Journal of Intelligent and Robotic Systems **24:** 253–268, 1999.

Neural Network Force Control for Industrial Robots

SHIH-TIN LIN and SHYI-JANG TZENG
Department of Mechanical Engineering, National Chung-Hsing University, Taichung, Taiwan, R.O.C.

(Received: 18 July 1997; accepted: 12 August 1998)

Abstract. In this paper, we present a hierarchical force control framework consisting of a high level control system based on neural network and the existing motion control system of a manipulator in the low level. Inputs of the neural network are the contact force error and estimated stiffness of the contacted environment. The output of the neural network is the position command for the position controller of industrial robots. A MITSUBISHI MELFA RV-M1 industrial robot equipped with a BL Force/Torque sensor is utilized for implementing the hierarchical neural network force control system. Successful experiments for various contact motions are carried out. Additionally, the proposed neural network force controller together with the master/slave control method are used in dual-industrial robot systems. Successful experiments are carried out for the dual-robot system handling an object.

Key words: neural network control, force control, industrial robots, dual-arm robots.

1. Introduction

For tasks performed by most industrial robot manipulators, such as spray painting, position controllers give adequate performance. However, when a contact is made between the end-effector and the environment, the interaction force must be controlled properly, since otherwise the arising contact forces may damage the object or the robot structure. Two force control methods have been studied extensively over recent years, namely, the hybrid position/force control [1–3] and impedance control [4–8]. As the name implies, the hybrid position/force controller can be used to track positional trajectories and force trajectories in different subspaces simultaneously. One difficulty in implementing the hybrid control method is that we usually do not have precise information on the shape of an object with which the end-effector contacts. To cope with the problem, several approaches to estimate the constraint surface for force control have been studied [9, 10]. Impedance control aims at developing a relationship between interaction forces and end-effector position. By controlling the end-effector position and specifying its relationship to the interaction forces, designers can ensure that the manipulator maneuver in a constrained environment while maintaining appropriate contact forces. In addition to these two methods, there are new approaches developed in recent years. See [11, 12] for details.

In order to implement the hybrid or impedance controllers, the control system must be allowed to have a direct access to actuator torques. However, most industrial robots have built-in position controllers and do not allow direct access to actuator torques. It will be helpful to accomplish force control by using the existing position controllers. This type of control is known as position-based force control [13]. In the applications of force control on industrial robots, Degoulange and Dauchez [14] described an external force control scheme on a non-modified industrial robot controller, and the implementation of this control scheme in PUMA 560 robots UNIMATE controller is carried out.

In most practical applications, conventional control theories are not intelligent enough to tackle more complicated tasks, such as nonlinear dynamics, unknown dynamics, and others. Thus the need for research of neural networks and fuzzy logic control has been noticed by many researchers and in various fields [15–23]. In this paper, we design a neural network force controller and integrate it with an industrial robot position control system. For robot contact tasks, it is difficult to model the environment dynamics accurately. The concept of neural-fuzzy has been applied to robot force control in the presence of environment uncertainties. Ventaraman et al. [24] used neural networks for identifying environments that a robot contacts with. Both function approximation and parameter identification results are presented. Kiguchi et al. [25] designed a controller using fuzzy logic in order to realize a human-like control and then modeled a neural network to adjust membership functions and rules in order to achieve a desired contact force in the presence of unknown environment. Katic and Vukobratovic [26] proposed a new learning control algorithm based on neural network classification of unknown dynamic environment models and neural network learning of robot dynamic model. In this paper, a neural network is used to learn the mapping between the contact force error, environmental stiffness and the accommodated position command to the position controller of an industrial robot.

Of late, the research on multi-arm robot control is an active area. Evidently, multi-arm systems yield greater dexterity and provide capability of transmitting heavy or large objects. It is well known that the coordinated operation of multiple robots opens up new applications in assembly automation and flexible manufacturing systems.

The control problems in coordinated robots are very complex and various control methods have been developed. Hu et al. [27] roughly classified these control approaches in two categories:

- master/slave control, in which one plays the role of the master and the rest is moved in conjunction with the master, and
- object-oriented control, in which users have the ability to specify the interaction between the object and the environment regardless of the robot details. In other words, the position and force trajectories can be specified regardless of the independent robot action.

For further details, see [28, 29]. In the pioneering work, variations of master/slave control have been proposed for cooperation of two robots, see [30, 31]. The basic concept of master/slave control scheme is that the force controlled slave arm follows the position controlled master arm. In other words, the position trajectories of the slave arm are determined by the position trajectories of the master arm. In this paper, the master/slave method is used because only one force/torque sensor is available.

2. Hierarchical Neural Network Force Control

In this paper, we propose a hierarchical control framework based on neural network to accomplish force control problem of a manipulator. The hierarchical force control is shown in Figure 1. The high level of the hierarchical system consists of two parts: neural network control and stiffness estimator. A built-in motion controller of an industrial manipulator is treated as a low level control system. The high level control system determines the motion control commands based on sensory information. The low level control system controls the motion of the robot according to control commands given by the high level control system.

2.1. NEURAL NETWORK CONTROL

The neural network is used to learn the mapping between the contact force error and the accommodated position command to the built-in position controller of the industrial robot. A popular feedforward error-back propagation neural network is used in this paper. The structure of the neural networks is shown in Figure 2, where $\Delta f_{\mathrm{da}} = f_{\mathrm{d}} - f_{\mathrm{a}}$ is the contact force error, K_n is the environmental stiffness and Δx_{c} is the position command to the built-in position controller of the industrial robot.

Figure 2 shows that the neural network consists of three layers, including input/output layers and one hidden layer which has ten neurons. The input to a hidden

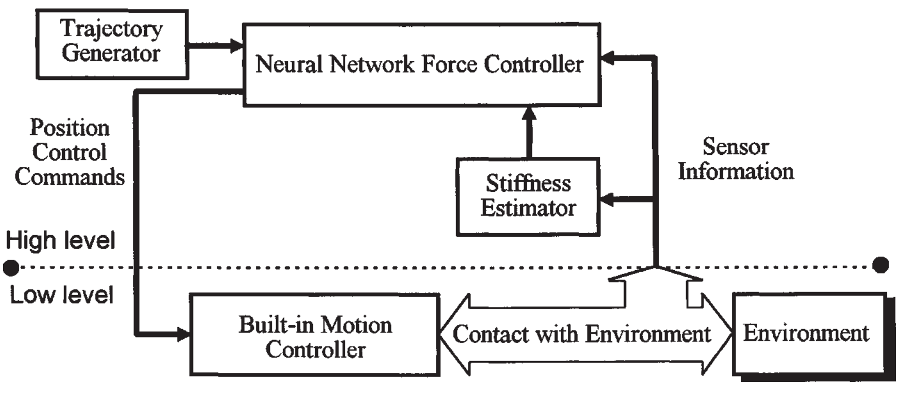

Figure 1. The structure of the hierarchical neural network force control.

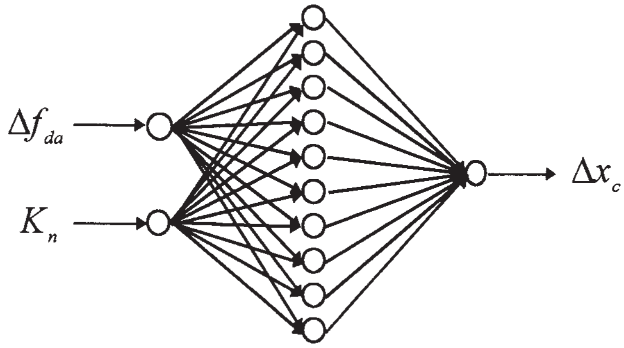

Figure 2. Structure of the neural network.

layer or output layer node is a weighted sum of the previous layer. This sum is passed through a nonlinear activation function $f(x)$. The input–output relationship of neuron unit are as follows:

$$net_i = \sum_{j=1}^{n} w_{ij} o_j, \tag{1}$$

$$o_i = f(net_i), \tag{2}$$

where net_i is the state of unit i, w_{ij} is the interconnection weight between units j and i, and o_i is the output of unit i. In this paper, a typical continuous activation function

$$f(net) = \frac{2}{1 + \exp(-\lambda net)} - 1 \tag{3}$$

is used. The input layer has two nodes and the output layer has one node. Input data consists of contact force error and estimated environmental stiffness. The environmental stiffness is included in the input data to adapt various contact conditions. The output data is the position command to the position controller so that the contact force error will converge to zero.

The neural network is trained using the generalized δ rule [32] to learn the mapping between inputs and outputs of the system. Four different kinds of material labeled as Spring, Sponge, Rubber1, and Rubber2 are used to obtain the training data. A complete training data are listed in Table I. For the generalized δ rule, the weight between ith and jth unit is changed by the following equation:

$$w_{ij}(k) = w_{ij}(k-1) + \eta \delta_i o_j, \tag{4}$$

where η is the learning rate and δ_i is given by the following equation:

$$\delta_i = (d_i - o_i) f'(net_i) \text{ for output units}, \tag{5}$$

$$\delta_i = f'(net_i) \sum_k w_{ki} \delta_k \text{ for other units}, \tag{6}$$

Table I. Training data of the neural network

K_n	Δf_{da}	Δx_c	K_n	Δf_{da}	Δx_c
0.250	–0.600	–0.800	0.889	–0.600	–0.225
0.250	–0.525	–0.700	1.060	–0.525	–0.165
0.250	–0.450	–0.600	1.049	–0.450	–0.143
0.250	–0.375	–0.500	1.041	–0.375	–0.120
0.250	–0.300	–0.400	1.020	–0.300	–0.098
0.250	–0.225	–0.300	1.000	–0.225	–0.075
0.250	–0.150	–0.200	0.943	–0.150	–0.053
0.250	–0.075	–0.100	1.086	–0.075	–0.023
0.250	0.000	0.000	1.086	0.000	0.000
0.250	0.075	0.100	1.086	0.075	0.023
0.250	0.150	0.180	1.667	0.150	0.030
0.250	0.225	0.270	1.974	0.225	0.038
0.250	0.300	0.370	1.887	0.300	0.053
0.250	0.375	0.490	1.667	0.375	0.075
0.250	0.450	0.590	1.530	0.450	0.098

Spring Sponge

K_n	Δf_{da}	Δx_c	K_n	Δf_{da}	Δx_c
0.454	–0.600	–0.441	1.429	–0.600	–0.140
0.526	–0.525	–0.333	1.446	–0.525	–0.121
0.575	–0.450	–0.261	1.613	–0.450	–0.093
0.604	–0.375	–0.207	1.488	–0.375	–0.084
0.654	–0.300	–0.153	1.538	–0.300	–0.065
0.695	–0.225	–0.108	1.596	–0.225	–0.047
0.695	–0.150	–0.072	1.785	–0.150	–0.028
0.926	–0.075	–0.027	1.316	–0.075	–0.019
0.833	0.000	0.000	1.136	0.000	0.000
0.758	0.075	0.033	1.000	0.075	0.025
0.980	0.150	0.051	1.471	0.150	0.034
1.041	0.225	0.072	1.786	0.225	0.042
1.010	0.300	0.099	1.923	0.300	0.052
1.096	0.375	0.114	1.667	0.375	0.075
1.111	0.450	0.135	1.686	0.450	0.089

Rubber1 Rubber2

where d_i is the desired output of the unit, $f'(\cdot)$ is the derivative of the function $f(\cdot)$ with respect to *net*, and k denotes the number of units in the succeeding layer.

2.2. STIFFNESS ESTIMATOR

In order to control the contact force between the end-effector (tool) and the uncertain environment, estimation of the stiffness of the environment becomes very important. If the environment is modeled as a spring, a position command Δx_{c} corresponding to an actual reaction force Δf_{a} can be used to determine the stiffness K_n:

$$K_n = \frac{\Delta f_{\mathrm{a}}(k)}{\Delta x_{\mathrm{c}}(k)}. \tag{7}$$

To estimate the stiffness of the system, the recursive least-squares (RLS) method is considered [33]:

$$K_n(k) = K_n(k-1) + R(k)\big(\Delta f_{\mathrm{a}}(k) - \Delta x_{\mathrm{c}}(k) K_n(k-1)\big), \tag{8}$$

$$R(k) = P(k-1)\Delta x_{\mathrm{c}}(k)\big(1 + P(k-1)\Delta x_{\mathrm{c}}^2(k)\big)^{-1}, \tag{9}$$

$$P(k) = \big(1 - R(k)\Delta x_{\mathrm{c}}(k)\big)P(k-1), \tag{10}$$

where incremental force $\Delta f_{\mathrm{a}}(k)$ is a measured data corresponding to an incremental position command $\Delta x_{\mathrm{c}}(k)$. The initial condition is set to be $K_n(0) = 1300$. Although the resolution (the smallest command allowed) of the RV-M1 industrial robot motion controller is 0.1 mm, the position repeatability of the motion controller is only 0.3 mm, hence the estimated $K_n(k) = K_n(k-1)$ if $\Delta x_{\mathrm{c}}(k) < 0.3$ mm.

3. Control Strategy For Dual-Robot Systems

An object is grasped symmetrically by two robots with open palms as indicated in Figure 3.

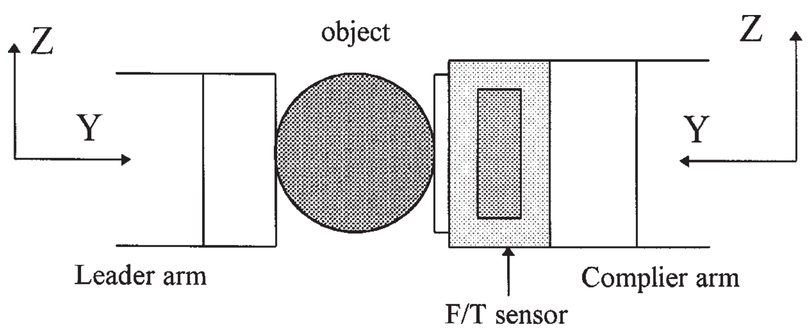

Figure 3. Symmetrical grasp configuration.

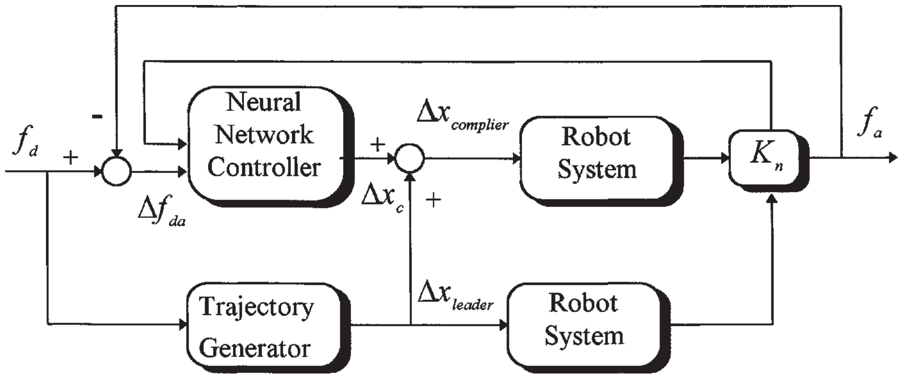

Figure 4. Block diagram of dual-arm robot force control system.

The contact surfaces have friction, thus the surfaces of the robot arms and the object surfaces did not slip if a sufficient squeezing force is maintained. The robot arm that is equipped with a F/T sensor is called the slave arm and the other, without F/T sensor, is called the master arm. The basic concept is that the master arm moves along the prescribed trajectory without concern over the desired squeezing force and the slave arm has to move in a way such that the squeezing force remains constant.

A world reference Cartesian coordinate system is fixed at the base of each robot as shown in Figure 3. The task is planed with respect to the reference frame of the leader arm. The position command $\Delta x_{\mathrm{complier}}$ for complier arm and position command $\Delta x_{\mathrm{leader}}$ for leader arm are given as

$$\Delta x_{\mathrm{complier}} = \Delta x_{\mathrm{G}} + \Delta x_{\mathrm{c}}, \qquad \Delta x_{\mathrm{leader}} = \Delta x_{\mathrm{G}}, \tag{11}$$

where Δx_{G} is the position command that is yielded by trajectory generator while Δx_{c} is generated by the neural network. Figure 4 shows the block diagram for the force control system of a two-robot system.

4. Experiment Results

The implementation of the hierarchical neural network force control scheme is carried out using a MITSUBISHI MELFA RV-M1 industrial robot, equipped with a BL Force/Torque sensor. The configuration of the force control system is shown in Figure 5.

The hierarchical neural network force controller is realized on a Pentium75 personal computer in C Language. The accommodated position command is computed by the force control laws, which is resided in the PC, according to the Force/Torque information. The Drive Unit takes responsibility on position/motion control. The primary function of the F/T sensor controller is to transform the strain gauge data from the transducer (labeled “F/T sensor” in Figure 5) into Cartesian force/torque components and transmit this force/torque information to the PC. Since the position

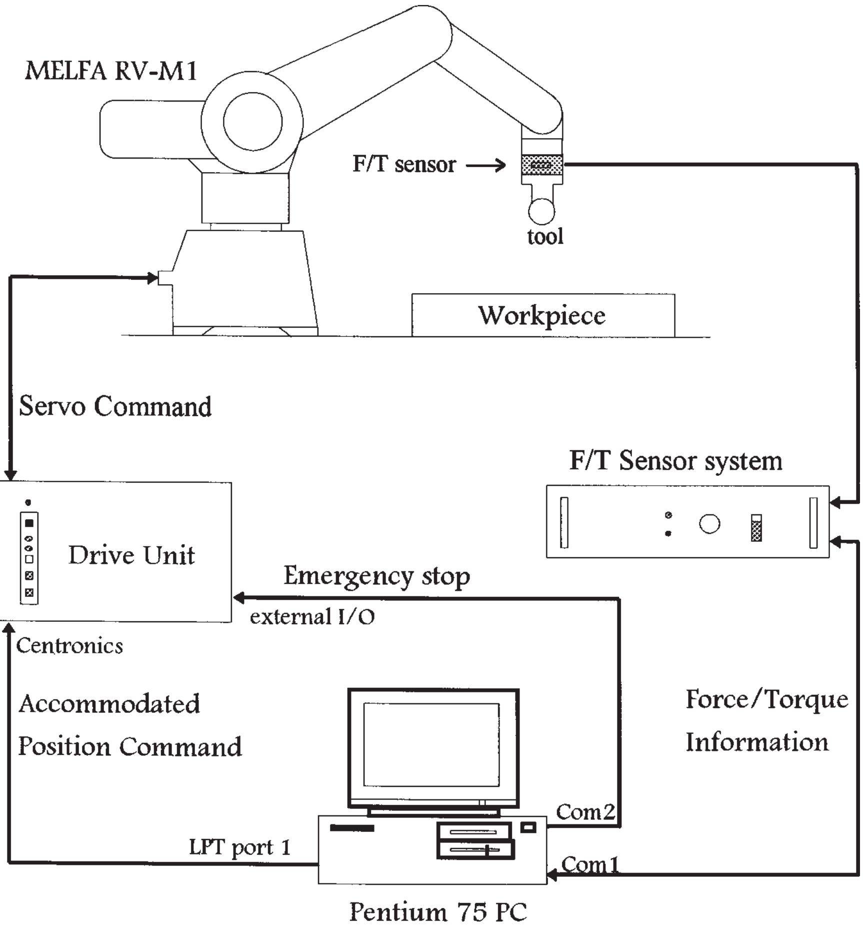

Figure 5. Experimental setup.

controller of the industrial robot do not accept a new position command until the previous one is completed, the neural network force controller sends a new position command only after the previous one is completed. The only exception is when the force sensor sensed a contact force large enough to damage the robot, the controller will send an emergency stop command to the robot. Because of this characteristic, the sampling frequency of the proposed force controller is not fixed.

In the following experiments, the learning rate $\eta = 0.1$ and $\lambda = 1$ in the activation function. The scaling factors of the input and output parameters of the neural network are 0.003 and 10, respectively.

4.1. PUSH AGAINST WORKPIECES

Four workpieces labeled as Spring, Sponge, Rubber1 and Rubber2 are used for the experiment. The results are shown in Figure 6. This figure shows that the

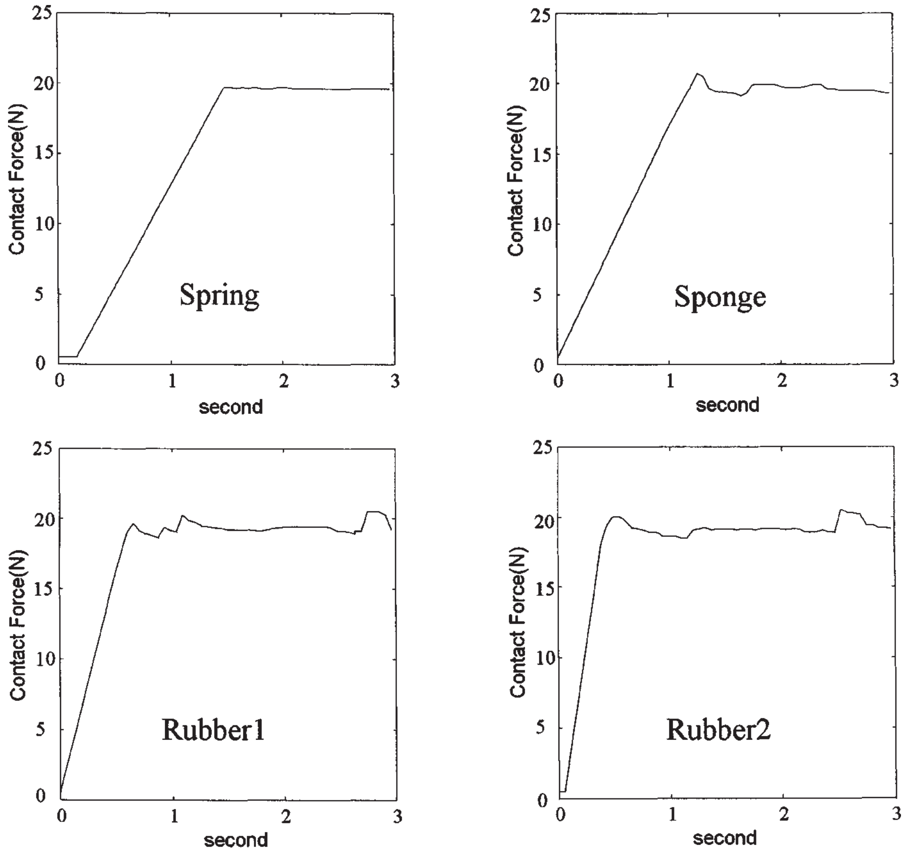

Figure 6. Experimental results: various kinds of materials.

response of the system using Spring has the best performance. The reason for this phenomenon is that Sponge and Rubber can not be modeled as just a spring. However, throughout the experiment, we find that although the response of the system using Sponge or Rubber is not as good as that using Spring, it is still acceptable. Figure 7 shows that the estimated environmental stiffness using the proposed stiffness estimator is close to the true value (1086 kg/m) when the robot pushes against the Rubber1 material. Note that although the performance of the proposed neural network force control method is satisfactory in this experiment, it is available only for soft workpieces due to the limitation of precision of the robot's position controller.

To see the effect of the stiffness estimator, Figure 8 shows the response of the system if we turn off the stiffness estimator and command the robot to push against the Rubber1 material. This figure indicates that a large overshoot and oscillation occurs.

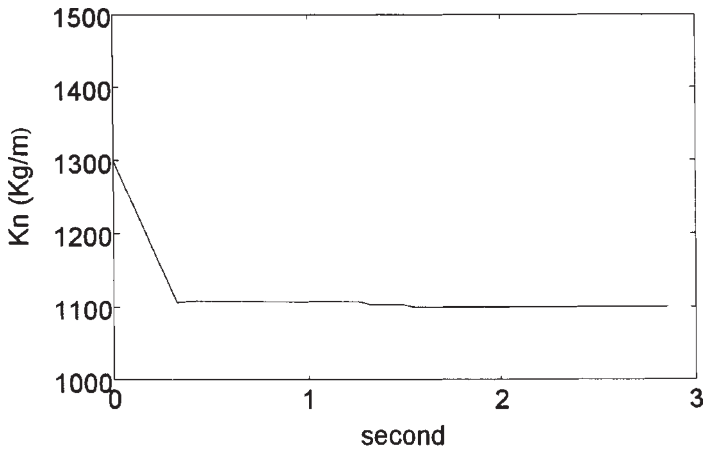

Figure 7. Estimated K_n with recursive least-squares (RLS).

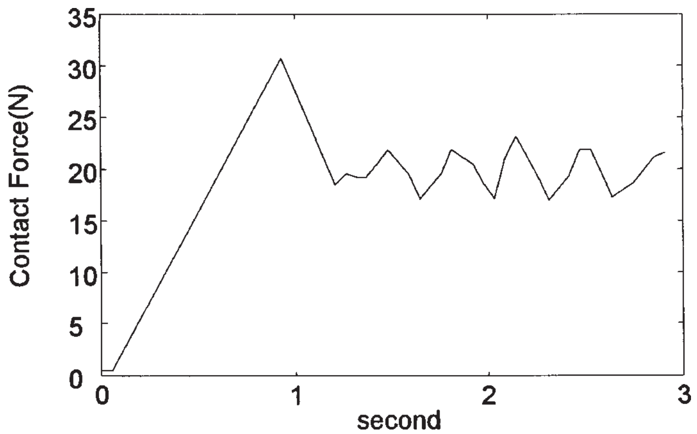

Figure 8. Experiment result if the stiffness estimator is off.

4.2. TRACKING THE SURFACE OF A WORKPIECE

In this experiment, the robot is commanded to track the surface of a workpiece with prescribed normal force 20 N. In order to minimize the resistant force between the end-effector and the contact surface, a roller is mounted to the end-effector. Note that the proposed neural network force control is used only in the normal (X) direction of the workpiece. The original motion controller of the industrial robot is used in the tracking (Y) direction. Figures 9 and 10 show the force response of the system for tracking the surface of two different material, Rubber1 and a mouse pad, respectively. Note that the mouse pad is not included in the training material. These two figures show that the proposed neural network force controller performs well for both trained and untrained materials.

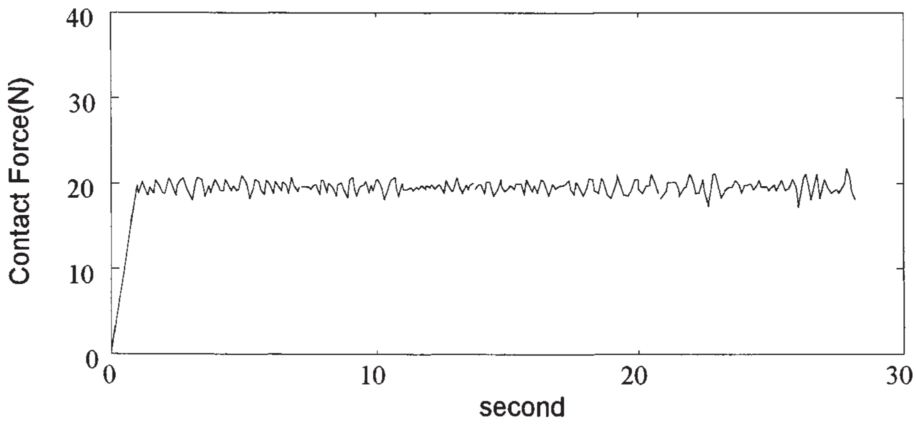

Figure 9. Experiment result by using Rubber1.

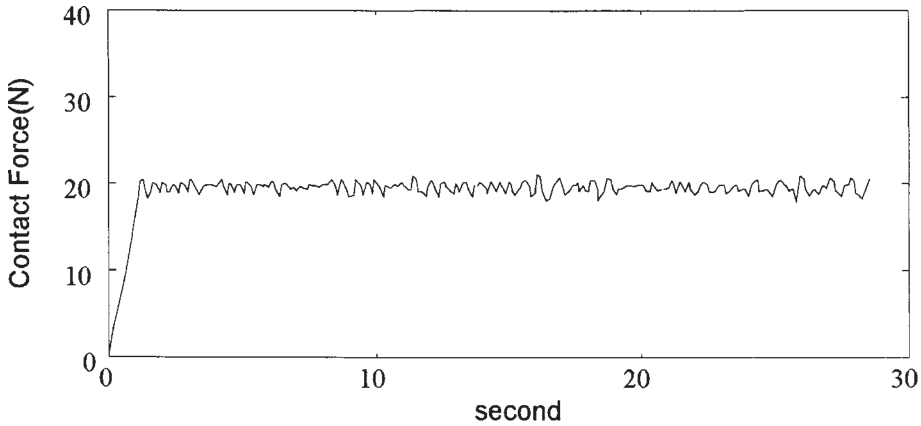

Figure 10. Experiment result by using a mouse pad.

4.3. A DUAL-ROBOT SYSTEM HANDLING AN OBJECT

In this experiment, two MELFA RV-M1 robots (one is equipped with a F/T sensor and the other is not) were used to hold an object, and to move it along a prescribed trajectory, by maintaining squeezing force and constant orientation throughout the motion. In the experiment, the same neural network trained in the single robot system is used for the slave robot. The configuration of the two-robot system is shown in Figure 11. The dual-robot system grasps a tennis ball symmetrically and moves the tennis ball along the semicircular trajectory with diameter 6 cm, and maintains the orientation of tennis ball as shown in Figure 12. At the initial stage, a small bias squeezing force about 2 N to 3 N is applied to the object to prevent the object from falling down. Two cases are considered as follows:

Case 1: The desired squeezing force $f_{\rm d} = 20$ N while moving along the semicircular trajectory. The experimental result is shown in Figure 13. Disturbance force

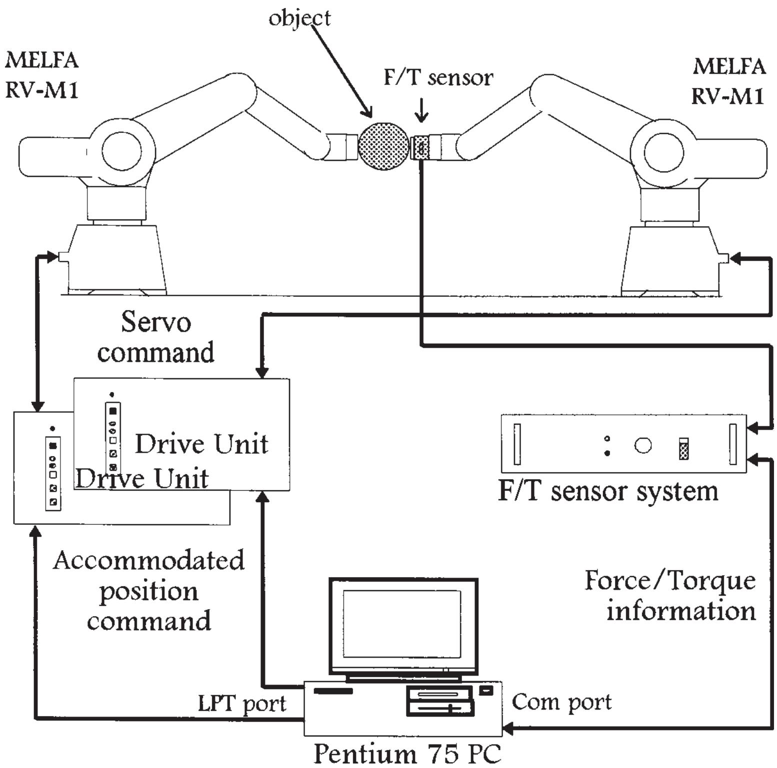

Figure 11. The experiment setup of a two-robot system.

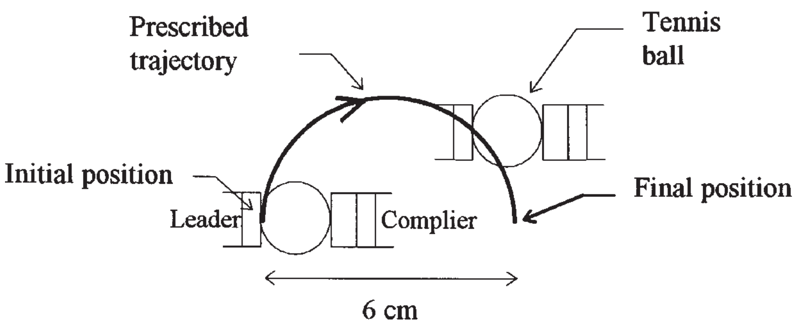

Figure 12. Prescribed semicircular trajectory.

was added to show the capability of disturbance rejection while the object was moved along the trajectory. The result is shown in Figure 14.

Case 2: Track the desired squeezing force $f_d = 1 + 0.5\sin(2\theta)$ while moving along the semicircular trajectory, where $0 \leqslant \theta \leqslant \pi$ as indicated in Figure 15. The experimental result is shown in Figure 16.

Experimental results show that the two-robot system has the capability in tracking force trajectory during the operation of moving object. The key factor is that the

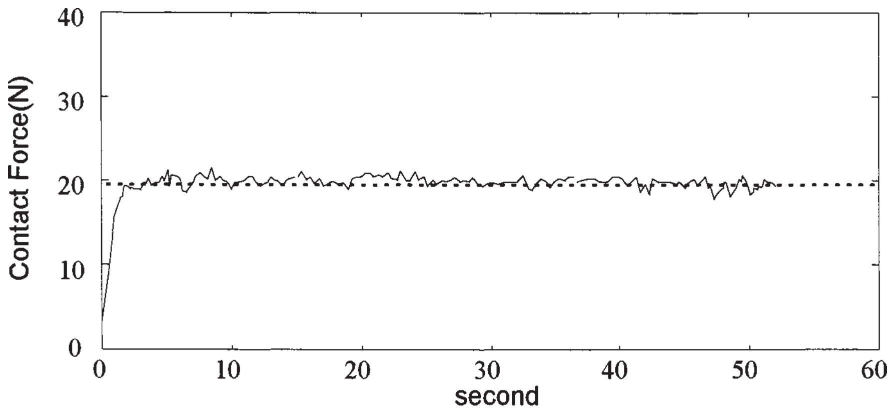

Figure 13. Move along semicircular trajectory with constant desired squeezing force.

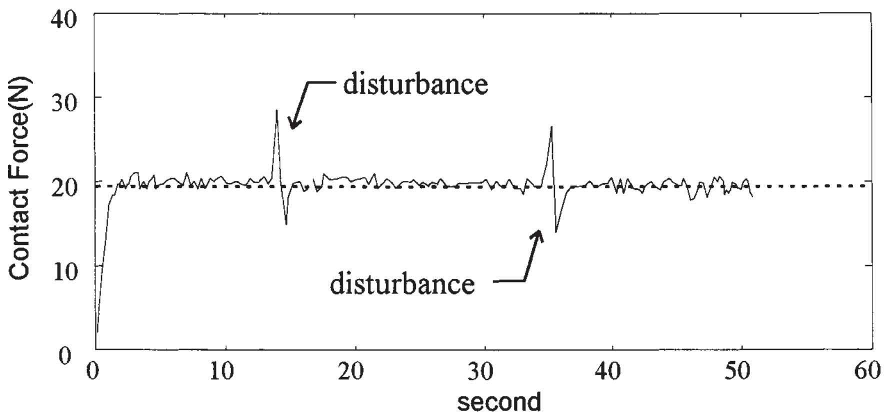

Figure 14. Experimental result of disturbance rejection.

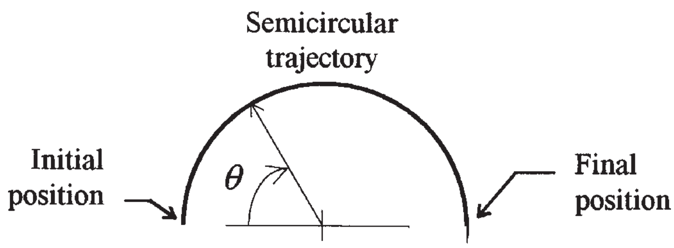

Figure 15. Semicircular trajectory.

neural network force control system generates reasonable accommodated position command Δx_c to maintain the squeezing force. It is verifiable that if the neural network force controller is turned off, the response of the squeezing force diverges as indicated in Figure 17. This situation illustrates that the squeezing force is not under control.

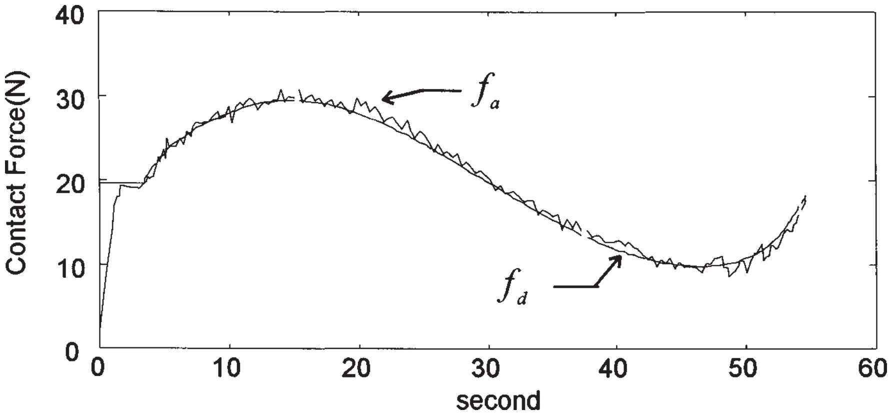

Figure 16. Experiment result with sinusoid desired force trajectory.

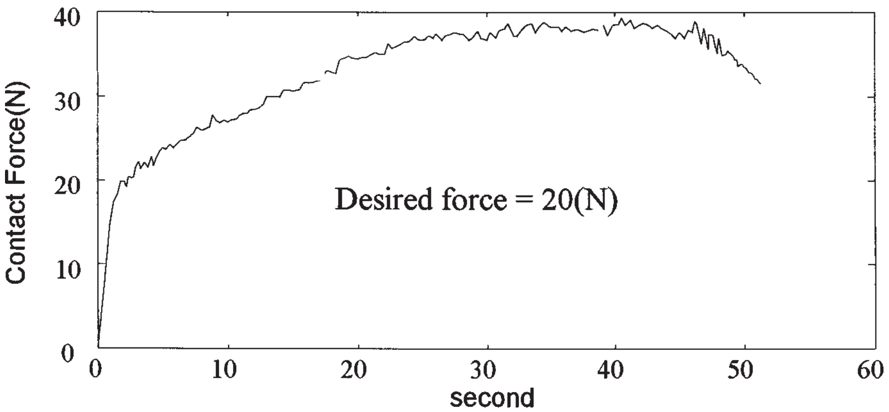

Figure 17. Experiment result if the neural network force controller is switched off.

To conclude, the neural network force control with master-slave algorithm system can be easily applied to the dual-robot system and the control response is quite satisfactory in some applications.

5. Conclusion

In this paper, a hierarchical neural network force control strategy has been applied to industrial robots. The advantage of designing a neural network force controller is that mathematical models of the robot manipulator and contacted environment are not needed. The concept of hierarchical control is more easily applied to the existing industrial manipulators equipped with the motion control system only. For dual-arm robots, the master-slave scheme provides acceptable performance and it could be applied to practical tasks, for example, only one F/T sensor could be equipped what is economically advantageous.

References

1. Mason, M. T.: Compliance and force control for computer controlled manipulators, *IEEE Trans. Systems Man Cybernet.* **11** (1981), 418–432.
2. Raibert, M. and Craig, J.: Hybrid position/force control of manipulators, *ASME J. Dyn. Systems Meas. Control* **102** (1981), 126–132.
3. Koo, C. Y. and Wang, S. P. T.: Nonlinear robust hybrid control of robotic manipulators, *ASME J. Dyn. Systems Meas. Control* **112** (1990), 48–54.
4. Hogan, N.: Impedance control: An approach to manipulation: Part I – Theory, Part II – Implementation, and Part III – Application, *ASME J. Dyn. Systems Meas. Control* **107** (1985), 1–24.
5. Hogan, N.: Stable execution of contact tasks using impedance control, in: *IEEE Internat. Conf. on Robotics and Automation*, 1987, pp. 595–601.
6. Kazerooni, H., Sheridan, T. B., and Houpt, P. K.: Robust compliant motion for manipulator, Part 1: The fundamental concepts, Part 2: Design method, *IEEE J. of Robotics and Automation* **2**(2) (1986), 82–105.
7. Kazerooni, H.: Robust non-linear impedance control for robot manipulators, in: *IEEE Internat. Conf. on Robotics and Automation*, Vol. 2, 1987, pp. 741–750.
8. Kelly, R., Carelli, R., Amestegui, M., and Ortega, R.: Adaptive impedance control of robot manipulators, *Internat. J. Robotics Automat.* **4**(3) (1989), 134–141.
9. Merlet, J. P.: C-surface applied to the design of a hybrid force-position robot controller, in: *Proc. of the 1987 IEEE Internat. Conf. on Robotics and Automation*, Vol. 2, 1987, pp. 1055–1059.
10. Yoshikawa, T. and Sudou, A.: Dynamic hybrid position-force control of robot manipulators: On-line estimation of unknown constraint, *IEEE*, CH2876-1/90/0000/1231, 1990.
11. Zweng, G. and Hemami, A.: An overview of robot force control, *Robotica* **15**(5) (1997), 473–482.
12. Vukobratovic, M. and Ekalo, Y.: New approach to control of robotic manipulators interacting with dynamic environment, *Robotica* **14**(1) (1996), 31–39.
13. Volpe, R. and Khosla, P.: A theoretical and experimental investigation of explicit force control strategies for manipulators, *IEEE Trans. Automat. Control* **38**(11) (1993), 1634–1650.
14. Degoulange, E. and Dauchez, P.: External force control of an industrial PUMA 560 Robot, *J. Robotic Systems* **11**(6) (1994), 523–540.
15. Lin, S. T. and Huang, A. K.: Position-based fuzzy force control for dual industrial robot, *J. Intelligent Robotic Systems* **19**(4) (1997), 393–409.
16. Chan, S. P.: A neural network compensator for uncertainties in robot assembly, *J. Intelligent Robotic Systems* **13** (1995), 127–141.
17. Corbet, T., Sepehri, N., and Lawrence, P. D.: Fuzzy control of a class of hydraulically actuated industrial robots, *IEEE Trans. Control Systems Technol.* **4**(4) (1996), 419–426.
18. Fukuda, T., Kurihara, T, Shibata, T, Tokita, M., and Mitsuoka, T.: Application of neural network-based servo controller to position, force and stabbing control by robotic manipulator, *JSME Internat. J., Series III* **34**(2) (1991), 303–309.
19. Ishiguro, A., Furuhashi, T., and Okuma, S.: A neural network compensator for uncertainties of robotics manipulators, *IEEE Trans. Industr. Electronics* **39**(6) (1992), 565–570.
20. Liu, M. H.: Force-controlled fuzzy-logic-based robotic deburring, *Control Engrg. Practice* **3**(2) (1995), 189–201.
21. Tang, J., Watanabe, K., and Nakamura, M.: Block hierarchical fuzzy-neural networks and their application to a mobile robot control, *J. Robotics Mechatronics* **7**(1) (1995), 12–20.
22. Watanabe, K. and Tzafestas, S. G.: Fuzzy logic controller as a compensator in the problem of tracking control of manipulators, in: *Proc. of the IFToMM-jc Internat. Symp. on Theory of Machines and Mechanisms*, 1992, pp. 98–103.
23. Katic, D. and Vukobratovic, M.: A connectionist based robot control: An overview, in: *Proc. of the 13th IFAC World Congress*, San Francisco, 1996, pp. 169–174.

24. Venkataramam, S., Gulati, S., Barhen, J., and Toomarian, N.: A neural network-based identification of environment models for compliant control of space robot, *IEEE Trans. Robotic Automat.* **9**(5) (1993), 685–697.
25. Kiguchi, K. and Fukuda, T.: Robot manipulator contact force control application of fuzzy-neural network, in: *Proce. of the IEEE Internat. Conf. on Robotic and Automation*, Nagoya, 1995, pp. 875–880.
26. Katic, D. and Vukobratovic, M.: Robot compliance control algorithm based on neural network classification and learning of robot-environment dynamic model, in: *Proc. of the IEEE Internat. Conf. on Robotics and Automation*, Albuquerque, 1997, pp. 2632–2637.
27. Hu, Y. R., Goldenberg, A. A., and Zhou, C.: Motion and force control of coordinated robots during constrained motion tasks, *Internat. J. Robotics Res.* **14**(4) (1995), 351–365.
28. Hayati, S.: Hybrid position/force control of multi-arm cooperating robots, in: *Proc. of the IEEE Internat. Conf. on Robotics and Automation*, Los Alamitos, CA, 1986, pp. 82–89.
29. Yoshikawa, T. and Zheng, X. Z.: Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object, *Internat. J. Robotics Res.* **12**(3) (1993), 219–230.
30. Fujii, S. and Kurono, S.: Coordinated computer control of a pair of manipulators, *Industr. Robot* (1975), 155–161.
31. Ishida, T.: Force control in coordination of two arms, in: *Proc. of the 5th Internat. Joint Conf. on Artificial Intelligence*, 1977, pp. 717–722.
32. Zurada, J. M.: *Introduction to Artificial Neural Network Systems*, West Info Access, Singapore, 1992.
33. Astrom, K. J. and Wittenmark, B.: *Adaptive Control*, 2nd ed., Addison-Wesley, Reading, MA, 1995.