



Reinforcement Learning and Robust Control for Robot Compliance Tasks [★]

CHENG-PENG KUAN and KUU-YOUNG YOUNG

Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan; e-mail: kyoung@cc.nctu.edu.tw

(Received: 4 January 1998; accepted: 4 March 1998)

Abstract. The complexity in planning and control of robot compliance tasks mainly results from simultaneous control of both position and force and inevitable contact with environments. It is quite difficult to achieve accurate modeling of the interaction between the robot and the environment during contact. In addition, the interaction with the environment varies even for compliance tasks of the same kind. To deal with these phenomena, in this paper, we propose a reinforcement learning and robust control scheme for robot compliance tasks. A reinforcement learning mechanism is used to tackle variations among compliance tasks of the same kind. A robust compliance controller that guarantees system stability in the presence of modeling uncertainties and external disturbances is used to execute control commands sent from the reinforcement learning mechanism. Simulations based on deburring compliance tasks demonstrate the effectiveness of the proposed scheme.

Key words: compliance tasks, reinforcement learning, robust control.

1. Introduction

Compared with robot tasks involving position control, planning and control of robot compliance tasks are much more complicated [15, 22]. In addition to simultaneous control of both position and force in compliance task execution, another major reason for the complexity is that contact with environments is inevitable. It is by no means an easy task to model the interaction between the robot and the environment during contact. Furthermore, compliance tasks of the same kind may even evoke different interactions with the environment. For instance, burrs vary in size and distribution on various target objects in deburring compliance tasks [13]. These hinder the planning of compliance tasks and the development of compliance control strategies for task execution.

Among previous studies in this area, Mason formalized constraints describing the relationship between manipulators and task geometries [22]. Lozano-Perez et al. extended the concept described in [22] to the synthesis of compliant motion strategies from geometric descriptions of assembly operations [18]. Related studies include automatic compliance control strategy generation for generic assembly

[★] This work was supported in part by the National Science Council, Taiwan, under grant NSC 86-2212-E-009-020.

operation [28], automatic task-frame-trajectory generation for varying task frames [8], on-line estimation of unknown task frames based on position and force measurement data [30], among others. The formulation in [22] also leads to simple control strategy specification using hybrid control, in which position and force are controlled along different degrees of freedom [23]. A famous compliance control scheme, the impedance control, deals with control of dynamic interactions between manipulators and environments as wholes instead of controlling position and force individually [12]. Extensions from hybrid control and impedance control include hybrid impedance control for dealing with different types of environments [1], the parallel approach to force/position control for tackling conflicting situations when both position and force control are exerted in the same direction [7], generalized impedance control for providing robustness to unknown environmental stiffnesses [17], among others. Studies have also been devoted to stability analysis and implementation of the hybrid and impedance control schemes [14, 19, 20, 27].

Another methodology for compliance task execution is human control strategy emulation, which arises from human excellence at performing compliance tasks [24]. Since humans can perform delicate manipulations while remaining unaware of detailed motion planning and control strategies, one approach to acquiring human skill uses direct recording. Neural networks, fuzzy systems, or statistical models are used to extract human strategies and store them implicitly in the neural networks, in the fuzzy rules, or in the statistical models [4, 16, 29]. This avoids the necessity for direct compliance task modeling, and transfers human skill at performing compliance tasks via teaching and measurement data processing. Following this concept, Hirzinger and Landzettel proposed a system based on hybrid control for compliance task teaching [11]. Asada et al. presented a series of research results that concerned human skill transfers based on using various methods, such as learning approaches and signal processing, and various compliance control schemes, such as hybrid control and impedance control [2–4].

From past studies, we found that it is quite difficult to achieve accurate, automatic modeling for compliance task planning. The stability and implementation issues also remain as challenges for hybrid control and impedance control. Human skill transfer is limited by incompatibilities between humans and robot manipulators, since they are basically different mechanisms with different control strategies and sensory abilities [4, 21, 24]. Accordingly, in this paper, we propose a reinforcement learning and robust control scheme for robot compliance tasks. A reinforcement learning mechanism is used to deal with variations among compliance tasks of the same kind. The reinforcement learning mechanism can adapt to compliance tasks via an on-line learning process, and generalize existing control commands to tackle tasks not previously encountered. A robust compliance controller that guarantees system stability in the presence of modeling uncertainties and external disturbances is used to execute control commands sent from the reinforcement learning mechanism. Thus, the complexity in planning and control of robot compliance tasks is shared by a learning mechanism for command generation at a higher

level, and a robust controller for command execution at a lower level. The rest of this paper is organized as follows. System configuration and implementation of the proposed reinforcement learning and robust control scheme are described in Section 2. In Section 3, simulations of deburring compliance tasks are reported to demonstrate the effectiveness of the proposed scheme. Finally, conclusions are given in Section 4.

2. Proposed Scheme

The system organization of the proposed scheme is as shown in Figure 1. The reinforcement learning mechanism generates commands C_d for an input compliance task according to evaluation of the difference between the current state and the task objective that must be achieved. Task objectives can be, for instance, to reach desired hole locations for peg-in-hole tasks. In turn, the robust compliance controller modulates the commands sent from the reinforcement learning mechanism and generates torques to move the robot manipulator for task execution in the presence of modeling uncertainties and external disturbances. The positions, velocities, and forces induced during interactions between the robot manipulator and the environment are fed back to the reinforcement learning mechanism and the robust compliance controller. System structures of the two major modules in the proposed scheme, the reinforcement learning mechanism and the robust compliance controller, are as shown in Figure 2, and are discussed below.

2.1. REINFORCEMENT LEARNING MECHANISM

As mentioned above, learning mechanisms are used to tackle variations present in compliance tasks of the same kind. There are three basic classes of learning paradigms: supervised learning, reinforcement learning, and unsupervised learning [10, 26]. Supervised learning is performed under the supervision of an external teacher. Reinforcement learning involves the use of a critic that evolves through a trial-and-error process. Unsupervised learning is performed in a self-organized manner in that no external teacher or critic is required to guide synaptic changes in the network. We adopted the reinforcement learning as the learning mechanism

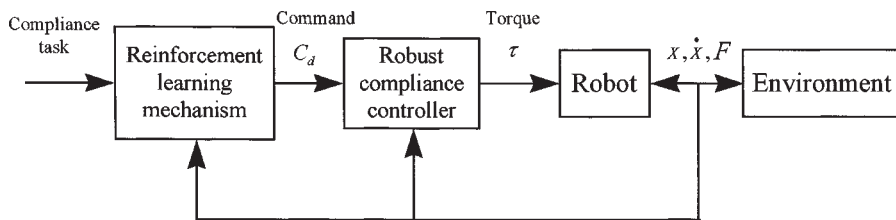
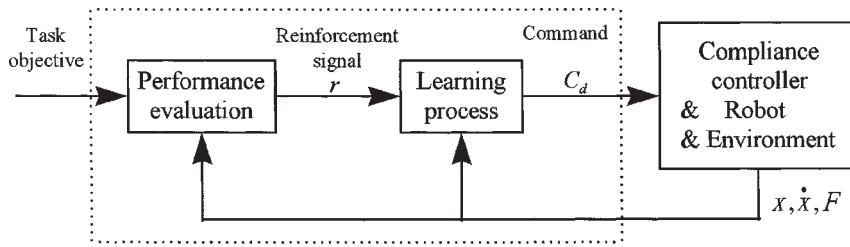
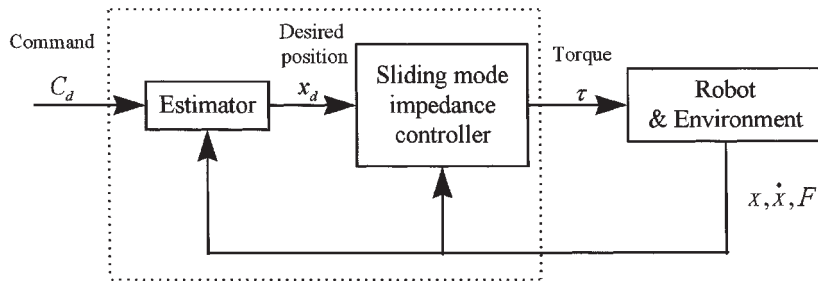


Figure 1. System organization of the proposed reinforcement learning and robust control scheme.



(a) The reinforcement learning mechanism



(b) The robust compliance controller

Figure 2. (a) The reinforcement learning mechanism. (b) The robust compliance controller.

for the proposed scheme, because in addition to environmental uncertainties and variations in most compliance tasks, the complexity of the combined dynamics of the robot controller, the robot manipulator, and the environment makes it difficult to obtain accurate feedback information concerning how the system should adjust its parameters to improve performance.

Figure 2(a) shows the structure of the reinforcement learning mechanism, which executes two main functions: performance evaluation and learning. The reinforcement learning mechanism first evaluates system performance using a scalar performance index, called a reinforcement signal r , which indicates the closeness of the system performance to the task objective. Because this reinforcement signal r is only a scalar used as a critic, it does not carry information concerning how the system should modify its parameters to improve its performance; by contrast, the performance measurement used for supervised learning is usually a vector defined in terms of desired responses by means of a known error criterion, and can be used for system parameter adjustment directly. Thus, the learning in the reinforcement learning mechanism needs to search for directional information by probing the environment through the combined use of trial and error and delayed reward [5, 9].

The reinforcement signal r is usually defined as a unimodal function with a maximal value, indicating the fulfillment of the task objective. We take the debur-

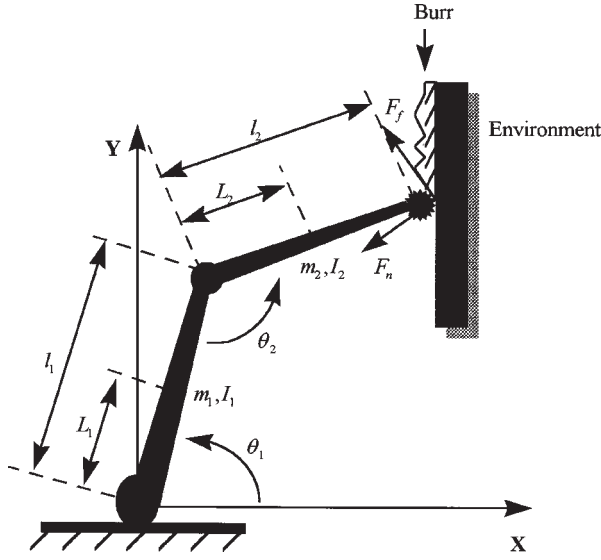


Figure 3. The deburring compliance task.

ring task shown in Figure 3 as an example. The task objective is to remove the burrs from the desired surface. We then define an error function E as

$$E = \frac{1}{2}(x - x_d)^2, \quad (1)$$

where x is the current grinding tool position and x_d is the position of the desired surface. Thus, the reinforcement signal r can be chosen as

$$r = -E. \quad (2)$$

This choice of r has an upper bound at zero, and the task objective is fulfilled when r reaches zero.

In the next stage, the learning process is used to generate commands C_d that cause the reinforcement signal r to approach its maximum. Before the derivation, we first describe a simplified deburring process for the deburring task shown in Figure 3 [4, 13]. When the grinding tool is removing a burr, the speed of burr removal \dot{x} increases as the product of the contact force F and the rotary speed of the grinding tool ω_r increases, but decreases as the grinding tool's velocity in the Y direction \dot{y} increases. And the increase in the contact force F and the grinding tool's velocity in the Y direction \dot{y} both decrease the rotary speed of the grinding tool ω_r . This simplified deburring process is formalized in the following two equations:

$$\dot{x} = K_{F\omega} F \omega_r - K_{xy} \dot{y}, \quad (3)$$

$$\dot{\omega}_r = -K_F F - K_{\omega y} \dot{y}, \quad (4)$$

where $K_{F\omega}$, K_{xy} , K_F , and $K_{\omega y}$ are parameters related to burr characteristics.

To deal with the deburring task, the command C_d is specified as the desired force that pushes the grinding tool to deburr, and defined as a function of ω_r , x , and \dot{y} :

$$C_d = W_\omega \omega_r + W_x x + W_y \dot{y}, \quad (5)$$

where W_ω , W_x , and W_y are adjustment weights. The command C_d is sent to the robust compliance controller, which in turn generates torques to move the robot manipulator for deburring. System performance is then evaluated, and desired commands C_d to increase the reinforcement signal r are obtained by adjusting the weights, W_ω , W_x , and W_y , via the learning process described below:

$$W_\omega(k+1) = W_\omega(k) + \eta_\omega \frac{\partial r}{\partial C_d} \omega_r, \quad (6)$$

$$W_x(k+1) = W_x(k) + \eta_x \frac{\partial r}{\partial C_d} x, \quad (7)$$

$$W_y(k+1) = W_y(k) + \eta_y \frac{\partial r}{\partial C_d} \dot{y}, \quad (8)$$

where k is the time step, and η_ω , η_x , and η_y the learning rates. Because a concise form of the combined inverse dynamic model of the robust compliance controller, the robot manipulator, and the environment is not available, $\partial r / \partial C_d$ in Equations (6)–(8) cannot be obtained directly, and it is approximated by

$$\frac{\partial r}{\partial C_d} \approx \frac{r(k) - r(k-1)}{C_d(k) - C_d(k-1)}. \quad (9)$$

By applying the learning process described in Equations (6)–(8) repeatedly, the reinforcement signal r will approach its maximum gradually. Note that in Equations (6)–(8), only one scalar signal r is used to adjust three weight parameters, W_ω , W_x , and W_y . Therefore, learning of these three weights may create conflicts. This is a characteristic of reinforcement learning, and can be resolved by using proper learning rates, η_ω , η_x , and η_y .

2.2. ROBUST COMPLIANCE CONTROLLER

The robust compliance controller is used to execute commands sent from the reinforcement learning mechanism, and it is designed to achieve system stability in the presence of modeling uncertainties and external disturbances. Figure 2(b) shows the block diagram of the robust compliance controller, which consists of two modules: the estimator and the sliding mode impedance controller. The sliding mode impedance controller is basically an impedance controller implemented using a sliding mode control concept [20]; impedance control is used because it is a unified approach to handling free and constrained motions; sliding mode control

is used because of its robustness to environmental uncertainties. The sliding mode impedance controller accepts position commands and yields forces imposed upon the environment. The purpose of the estimator is to provide proper desired positions that cause the controller to generate desired contact forces for task execution.

The robust compliance controller functions as follows. As Figure 2(b) shows, input commands C_d are sent from the reinforcement learning mechanism; when the motion is in free space, i.e., C_d is a position command, the estimator executes no operation, and just allows C_d to pass directly to the sliding mode impedance controller as desired positions x_d ; when the motion is in constrained space, i.e., C_d is in the form of desired force commands, the estimator first estimates the environmental parameters, and then derives the desired positions x_d accordingly. The desired positions x_d are sent to the sliding mode impedance controller, which in turn generates torques τ to move the robot manipulator to impose the desired contact forces specified by C_d on the environment.

A. The Estimator

Again, we take the deburring task in Figure 3 as an example. The desired surface to reach after deburring is specified as being parallel to the Y axis. The environment, i.e., the desired surface with burrs, is modeled as a spring with stiffness K_e , which may vary at different places on the surface. The surface is approximated by lines defined as $ax + by + c = 0$ for planar cases, where a , b , and c are parameters varying along with the surface curvature. The contact force F_e can then be derived as

$$F_e = F_n + F_f \quad (10)$$

with

$$F_n = K_e \frac{ax + by + c}{\sqrt{a^2 + b^2}}, \quad (11)$$

$$F_f = \rho F_n, \quad (12)$$

where F_n and F_f are, respectively, the forces normal and tangential to the line defined by $ax + by + c = 0$, and ρ is the friction coefficient. The function, E_f , that specifies the error between F_e and a desired contact force, F_d , is defined as

$$E_f = \frac{1}{2} (\|F_d\| - \|F_e\|)^2. \quad (13)$$

To tackle this deburring task, the estimator employs a learning process to derive the desired position x_d that causes F_e to approach F_d . Since F_d is generated when the deburring tool pushes upon the environment to a certain depth, it can be formulated as a function of a location in the environment:

$$F_d = K_x x + K_y y + K_c \quad (14)$$

where K_x , K_y , and K_c are adjustment parameters. Thus, the learning process becomes one of finding proper K_x , K_y , and K_c that cause x to approach some desired

x_d corresponding to the desired F_d . It is straightforward to use the gradient descent method and the error function E_f in Equation (13) to update K_x , K_y , K_c , x , and y in Equation (14) in a recursive manner, such that F_e gradually converges to F_d , and $x = x_d$ is realized. Simultaneously, this learning process also implicitly identifies the environmental stiffness parameter K_e that defines the relationship between x_d and F_d .

B. The Sliding Mode Impedance Controller

Using the desired position x_d provided by the estimator, the sliding mode impedance controller generates torques to move the robot manipulator to push upon the environment with the desired contact force F_d . The basic concept of sliding mode control is to force system states toward a sliding surface in the presence of modeling uncertainties and external disturbances [20, 25]. Once system states are on the sliding surface, the states will then slide along the surface toward the desired states. Because an impedance controller is used, we select the sliding surface $s(t)$ to

$$s(t) = M\dot{x} + B(x - x_d) + K \int_0^t (x(\sigma) - x_d) d\sigma, \quad (15)$$

where the impedance parameter matrices, M , B , and K , are set positive definite, and they determine the convergence speed for states on the sliding surface. This sliding surface $s(t)$ leads to $x = x_d$, when $s \equiv 0$, as demonstrated in Theorem 1.

THEOREM 1. *Given the sliding surface $s(t)$ defined in Equation (15) with M , B , and K positive definite matrices, if s converges to zero, x will approach the desired position x_d .*

The proof of this theorem is given in Appendix A. The control law for this sliding mode impedance controller can be derived as follows. We first choose a Lyapunov function L as

$$L = \frac{1}{2}s^T s. \quad (16)$$

Obviously, $L \geq 0$ and $L = 0$, when $s = 0$. By differentiating L , we obtain

$$\dot{L} = s^T \dot{s} = s^T [M\ddot{x} + B\dot{x} + K(x - x_d)]. \quad (17)$$

And by incorporating the robot dynamics into Equation (17), \dot{L} becomes

$$\begin{aligned} \dot{L} &= s^T [MJ\ddot{q} + M\dot{J}\dot{q} + BJ\dot{q} + K(x - x_d)] \\ &= s^T [MJH^{-1}(\tau + J^T F + D_u - V - G) + M\dot{J}\dot{q} + BJ\dot{q} + K(x - x_d)] \end{aligned} \quad (18)$$

with the robot dynamic equation formulated as

$$H(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau + J^T(q)F + D_u, \quad (19)$$

where $H(q)$ is the inertial matrix, $V(q, \dot{q})$ is the centrifugal and Coriolis term vector, $G(q)$ is the gravity term vector, $J(q)$ is the Jacobian matrix, τ is the joint torque vector, and D_u stands for uncertainties and disturbances. By making the control law

$$u = W_1 \|MJH^{-1}\| D_{\text{umax}} \frac{s}{\|s\|}, \quad W_1 \geq 1, \quad (20)$$

$$\tau = V + G - J^T F - HJ^{-1}M^{-1}[MJ\dot{q} + BJ\dot{q} + K(x - x_d) + u], \quad (21)$$

it can be shown that

$$\dot{L} = s^T(-u + MJH^{-1}D_u) < 0, \quad (22)$$

where D_{umax} in Equation (20) is the upper bound on D_u . Thus, with this control law, L will approach zero under bounded uncertainties and disturbances; then, $s \rightarrow 0$, and $x \rightarrow x_d$ according to Theorem 1. Note that chattering may result from control discontinuity in Equation (20), and saturation functions can be used to replace $s/\|s\|$ in Equation (20) to smooth control inputs [25].

3. Simulation

To demonstrate the effectiveness of the proposed scheme, simulations were performed using a two-joint planar robot manipulator to execute the deburring compliance task shown in Figure 3. The dynamic equations for this two-joint robot manipulator are as follows:

$$\tau_1 = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 - H\dot{\theta}_2^2 - 2H\dot{\theta}_1\dot{\theta}_2 + G_1, \quad (23)$$

$$\tau_2 = H_{21}\ddot{\theta}_1 + H_{22}\ddot{\theta}_2 + H\dot{\theta}_1^2 + G_2, \quad (24)$$

where

$$H_{11} = m_1L_1^2 + I_1 + m_2[l_1^2 + L_2^2 + 2l_1L_2 \cos(\theta_2)] + I_2, \quad (25)$$

$$H_{22} = m_2L_2^2 + I_2, \quad (26)$$

$$H_{12} = m_2l_1L_2 \cos(\theta_2) + m_2L_2^2 + I_2, \quad (27)$$

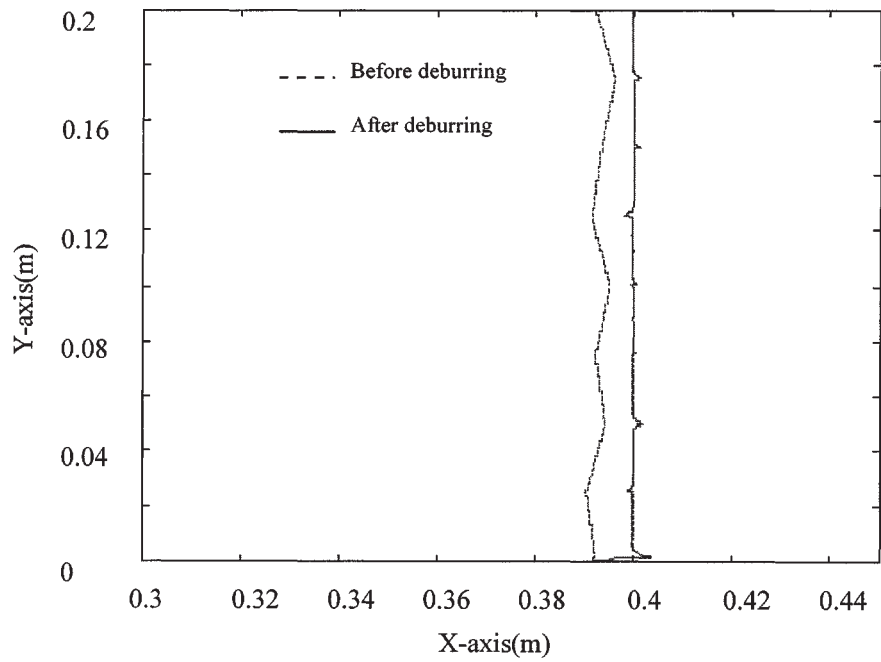
$$H_{21} = H_{12}, \quad (28)$$

$$H = m_2l_1L_2 \sin(\theta_2), \quad (29)$$

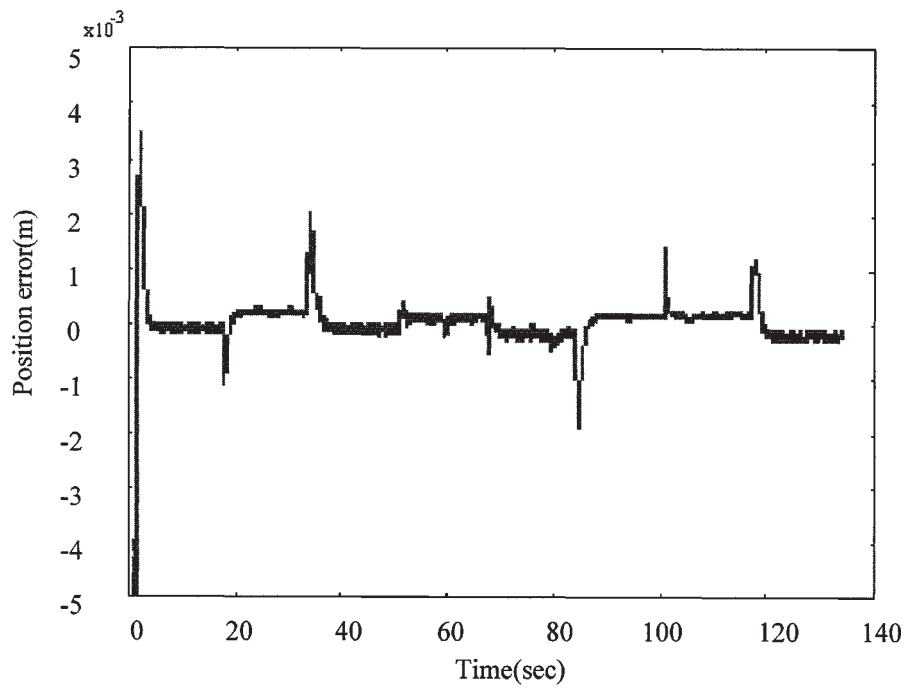
$$G_1 = m_1L_1g \cos(\theta_1) + m_2g[L_2 \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1)], \quad (30)$$

$$G_2 = m_2L_2g \cos(\theta_1 + \theta_2) \quad (31)$$

with $m_1 = 2.815$ kg, $m_2 = 1.640$ kg, $l_1 = 0.30$ m, $l_2 = 0.32$ m, $L_1 = 0.15$ m, $L_2 = 0.16$ m, and $I_1 = I_2 = 0.0234$ kgm². The effects of gravity were ignored in the simulations. Modeling uncertainties and external disturbances D_u acting on the robot manipulator were unknown, but were bounded, and formalized in a function



(a) Position response



(b) Position error plot

Figure 4. Simulation results using the proposed scheme: (a) position response, (b) position error plot.

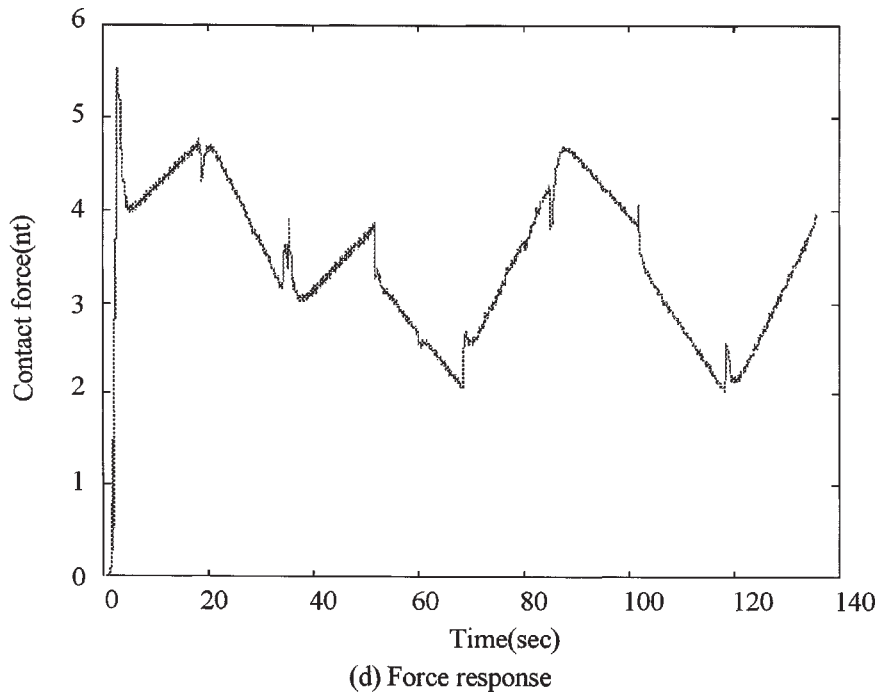
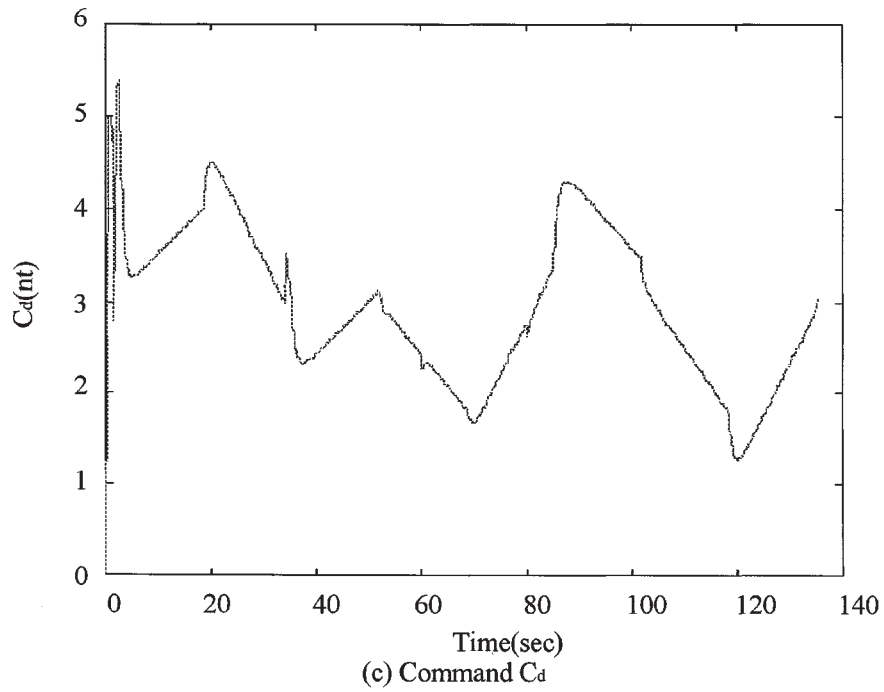
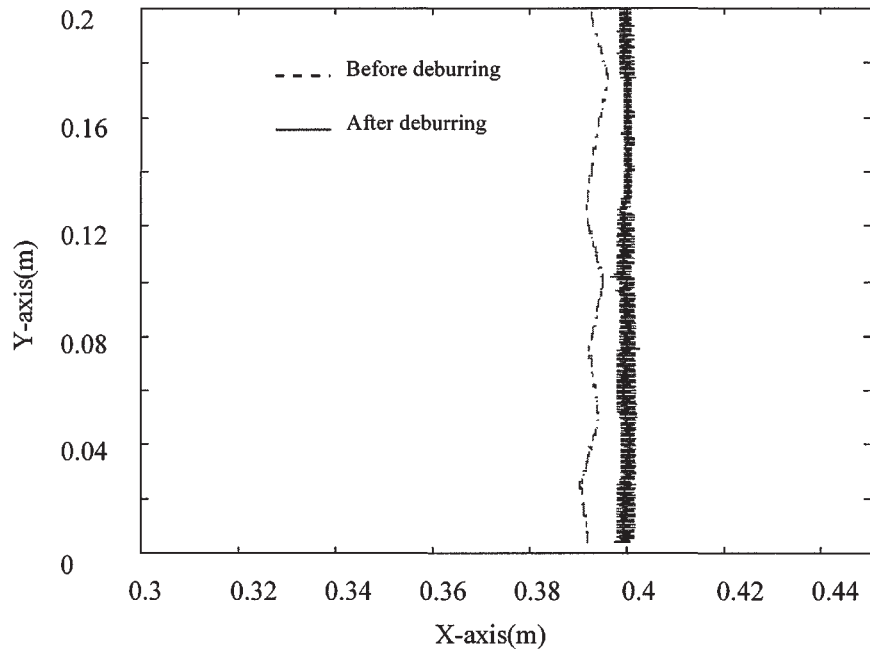


Figure 4. (Continued) (c) command C_d , and (d) force response.

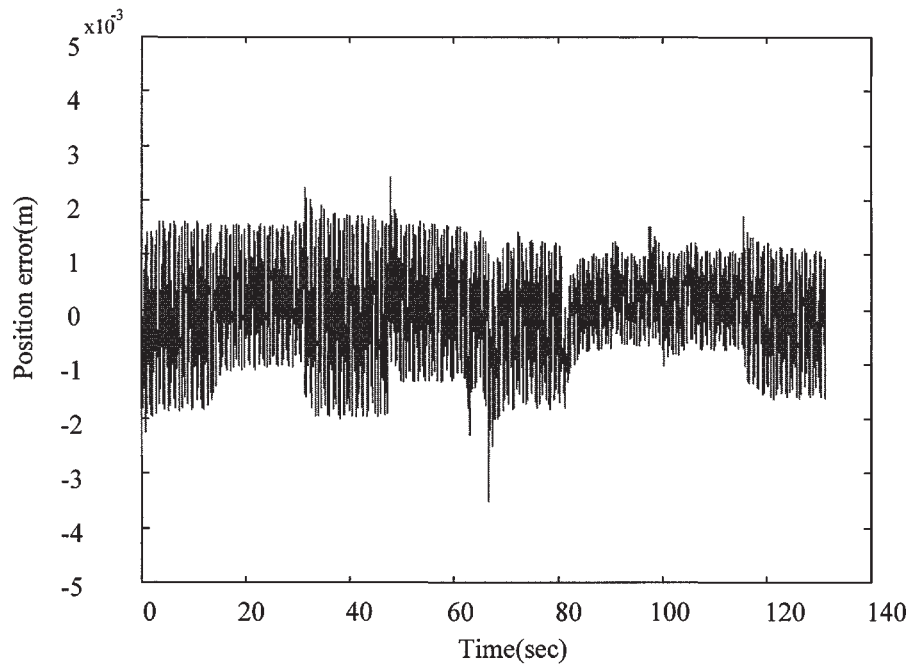
involving sin and cos functions, with $D_{\text{umax}} = 0.4$ N. The deburring process, burr characteristics, and the designs of the reinforcement learning mechanism and the robust compliance controller for deburring were as all described in Section 2. In all simulations, the environmental stiffness, $K_e \in [8000, 10000]$ (N/m), varied along with burr distribution, and the parameters describing burr characteristics, $K_{F\omega} \in [0.008, 0.012]$ (m/(N · rad)), $K_{xy} \in [0.9, 1.1]$, $K_F \in [0.9, 1.1]$ (rad/(N · s²)) and $K_{\omega y} \in [0.9, 1.1]$ (rad/m · s), also varied at different places on the surface.

In the first set of simulations, we used the proposed scheme to perform deburring. The simulation results are as shown in Figure 4. As Figure 4(a) shows, the grinding tool was moved from an initial location [0.3, 0.0] (m) in free space and contacted the burrs beyond the line $x = 0.4$ m, where the desired surface to reach was located. In Figure 4(a), the surface before deburring is indicated by dashed lines, and that after deburring by solid lines. Figure 4(b) shows the position errors between the surface after deburring and the desired surface; the errors were quite small with an average deviation of 0.29 mm in the X direction. Figure 4(b) shows a few points with larger errors, which occurred at locations where burr characteristics or burr surfaces varied abruptly. Figure 4(c) shows the commands C_d generated by the reinforcement learning mechanism, and Figure 4(d), the resultant force responses produced when the robust compliance controller executed the commands C_d to move the robot manipulator in contact with the environment. The closeness of the trajectories in Figures 4(c) and (d) demonstrates that the robust compliance controller successfully imposed the desired contact forces specified by the commands C_d on the environment. To investigate the influence of initial weight selection on the reinforcement learning mechanism in command generation, i.e., the selection of W_ω , W_x , and W_y in Equation (5), we also performed simulations using different sets of initial weights. Results similar to those in Figure 4 were obtained, and we deemed that initial weight selection did not greatly affect the performance of the reinforcement learning mechanism.

In the second set of simulations, we replaced the sliding mode impedance controller in the robust compliance controller of the proposed scheme with an impedance controller, and still used the estimator in the robust compliance controller to transform commands C_d from the reinforcement learning mechanism into desired positions x_d . The purpose was to investigate the difference between the sliding mode impedance controller and the impedance controller. The same deburring task (with the same burr characteristics and distributions) used in the first set of simulations was used here, and the same impedance parameter matrices, M , B , and K , used in the sliding mode impedance controller were used in the impedance controller. The simulation results are as shown in Figure 5. As shown in Figure 5(b), the position errors between the surface after deburring and the desired surface oscillated and were larger than those in Figure 4(b), having an average deviation of 0.89 mm in the X direction. Figures 5(c) and (d) show the commands C_d and the resultant force responses. Figure 5(d) shows oscillations in the force responses similar to those in the position responses, indicating that the impedance controller



(a) Position response



(b) Position error plot

Figure 5. Simulation results using the proposed scheme with the sliding mode impedance controller replaced by an impedance controller: (a) position response, (b) position error plot.

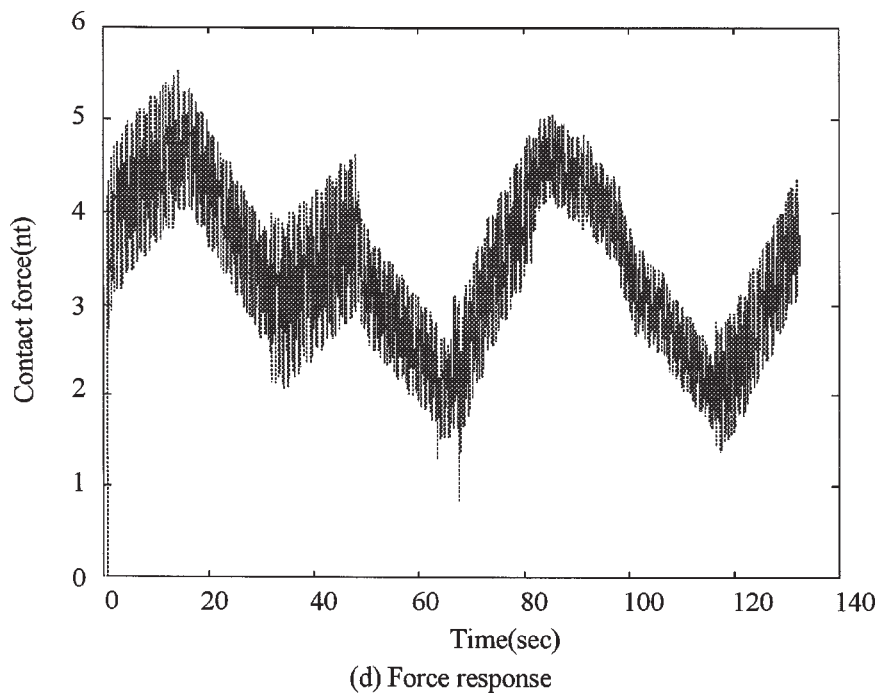
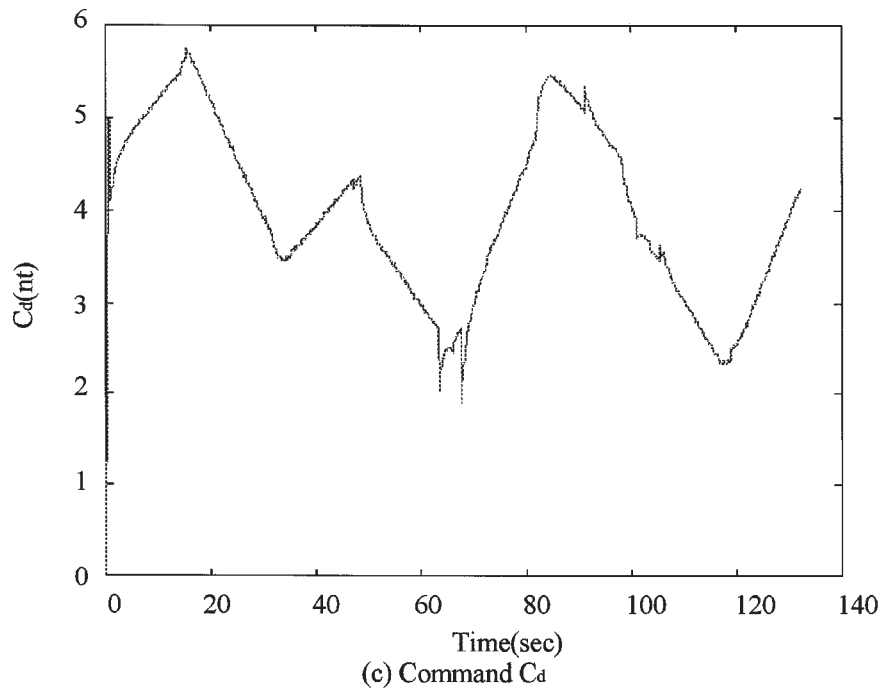


Figure 5. (Continued) (c) command C_d , and (d) force response.

did not respond well to uncertainties and disturbances. Judging by the position and force responses in Figures 4 and 5, we deemed that the sliding mode impedance controller performed better than the impedance controller in dealing with uncertainties and disturbances. One point to note here is that the command trajectories (C_d) in Figures 4(c) and 5(c) are not exactly the same, although they do have similar shapes. This is because different compliance controllers were used, leading to different performances, thus, the reinforcement learning mechanism adaptively responded with different commands, although the same deburring task was tackled.

In the third set of simulations, we used the robust compliance controller alone to perform the deburring task. The purpose was to further investigate the importance of the reinforcement learning mechanism in the proposed scheme. Since the reinforcement learning mechanism was not included for command generation, the estimator in the robust compliance controller was in fact not in use, and fixed position commands were sent directly to the sliding mode impedance controller. The simulation results show that the surface after deburring was in general irregular, and the position errors between the surface after deburring and the desired surface were much larger than those in Figure 4, indicating that fixed position commands were not appropriate for varying burr characteristics and surfaces. Thus, we deemed that the reinforcement learning mechanism did provide proper commands corresponding to task variations for the robust compliance controller to follow; otherwise the robust compliance controller might have faced too wide environmental variations.

4. Conclusion

In this paper, we have proposed a reinforcement learning and robust control scheme for robot compliance tasks. Due to variations present in compliance tasks of the same kind, the reinforcement learning mechanism is used to provide corresponding varying commands. The robust compliance controller is then used to execute the commands in the presence of modeling uncertainties and external disturbances. The cooperation of the reinforcement learning mechanism and the robust compliance controller in the proposed scheme successfully tackled the complexity of planning and controlling compliance tasks. The deburring compliance task was used as a case study, and the simulation results demonstrate the effectiveness of the proposed scheme. Nevertheless, it is quite straightforward to apply the proposed scheme to different kinds of compliance tasks. In future works, we will verify the proposed scheme with experimentation.

Appendix A. Proof of Theorem 1

THEOREM 1. *Given the sliding surface $s(t)$ defined as*

$$s(t) = M\dot{x} + B(x - x_d) + K \int_0^t (x(\sigma) - x_d) d\sigma \quad (A1)$$

with M , B , and K positive definite matrices, if s converges to zero, x will approach the desired position x_d .

Proof. By employing the control law defined in Equations (20) and (21), $s(t)$ is bounded and will converge to zero, as shown in Section 2.2. Let s_u be the upper bound on $s(t)$, i.e., $\|s(t)\| \leq s_u$, $\forall t$, and define

$$y = [y_1, y_2]^T, \quad \text{with } y_1 = \int_0^t (x(\sigma) - x_d) d\sigma,$$

and $y_2 = \dot{y}_1 = x - x_d$. We then have

$$\dot{y}_2 = -M^{-1}B y_2 - M^{-1}K y_1 + s \tag{A2}$$

and

$$\dot{y} = Ay + Cs \tag{A3}$$

with

$$A = \begin{bmatrix} 0 & 1 \\ \frac{-K}{M} & \frac{-B}{M} \end{bmatrix} \tag{A4}$$

and

$$C = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{A5}$$

And, $y(t)$ can be solved as

$$y(t) = e^{At} y(0) + \int_0^t e^{A(t-\tau)} C s(\tau) d\tau. \tag{A6}$$

Because M , B , and K are positive definite, leading to A in Equation (A4) exponentially stable, the first term $e^{At} y(0)$ in Equation (A6) will converge to zero as $t \rightarrow \infty$. By using the Lebesgue dominated convergence theorem [6] and $\|s(t)\| \leq s_u$, $\forall t$, we can show that the second term $\int_0^t e^{A(t-\tau)} C s(\tau) d\tau$ will also converge to zero. Thus, $y(t)$ will approach zero, and $x(t)$ will approach x_d . \square

References

1. Anderson, R. J. and Spong, M. W.: Hybrid impedance control of robotic manipulators, *IEEE J. Robotics Automat.* **4**(5) (1988), 549–556.
2. Asada, H. and Asari, Y.: The direct teaching of tool manipulation skills via the impedance identification of human motions, in: *IEEE Int. Conf. on Robotics and Automation*, 1988, pp. 1269–1274.
3. Asada, H. and Izumi, H.: Automatic program generation from teaching data for the hybrid control of robots, *IEEE Trans. Robotics Automat.* **5**(2) (1989), 166–173.

4. Asada, H. and Liu, S.: Transfer of human skills to neural net robot controllers, in: *IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 2442–2448.
5. Barto, A. G.: Reinforcement learning and adaptive critic methods, in: White and Sofge (eds), *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, 1992, pp. 469–491.
6. Callier, F. and Desoer, C.: *Linear System Theory*, Springer, New York, 1991.
7. Chiaverini, S. and Sciavicco, L.: The parallel approach to force/position control of robotic manipulators, *IEEE Trans. Robotics Automat.* **9**(4) (1993), 361–373.
8. De Schutter, J. and Leysen, J.: Tracking in compliant robot motion: Automatic generation of the task frame trajectory based on observation of the natural constraints, in: *Int. Symp. of Robotics Research*, 1987, pp. 215–223.
9. Gullapalli, V., Franklin, J. A., and Benbrahim, H.: Acquiring robot skills via reinforcement learning, *IEEE Control Systems Magazine* **14**(1) (1994), 13–24.
10. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, Macmillan, New York, 1994.
11. Hirzinger, G. and Landzettel, K.: Sensory feedback structures for robots with supervised learning, in: *IEEE Int. Conf. on Robotics and Automation*, 1985, pp. 627–635.
12. Hogan, N.: Impedance control: An approach to manipulation. Part I: Theory; Part II: Implementation; Part III: Application, *ASME J. Dyn. Systems Meas. Control* **107** (1985), 1–24.
13. Kazerooni, H., Bausch, J. J., and Kramer, B.: An approach to automated deburring by robot manipulators, *ASME J. Dyn. Systems Meas. Control* **108** (1986), 354–359.
14. Kazerooni, H., Sheridan, T. B., and Houpt, P. K.: Robust compliant motion for manipulators. Part I: The fundamental concepts of compliant motion. Part II: Design method, *IEEE J. Robotics Automat.* **2**(2) (1986), 83–105.
15. Khatib, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation, *IEEE J. Robotics Automat.* **3**(1) (1987), 43–53.
16. Kuniyoshi, Y., Inaba, M., and Inoue, H.: Learning by watching: extracting reusable task knowledge from visual observation of human performance, *IEEE Trans. Robotics Automat.* **10**(6) (1994), 799–822.
17. Lee, S. and Lee, H. S.: Intelligent control of manipulators interacting with an uncertain environment based on generalized impedance, in: *IEEE Int. Symp. on Intelligent Control*, 1991, pp. 61–66.
18. Lozano-Perez, T., Mason, M. T., and Taylor, R. H.: Automatic synthesis of fine-motion strategies for robots, *Internat. J. Robotics Res.* **3**(1) (1984), 3–24.
19. Lu, W.-S. and Meng, Q.-H.: Impedance control with adaptation for robotic manipulations, *IEEE Trans. Robotics Automat.* **7**(3) (1991), 408–415.
20. Lu, Z., Kawamura, S., and Goldenberg, A. A.: An approach to sliding mode-based impedance control, *IEEE Trans. Robotics Automat.* **11**(5) (1995), 754–759.
21. Lumelsky, V.: On human performance in telerobotics, *IEEE Trans. Systems Man Cybernet.* **21**(5) (1991), 971–982.
22. Mason, M. T.: Compliance and force control for computer controlled manipulators, *IEEE Trans. Systems Man Cybernet.* **11**(6) (1981), 418–432.
23. Raibert, M. H. and Craig, J. J.: Hybrid position/force control of manipulators, *ASME J. Dyn. Systems Meas. Control* **102** (1981), 126–133.
24. Schmidt, R. A.: *Motor Control and Learning: A Behavioral Emphasis*, Human Kinetics Publishers, Champaign, IL, 1988.
25. Slotine, J.-J. E.: Sliding controller design for nonlinear systems, *Internat. J. Control* **40**(2) (1984), 421–434.
26. Sutton, R. S., Barto, A. G., and Williams, R. J.: Reinforcement learning is direct adaptive optimal control, *IEEE Control Systems Magazine* **12**(2) (1992), 19–22.
27. Wen, J. T. and Murphy, S.: Stability analysis of position and force control for robot arms, *IEEE Trans. Automat. Control* **36**(3) (1991), 365–371.

28. Wu, C. H. and Kim, M. G.: Modeling of part-mating strategies for automating assembly operations for robots, *IEEE Trans. Systems Man Cybernet.* **24**(7) (1994), 1065–1074.
29. Yang, T., Xu, Y., and Chen, C. S.: Hidden Markov model approach to skill learning and its application to telerobotics, *IEEE Trans. Robotics Automat.* **10**(5) (1994), 621–631.
30. Yoshikawa, T. and Sudou, A.: Dynamic hybrid position/force control of robot manipulators – On-line estimation of unknown constraint, *IEEE J. Robotics Automat.* **9**(2) (1993), 220–226.