximation for the tangential angle. Also, we would like to keep the cost of fitting low, which is needed for frequent updates.

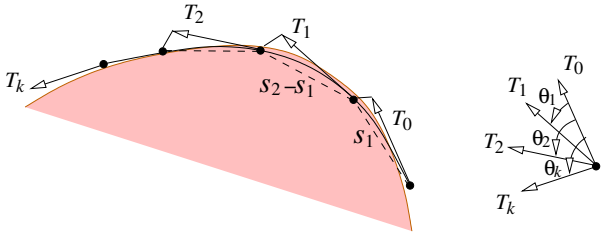



**Figure 4**: Determining the next moving direction $T_k$ of the sensor by extrapolating a polynomial fit to the tangential angles at the last $k$ updates with arc length as the variable. Between the updates, the sensor moves in the same direction while sampling contacts.

Tangential angle estimates $\theta_0 = 0, \ldots, \theta_{k-1}$ at $s_0, \ldots, s_{k-1}$ were already computed in the previous $k$ updates (see Figure 4). We solve the least-squares formulation for $a, b, c$:

$$\min_{a,b,c} \sum_{i=0}^{k-1} (\theta(s_i) - \theta_i)^2 .$$

Then $\theta(s_k)$ is an estimate of the tangential angle at the current contact $s_k$ based on the recent tracking history.

To make the estimation less biased, we also consider the measurement $\bar{\theta}$ inferred directly from the current force reading[2] $f$ and choose the average:

$$\theta_k = \frac{\theta(s_k) + \bar{\theta}}{2}. \tag{3}$$

So $T_k = (\cos\theta_k, \sin\theta_k)$ will be the estimated tangent that determines the sensor motion until the next update. Next, $\theta_k$ is added to the queue while $\theta_0$ is removed. Figure 5 is an example of the above prediction process.

## 3.2 Control of Tracking Speed

Curvature information is needed while tracking a highly curved portion or a corner. As curvature of the contour increases, tracking needs to slow down to sample contour data more densely.

[1] In our implementation, $k = 14$ is chosen to balance between the computational cost and the quality of local shape approximation.

[2] that is, $f \times (\cos\bar{\theta}, \sin\bar{\theta}) > 0$ and $f \cdot (\cos\bar{\theta}, \sin\bar{\theta}) = 0$.

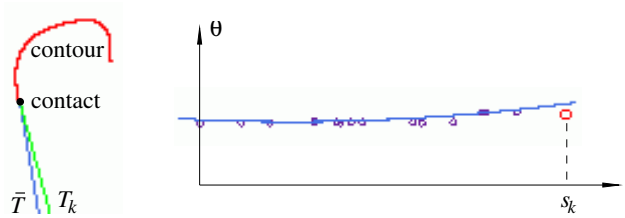



**Figure 5**: Prediction of the tracking motion: (a) the direction $\bar{T} = (\cos\bar{\theta}, \sin\bar{\theta})$ orthogonal to the measured contact force, and the adjusted direction $T_k$ according to (3); and (b) the quadratic t $\theta(s)$ over the last 14 tracking directions to estimate the direction at the current contact $s_k$.

For this reason, we did not t to the position data directly in Section 3.1.

The same quadratic t $\theta(s)$ in (1) is used as long as the tangent has not been updated. Only $s$ has increased. We estimate the contact curvature as $\theta'(s)$ according to (2) and monitor the change of geometry while tracking. Figure 6 shows how to detect that a corner is being approached.

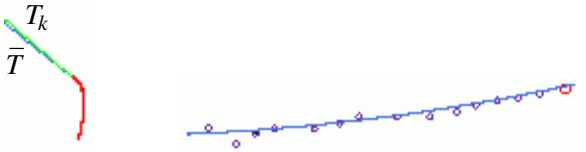



**Figure 6**: Tracking around the rst corner on the contour in Figure 5(a). The slope of the quadratic t $\theta(s)$ is increasing.

Thus rough curvature estimates are obtained while tracking. Figure 7 compares the curvature function for the part of the real shape corresponding to the contour in Figure 5(a), and the set of curvature estimates acquired in tracking. Although the scales of arc length in the two diagrams are slightly different, we can still see the correspondences between the two major peaks in (b) and those in (a).

After the entire shape has been tracked, much more reliable curvature information can be obtained through differentiating a curve t to the contour data.

# 4 Contour Adjustment

The contact between the shape and the joystick is estimated from the sensor's radius and the location of the Adept's end effector. Due to the joystick bending, the projection of its upper end, which is attached to the end effector (see Figure 2(a)), is not tangent to the shape but rather intersects it. This is illustrated in Figure 8. The larger the contact force, the more inside the shape boundary the estimated contact position. As a result, the generated contour would be slightly smaller than the original one without any adjustment.

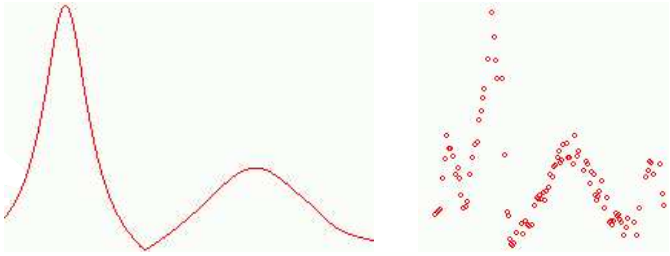



**Figure 7**: Estimating curvature on the y: (a) the curvature function of a real contour; and (b) corresponding curvature estimates according to (2).

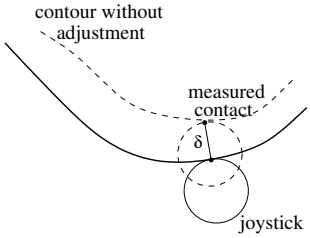



**Figure 8**: Contour adjustment

In the meantime, the tracking motion described in Section 3 generates a zig-zagging contour, as shown in Figure 9(a) and (b). Some adjustment must be performed.

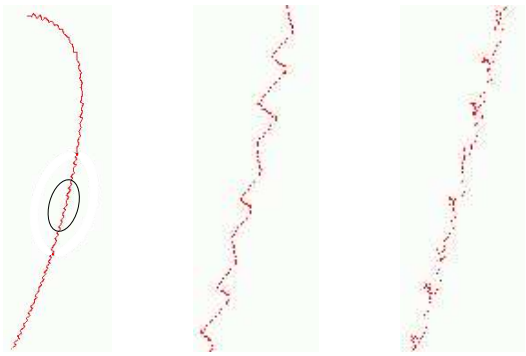


**Figure 9**: Contour adjustment: (a) a contour generated by tracking; (b) an enlargement of the encircled section in (a); and (c) the same section after contour adjustment.

The contour adjustment module is called upon to "grow" the contour outward based on force readings and reduce the zigzagging effect. We pull every measured contact point outward along the normal direction. This is done by adding a vector offset $\delta \cdot f/\|f\|$, where $f$ is the force reading and $\delta$ is the magnitude of the offset to be determined very shortly. The joystick sensor has discrete readings independent of the force direction. We calibrate $\delta$ as a function of $\|f\|$, which is plotted in Figure 10 over the frequent range of readings. From the gure we see that the relationship between $\delta$ and $\|f\|$ is not linear.

As an example, the zigzagged contour section in Figure 9(b) becomes smoother in (c).

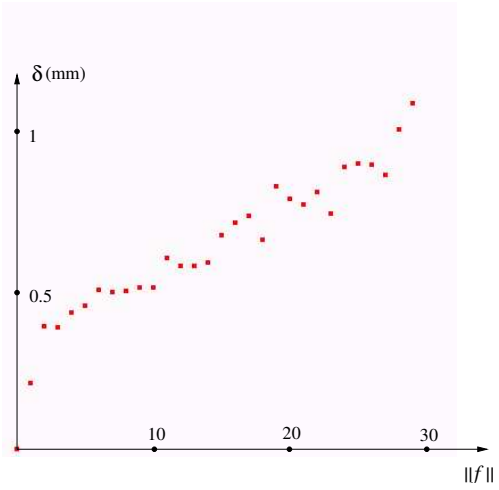
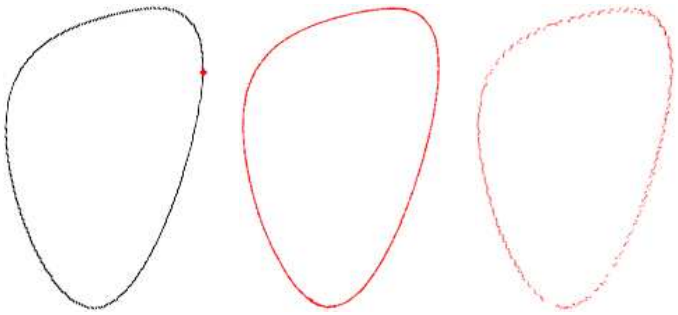



**Figure 10**: Contact adjustments according to sensor readings. An offset of $\delta \cdot f/\|f\|$ is added to the robot positional reading.

# 5 Experiments

Figure 11 displays the results of tracking a shape (a) at two different speeds. The contour in (b) was generated in time $16'30''$



(a) (b) (c)

**Figure 11**: Tracking at different speeds: (a) the original shape; (b) a contour generated with the tracking speed 0.3mm/s; (c) a contour generated with the tracking speed 5.0mm/s (the discontinuities result from downscaling to t the column not from generation). The scale is 1:1.4.

with 20,328 data points while the contour in (c) in time $1'18''$ with only 1,170 data points. The two contours approximate the original shape with almost no visual difference that they are displayed side by side instead of superposed on the shape. We have found the accuracy to be within 0.1mm. For the purpose of t- ting in Section 5.1, the speed can still be improved several times with barely any distortion of the original shape.

The results of tracking ve more shapes are displayed in Figure 12. The concavity of the rst shape did not pose any dif- culty for tracking due to the small size of the joystick sensor. Comparisons between the contours and the original shapes yield almost no difference.

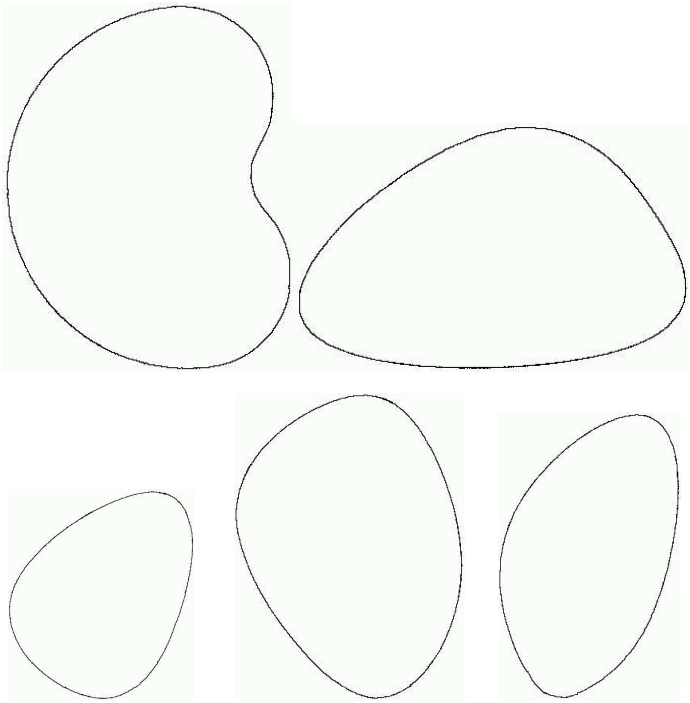


**Figure 12**: Contours of ve tracked shapes (scale 1:1.6).

## 5.1 Shape Representation

The shape data generated from tracking are often too large to store or perform any operations on. Their discrete nature also presents dif culty for reliably extracting geometric quantities like curvature that require differentiations. An analytical description of the shape is thus desired.

One solution is to t an implicit polynomial curve to the shape data. We have implemented the $3L$ algorithm in [3] which uses the following polynomial family:

$$\begin{aligned} f(x,y) &= \sum_{0\le i,j,0\le i+j\le d} a_{ij}x^i y^j \\ &= a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + \cdots + a_{0d}y^d. \end{aligned}$$

For the shapes in Figure 12, all except one of which are cubic splines, the orders $d$ of their polynomial ts are set to be 4 or 5.

# 6 Discussion

We have described a hybrid force/position control system that tracks curved contours very precisely with a simple joystick sensor. Taking advantage of the Adept robot's kinematics, the system is able to achieve high accuracy on contact measurements. This fact has two consequences over the control design. First, force control is implemented as a simple feedback loop without resorting to dynamics so that unnecessary oscillations are reduced. Second, position control over tangential movements is based on approximating the local geometry with

a polynomial t to the most recent contour data. Experiments have demonstrated high quality of tracking.

Curvature is estimated at the same time of tracking, which allows dynamic adjustment of the tracking speed based on local geometry. This has eliminated the need for a vision system to guide corner turns as in some previous work.

Latencies in robot acceleration and deceleration constitute an obstacle to improvement on the tracking speed. Communication delays between the PC and the sensor and between the PC and the robot also slow down the tracking. One solution is to code the control directly into hardware. We are planning to use stepper motors controlled by an FPGA (Field Programmable Gate Arrays) board to replace the robot in controlling the joystick motions directly.

Extension to surface tracking lies in reliable estimation of surface normal (i.e., the tangent plane at contact). We may use tactile array sensors for the purpose. But some additional force sensor may be needed to carry out force control. Again, we could t a function to the unit surface normals on the most recent tracking path. Then use extrapolation to predict the current normal direction (and subsequently choose the next tangential direction). To reconstruct the surface, some high level planning tracking trajectories may also be needed.

**Acknowledgment** Support for this research has been provided in part by Iowa State University, and in part by the National Science Foundation through a CAREER award IIS-0133681. Thanks to Neil Kronlage for a curvature plot, to Oussama Khatib for his feedback on hybrid control, and to the anonymous reviewers for their valuable comments.

# References

[1] P. K. Allen and P. Michelman. Acquisition and interpretation of 3-D sensor data from touch. *IEEE Trans. Robot. & Automat.*, 6(4):397–404, 1990.

[2] J. Baeten and J. De Schutter. Hybrid vision/force control at corners in planar robotic-contour following. *IEEE/ASME Trans. Mechatronics*, 7(2):143–151, 2002.

[3] M. M. Blane *et al.* The $3L$ algorithm for tting implicit polynomial curves and surfaces to data. *IEEE Trans. Pattern Analys. & Mach. Intell.*, 22(3):298–313, 2000.

[4] R. E. Ellis and M. Qin. Singular-value and nite-element analysis of tactile shape recognition. In *Proc. IEEE Intl. Conf. Robot. & Automat.*, pp. 2529–2535, 1994.

[5] R. S. Fearing. Tactile sensing for shape interpretation. In S. T. Venkataraman and T. Iberall, eds., *Dextrous Robot Hands*, pp. 209–238. Springer-Verlag, 1990.

[6] F. Jatta, G. Legnani, and A. Visioli. Hybrid force/velocity control of industrial manipulators with elastic transmissions. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Sys.*, pp. 3276-3281, 2003.

[7] Y.-B. Jia. Contact sensing for parts localization: sensor design and experiments. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Sys.*, pp. 516–522, 2003.

[8] O. Khatib and J. Burdick. Motion and force control of robot manipulators. *Proc. IEEE Intl. Conf. Robot. & Automat.*, pp. 1381-1386, 1996.

[9] F. Lange and G. Hirzinger. Learning force control with position controlled robots. In *Proc. IEEE Intl. Conf. Robot. & Automat.*, 1996.

[10] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Trans. Systems, Man, & Cybernetics*, 11(6):418–432, 1981.

[11] M. Moll and M. A. Erdmann. Reconstructing the shape and motion of unknown objects with active tactile sensors. In J.-D. Boissonnat *et al.*, eds., *Algorithmic Foundations of Robotics V*, pp. 293–309. Springer-Verlag, 2004.

[12] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *J. Dynamic Systems, Measurement, & Control, Trans. of ASME*, 102:126–133, 1981.

[13] D. Xiao, B. K. Ghosh, N. Xi, and T. J. Tarn. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Trans. Control Sys. Tech.*, 8(4):635–645, 2000.

[14] T. Yoshikawa and A. Sudou. Dynamic hybrid position/force control of robot manipulators — on-line estimation of unknown constraint. *IEEE Trans. Robot. & Automat.*, 9(2):220–226, 1993.