On-line Learning of Robot Arm Impedance Using Neural Networks

Yoshiyuki Tanaka
Graduate School of Engineering, Hiroshima University,
Higashi-hiroshima, 739-8527, JAPAN
Email: ytanaka@bsys.hiroshima-u.ac.jp

Toshio Tsuji
Graduate School of Engineering, Hiroshima University,
Higashi-hiroshima, 739-8527, JAPAN
Email: tsuji@bsys.hiroshima-u.ac.jp

***Abstract*— Impedance control is one of the most effective methods for controlling the interaction between a robotic manipulator and its environments. This method can regulate dynamic properties of the manipulator's end-effector by the mechanical impedance parameters and the desired trajectory. In general, however, it would be difficult to determine them that must be designed according to tasks. In this paper, we propose a new on-line learning method using neural networks to regulate robot impedance while identifying characteristics of task environments by means of four kinds of neural networks: three for the position, velocity and force control of the end-effector; and one for the identification of environments. The validity of the proposed method is demonstrated via a set of computer simulations of a contact task by a multi-joint robotic manipulator.**

I. Introduction

When a manipulator performs a task in contact with its environment, the position and force control are required because of constraints imposed by the environmental conditions. For such contact tasks of the manipulator, the impedance control method [1] is one of the most effective methods for controlling a manipulator in contact with its environments. This method can realize the desired dynamic properties of the end-effector by the appropriate impedance parameters with the desired trajectory. In general, however, it would be difficult to design them according to tasks and its environments including nonlinear or time-varying factors.

There have been many studies by means of the optimization technique with an objective function depending on tasks. Those methods can adapt the desired trajectory of the end-effector according to the task, but there still remains to be accounted for how to design the desired impedance parameters. Besides, the methods cannot be applied into the contact task where the characteristics of its environment are nonlinear or unknown. For this problem, some methods using neural networks (NNs) have been proposed, which can regulate robot impedance properties through the learning of NNs in consideration of the model uncertainties of manipulator dynamics and its environments. Most of such methods using NNs assume that the desired impedance parameters are given in advance, while several methods try to obtain the desired impedance of the end-effector by regulating the impedance parameters as well as the reference trajectory of the end-effector according to tasks and environmental conditions. However, there does not exist an effective method to regulate impedance parameters that can be applied to the case where environmental conditions are changed during task execution.

In this paper, a new on-line learning method using NNs is proposed to regulate all impedance parameters as well as the desired trajectory at the same time. This paper is organized as follows: Section II describes related works on the impedance control method. Then, the proposed learning method using NNs is explained in Sections III and IV. Finally, effectiveness of the proposed method is verified through a set of computer simulations for the given task including the transitions from free to contact movements and the modeling error of environments.

II. Related Works

Asada [2] showed that the nonlinear viscosity of the end-effector could be realized by using a NN model as a force feedback controller. Cohen and Flash [3] proposed a method to regulate the stiffness and viscosity of the end-effector. Also, Yang and Asada [4] proposed a progressive learning method that can obtain the desired impedance parameters by modifying the desired velocity trajectory. Then, Tsuji et al. [5], [6] proposed the iterative learning methods using NNs that can regulate all impedance parameters and the desired end-point trajectory at the same time. Their method can realize smooth transitions of the end-effector from free to contact movements. Venkataraman et al. [7] then developed an on-line learning method using NN for compliant control of space robots according to the given target trajectory of the end-effector through identifying unknown environments with a nonlinear viscoelastic model.

The previous methods [2], [3], [7] cannot deal with a contact task including free movements, while the methods [4], [5], [6] can be applied to only cyclical tasks in which environmental conditions are constant because the learning is conducted in off-line. Considering to make a robot to perform realistic tasks in a general environment, the present paper develops a new method that the robot can cope with an unknown task by regulating the control properties of its movements according to changes of environmental circumstances including non-linear and uncertain factors in real-time.

III. IMPEDANCE CONTROL

In general, a motion equation of an m-joint manipulator in the l-dimensional task space can be expressed as

$$M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \tau + J^T(\theta)F_c, \quad (1)$$

where $\theta \in \Re^m$ denotes the joint angle vector; $M(\theta) \in \Re^{m\times m}$ the non-singular inertia matrix; $h(\theta, \dot{\theta}) \in \Re^m$ the nonlinear term including the joint torque due to the centrifugal, Colioris, gravity and friction forces; $\tau \in \Re^m$ the joint torque vector; $J \in \Re^{l\times m}$ the Jacobian matrix; and $F_c \in \Re^l$ is the external force exerted on the end-effector of the manipulator from the environment in contact movements. The external force F_c can be expressed with an environment model including time-varying and nonlinear factors as

$$F_c = g\left(dX_o, d\dot{X}_o, d\ddot{X}_o, t\right), \quad (2)$$

where $dX_o = X_o^e - X$ represents the displacement vector between the end-effector position X and the equilibrium position on the environment X_o^e; and $g(*)$ is a nonlinear and unknown function.

The desired impedance property of the end-effector can be given by

$$M_e d\ddot{X} + B_e d\dot{X} + K_e dX = F_d - F_c, \quad (3)$$

where $M_e, B_e, K_e \in \Re^{l\times l}$ are the desired inertia, viscosity and stiffness matrices of the end-effector, respectively; $dX = X - X_d \in \Re^l$ is the displacement vector between X and the desired position of the end-effector X_d; and $F_d \in \Re^l$ denotes the desired end-point force vector.

Applying the nonlinear compensation technique with

$$\tau = \left\{\hat{M}^{-1}(\theta)J^T M_x(\theta)J\right\}^T \hat{h}(\theta, \dot{\theta}) - J^T(\theta)F_c + J^T M_x(\theta)(F_{act} - \dot{J}\dot{\theta}) \quad (4)$$

to the nonlinear equation of motion given in (1), the following linear dynamics in the operational task space can be derived as

$$\ddot{X} = F_{act}, \quad (5)$$

where $\hat{M}(\theta)$ and $\hat{h}(\theta, \dot{\theta})$ are the estimated values of M and $h(\theta, \dot{\theta})$, respectively; $M_x(\theta) = (J\hat{M}^{-1}(\theta)J^{\mathrm{T}})^{-1} \in \Re^{l\times l}$ denotes a non-singular matrix as far as the arm is not in a singular posture; and $F_{act} \in \Re^l$ denotes the force control vector represented in the operational task space.

From (3) and (5), the following impedance control law can be designed [5], [6] as

$$F_{act} = F_t + F_f + \ddot{X}_d, \quad (6)$$
$$F_t = M_e^{-1}B_e d\dot{X} + M_e^{-1}K_e dX, \quad (7)$$
$$F_f = -M_e^{-1}(F_d - F_c). \quad (8)$$

Figure 1 shows the block diagram of the impedance control in the operational task space. Note that the force control loop does not exist during free movements because of $F_d = F_c = 0$, while the position and velocity control loop as well as the force one work simultaneously during contact movements. Using the above impedance controller for robotic manipulators, dynamic properties of the end-effector can be regulated by changing the impedance parameters. However, it is so much tough to design the appropriate impedance and the desired trajectory of the end-effector according to given tasks and unknown environmental conditions.

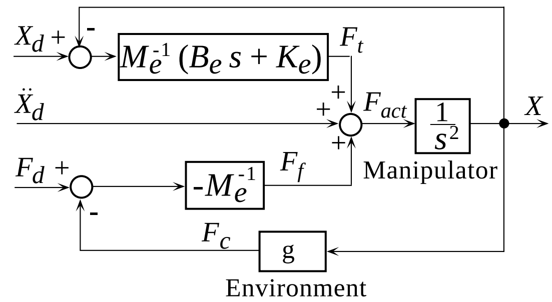


Fig. 1. Impedance control system represented in the operational space

IV. ON-LINE LEARNING OF END-EFFECTOR IMPEDANCE USING NNS

A. Structure of Control System

The proposed control system contains four NNs; three NNs for regulating impedance parameters of the end-effector, and one NN for identifying a task environment. Fig. 2 shows the structure of the proposed impedance control system including three multi-layered NNs: Position Control Network (PCN) for controlling the end-effector position, Velocity Control Network (VCN) for controlling the end-effector velocity, and Force Control Network (FCN) for controlling the end-effector force. The inputs to these NNs include X and dX, and in addition to the end-point force F_c to the FCN. When the learning of the NNs is completed, it can be expected to obtain the optimal impedance parameters $M_e^{-1}B_e$ by PCN, $M_e^{-1}K_e$ by VCN, and M_e^{-1} by FCN, which correspond to the gain matrices of the designed controller in (6), (7), (8).

The linear function is utilized in the input units of NNs, and the sigmoidal function $\sigma_i(x)$ is used in the hidden and output units given by

$$\sigma_i(x) = a_i \tanh(x), \quad (9)$$

where a_i denotes a positive constant. The output of NNs are represented by the following vectors:

$$O_p = \left(o_{p1}^T, o_{p2}^T, \cdots, o_{pl}^T\right)^T \in \Re^{l^2}, \quad (10)$$
$$O_v = \left(o_{v1}^T, o_{v2}^T, \cdots, o_{vl}^T\right)^T \in \Re^{l^2}, \quad (11)$$
$$O_f = \left(o_{f1}^T, o_{f2}^T, \cdots, o_{fl}^T\right)^T \in \Re^{l^2}, \quad (12)$$

where o_{pi}, o_{vi} and $o_{fi} \in \Re^l$ are the vectors that consist of the output values of the PCN, VCN, and FCN. It should be noted that the maximum output value of each NN can be regulated by the parameter a_i in (9).

Learning of the NNs is progressed as follows: First, the PCN and VCN are trained by the iterative learning to improve the tracking control ability of the end-effector to follow the desired

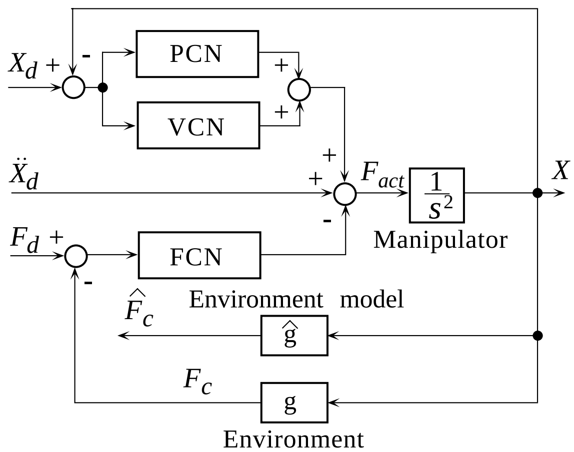


Fig. 2. Proposed impedance control system including three neural networks

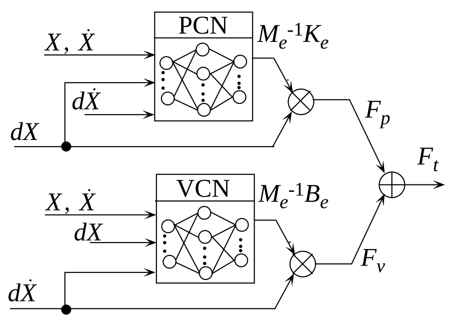


Fig. 3. The structure of the tracking control part using neural networks

trajectory during free movements. The FCN is then trained under the PCN and VCN are fixed during contact movements, where the desired trajectory X_d is also regulated to reduce errors of the end-point force as small as possible.

B. Learning During Free Movements

Figure 3 shows the detailed structure of the tracking control part using PCN and VCN in the proposed impedance control system. The force control input F_{act} during free movements is defined by

$$F_{act} = F_t + \ddot{X}_d = F_p + F_v + \ddot{X}_d$$
$$= \begin{bmatrix} o_{p1}^T \\ o_{p2}^T \\ \cdots \\ o_{pl}^T \end{bmatrix} dX + \begin{bmatrix} o_{v1}^T \\ o_{v2}^T \\ \cdots \\ o_{vl}^T \end{bmatrix} d\dot{X} + \ddot{X}_d, \qquad (13)$$

where F_p, $F_v \in \Re^l$ are the control vectors computed with the output of PCN and VCN, respectively.

The energy function for the learning of PCN and VCN is given by

$$E_t(t) = \frac{1}{2} dX(t)^T dX(t) + \frac{1}{2} d\dot{X}(t)^T d\dot{X}(t). \qquad (14)$$

The synaptic weights in the PCN, $w_{ij}^{(p)}$, and the VCN, $w_{ij}^{(v)}$, are modified in the direction of the gradient descent reducing E_t as

$$\Delta w_{ij}^{(p)}(t) = -\eta_p \frac{\partial E_t(t)}{\partial w_{ij}^{(p)}(t)}, \qquad (15)$$

$$\Delta w_{ij}^{(v)}(t) = -\eta_v \frac{\partial E_t(t)}{\partial w_{ij}^{(v)}(t)}, \qquad (16)$$

$$\frac{\partial E_t(t)}{\partial w_{ij}^{(p)}(t)} = \frac{\partial E_t(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_p(t)} \frac{\partial F_p(t)}{\partial O_p(t)} \frac{\partial O_p(t)}{\partial w_{ij}^{(p)}(t)}, \qquad (17)$$

$$\frac{\partial E_t(t)}{\partial w_{ij}^{(v)}(t)} = \frac{\partial E_t(t)}{\partial \dot{X}(t)} \frac{\partial \dot{X}(t)}{\partial F_v(t)} \frac{\partial F_v(t)}{\partial O_v(t)} \frac{\partial O_v(t)}{\partial w_{ij}^{(v)}(t)}, \qquad (18)$$

where η_p and η_v are the learning rates for PCN and VCN, respectively. The partial differential computations $\frac{\partial E_t(t)}{\partial X(t)}$, $\frac{\partial E_t(t)}{\partial \dot{X}(t)}$, $\frac{\partial F_p(t)}{\partial O_p(t)}$, and $\frac{\partial F_v(t)}{\partial O_v(t)}$ can be derived by (13) and (14), while $\frac{\partial O_p(t)}{\partial w_{ij}^{(p)}(t)}$ and $\frac{\partial O_v(t)}{\partial w_{ij}^{(v)}(t)}$ can be obtained by the error back propagation method [8]. However, $\frac{\partial X(t)}{\partial F_p(t)}$ and $\frac{\partial \dot{X}(t)}{\partial F_v(t)}$ cannot be computed directly because of dynamics of the manipulator. To deal with such computational problems, $\frac{\partial X(t)}{\partial F_p(t)}$ and $\frac{\partial \dot{X}(t)}{\partial F_v(t)}$ are approximated by the finite variations [9] as follows:

$$\frac{\partial X(t)}{\partial F_p(t)} \approx \Delta t_s^2 I, \qquad (19)$$

$$\frac{\partial \dot{X}(t)}{\partial F_v(t)} \approx \Delta t_s I, \qquad (20)$$

where Δt_s is the sampling interval, and I is the l dimensional unit matrix.

When the learning is finished, the trained PCN and VCN may express the optimal impedance parameters $M_e^{-1}K_e$ and $M_e^{-1}B_e$ as the output values of the networks $O_p(t)$ and $O_v(t)$, respectively.

C. Identification of Environments by NN

In the proposed learning method for contact movements, the FCN is trained by using the estimated positional error of the end-effector from the force error information without any positional information on the environment. Consequently, it needs to identify an environment model $\hat{F}_c$ to generate the inputs to the FCN:

$$\hat{F}_c = \hat{g}\left(dX_o, d\dot{X}_o, d\ddot{X}_o, t\right). \qquad (21)$$

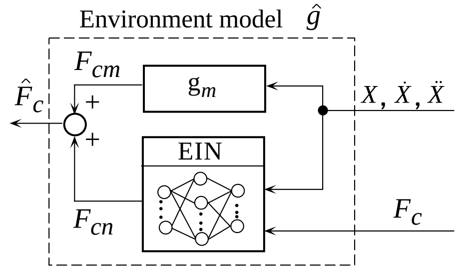


Fig. 4. Identification of the environment using EIN

In some target tasks, the environment model for the manipulator can be expressed with the following linear and time-invariant model as:

$$F_{cm} = g_m\left(dX_o, d\dot{X}_o, d\ddot{X}_o\right) = K_c dX_o + B_c d\dot{X}_o + M_c d\ddot{X}_o, \tag{22}$$

where K_c, B_c, and $M_c \in \Re^{l\times l}$ denote the stiffness, viscosity, and inertia matrices of the environment model, respectively. However, it is much likely that there may exist some modeling errors between the real environment g in (2) and the defined environment model g_m. In the proposed control system, such modeling errors on the environment is dealt with the Environment Identification Network (EIN) which is put in parallel with the environment model g_m as shown in Fig. 4. The inputs to the EIN is the end-point force, position, velocity, and acceleration, while the output is the force compensation F_{cn} for the force error caused by the modeling errors. Accordingly, the end-point force $\hat{F}_c$ is estimated as follows:

$$\hat{F}_c = F_{cm} + F_{cn}. \tag{23}$$

The energy function for the learning of EIN is defined as

$$E_e(t) = \frac{1}{2}\left\{\hat{F}_c(t) - F_c(t)\right\}^T \left\{\hat{F}_c(t) - F_c(t)\right\}. \tag{24}$$

The synaptic weights in the EIN, $w_{ij}^{(e)}$, are modified in the direction of the gradient descent reducing E_e as follows:

$$\Delta w_{ij}^{(e)}(t) = -\eta_e \frac{\partial E_e(t)}{\partial w_{ij}^{(e)}(t)}, \tag{25}$$

$$\frac{\partial E_e(t)}{\partial w_{ij}^{(e)}(t)} = \frac{\partial E_e(t)}{\partial \hat{F}_c(t)} \frac{\partial \hat{F}_c(t)}{\partial F_{cn}(t)} \frac{\partial F_{cn}(t)}{\partial w_{ij}^{(e)}(t)}, \tag{26}$$

where η_e is the learning rate for the EIN. The terms $\frac{\partial E_e(t)}{\partial \hat{F}_c(t)}$ and $\frac{\partial \hat{F}_c(t)}{\partial F_{cn}(t)}$ can be computed by (23) and (24), while $\frac{\partial F_{cn}(t)}{\partial w_{ij}^{(e)}}$ by the error back propagation learning.

The condition $\hat{F}_c = F_c$ should be established at minimizing the energy function $E_e(t)$ by the EIN, so that $\hat{F}_c$ can be utilized for the learning of contact movements even if the exact environment model is unknown.

D. Learning During Contact Movements

The FCN is trained to realize the desired end-point force F_d by utilizing the estimated end-point force $\hat{F}_c$. Note that the synaptic weights of the PCN and VCN are fixed during the FCN learning, and no positional information on the environment is given to the manipulator.

The learning in this stage is performed by exchanging the force control input F_{act} given in (13) with

$$F_{act} = F_t + F_f + \ddot{X}_d = F_t - \begin{bmatrix} o_{f1}^T \\ o_{f2}^T \\ \cdots \\ o_{fl}^T \end{bmatrix} (F_d - F_c) + \ddot{X}_d. \tag{27}$$

The energy function for the FCN learning is defined as

$$E_f(t) = \frac{1}{2}\left\{F_d(t) - F_c(t)\right\}^T \left\{F_d(t) - F_c(t)\right\}. \tag{28}$$

The synaptic weights in the FCN, $w_{ij}^{(f)}$, are modified in the direction of the gradient descent reducing E_f as follows:

$$\Delta w_{ij}^{(f)}(t) = -\eta_f \frac{\partial E_f(t)}{\partial w_{ij}^{(f)}(t)}, \tag{29}$$

$$\frac{\partial E_f(t)}{\partial w_{ij}^{(f)}(t)} = \frac{\partial E_f(t)}{\partial F_c(t)} \left\{ \frac{\partial F_c(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_f(t)} + \frac{\partial F_c(t)}{\partial \dot{X}(t)} \frac{\partial \dot{X}(t)}{\partial F_f(t)} + \frac{\partial F_c(t)}{\partial \ddot{X}(t)} \frac{\partial \ddot{X}(t)}{\partial F_f(t)} \right\} \frac{\partial F_f(t)}{\partial O_f(t)} \frac{\partial O_f(t)}{\partial w_{ij}^{(f)}(t)}, \tag{30}$$

where η_f is the learning rate for the FCN. The terms $\frac{\partial E_f(t)}{\partial F_c(t)}$ and $\frac{\partial F_f(t)}{\partial O_f(t)}$ can be computed by (27) and (28), and $\frac{\partial O_f(t)}{\partial w_{ij}^{(f)}(t)}$ by the error back propagation learning. Moreover, $\frac{\partial X(t)}{\partial F_f(t)}$, $\frac{\partial \dot{X}(t)}{\partial F_f(t)}$, and $\frac{\partial \ddot{X}(t)}{\partial F_f(t)}$ can be approximated as $\frac{\partial X(t)}{\partial F_f(t)} \approx \Delta t_s^2 I$, $\frac{\partial \dot{X}(t)}{\partial F_f(t)} \approx \Delta t_s I$, and $\frac{\partial \ddot{X}(t)}{\partial F_f(t)} = I$, respectively, in the similar way to the learning rules for free movements. On the other hand, $\frac{\partial F_c(t)}{\partial X(t)}$, $\frac{\partial F_c(t)}{\partial \dot{X}(t)}$ and $\frac{\partial F_c(t)}{\partial \ddot{X}(t)}$ are computed by using the estimated end-point force $\hat{F}_c$ in order to concern dynamic characteristics of the environment during contact movements. When the EIN is trained enough to realize $\hat{F}_c = F_c$, those partial differential computations can be carried out with (22) and (23).

On the other hand, the desired trajectory, as well as the impedance parameters, is regulated to reduce learning burdens of the FCN as small as possible. The modification of the desired trajectory $\Delta X_d(t)$ is executed by

$$\Delta X_d(t) = -\eta_d \frac{\partial E_f(t)}{\partial X_d(t)}, \tag{31}$$

$$\frac{\partial E_f(t)}{\partial X_d(t)} = \frac{\partial E_f(t)}{\partial \hat{F}_c(t)} \frac{\partial \hat{F}_c(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_f(t)} \frac{\partial F_f(t)}{\partial X_d(t)}, \tag{32}$$

where η_d is the rate of modification. The desired velocity trajectory is also regulated in the same way. At minimizing the force error of the end-point with the designed learning rule, the FCN may express the optimal impedance parameter M_e^{-1} as the output value of the network $O_f(t)$.

The designed learning rules during contact movements can be utilized under that the EIN has been trained enough to $\hat{F}_c \approx F_c$. However, there is the possibility that the estimated error of $\hat{F}_c$ would be much increased because of the unexpected changes of the environment or the learning error of the EIN. Therefore, the learning rates η_f and η_d during contact movements are defined with the time-varying functions of $E_e(t)$ given in (24) as follows:

$$\eta_f(t) = \frac{\eta_f^{\mathrm{MAX}}}{1 + pE_e(t)}, \tag{33}$$

$$\eta_d(t) = \frac{\eta_d^{\mathrm{MAX}}}{1 + pE_e(t)}, \tag{34}$$

where η_f^{MAX} and η_d^{MAX} are the maximum values of $\eta_f(t)$ and $\eta_d(t)$, respectively; and p is a positive constant. It can be expected that the learning rates defined here become small automatically to avoid the wrong learning when the learning error $E_e(t)$ is large.

V. Computer Simulations

A series of computer simulations is performed with a four-joint planar manipulator to verify the effectiveness of the proposed method, in which the length of each link of the manipulator is 0.2 [m], the mass 1.57 [kg] and the moment of inertia 0.8 [kgm^2]. The given task for the manipulator is a circular motion of the end-effector including free movements and contact movements as shown in Fig. 5, in which the end-effector rotates counterclockwise in 8 [s] and its desired trajectory is generated by using the fifth-order polynomial with respect to the time t. The impedance control law used in this paper is designed by means of the multi-point impedance control method for a redundant manipulator [10], and the sampling interval of dynamics computations was set at Δt_s = 0.001 [s].

The PCN and VCN used in the simulations were of four layered networks with eight input units, two hidden layers with twenty units and four output units. The initial values of synaptic weights were randomly generated under $|w_{ij}^{(p)}|$, $|w_{ij}^{(v)}| < 0.01$, and the learning rates were set at η_p = 13000, η_v = 15. The parameter a_i in (9) was determined so that the output of NNs was limited to -100 $\sim$ 100. The structures of FCN and EIN were settled as four layered networks with ten and eight input units, respectively, two hidden layers with twenty units, and four output units. The parameters for the learning of the NNs were set as $\eta_f^{MAX} = 0.0001$, $\eta_d^{MAX} = 0.01$, $p = 10$, $\eta_e = 0.001$, and the desired end-point force at $F_d = (0, 5)^T$ [N].

In the computer simulation, the estimated environment model under $x < -0.1$ [m] was the same as the task environment g_m in (22) with $K_c = \mathrm{diag.}(0, 10000)$ [N/m], $B_c = \mathrm{diag.}(0, 20)$ [Ns/m], $M_c = \mathrm{diag.}(0, 0.1)$ [kg], but otherwise the following nonlinear characteristics was assumed as modeling errors:

$$F_{cy} = K_{cy} dy_o^2 + B_{cy} d\dot{y}_o^2 + M_{cy} d\ddot{y}_o, \qquad (35)$$

where F_{cy} is the normal force from the environment to the end-effector; and the tangential force F_{cx} = 0. Simulations were conducted under K_{cy} = 1000000 [N/m^2], B_{cy} = 2000 [Ns/m^2], and M_{cy} = 0.1 [kg].

Figure 6 shows the changes of arm postures and end-point forces F_c of the manipulator in the process of learning during contact movements after the learning of free movements. A large interaction force is observed until the learning of the FCN is enough progressed, because the manipulator tries to follow the initial desired trajectory with the trained PCN and VCN for free movements. Then, the end-point force converges to the desired one (5 [N]) through regulating the FCN and the desired trajectory of the end-effector at the same time as shown in Figs. 7, 8, where (i, j) in Fig. 7 represents the element of

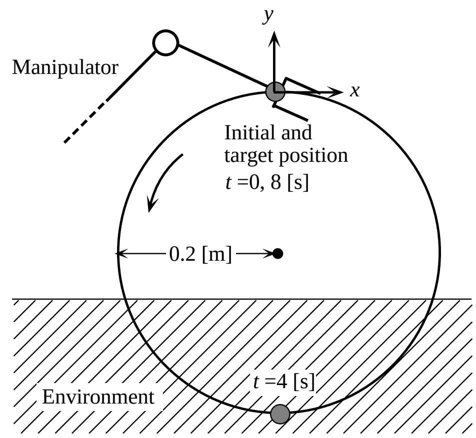
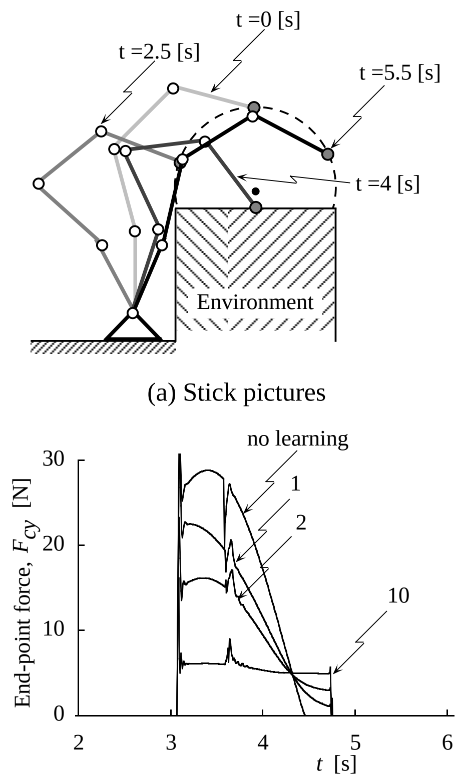


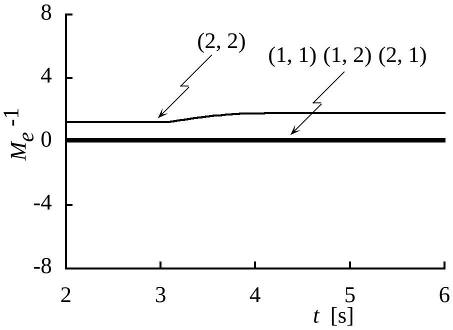
Fig. 5. An example of a contact task



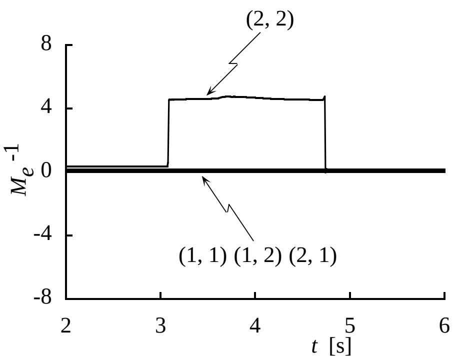
(b) End-point force

Fig. 6. End-point force of the manipulator during learning of a contact movement

M_e^{-1}. It can be found that the manipulator can generate the desired end-point force just after contacts with the environment when the learning on contact movements was completed (at the 10-th trial). However, the force error still remains only in a moment when the characteristics of the environment changes discontinuously, because the estimated environment model by the trained EIN includes some identification errors.



(a) First trial



(b) 10th trial

Fig. 7. Impedance parameters before and after learning of a contact movement

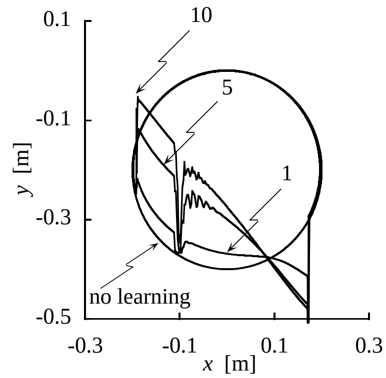


Fig. 8. Virtual trajectories of the manipulator during learning of a contact movement

VI. Conclusions

In this paper, a new learning method using NNs has been proposed to regulate impedance properties of robotic manipulators in real time according to task environmental conditions. The method can regulate all impedance parameters as well as the desired trajectory of the end-effector while identifying the environment model, even if the manipulator contacts with the unknown environment including nonlinear characteristics.

Future research will be directed to analyze the stability and convergence of the proposed control system during the learning, and to develop more appropriate structure of NNs for improving learning efficiency.

References

[1] N. Hogan: "Impedance Control: An Approach to Manipulation, Parts I, II, III," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, Vol.107, No.1, pp.1–24, 1985.

[2] H. Asada: "Teaching and learning of Compliance Using NeuralNets: Representation and Generation of Nonlinear Compliance," in Proceedings of *the 1990 IEEE International Conference on Robotics and Automation*, pp.1237–1244, 1990.

[3] M. Cohen and T. Flash: "learning Impedance Parameters for Robot Control Using an Assosiative Search Networks," *IEEE Transactions on Robotics and Automation*, Vol.7, No.3, pp.382–390, June 1991.

[4] B.-H. Yang and H. Asada: "Progressive Learning and Its Application to Robot Impedance Learning," *IEEE Transactions on Neural Networks*, Vol.7, No.4, pp.941–952, 1996.

[5] T. Tsuji, M. Nishida, and K. Ito: "Iterative Learning of Impedance Parameters for Manipulator Control Using Neural Networks," *Transactions of the Society of Instrument and Control Engineers*, Vol.28, No.12, pp.1461–1468, 1992. (in Japanese)

[6] T. Tsuji, K. Ito, and P. G. Morasso: "Neural Network Learning of Robot Arm Impedance in Operational Space," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, Vol.26, No.2, pp.290–298, April 1996.

[7] S. T. Venkataraman, S. Gulati, J. Barhen, and N. Toomarian: "A NeuralNetwork Based Identification of Environments Models for Compliant Control of Space Robots," *IEEE Transactions on Robotics and Automation*, Vol.9 No.5, pp.685–697, October 1993.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams: "Learning Representations by Error Propagation", in *Parallel Distributed Processing*, MIT Press, Vol.1, pp.318–362, 1986.

[9] T. Tsuji, B. H. Xu and M. Kaneko: "Neuro-Based Adaptive Torque Control of a Flexible Arm," in Proceedings of *the 12th IEEE International Symposium on Intelligent Control*, pp.239-244, 1997.

[10] T. Tsuji, A. Jazidie, and M. Kaneko: "Multi-Point Impedance Control for Redundant Manipulators", *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, Vol.26, No.5, pp.707–718, October 1996.