Intelligent Position/Force Controller for Industrial Robot Manipulators—Application of Fuzzy Neural Networks

Kazuo Kiguchi, *Member, IEEE*, and Toshio Fukuda, *Fellow, IEEE*

***Abstract*— An intelligent controller, which consists of an intelligent planner and an adaptive fuzzy neural position/force controller, is proposed for a robot manipulator. The proposed controller deals with the human expert knowledge and skills for planning and control. In this paper, it is applied to the task of deburring with an unknown object. The effectiveness of the proposed controller is evaluated by computer simulations.**

***Index Terms*— Fuzzy neural networks, intelligent robots, manipulators, manufacturing.**

I. Introduction

MANY robot manipulators are used in factories these days. One of the most important and fundamental tasks of the robot manipulators is position/force control. Most of the industrial robots, however, are used only for positioning of objects, since there are some problems in realizing force control for practical uses. Recently, robot manipulators have been expected to perform more sophisticated tasks than just the positioning of objects, even in factories. In order to realize some sophisticated tasks, such as grinding, deburring, wiping, or assembling of objects with the robot, a proper amount of force has to be applied with a proper tool feedrate. As a matter of fact, there are difficulties not only in designing an effective position/force controller, but also in obtaining the desired force and the desired tool feedrate for the tasks. In order to apply force to an object with the robot, the desired force to be controlled has to exist. Furthermore, the desired tool feedrate has to exist in order to perform demanded tasks. However, the desired force and the desired tool feedrate are difficult to decide practically for tasks such as deburring, since the human experts usually control the force and the tool feedrate properly, depending on the property of the object and the size of the burr, with their *a priori* knowledge. Therefore, effective and flexible control rules can be realized if we can obtain the model of the knowledge of experts. It is difficult to make a mathematical model of control rules from the knowledge of experts, because the knowledge of human experts is expressed in vague linguistic rules.

Manuscript received May 20, 1996; revised June 11, 1997.

K. Kiguchi is with the Department of Industrial and Systems Engineering, Niigata College of Technology, Niigata, 950-21 Japan.

T. Fukuda is with the Department of Micro System Engineering, Nagoya University, Nagoya, 464-01 Japan.

Publisher Item Identifier S 0278-0046(97)07761-7.

Some research has been done on transferring the knowledge of human experts to robots. Yang and Asada [1] introduced the linguistic approach of a skill transfer method from human experts to robots using mapping between sensor space and control command. Liu and Asada [2] introduced a skill transfer method using a neural network to store the mapping between task-level process parameters and control strategy parameters. However, it is difficult to handle unexpected situations beyond the training set with their method. Xu and Yang [3] applied the hidden Markov model to model and transfer human skills. These methods require previous experiments to obtain some sensor data caused by human task and prelearning to make a model or a mapping. It is time consuming to do some experiments and/or train the model.

Recently, fuzzy logic or neural networks have been applied for control. It is known that fuzzy control [4], [5] is able to deal with human knowledge. Therefore, a precise mathematical model of the plant and its object are not required for designing the controller. Designing the fuzzy controller is the same as transferring human expert control skills with linguistic rules. The fuzzy control rules are approximations of human control rules. Lingarkar *et al.* [6] applied this ability of the fuzzy control to the force regulation in robotic machining. It is difficult, however, to design the fuzzy controller systematically. Furthermore, once fuzzy control rules and membership functions are designed, usually they will not be modified even if the controller is not perfect. On the other hand, it is known that neural network controllers [7]–[9] have the ability to learn from their experiments and adapt to a new environment. Because of this ability, the controller is especially effective when it has to deal with an unknown object. The neural network controller, however, requires prelearning or time for adapting. If the adaptive-type neural network controller was applied for the control of a robot manipulator, the robot manipulator might cause some damage to the environment before the controller adapts to an unknown environment. Furthermore, the meaning of each weight value of the neural network is not understandable to the users. A fuzzy neural controller, the combination of a fuzzy controller and a neural network controller, is one of the best controllers to overcome these problems [10]. Fuzzy neural networks have been applied in many fields of robotics these days to make the controller intelligent [10]–[13].

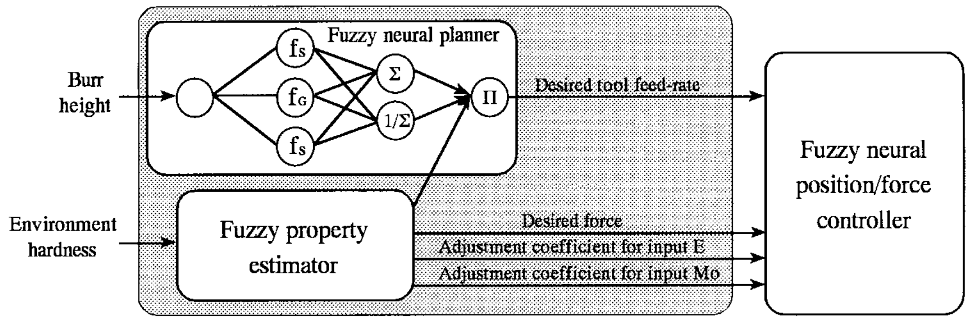

Fig. 1. Proposed intelligent planner.

In this paper, the intelligent method, which does not require either previous experimentation or prelearning, is introduced to acquire the knowledge of experts in order to decide the desired force and the desired tool feedrate for the deburring task, as an example task. The proposed intelligent planner (task-level controller) consists of the fuzzy neural planner and the fuzzy property estimator. The fuzzy neural planner, which is adjusted on-line during the task using an effective evaluation function, is proposed to decide the desired tool feedrate. The fuzzy property estimator is proposed to estimate the material property of an unknown object, generate the desired force to be applied, and generate the input adjustment coefficient for the adaptive fuzzy neural force controller (internal-level controller) to make it adapt immediately to the object.

Choosing an effective evaluation function is an important issue for the learning of the fuzzy neural planner. The fuzzy neural planner should learn only when it has to learn. For example, the desired force and the desired tool feedrate, the outputs from the intelligent planner (Fig. 1), should not be adjusted until the fuzzy neural force controller catches up with it. In this paper, the fuzzy controlled evaluation function is introduced for the effective learning of the fuzzy neural planner.

The effectiveness of the proposed method is evaluated with a three-degree-of-freedom (3DOF) planar robot manipulator (Fig. 2) by computer simulation.

II. Position/Force Control

In order to control force to an object and position on the surface of the object simultaneously with a robot manipulator, a hybrid position/force controller seems to be a proper controller [7], [14]. The basic idea of the hybrid control is separate directions for the force control (orthogonal to the constraint surface of the object) and for the position control (moving along the constraint surface in a Cartesian coordinate system). The coordinate frames used for 3DOF planar robot manipulator position/force control is depicted in Fig. 2.

The dynamics equation of the planar robot is written as

$$M(q)\ddot{q} + h(q, \ddot{q}) + F_{jc}\,\mathrm{sgn}(\dot{q}) = \tau - J^T f \tag{1}$$

where M is the inertia matrix, h is Coriolis and centrifugal components, F_{jc} is Coulomb friction of the robot manipulator joint, t is output torque, J is Jacobian, f is the applied force to the object, and q is an angular position vector.

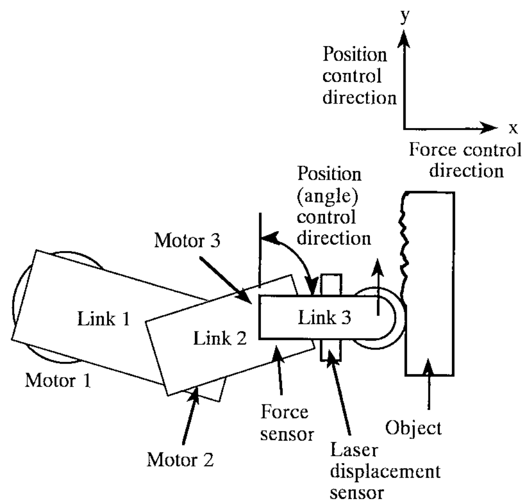

Fig. 2. Deburring task with 3DOF robot manipulator.

From (1), the required motor torque equation becomes

$$\tau = M(q)\ddot{q} + h(q, \dot{q}) + F_{jc}\,\mathrm{sgn}(\dot{d}) + J^T f. \tag{2}$$

The acceleration of the end-effector of the robot manipulator in a Cartesian coordinate system is written as

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \tag{3}$$

where x is a position vector in the Cartesian coordinate system.

From (2) and (3) and the selection matrix S, which selects the direction for the force control that is normal to the constraint surface of the object and for the position control that is along the constraint surface in the Cartesian coordinate system, the following equation can be written for hybrid control:

$$\begin{aligned}\tau =& M(q)J^{-1}[(I - S)u_x - \dot{J}\dot{q}] + h(q, \dot{q}) \\ &+ F_{jc}\,\mathrm{sgn}(\dot{q}) + J^T f + J^T S u_f\end{aligned} \tag{4}$$

where u_x is a command vector for position control and u_f is a command vector for force control.

The block diagram of the hybrid controller is shown in Fig. 3. This controller controls the force applied to the object, the position in the y direction, and the angle of the end-effector (see Fig. 2).

In this paper, the angle of the end-effector is controlled using the resolved acceleration control method [15], since no adaptation is required for the end-effector angle control.

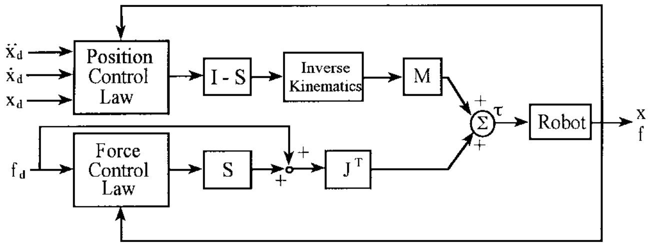

Fig. 3. Block diagram of hybrid control.

The driving force command for the end-effector angle in the Cartesian coordinate system u_{xe}, one of the components of the command vector for position control u_x in (4), is written as

$$u_{se} = \ddot{\theta}_d + K_v(\dot{\theta}_d - \dot{\theta}) + K_p(\theta_d - \theta). \quad (5)$$

Generally, obtaining desired position/force using conventional control methods is difficult because of the following reasons. First, it is very difficult to make perfect mathematical models of the robot manipulator, the tool, and the object. Second, as far as the deburring task has to be performed on several different kinds of objects, the dynamic properties of which are unknown, a high feedback gain for the force control should not be used, since the dynamics of the tool and the object affect the stability of the whole system, while the fixed tool on the robot manipulator is in contact with the object. Third, friction in the robot manipulator joints and between the tool and the object cannot be compensated for completely. Especially in the case where the applied force to the object varies, the amount of Coulomb friction also changes over time. Fourth, external disturbance cannot be avoided. Furthermore, if the object is made of rubber, plastic, wood, etc., the property changes over time, since it is affected by the temperature and humidity of the air. Consequently, adaptation abilities are required for the hybrid controller to overcome these problems.

III. Fuzzy Neural Network

A fuzzy neural network is a combination of fuzzy reasoning and a neural network. The first step in constructing the fuzzy neural network is designing fuzzy rules using the knowledge of human experts. The fuzzy rules represent the knowledge of human experts. In the next step, the designed fuzzy rules are converted to a neural network in order to obtain learning ability. This neural network is called the fuzzy neural network, since its structure is made from fuzzy reasoning.

Making fuzzy control rules means approximations of human knowledge and skill using the human linguistic rules. We cannot expect that human knowledge and skills can be transferred perfectly to fuzzy control rules using the vague human linguistic rules. Therefore, adjustment of the designed fuzzy control rules is required to make the controller more effective. Once the fuzzy control rules are converted to the neural network, they can be adjusted using the backpropagation learning algorithm. Learning of the fuzzy neural network means adjustment for the approximated human knowledge and skills. By adjusting the fuzzy neural networks, the human expert rules are able to be acquired [13]. In this paper, the fuzzy neural network is applied not only for acquisition of human expert knowledge and skill of the deburring task (task-level control), but also for control of the robot manipulator (internal-level control). The block diagram of the proposed system is depicted in Fig. 4.

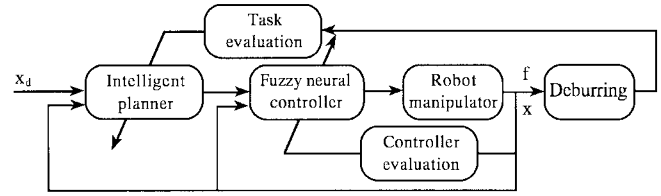

Fig. 4. Block diagram of proposed system.

IV. Fuzzy Neural Position Controller

In order to perform the deburring task with the robot manipulator, the Coulomb friction between the deburring tool and the object has to be taken into account to realize smooth motion. Coulomb friction varies according to the amount of applied force to the object. If the value of the friction coefficient is known previously, it might not be very difficult to compensate for the friction. However, it is difficult to obtain the coefficient of the friction between the tool and an unknown object. The equation of the Coulomb friction F_{ot} is written as

$$F_{ot} = \mu_k f \quad (6)$$

where μ_k is a coefficient of kinematic friction and f is the applied force perpendicular to the object.

The fuzzy neural position controller, which contains a specialized neuron for friction compensation, has been proposed to compensate for unknown friction [12]. In this section, we briefly explain about the proposed fuzzy neural position controller. The architecture of the fuzzy neural position controller is shown in Fig. 5(a). Here, Σ means sum of the inputs and Π means multiplication of the inputs.

There are five kinds of fuzzy numbers (positive big: PB, positive small: PS, zero: ZO, negative small: NS, and negative big: NB) for each input variable (error and manipulator momentum). The membership functions are obtained through the fuzzifier layer of the neural network using the Gaussian function, which is written as (7), and the sigmoidal function, which is written as (9). The simple fuzzifier layer is realized by using these two kinds of functions:

$$f_S(u_S) = \frac{1}{1 + e^{-u_S}} \quad (7)$$

$$u_S(x) = w_o + w_i x \quad (8)$$

$$f_G(u_G) = e^{-u_G^2} \quad (9)$$

$$u_G(x) = \frac{w_o + x}{w_i} \quad (10)$$

where w_i is the threshold value and w_i is the weight. In the Gaussian function, w_0 is a mean value and w_i is a deviation of the membership function.

Calculated membership functions from the fuzzifier layer are sent to the rule layer and multiplied in the neuron in the rule layer according to fuzzy IF–THEN rules. There are two outputs from each neuron in the rule layer. One of them is

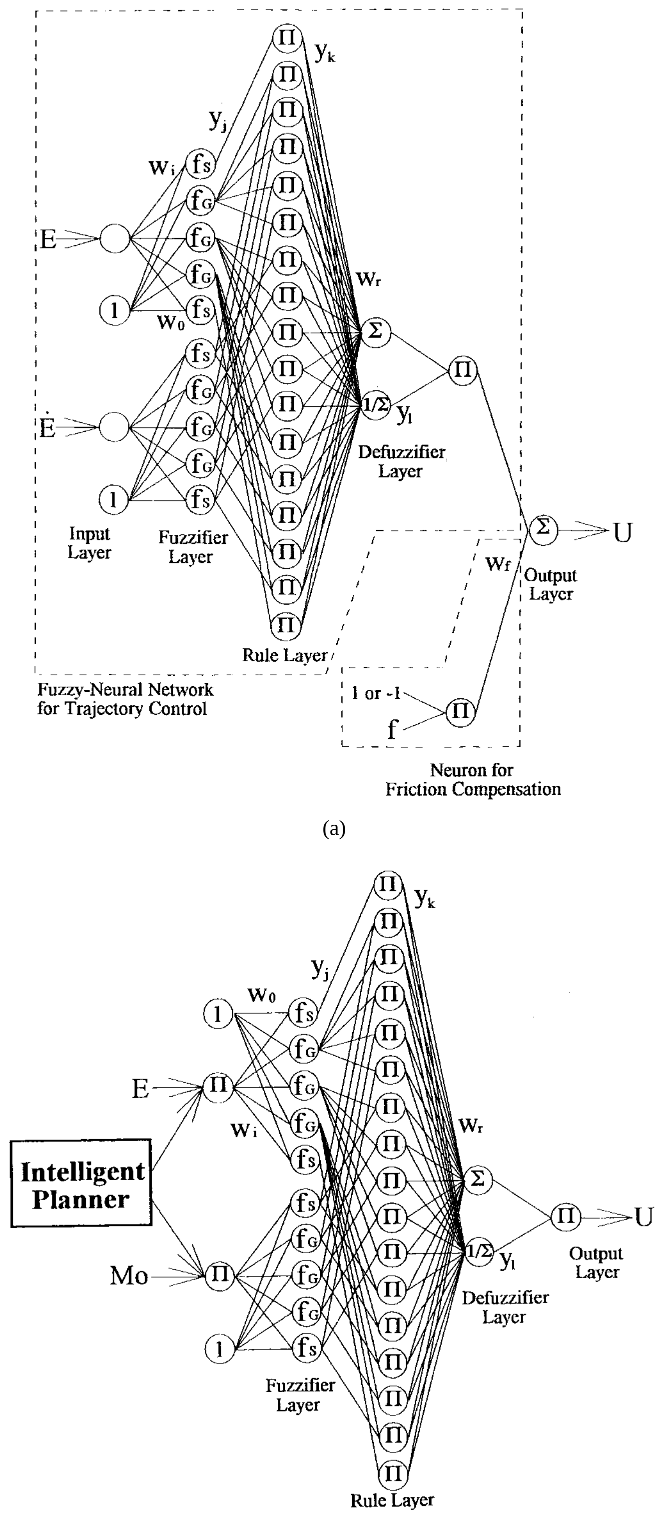

Fig. 5. Architecture of the fuzzy neural controller. (a) Position controller. (b) Force controller.

multiplied by the weight and summed in the next layer. The other is just summed and then inverted in the defuzzifier layer. The multiplied value of these outputs will be the output of the fuzzy neural network. Note that the output of the controller is real value, not fuzzy. The output from the controller is the force command in the Cartesian coordinate.

The fuzzy neural position controller consists of a fuzzy neural network division for trajectory control and a specialized neuron division for friction compensation. Here, the friction means friction between the tool fixed to the robot manipulator and the object. Controller adaptation is carried out at every sampling time. In this controller, however, these two divisions do not learn (adjust each weight value) simultaneously. In other words, learning is switched between the two divisions in the controller depending on the situation, as depicted in Fig. 6. The specialized neuron for friction compensation is able to learn (adjust the weight value) using the backpropagation learning algorithm only when the tool cannot move on the object in spite of the applied driving force. Once the tool begins to move on the object, the specialized neuron for friction compensation stops learning, and the fuzzy neural network for trajectory control begins to learn in turn. On the contrary, if the tool gets stuck on the object, although the controller is still applying the driving force, the fuzzy neural network for trajectory control stops learning, and the friction neuron begins to learn again. While the tool is moving, the friction neuron keeps outputting the force for friction compensation according to the value of the friction coefficient that has been learned. If the direction of the tool movement is changed, the direction of the force for friction compensation is also changed, since $+1/-1$ of the input to the friction neuron changes according to the direction of the robot manipulator movement. We call this learning method switch learning.

The squared error between desired position and measured (calculated) position is adopted for the evaluation function. The objective of the backpropagation algorithm is to minimize this function. The equation of this function is

$$Y = \frac{1}{2}(x_d - x)^2 \tag{11}$$

where x_d is the desired position and x is the measured (calculated) position.

The equation of weight adjustment of the friction neuron, in the case where the direction in which the friction compensation is needed is 1, is

$$\Delta w_f = \eta(x_d - x)f \tag{12}$$

where η is learning rate of the friction neuron. If the direction is -1, the equation is changed to

$$\Delta w_f = -\eta(x_d - x)f. \tag{13}$$

The equation of individual weight adjustment in the defuzzifier layer of the fuzzy neural network for trajectory control is

$$\Delta w_r = \eta(f_d - f)y_t y_k. \tag{14}$$

The equation of individual weight adjustment in the fuzzifier layer of the fuzzy neural network for trajectory control is

$$\Delta w_i = \eta \Sigma \frac{\partial e_{ij}}{\partial y_{ij}} f'_{(u)} u' \tag{15}$$

where y_{ij} is output from the fuzzifier layer, e_{ij} is output error in the fuzzifier layer, and $f'(u)$ and u' are derivatives of

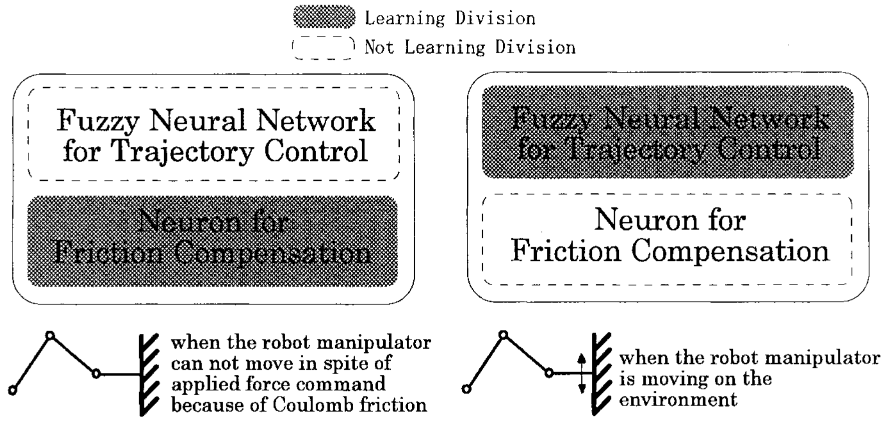

Fig. 6. Switch-learning algorithm.

activation function and input to the function [see (7)–(10)], respectively.

The output from the fuzzy neural position controller, the combination of the output from the fuzzy neural network division for trajectory control and from the specialized neuron division for friction compensation, is the force command for the robot manipulator in the Cartesian coordinate system.

V. Fuzzy Neural Force Controller

In order to apply desired force to an object, the dynamic property of which is unknown, adaptive control such as the adaptive-type neural network control has to be applied. The adaptive-type neural network controller, however, cannot adapt to the unknown object immediately [7]. Consequently, overshooting, which gives some damage to the object, might occur. The authors have proposed the fuzzy neural force controller [10], [11] for robot manipulators to solve this problem. In order to make the fuzzy neural force controller, the fuzzy force controller, which is able to avoid unexpected overshooting, has to be designed first, using human expert knowledge. Then, it is converted to the neural network, which is able to adapt to an unknown object using the backpropagation learning algorithm. The architecture of the fuzzy neural force controller is depicted in Fig. 5(b). Here, Σ means the sum of the inputs and Π means multiplication of the inputs. The fuzzifier layer, in which each neuron represents a membership function for the input, consists of ten neurons, the rule layer consists of 17 neurons, and the defuzzifier layer consists of two neurons.

Usually, fuzzy control uses error and its change rate as the input information to the controller. In the case of force control, however, it is not easy to use error change rate, since the signals from a force sensor are noisy. In this case, the manipulator's pushing velocity against the object might be used instead of force error change rate. As far as robot manipulators are concerned, however, the sense of the velocity depends on the amount of their inertia. When the robot manipulator changes its configuration, the inertia matrix of the robot is also changed. This means that the sense of velocity might be different, even with the same robot manipulator, if its configuration is different. Therefore, the same membership functions of the velocity should not be used if the configuration of the robot manipulator is different. In order to avoid this kind of problem, the authors proposed using the momentum of the robot manipulator instead of the velocity of the robot manipulator [10], [11]. It might also be an effective method to use the kinetic energy of the robot manipulator instead of the velocity. However, dealing with the squared velocity for the input information to the fuzzy neural controller is difficult, since the definition of its membership function becomes very sensitive. Therefore, the momentum of the robot manipulator, the equation of which follows, and the error between the desired force and the measured force are used as input variables to the controller in this paper:

$$M_o = M_x(q)v \tag{16}$$

where M_o represents the momentum of the manipulator in the direction of force control, v is the velocity of the robot manipulator in the direction of force control, and M_x is the inertia matrix in the Cartesian coordinate when the angular position vector of the robot manipulator is q. The inertia matrix in the Cartesian coordinate M_x is written as the following equation, using the inertia matrix in the manipulator angular coordinate M and Jacobian J:

$$M_x(q) = J^{-T}(q)M(q)J^{-1}(q). \tag{17}$$

The learning algorithm of the fuzzy neural force controller is basically the same as that of the fuzzy neural position controller, except for the evaluation function. The equation of the evaluation function for the fuzzy neural force controller is written as

$$Y = \tfrac{1}{2}(f_d - f)^2 \tag{18}$$

where f_d represents the desired force and f is the measured force. Each and every weight of the controller is adjusted at each and every sampling time by the backpropagation learning algorithm to minimize the evaluation function.

Even if the designed fuzzy force control law is not perfect initially, compensation for the friction of the robot manipulator joints is not perfect, external disturbance exists, and/or the property of the environment changes, the proposed force controller is able to apply the desired amount of force to an unknown object with this learning algorithm.

VI. Intelligent Position/Force Controller

The fuzzy neural network is effective both for controlling the robot manipulator and for acquiring the expert knowledge and skill for the sophisticated tasks of the robot manipulator. In order to make the robot manipulator intelligent, a task-level controller (intelligent planner) and an internal-level controller (position/force controller) are required, and both of them should be intelligent. In the intelligent planner (see Fig. 1), the desired force and the desired tool feedrate are decided, depending on the hardness of the object, by the fuzzy property estimator. The decided desired tool feedrate in the fuzzy property estimator is modified, based on the burr size, by the fuzzy neural planner. The intelligent planner also adjusts the two inputs to the internal-level fuzzy neural force controller, in order to make it adapt immediately to an unknown object.

When the material property of the object to be deburred is harder, the expert applies more force and moves the tool slower. When the size of the burr is larger, the expert moves the tool slower. This kind of knowledge is applied to design the intelligent planner rules. Usually, the designed rules are not perfect, since they are converted from vague linguistic rules. In this case, some kinds of rule adaptation abilities are required. If the deburred amount is too little or too much during the deburring task, the generated desired force and/or the desired tool feedrate in the intelligent planner has to be adjusted. In this paper, the desired tool feedrate planning is adapted in order to allow deburring the proper amount by adjusting the membership function and the fuzzy rules in the fuzzy neural planner at every sampling time during the task.

VII. Fuzzy Property Estimator

The fuzzy property estimator in the intelligent planner evaluates the hardness of an unknown object and generates the desired force and the desired tool feedrate for the object. Using the measured information of pushed distance and applied force, the spring constant of the object can be obtained from the following equation:

$$K_s = f/x_e \tag{19}$$

where K_s is the spring constant of the object, f is the applied force to the object, and x_e represents the deformed distance of the object. Usually, the spring constants of objects are spread over a wide range. In this case, the fuzzy numbers of the input variable for the fuzzy property estimator are difficult to define, since the order of the spring constant is different. In this paper, the order of the spring constant K_f, the equation of which follows, is used for the definition of the fuzzy numbers of the input variable for the fuzzy property estimator:

$$K_f = \log(f/x_e). \tag{20}$$

There are two kinds of fuzzy rules for desired signals generation in the fuzzy property estimator. One is for deciding the desired force to be applied for the task, and the other is for deciding the desired tool feedrate depending on the hardness of the object. The fuzzy rules of this estimator have been designed based on the knowledge and the experience of the human experts.

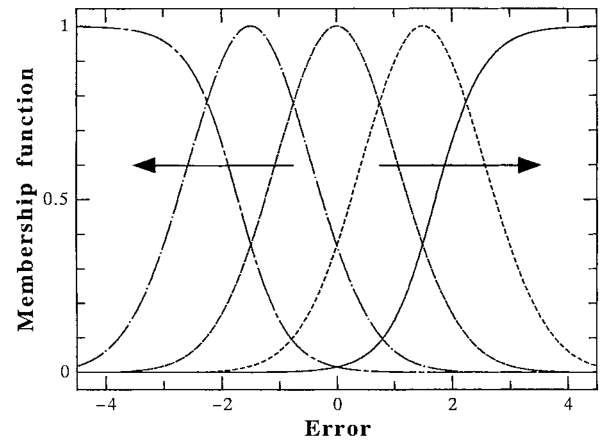

Fig. 7. Change of membership function.

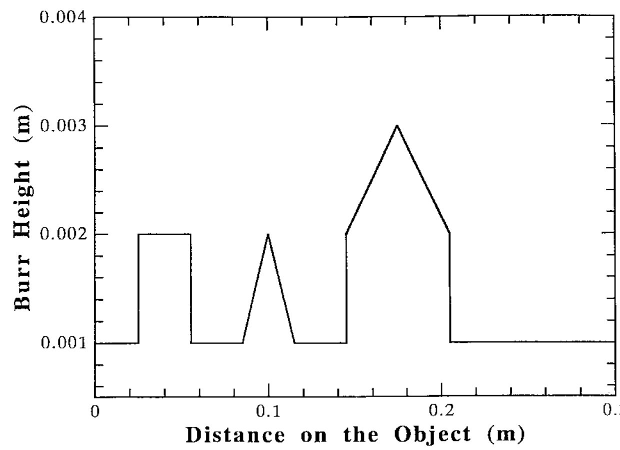

Fig. 8. Assumed burr height.

The other role of the fuzzy property estimator is generating the input adjustment coefficients for the adaptive fuzzy neural force controller. When the hardness of an unknown object is equal, softer, or a little harder than the estimated material property of the object which is used for designing fuzzy rules of the fuzzy neural force controller, the fuzzy neural force controller is able to add the desired force without overshooting. However, if the hardness of an unknown object is much harder than estimated, a large overshoot might occur at the beginning. In this case, the fuzzy neural force controller can be adjusted immediately by multiplying the adjustment coefficient which is smaller than one to one of the input signals E and the adjustment coefficient which is larger than one to the other input signal M_o of the controller. In other words, the antecedent of the fuzzy force control law can be adjusted immediately, according to the hardness of the object, because adjusting the membership function of the input variables has the same effect as changing the shape of the membership function wider or narrower, as shown in Fig. 7. For input signal E of the controller, as an example, the precise control can be realized by this effect as if the error is small when the actual error is large.

Choosing the proper amount of adjustment coefficients is an important issue. It has to be carried out using the knowledge of human experts.

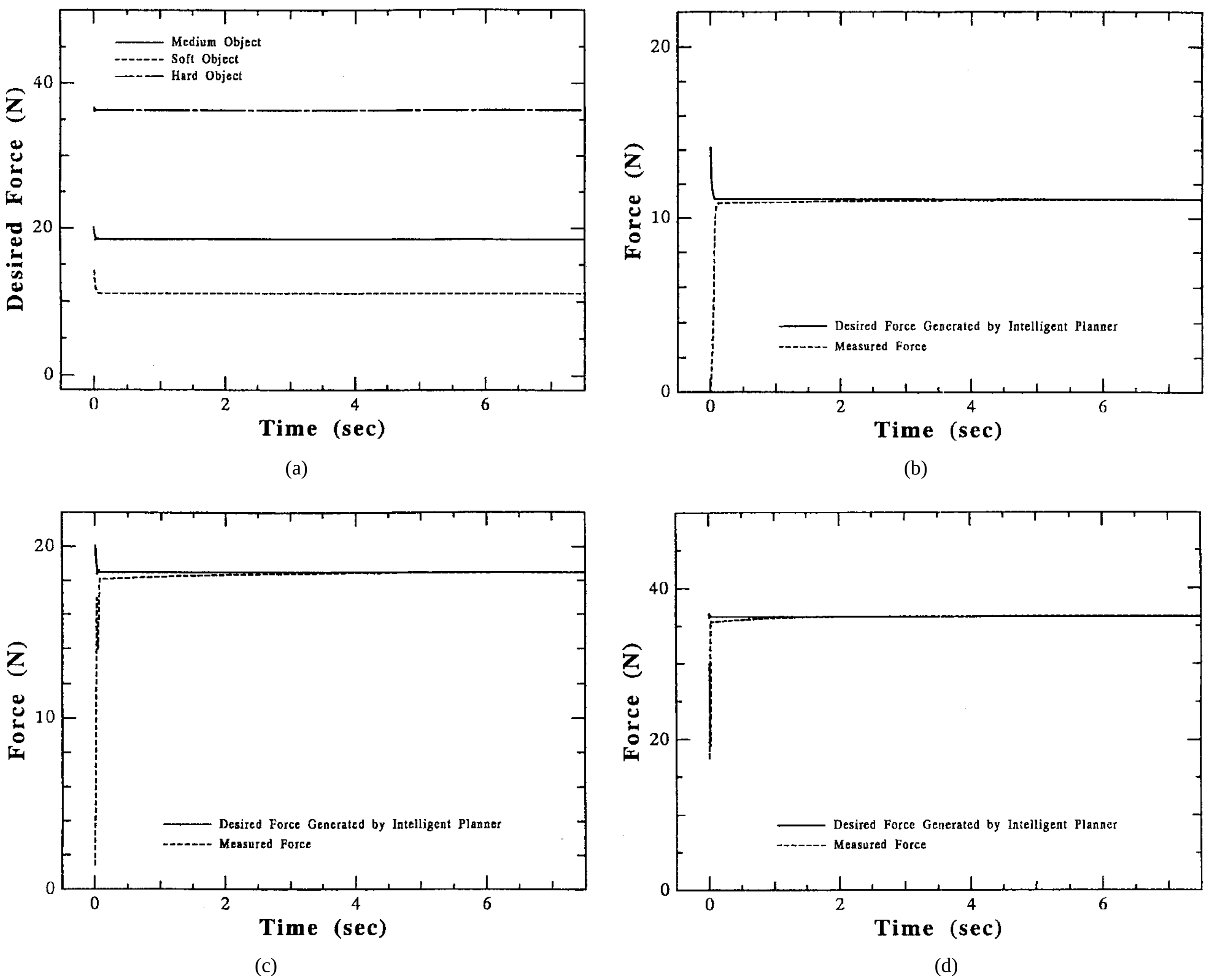

Fig. 9. Simulation results 1. (a) Generated desired force. (b) Force response with a soft object. (c) Force response with a medium object. (d) Force response with a hard object.

VIII. Fuzzy Neural Planner

The fuzzy neural planner in the intelligent planner plays an important role in the deburring task. The fuzzy neural planner generates the desired tool feedrate, depending on the size of the burr and the hardness of the object. The input variable to the fuzzy neural planner is burr size. First, the fuzzy neural planner generates the desired tool feedrate, depending on the burr size. Then, it is multiplied by the generated desired tool feedrate depending on the hardness of the object from the fuzzy property estimator and outputs the modified desired tool feedrate. Therefore, both burr size and hardness of the object are taken into account to decide the desired tool feedrate in the planner. The rules in the fuzzy neural planner have to be adjusted to perform a more effective deburring task using the backpropagation learning algorithm, since the initial rules in the planner are made from vague linguistic human expert knowledge. The squared error between the burr height and the deburred height is used for the evaluation function of the fuzzy neural planner, assuming that both of them are measurable using a camera [16] or a laser displacement sensor [17]. An efficient deburring task can be expected with the robot by minimizing this evaluation function.

A. Fuzzy-Controlled Evaluation Function

The fuzzy neural planner should not learn (should not adjust each weight value) until the reaction force applied by the fuzzy neural force controller and the controlled position by the fuzzy neural position controller catch up with the current desired force and current desired position, respectively. In order to make the fuzzy neural planner learn effectively, considering the situation, the fuzzy-controlled evaluation function is introduced in this paper. Actually, it is a coefficient of the evaluation function, not the evaluation function itself, which is fuzzy controlled. Input variables of the evaluation function fuzzy controller are errors of the internal controller (error of the force controller and the position controller). The coefficient of the evaluation function is controlled to be zero, or very small, when both input variables are large (when the errors of the internal controller are large), and one when both input variables are zero (the internal controller follows exactly the desired trajectory/force). By applying the proposed fuzzy-controlled evaluation function to the learning algorithm, the fuzzy neural planner is able to adapt effectively for the task of deburring an unknown object.

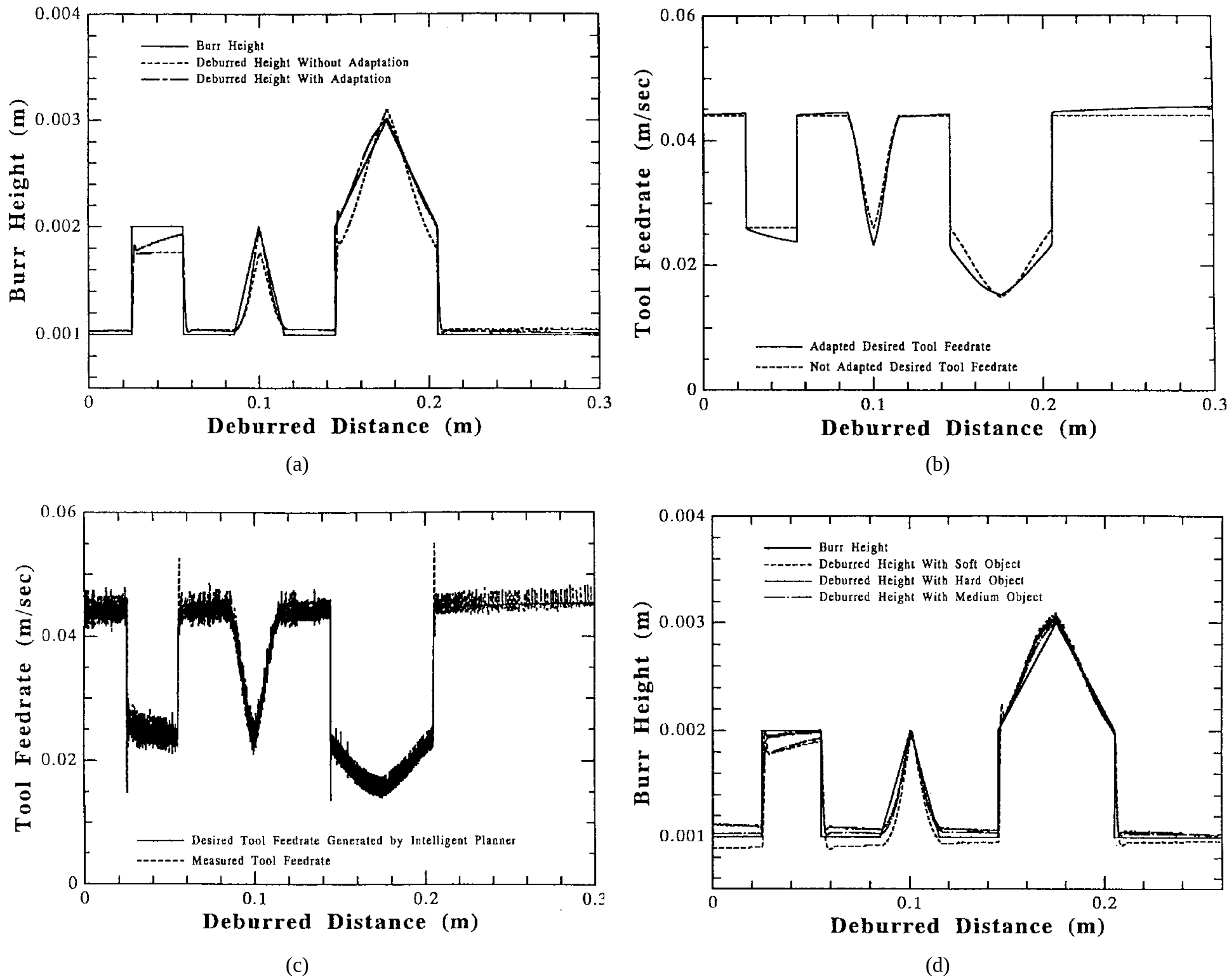

Fig. 10. Simulation results 2. (a) Deburring task. (b) Generated desired tool feedrate. (c) Tool feedrate response. (d) Deburring task with several objects.

IX. Simulation

In order to evaluate the proposed intelligent controller, computer simulation has been performed with a 3DOF planar robot manipulator (Fig. 2). Practically, the height of the burr is between about 1–3 mm. In the simulation, the height of the burr is assumed to be like the shape shown in Fig. 8, considering the practical burr height. The burr height to be deburred and the deburred height are assumed to be able to be monitored using sensors, such as laser displacement sensors, in the simulation. The sampling time is set to 1 ms for every simulation. The angle of the robot manipulator end-effector is controlled to be always perpendicular to the object, using the resolved acceleration control method. The applied force to the object and the position on the object are controlled by the proposed intelligent position/force controller.

In order to evaluate the proposed intelligent planner, some deburring task simulations were carried out with unknown soft, medium, and hard objects. The fuzzy property estimator in the intelligent planner is supposed to generate 20 N for the standard object (the material which is used for design of fuzzy rules of the controller), generate higher desired force for harder objects, and generate lower desired force for softer objects. Fig. 9(a) shows the desired force for three kinds of objects generated by the intelligent planner. In this simulation, the material property of the medium object is the same as that of the standard object, the soft object has a spring constant ten times smaller than the standard object, and the hard object has a spring constant ten times larger than the standard object. We can see that the intelligent planner generates the proper desired force for each material. The applied force responses to the soft object, medium object, and hard object are shown in Fig. 9(b)–(d), respectively. These simulation results show the effectiveness of the adaptive fuzzy neural force controller.

Next, the deburring task was evaluated with the medium object, assuming the object is deburred directly proportional to applied force and inversely proportional to tool feedrate. The motion of the robot manipulator is started a little before the deburring area, since it takes a little time to catch up with the desired tool feedrate. In order to verify the learning ability of the fuzzy neural planner in the intelligent planner, the same deburring task simulation was performed with and without adaptation. The results of this simulation are shown in Fig. 10(a). It is confirmed that the deburred height is well adjusted by the learning of the fuzzy neural planner in the intelligent planner to reduce the deburring error. The desired tool feedrate generated by the intelligent planner with and without adaptation in this simulation is shown in Fig. 10(b). Fig. 10(c) shows the tool feedrate controlled by the fuzzy neural position controller to follow the desired tool feedrate. The results of this simulation show the effectiveness of the adaptive fuzzy neural position controller.

For the last simulation, the proposed controller is applied to the deburring task with a soft, medium, and hard object. The results of the simulation of these tasks are shown in Fig. 10(d). We can see that the deburring error is reduced over time in the case of any object, although the initial error is different, since initially designed fuzzy rules are not perfect.

X. Conclusion

The intelligent position/force controller for a robot manipulator was proposed using fuzzy neural approaches. The proposed controller was applied to the task of deburring as an example task. The controller is able to deal with the human expert knowledge and skills for planning and control of sophisticated tasks.

Some computer simulations of a deburring task for an unknown object were performed with the 3DOF robot manipulator. The simulation results showed the effectiveness of the proposed intelligent controller for any object.

References

[1] B. Yang and H. Asada, "Hybrid linguistic/numeric control of deburring robots based on human skills," in *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 1467–1474.
[2] S. Liu and H. Asada, "Transfer of human skills to robots: Learning from human demonstrations for building an adaptive control system," in *Proc. 1993 American Control Conf.*, pp. 1414–1417.
[3] Y. Xu and J. Yang, "Toward human-robot coordination: Skill modeling and transferring via hidden markov model," in *Proc. 1995 IEEE Int. Conf. Robotics and Automation*, pp. 1906–1911.
[4] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamical plants," *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585–1588, 1974.
[5] W. Pedrycz, *Fuzzy Set Engineering*. Boca Raton, FL: CRC Press, 1995.
[6] R. Lingarkar, L. Liu, G. Bone, M. A. Elbestawi, and N. K. Sinha, "Fuzzy controller for force regulation in robotic machining," in *Proc. Canadian Conf. Electrical and Computer Engineering*, 1988, pp. 595–597.
[7] K. Kiguchi and D. Necsulescu, "Control of multi-DOF robot using neural networks," in *Proc. Knowledge-Based Systems and Robotics Workshop*, 1993, pp. 747–754.
[8] T. Shibata, T. Fukuda, K. Kosuge, F. Arai, M. Tokita, and T. Mitsuoka, "Hybrid symbolic and neuromorphic control for hierarchical intelligent control," in *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 2081–2086.
[9] T. Yabuta and T. Yamada, "Possibility of neural networks controller for robot manipulators," in *Proc. 1990 IEEE Int. Conf. Robotics and Automation*, pp. 1686–1691.
[10] K. Kiguchi and T. Fukuda, "Fuzzy neural controller for robot manipulator force control," in *Proc. Joint Conf. 4th IEEE Int. Conf. Fuzzy Systems and 2nd Int. Fuzzy Engineering Symp. (FUZZ-IEEE/IFES'95)*, 1995, vol. 2, pp. 869–874.
[11] K. Kiguchi and T. Fukuda, "Robot manipulator contact force control application of fuzzy-neural network," in *Proc. 1995 IEEE Int. Conf. Robotics and Automation*, vol. 1, pp. 875–880.
[12] K. Kiguchi and T. Fukuda, "Fuzzy neural friction compensation method of robot manipulation during position/force control," in *Proc. 1996 IEEE Int. Conf. Robotics and Automation,* pp. 372–377.
[13] I. Hiraga, T. Furuhashi, Y. Uchikawa, and S. Nakayama, "An acquisition of operator's rules for collision avoidance using fuzzy neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 280–287, Aug. 1995.
[14] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Trans. ASME, J. Dyn. Syst. Meas. Contr.*, vol. 102, pp. 126–133, June 1981.
[15] J. Y. S. Luh, W. M. Walker, and R. P. C. Paul, "Resolved acceleration control of manipulators," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 468–474, Mar. 1980.
[16] D. E. Whitney, A. C. Edsall, A. B. Todtenkopf, T. R. Kurfess, and A. R. Tate, "Development and control of an automated robotic weld bead grinding system," *Trans. ASME, J. Dyn. Syst. Meas. Contr.*, vol. 112, pp. 166–176, June 1990.
[17] K. Shimokura, H. Asada, and S. Liu, "A study of human skill transfer for teaching a deburring robot," *J. Robot. Soc. Jpn.*, vol. 14, no. 1, pp. 144–149, 1996.

Kazuo Kiguchi (S'92–M'93) received the B.E. degree in mechanical engineering from Niigata University, Niigata, Japan, in 1986, the M.A.Sci. degree in mechanical engineering from the University of Ottawa, Ottawa, Ont., Canada, in 1993, and the D.E. degree from the Department of Mechano-Informatics and Systems, Nagoya University, Nagoya, Japan, in 1997.

He was a Research Engineer with Mazda Motor Company between 1986–1989 and with MHI Aerospace Systems Co. between 1989–1991. He is currently an Assistant Professor in the Department of Industrial and Systems Engineering, Niigata College of Technology, Niigata, Japan. His research interests include intelligent robots, application of fuzzy logic and neural networks for robot control, and application of robotics for medicine.

Toshio Fukuda (M'83–SM'93–F'95) received the B.M.E. degree from Waseda University, Tokyo, Japan, in 1971 and the M.M.E. and D.E. degrees from the University of Tokyo, Tokyo, Japan, in 1973 and 1977, respectively. He also studied at the graduate school of Yale University, New Haven, CT, from 1973 to 1975.

In 1977, he joined the National Mechanical Engineering Laboratory, Japan. From 1979 to 1980, he was a Visiting Research Fellow at the University of Stuttgart, Stuttgart, West Germany. He joined the Science University of Tokyo in 1981 and then, in 1989, he joined Nagoya University, Nagoya, Japan, where he is currently a Professor in the Department of Micro System Engineering and the Department of Mechano-Informatics and Systems. He is mainly engaged in the research fields of intelligent robotic systems, self-organizing systems, microrobotics, robotic systems under hostile environments, biorobotic systems, artificial life, neuromorphic intelligent control, fuzzy control, control of mechanical systems, and technical diagnosis.

Dr. Fukuda has been the Vice President of the IEEE Industrial Electronics Society since 1990. He is the Chairman of the Meeting Committees of the IEEE Robotics and Automation Society, IEEE Industrial Electronics Society, and IEEE Systems, Man, and Cybernetics Society. He was the Publication Chairman (1991–1992) and the Secretary of the IEEE Neural Network Council (1992–1993). He was the Chairman of the Division of Robotics and Mechatronics, Japan Society of Mechanical Engineering (JSME), Chairman of the Technical Committee of Robotics, Society of Instrument and Control Engineers (SICE), and Chairman of many other technical committees. He was also the Steering Chairman of the International Joint Conference on Neural Networks (IJCNN'93-NAGOYA). He was the Founding General Chairman of the IEEE International Workshop on Intelligent Robots and Systems (IROS) held in Tokyo, Japan, in 1988 and Program Chairman of IJCNN'91, Singapore, in 1991. Also, in 1995, he was the General Chairman of the IEEE International Conference on Robotics and Automation and the Program Co-Chairman of the IEEE International Conference on Fuzzy Systems. He is the President-Elect for the IEEE Robotics and Automation Society.