Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

A New Adaptive Fuzzy Hybrid Force/Position Control for Intelligent Robot Deburring

Feng-Yih Hsu[1] and Li-Chen Fu[1,2]

Dept. of Electrical Engineering[1]
Dept. of Computer Science & Information Engineering[2]
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

The major control problems for robot deburring mainly arise from uncertainty of the robot manipulators and complex deburring process. In this paper, a new design of hybrid force/position control of robot manipulators via adaptive fuzzy approach is proposed to solve these problems. The control architecture consists of an outer-loop command generator which can automatically determine the desired robot motion profile and an inner-loop adaptive fuzzy hybrid force/position controller which can realtime achieve the command. To demonstrate the effectiveness of the present work, it is applied to the control of a five degree-of-freedom (DOF) articulated robot manipulator for deburring tasks.

1 Introduction

Applying automated robot manipulators to replace manual deburring operation has become more and more important owing to the high cost of deburring, for some of a variety of cast parts which sometimes amounts to 35 percent of total part's cost [1]. In general, deburring tasks are meant to remove burrs from part's edges and to maintain the final geometry of deburred part's edges within some allowable tolerances. To achieve that, a cutting tool with fixed spindle speed has to chamfer the part edges while undergoing compliant contour-following motion. When driving the cutting tool to perform a deburring task, the deburring robots have to implement two major motions: one is to apply suitable chamfering force to the part's edge so as to remove burrs and to avoid damageto the part, and the other is to perform a contour-following motion to assure that the cutting tool will continue to contact all burrs that spread out over the chamfer.

With an aim to achieving this goal, a heuristic strategy that is often applied by experienced engineers is to control the feedrate of the cutting tool to attain the desired chamfering force. This heuristic is simply slowing down the feedrate of the cutting tool when the encountered burr is large, but speeding it up if otherwise [1], [4].On the other hand, the chamfering force for a constant feedrate and a specific chamfer depth is proportional to the cross sectional area of the burr [2], which motivates us to command the feedrate of the cutting tool so as to control the chamfering force. Human operators, however, can not manipulate the cutting tool so precisely to produce the desired chamfer depth as a deburring robot can. Alternatively, some force control schemes such as impedance control, hybrid force control, and other force control schemes had been applied to compliantly yield a desired chamfer depth [3]-[9]. The common feature of these approaches is to construct a model of the contact force that describes the relationship between the position of the cutting tool and the magnitude of the contact force. However, it is fairly hard to construct a precise model and as a result a deburring scheme based on a human-skill model was proposed [4]. Furthermore, a neural network controller which is equipped with the ability of refining the deburring behavior by learning from the skillful engineers was also developed to solve this problem [5].

To sum up, to achieve automated deburring, the guideline for human operator in performing the deburring task needs to be incorporated into the controller and the self-tuning capability needs to be endowed so as to adaptively correct the control law for coping with uncertainties in deburring dynamics. On the other hand,the fuzzy variable structure control can provide the stability and smoothness at the same time of a fuzzy control if the fuzzy control is formulated in a form of variable structure control [17]-[19], or if a variable structure control is augmented with some rule-parameter setting mechanism [20]-[23]. In this paper, we present an adaptive fuzzy hybrid force/position controller via variable structure control approach to

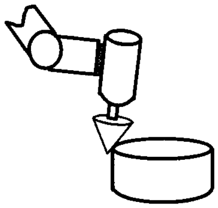

Figure 1: The deburring robot performing the contour-following task

achieve the aforementioned goal in this paper.

2 Dynamic Model of a Deburring Robot in Cartesian Frame

Consider an n degree-of-freedom articulated deburring robot whose cutting tool to is commanded to perform contour-following motion in order to remove burrs from a part, as depicted in Fig. 1. Its dynamic model in joint coordinates can be derived as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + D(\dot{q}) = \tau + \tau_f, \qquad (1)$$

where $q \in \Re^n$ is the joint vector, $M(q) \in \Re^{n\times n}$ is the inertia matrix, $C(q,\dot{q})\dot{q}$ is the vector representing the centrifugal and Coriolis forces satisfying $\dot{M} - 2C$ is a skew-symmetric matrix, $G(q)$ is the vector of gravitational forces, $D(\dot{q})$ is the vector of friction forces, τ_f is the vector of cutting forces and moments, and τ is the vector of control input forces and moments.

To ease the controller design for the deburring task, we re-express the dynamic model in the Cartesians coordinates. First, we assume that the Cartesians coordinate of the cutting tool, namely, x, is with respect to the world frame $\{W\}$, so that x can be represented as a function of its joint coordinates, q in the reference frame, i.e.,

$$x = H(q), \qquad (2)$$

where $x = [x_1, \cdots, x_6]^T = [x_p^T, x_o^T]^T$, with $x_p \in \Re^3$ being its position vector of cutting tool and $x_o \in \Re^3$ being the orientation vector, and $q = [q_1, \cdots, q_n]^T$. Differentiating equation (2), we then get

$$\dot{x} = \frac{\partial H(q)}{\partial q}\dot{q} = J(q)\dot{q}, \qquad (3)$$

where $J(q) \in \Re^{6\times n}$ is a Jocabin transform matrix and is assumed to be of full rank for q lying in a compact set in the joint space, so that there exists a one-to-one mapping between x and q in a properly defined compact set [11]. Thus, J has a sudo-inverse matrix J^+, satisfying $JJ^+ = I$. Then, letting $M_x = J^{+T}MJ^+$, $C_x = J^{+T}CJ^+ - J^{+T}MJ^+\dot{J}J^+$, $G_x = J^{+T}G$, $D_x = J^{+T}D$, $f_\tau = J^{+T}\tau$ and $f = J^{+T}\tau_f$, we can derive the dynamics of the robot manipulator in the world frame as follows:

$$M_x(x)\ddot{x} + C_x(x,\dot{x})\dot{x} + G_x(x) + D_x(\dot{x}) = f_\tau + f, \quad (4)$$

Here, the torque vector τ in joint coordinates can be derived as $\tau = J^T f_\tau$. To simplify the underlying control problem, we will assume that the cutting tool is in contact with the part only at a single point so that the moment exerted by the cutting tool equals zero. Therefore, the cutting force/moment, f, can be denoted as $f = [f_n^T + f_t^T, 0, 0, 0]^T$, where $f_n \in \Re^3$ is the vector of contact force perpendicular to the chamfer surface of the part, and $f_t \in \Re^3$ is the vector of the chamfering force tangential to the chamfer contour of the part.

Now, we need to derive the dynamics of the chamfering force and the contact force during the deburring process. Let x_s represent the vector of the cutting tool position on the part's edge when the chamfer depth is zero (see Fig. 2), namely, x_s is subject to a constraint equation $\phi(x_s) = 0$. To express f_n and f_t on the chamfer surface, we define two unit orthogonal vectors, n_1 and n_2. The former is opposite to the direction of the contact force and can be constructed as $n_1 = \frac{-\nabla\phi(x_s)}{\|\nabla\phi(x_s)\|}$, where ∇ represents a gradient operator. However n_1 can not be constructed directly from the above equation since x_s is not measurable. Therefore, we approximate n_1 by $n_1 \cong \frac{-\nabla\phi(x_p)}{\|\nabla\phi(x_p)\|}$, when the chamfer depth, say, $\delta \in \Re^1$, very small, where x_p is defined to be the position of the cutting tool, which is attributed to the fact that $\nabla\phi(x_p) \times \nabla\phi(x_s) \cong 0$. On the other hand, n_2 is defined to be opposite to the direction of contour-following motion on a chamfer with zero depth, satisfying $n_1 \perp n_2$, then it can be expressed as

$$n_2 = \frac{\dot{x}_p - n_1 n_1^T \dot{x}_p}{\|\dot{x}_p - n_1 n_1^T \dot{x}_p\|} = \frac{(I - n_1 n_1^T)\dot{x}_p}{\|(I - n_1 n_1^T)\dot{x}_p\|} \qquad (5)$$

As a result, we can rewrite the contact force and the chamfering force, respectively, as $f_n = -\|f_n\|n_1$ and $f_t = -\|f_t\|n_2$. Furthermore, it is shown that the magnitude of f_t is found to satisfy the following [4]:

$$\begin{aligned}\|f_t\| &= g|n_2^T\dot{x}_p| + h|n_1^T\dot{x}_p| + P'_{th} \\ &= g\|(I - n_1 n_1^T)\dot{x}_p\| + h|n_1^T\dot{x}_p| + P'_{th}, \end{aligned} \qquad (6)$$

where g and h are coefficients which vary with different burrs and chamfer depth, and P'_{th} is a constant value representing the threshold power of the cutting tool. However, it is very much difficult to establish an exact model for the contact force. Here, we assume that the deburring robot is rigid enough and the spindle of the cutting tool rotates at fast speed, so that the magnitude of the contact force will roughly obey the following relations:

$$\|f_n\| = \begin{cases} c_2\dot{\delta} + c_1\delta + c_0, & \text{as } \delta >= 0; \\ 0, & \text{otherwise,} \end{cases} \qquad (7)$$

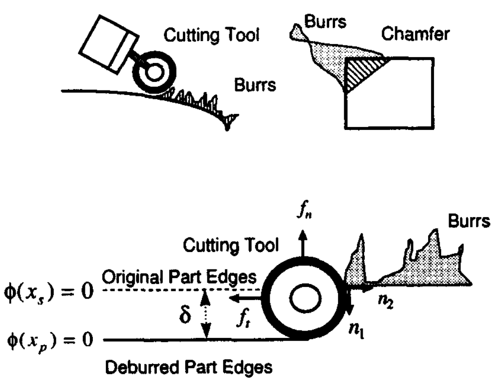

Figure 2: The geometry of part edges durring the deburring process

where c_0 is a threshold as the cutting tool contacts the chamfer, c_1 is the stiffness of the part, and c_2 is the damping ratio of the cutting tool moving into the chamfer.

3 Control Architecture of the Deburring Robot

Thus far, we have derived the dynamics of the robot which performs deburring operation. To accomplish the deburring task, a well designed controller to drive the robot system is needed. Here, the proposed control architecture consists of an outer-loop command generator and an inner-loop controller, as depicted in Fig. 3. The former determines a positional profile command such that the desired chamfering force and contour-following motion can be realizable, whereas the latter aims at driving the robot to execute that command in a compliant manner so as to yield a desired chamfer depth. Generally speaking, since the servo-rate of the inner-loop controller is much faster than that of the outer-loop command generator, we assume that the outer-loop has a nonzero servo-period T whereas the inner-loop controller is considered as a continuous type for simplicity, i.e. its servo-period is zero. The motion commands which are not so much time critical from the viewpoint of the inner-loop controller, like the chamfering force and the locus of the contour-following motion, are handled in the outer-loop. The positional command from the command generator is then transferred to a continuous input via passing through a first-order hold mechanism for the inner-loop controller.

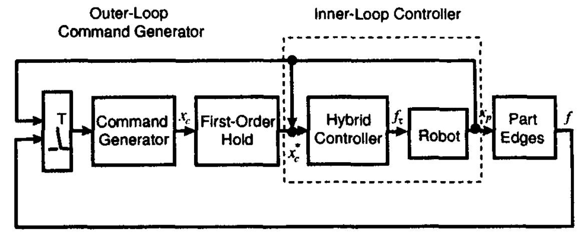

Figure 3: The control architeture of the deburring robot

3.1 Outer-Loop Command Generator

The function of the outer-loop command generator is mainly to ask the inner-loop controller to control the attached cutting tool to produce a desired positional profile. Hence, our goal here is to design such a desired position command, whereby the following objectives can be simultaneously realized provided the cutting tool can implement the command.

- maintaining a desired constant chamfering force
- performing a contour-following motion

First, simplifying the model of the chamfering force as in equation (6), we consider the case where the cutting tool velocity in the direction of n_1 is much smaller than that in the direction of n_2, i.e., $|n_1^T \dot{x}_p| << |n_2^T \dot{x}_p|$. As much, we let $\|f_t\|$ tend to a constant $\|f_{td}\|$ by letting the cutting tool position x_p follow a desired positional profile command x_c during the sampled-period, expressed as follows:

$$x_c(kT) = \begin{cases} x_p(kT) + \frac{\|n_2^T \dot{x}_p(kT)\|(\|f_{td}\| - P'_{th})}{\|f_t(kT)\| - P'_{th}} T n_2(kT), \\ \text{if} \|f_t(kT)\| - P'_{th} > v_\epsilon; \\ x_p(kT) + v_0 T n_2(kT), \quad \text{otherwise,} \end{cases}$$

where T is the servo-period of the outer loop, $k = 1, 2, \cdots$; v_0 is the default feedrate, set to prevent x_p from changing too fast to damage the part as $\|f_t\| - P'_{th}$ becomes excessively small, where $v_\epsilon > 0$ is its offset. Clearly, when $\dot{x}_p \rightarrow \frac{x_c(kT) - x_c(kT-T)}{T}$, we can get a result that $\|f_t\|$ closely approximates $\|f_{td}\|$.

Finally, passing $x_c(kT)$ through a modified first-order hold filter to the inner-loop controller, we can get a continuous desired position trajectory $x_c^*(t)$ with an initial value $x_c^*(0) = x_c(0)$ as follows:

$$x_c^*(t) = x_c{}^*(kT) + \frac{x_c(kT) - x_c{}^*(kT - T)}{T}(t - KT) \tag{8}$$

3.2 Inner-Loop Hybrid Force/Position Controller

The inner-loop controller is to drive the robot to perform the aforementioned positional profile command in a compliant manner so as to yield a desired chamfer depth. Since the direction for measure of the desired chamfer depth is perpendicular to that of the contour-following motion, we can apply a hybrid force/position control to achieve the above goal.

In order to obtain the desired chamfer depth, denoted as δ_d^*, under the desired contact force, denoted as f_{nd}, the desired positional trajectory of the cutting tool $\delta_d(t)$, in the direction of the chamfer depth, i.e., n_1 needs to be carefully selected. Apparently, $\delta_d(t)$ is related to the contact force f_n, and it will eventually approach the desired chamfer depth δ_d^* provided the contact force approaches the desired one. To construct $\delta_d(t)$, first we let the magnitude of desired contact force f_{nd} be defined as follows:

$$||f_{nd}|| = (c_1\delta_d^* + c_0) \tag{9}$$

Then, $\delta_d(t)$ is defined such that its initial condition $\delta_d(0) = 0$ and its time derivative is given in the following:

$$\begin{aligned}\dot{\delta}_d &= k_\delta(||f_{nd}|| - ||f_n||) = k_\delta\left[-c_2\dot{\delta} + c_1(\delta_d^* - \delta)\right] \\ &= k_\delta e_f,\end{aligned} \tag{10}$$

where $k_\delta > 0$ and $e_f = ||f_{nd}|| - ||f_n||$ is defined as the current error of the contact force. Note that, when $\dot{\delta} \to \dot{\delta}_d$ and $\delta \to \delta_d$, we can rewrite the above equation as

$$(1 + k_\delta c_2)\dot{\delta}_d + k_\delta c_1\delta_d = k_\delta c_1\delta_d^*, \tag{11}$$

which implies that δ_d approaches δ_d^* exponentially and, hence, the error of the contact force, e_f, can approach zero.

Given $x_c^*(t)$ and δ_d, we can augment the contour-following position command by summing the desired chamfer depth trajectory $\delta_d n_1$ and the desired contour-following profile trajectory x_c^* together. Thus, an augmented desired position trajectory, x_{p_d}, can be expressed as

$$x_{p_d}(t) = x_c^*(t) + \delta_d(t)n_1, \tag{12}$$

from which we now define the position tracking error as $e_p(t) = x_{p_d} - x_p$. Similarly, we let the desired orientation of the cutting tool be given as x_{od} so that the collective position/orientation trajectory can be denoted as $x_d = [x_{p_d}^T, x_{o_d}^T]^T$. By this definition, we now define the vector of tracking position/orientation tracking error vector as $e(t) = x_d(t) - x(t) = [e_p^T, e_o^T]^T$, where $e_o = x_{od} - x_o$, which facilitates us to rewrite the dynamic model of the manipulators system (4) as follows:

$$M_x(x)\ddot{e} = M_x\ddot{x}_d + C_x(x,\dot{x})\dot{x} + G_x(x) + D_x(\dot{x}) - f_\tau - f, \tag{13}$$

Clearly, our objective is now transferred to the design of a control law f_τ that can drive e to zero so that the desired (constant) chamfering force f_{td} and the desired (constant) chamfer depth δ_d^* can be achieved simultaneously. To that end, we define a sliding variable vector $s = \dot{e} + \lambda e$, where $\lambda \in \Re^{6\times 6}$ is a diagonal positive definite matrix, and re-express the above dynamic model (13) as follows:

$$\begin{aligned}M_x\dot{s} + (K + C_x)s &= M_x(\ddot{x}_d + \lambda\dot{e}) + G_x + D_x + C_x \\ &\quad (\dot{x}_d + \lambda e) + Ks - f - f_\tau,\end{aligned} \tag{14}$$

for some diagonal positive matrix $K \in \Re^{6\times 6}$. Apparently, if the controller design can force the right hand side (RHS) of the equation (14) to be zero, then the resulting system will be subject to the closed-loop dynamics $M_x\dot{s} + (K + C_x)s = 0$, which not only assures the stability of the system, but also guarantees the exponential convergence of the tracking error e. As a result, an ideal control law can be designed as

$$f_\tau^* = M_x(\ddot{x}_d + \lambda\dot{e}) + G_x + D_x + C_x(\dot{x}_d + \lambda e) + Ks - f \tag{15}$$

However, it is difficult to implement this ideal controller due to various kinds of model uncertainty and, hence, the major control problem here is how to realize such an ideal controller.

4 Control Algorithm

As has been mentioned, the parameters associated with contact force, c_0, c_1 and c_2 are often estimated as $\widehat{c}_0$, $\widehat{c}_1$, and $\widehat{c}_2$, respectively. These estimate values will result in the estimate of the desired contact force apart from the true desired one, and will be referred to as the virtual desired force in contrast with the actual desired one. Denote this virtual desired contact force, $\widehat{f_{nd}}$, whose magnitude can be expressed as

$$||\widehat{f}_{nd}|| = \widehat{c}_0 + \widehat{c}_1\delta_d^* \tag{16}$$

Apparently, here our goal is to let $||\widehat{f_{nd}}||$ approach $||f_{nd}||$ so that $\delta_d(\infty) = \delta_d^*$. To achieve this goal, we build a virtual contact force $\widehat{f}_n$ and express its magnitude as

$$||\widehat{f}_n|| = \widehat{c}_0 + \widehat{c}_1\delta + \widehat{c}_2\dot{\delta}, \tag{17}$$

whereby the error between $||f_n||$ and $||\widehat{f_n}||$ can be derived as follows:

$$e_n = ||f_n|| - ||\widehat{f}_n|| = \widetilde{c}_0 + \widetilde{c}_1\delta + \widetilde{c}_2\dot{\delta}, \tag{18}$$

where $\widetilde{c}_0 = c_0 - \widehat{c}_0$, $\widetilde{c}_1 = c_1 - \widehat{c}_1$ and $\widetilde{c}_2 = c_2 - \widehat{c}_2$. To reduce this error, we update the parameters by the following update law:

$$\dot{\widehat{c}}_0 = re_n;\ \ \dot{\widehat{c}}_1 = r\delta e_n;\ \ \dot{\widehat{c}}_2 = r\dot{\delta}e_n, \tag{19}$$

for some $r > 0$.

Consider the robust controller as following:

$$\begin{aligned}f_\tau &= Ks - f + H(||x||, s) \\ H &= (b_1 + b_2||x|| + b_3||x||^2)sgn(s) + (b_4 + b_5||x||)s,\end{aligned} \tag{20}$$

where $b_1, \cdots$, and b_5 are some constants. Then, the following theorem is valid.

Theorem 4.1 *If the control law is given as in* (20), *then the tracking error e will be driven to zero exponentially in term t.*

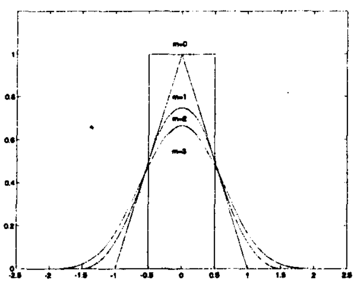

Figure 4: The m-order B-spline basis for m=0, 1, 2, and 3

4.1 Adaptive Fuzzy Hybrid Control

Howecer, gain parameters $b_1,\ldots,b_5$ of $H(\|x\|,s)$. are very difficult to be obtained. Here, we introduce an adaptive fuzzy control algorithm as our control solution. Let the actual control law f_τ be rewritten as follows:

$$f_\tau = Ks - f + f_s(\|x\|,s), \tag{21}$$

where Ks is a linear PD type compensator and f_s is the key adaptive fuzzy compensator. Here, our goal is to design a fuzzy controller $u_f = [u_{f_1},\cdots,u_{f_n}]^T = f_s$ which can compensate for the uncertainties. Then, consider a fuzzy controller u_f, consisting of n ($n = 6$) two-input single-output (MISO) fuzzy controllers, which are respectively characterized by

$$u_{f_i} \triangleq u_{f_i}(\|x\|,s_i) \equiv u_{f_i}(y_0,y_i) : \Omega_0 \times \Omega_i \rightarrow \Re$$

where u_{f_i} is the i-th fuzzy controller, $y = [y_0,y_1,\cdots,y_n]^T = [\|x\|,s_1,\cdots,s_n]^T$ and $\|x\|$ and s_i (i.e., y_0 and y_i) are the i-th input fuzzy variables, $\Omega_0 \equiv [-\Upsilon\Delta_0,\Upsilon\Delta_0]$, $\cdots$, $\Omega_n \equiv [-\Upsilon\Delta_n,\Upsilon\Delta_n]$ with Υ being a positive integer and can be set arbitrarily large to constitute enlarge enough compact sets, and $\Delta_0,\cdots,\Delta_n$ being some positive real numbers. Here, each of the membership functions is given as an m-th order multiple dimension central B-spline function (as depicted in Fig.4), of which the k-th dimension is defined as follows:

$$N_{mk}(y_k) = \sum_{l=0}^{m+1} \frac{(-1)^l}{m!}\binom{m+1}{l}[(y_k + (\frac{m+1}{2} - l)\Delta_k)+]^m \tag{22}$$

where we use the notation

$$x_+ := max(0,x) \tag{23}$$

The m-th order B-spline type of membership function has the following properties:

- an $(m-1)$-th order continuously differentiable function, i.e., $N_{mk}(y_k) \in C^{m-1}$;
- local compact support, i.e., $N_{mk}(y_k) \neq 0$ only for $y_k \in [-\frac{m+1}{2}\Delta_k, \frac{m+1}{2}\Delta_k]$
- $N_{mk}(y_k) > 0$ for $y_k \in (-\frac{m+1}{2}\Delta_k, \frac{m+1}{2}\Delta_k)$
- symmetric with respect to the center point (zero point)
- $\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} N_{m0}(y_0 - i\Delta_0)N_{mk}(y_k - j\Delta_k) = 1$, for $k \in \mathcal{Z}^+$

Then the membership functions for the k-th fuzzy variable y_k are defined as follows:

$$\mu_{k_i}(y_k) = N_{mk}(y_k - i\Delta_k) \tag{24}$$

whose compact support is given as:

$$\Omega_{k_i} = [(i - \frac{m+1}{2})\Delta_k,\ (i + \frac{m+1}{2})\Delta_k], \tag{25}$$

for $k = 0,\cdots,n$, and $i = -\Upsilon,\cdots,0,\cdots,\Upsilon$, which means that $y_k \in int(\Omega_{k_i})$ implies that $\mu_{k_i}(y_k) > 0$ and $\Omega_k \equiv \cup_{i\in\{-\Upsilon,\cdots,\Upsilon\}}\Omega_{k_i}$. Apparently, it is possible that $\Omega_{k_i} \cap \Omega_{k_j} \neq \emptyset$, for some $i \neq j$, i.e., y_k can simultaneously fall into several compact supports. From [15], we can represent the above fuzzy controllers as follows:

$$\begin{aligned} u_{f_k} &= \frac{\sum_{i=-\Upsilon}^{\Upsilon}\sum_{j=-\Upsilon}^{\Upsilon}\mu_{0_i}(y_0)\mu_{k_j}(y_k)\theta_{k(ij)}}{\sum_{i=-\Upsilon}^{\Upsilon}\sum_{j=-\Upsilon}^{\Upsilon}\mu_{0_i}(y_0)\mu_{k_j}(y_k)} \\ &= \sum_{i=-\Upsilon}^{\Upsilon}\sum_{j=-\Upsilon}^{\Upsilon}\nu_{k(ij)}(y_0,y_k)\theta_{k(ij)} = \theta_k{}^T\nu_k, \end{aligned} \tag{26}$$

where i,j are integer indices, $\nu_{k(ij)}(y_0,y_k)$ is the fuzzy basis function, $\theta_{k(ij)}$ is the parameter, $\theta_k = [\theta_{k(-\Upsilon-\Upsilon)},\cdots,\theta_{k(00)},\cdots,\theta_{k(\Upsilon\Upsilon)}]^T \in \Re^{(2\Upsilon+1)^2}$ and $\nu_k = [\nu_{k(-\Upsilon-\Upsilon)},\cdots,\nu_{k(00)},\cdots,\nu_{k(\Upsilon\Upsilon)}]^T \in \Re^{(2\Upsilon+1)^2}$.

Define the k-th element of a new vector $y_\Delta = [y_{\Delta 0},\cdots,y_{\Delta n}]^T$ as follows:

$$y_{\Delta k} = \begin{cases} y_k, & \text{as } y_k < -\Delta_k \text{ or } y_k > \Delta_k; \\ 0, & \text{otherwise (i.e. } y_k \in [-\Delta_k,\Delta_k]); \end{cases} \tag{27}$$

so that

$$\dot{y_{\Delta}}_k = \dot{y}_k \text{ for } y_{\Delta k} \neq 0.$$

Here, our goal is to design u_f to satisfy the following

$$\begin{aligned} u_{f_k}(\|x\|,s_k) &= \begin{cases} k_{f_k}(\|x\|)sgn(s_k), & \text{if } s_k \notin [-\Delta_k,\Delta_k]; \\ k_{sk}(\|x\|,s_k), & \text{otherwise;} \end{cases} \\ \equiv\ u_{f_k}(y_0,y_k) &= \begin{cases} k_{f_k}(y_0)sgn(y_k), & \text{if } y_k \notin [-\Delta_k,\Delta_k]; \\ k_{sk}(y_0,y_k), & \text{otherwise;} \end{cases} \end{aligned} \tag{28}$$

where $k_f(\|x\|,s) \geq |H(\|x\|,s)|$, $k_s(\|x\|,s) = [k_{s1},\cdots,k_{sn}]^T$ is a smooth function vector to make u_f smooth, and $[-\Delta_k,\Delta_k]$ is regarded as a designated dead-zone range which can be arbitrarily set.

Then the following adaptive law to update the parameters vector θ_k will be necessary so that the tracking error can be driven toward the dead-zone range:

$$\dot{\theta}_{k(ij)} = r\nu_{k(ij)}(y_0,y_k)y_{\Delta k}, \text{ for } y_k \in \Omega_k \tag{29}$$

where $r > 0$ is some positive constant.

Finally, we design f_τ as

$$f_\tau = Ks - f + diag(\zeta)f_s(\|x\|,s) + [I - diag(\zeta)]\overline{H}(\|x\|,s) \tag{30}$$

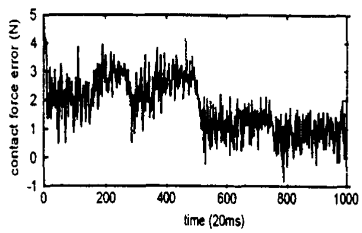

Figure 5: The tracking error of normal contact force

where $diag(\overline{H})sgn(s) > diag(H)sgn(s)$, and the k-th element of $\zeta \in \Re^6$ is defined as follows:

$$\zeta_k = \begin{cases} 1, & \text{as } y_k \in \Omega_k; \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

Then, the following theorem states the condition under which the above-mentioned adaptive fuzzy control law will yield satisfactory result.

Theorem 4.2 *If the update law and the control law are given as in equations* (29) *and* (30), *then the tracking errors will asymptotically converge to a neighborhood of zero.*

5 Experimental Results

A five degree-of-freedom (DOF) articulated deburring robot arm equipped with a Zebra force sensor and a cutting tool is set up in the Intelligent Robot Laboratory of Dept. of Computer Science & Information Engineering in National Taiwan University. The experiment is performed by controlling the robot arm to drive its cutting tool to chamfer the edge of the cylinder part which has a 30mm.radius for cross section. The control architecture based on an industrial PC (80486 dx2-66 CPU) has 2.5 ms sampling-period for the inner loop hybrid controller and 25 ms sampling period for the outer loop command generator. The desired contact force and the desired chamfering force are set to 5 Nt and 10 Nt, respectively. The number of fuzzy rules is set to equal $6 \times (2\Upsilon + 1)^2 = 6 \times 9 \times 9$. At the beginning, since initial parameters in matrix θ are set to zeros, which is similar to only use the PD controller to compensate for the uncertainties in the first period of deburring motion after the first period the force error is quickly driven toward zero as Fig. 5.

6 Conclusions

We had proposed an adaptive fuzzy hybrid force/position controller, which can update fuzzy rules to compensate for the robot dynamics along with the force dynamics induced by the contact between the cutting tool and the part's edge and identify the actual desired contact force. Various experimental results have been provided to verify the effectiveness of the developed work.

References

[1] H. Kazernooi, J. J. Kramer and B. M., "An approach to Automated Deburring by Robot Manipulators," *ASME Journal of Dynamic Systems Measurements And Control. vol. 108. no.4. DEC. 1986*

[2] M. G. Her and H. Kazernooi, "Automated Robotic Deburring of Parts Using Compliance Control," *ASME Journal of Dynamic Systems Measurements And Control. vol. 113. March. 1991*

[3] H. Kazernooi, "On the Robot Compliant Motion Control," *ASME Journal of Dynamic Systems Measurements And Control. vol. 111. September. 1989*

[4] Sheng Lu and H. Asada, "Transferring Manipulative Skills to Robots: Representation and Acquisition of Tool Manipulative Skills Using a Process Dynamics Control," *ASME Journal of Dynamic Systems Measurements And Control. vol. 114. June. 1992*

[5] Sheng Lu and H. Asada, "Teaching and Learning of Deburring Robots Using Neural Networks," *Proc. Of 1993 IEEE Int. Conf. On Robotics and Automation. pp339-345*

[6] G. M. Bone and M. A. Elbestawi " Robotic Force for Deburring Using an Active End Effector," *Robotica. vol. 7. pp-303-308. 1989*

[7] G. M. Bone, M. A. Elbestawi, R. Lingarkar and I. Liu " Force Control for Robotic Deburring," *ASME Journal of Dynamic Systems Measurements And Control. vol. 113. September. pp395-400.1991*

[8] M. S. Ali, M. N. Noori and J. Turi, "Automatic Deburring Utilizing a Real-Time Impedance Control Strategy," *Computer and Structure vol. 46. no.3 pp561-571. 1993*

[9] D. Jeon and M. Tomizuka, "Learning Hybrid Force and Position Control of Robot Manipulators," *IEEE Transactions on Robotics and Automation vo. 9. no. 4. pp. 423-432. Aug. 1993*

[10] L. X. Wang, "Stable Adaptive Fuzzy Control of Nonlinear Systems", *IEEE. Conf. on Decision and Control. 1992*

[11] J. Marsden, Elementary Classical Analysis. San Franciso: W. H. Freeman, 1974

[12] J-J E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice Hall, pp. 278-284, 1991.

[13] R. M. Sanner and J-J E. Slotine, "Gaussian Networks for Direcet Adaptive Control", *IEEE. Trans. on Neural Networks, vol. 3. no. 6. pp. 837-863. 1992.*

[14] S. Sastry and M. Bodson, *Adaptive Control: Stability. Convergence. and Robustness*, Prentice Hall, 1989.

[15] L.-X. Wang, Adaptive Fuzzy Systems and Control: Design and Stability analysis,NJ:Prentice Hall, 1994

[16] V. S. C. Raviraj and P. C. Sen, "Comparative Study of Proportional-Integral ,Sliding Mode, and Fuzzy Logic Controllers for Power Converters," *IEEE Trans. on Industry Applications. vol. 33. no. 2, pp. 518-524. 1997*

[17] Jacob S. Glower and Jeffery Muunighan, "Designing Fuzzy Controllers from a variable structures standpoint," *IEEE Trans. on Fuzzy Systems. vol.5. no. 1. pp.138-144. Feb. 1997*

[18] S.-C. Lin and Y.-Y. Chen, "Design of Self-Learning Fuzzy Sliding Mode Controllers Based on Genetic Algorithms," *Fuzzy Sets and Systems vol.86. no.2. pp. 139-153. Mar. 1997*

[19] J. C. Wu and T. S. Liu, "Fuzzy Control Stabilization with Applications to Motorcycle Control," *IEEE Trans. on Systems. Man. and Cybernetics. PartB: Cybernetics. vol. 26. no. 6. pp.836-847. Dec. 1996*

[20] Feng-Yih Hsu and Li-Chen Fu, "Adaptive Robust Fuzzy Control for Robot Manipulators", *IEEE Conference on Robotics and Automation. pp. 629-634. 1994*

[21] Feng-Yih Hsu and Li-Chen Fu, "A New Design of Adaptive Fuzzy Hybrid Force/Position Controller for Robot Manipulators", *IEEE Conference on Robotics and Automation. pp. 863-868. 1995*

[22] Feng-Yih Hsu and Li-Chen Fu, "An Adaptive Fuzzy Hybrid Control for Robot Manipulators Following Contours of an Uncertain Object", *IEEE Conference on Robotics and Automation. pp. 2232-2237. 1996*

[23] Feng-Yih Hsu and Li-Chen Fu, "Intelligent Robot Deburring Using Adaptive Fuzzy Hybrid Control" *Proc. 27th International Symposium on Industrial Robots. pp. 847-852, Milan. Italy. 1996*

[24] Feng-Yih. Hsu and Li-Chen Fu, "Recent Progress in Fuzzy Control," A Chapter in Control Problems in Robotics and Automation: Future Directions. Ed. Bruno Sicilliano, Springer-Verlag, London, 1997