

On-line learning of robot arm impedance using neural networks

Toshio Tsuji*, Yoshiyuki Tanaka

*Department of Artificial Complex Systems Engineering, Graduate School of Engineering, Hiroshima University,
1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 Japan*

Received 6 July 2004; received in revised form 7 June 2005; accepted 29 June 2005
Available online 3 August 2005

Abstract

Impedance control is an effective control method for a manipulator that is in contact with its environment. Nevertheless, the characteristics of force and motion control are determined by impedance parameters of the end-effector of the manipulator, which must be designed according to the given task. This report presents a method that uses neural networks to regulate impedance parameters of the manipulator's end-effector while identifying environmental characteristics through on-line learning. Four kinds of neural networks are used: three for the position, velocity and force control of the end-effector, and one for the identification of environments. First, the neural networks for the position and velocity control are trained during free movements. Then, the neural networks for the force control and identification of environments are trained during contact movements. Computer simulations show that the method can regulate stiffness, viscosity and inertia parameters of the end-effector and identify unknown properties of the environments through on-line learning.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Impedance control; Robot manipulator; Neural network; On-line learning

1. Introduction

When a manipulator performs a task in contact with its environment, position and force control are required because of constraints imposed by the environment. The impedance control method [1] is an effective control approach for such contact tasks of the manipulator. This method can realize the desired dynamic properties of the end-effector by regulating the mechanical

impedance parameters, i.e., inertia, viscosity, stiffness, and the desired trajectory of the end-effector. However, in general, it is extremely difficult to design them according to the target task and the environmental conditions including nonlinear and time-varying factors.

Many studies have aimed at regulating the impedance property and the desired trajectory of the end-effector by utilizing optimization techniques. 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 character-

* Corresponding author. Tel.: +81 824247676;
fax: +81 824242387.

E-mail address: tsuji@bsys.hiroshima-u.ac.jp (T. Tsuji).

istics 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.

This paper proposes a new on-line learning method using NNs to regulate all impedance parameters and the desired trajectory by extending the off-line learning methods proposed by Tsuji et al. [2,3]. The proposed method can realize the on-line learning of contact tasks by introducing another NN only for identifying the unknown environment model. This paper is organized as follows: Section 2 describes related works on the impedance control method. Then, the proposed learning method using NNs is explained in Sections 3 and 4. Finally, the effectiveness of the proposed method is verified by simulation experiments of contact tasks including the transition from free to contact movements and the modeling error of environments in Section 5.

2. Related works

Asada [4] developed a learning method to obtain the nonlinear viscous compliance of the end-effector by applying NN as a force feedback controller. Cohen and Flash [5] proposed a method using NN to regulate the stiffness and viscosity of the end-effector, in which the NN is trained to minimize a cost function on force and velocity while the desired velocity trajectory of the end-effector is modified to improve the learning performance. Likewise, Yang and Asada [6] proposed a progressive learning method using NN which can obtain the target impedance parameters by modifying the desired velocity trajectory. Against these previous methods using NN, Tsuji et al. [2,3] proposed the iterative learning methods using NNs which can regulate all impedance parameters and the desired

end-point trajectory at the same time. Then, Xiao and Todo [7] developed a discrete-time impedance control algorithm using NN for adapting robot impedance parameters to the unknown contact environment in on-line. Moreover, Venkataraman et al. [8] proposed an on-line learning method using NN which can realize the desired contact force while identifying the characteristics of environment, in which the environment is expressed with a nonlinear viscoelastic model and the desired trajectory is given beforehand.

The previous methods [4,5,8] cannot deal with a contact task including free movements. The method [7] focused on the control of the contact force only in the normal direction of the contact plane for a simple pushing task and cannot regulate the desired end-point trajectory. The methods [2,3,6] 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 nonlinear and uncertain factors in real-time.

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

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

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 . $g(*)$ is a nonlinear and unknown function.

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

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

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

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

to the nonlinear equation of motion 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 estimated values of M and $h(\theta, \dot{\theta})$, respectively. $M_x(\theta) = (J\hat{M}^{-1}(\theta)J^T)^{-1} \in \mathfrak{N}^{l \times l}$ denotes a non-singular matrix insofar as the arm is not in a singular posture. $F_{act} \in \mathfrak{N}^l$ denotes the force control vector represented in operational space. From (3) and (5), the following impedance control law can be designed [2,3,9] as

$$F_{act} = F_t + F_f + \ddot{X}_d, \quad (6)$$

$$F_t = M_e^{-1} B_e d\dot{X} + M_e^{-1} K_e dX, \quad (7)$$

$$F_f = -M_e^{-1}(F_d - F_c). \quad (8)$$

Fig. 1 shows a block diagram of the impedance control in the operational task space. Note that the force control loop does not exist during free movements because $F_d = F_c = 0$. The force control loop functions together with the position and velocity control loop during contact movements. Using the designed impedance controller for robotic manipulators, dynamic properties of the end-effector can be regulated by impedance parameters. In general, however, it is extremely difficult to design appropriate impedance and the desired trajectory of the end-effector according to a given task with

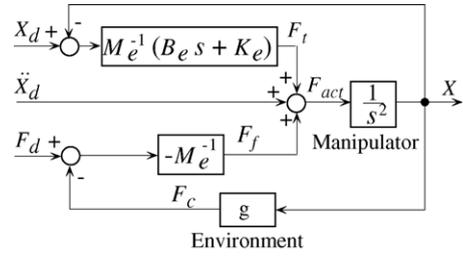


Fig. 1. The block diagram of the impedance control represented in the task space.

an unknown environment by trial and error. Therefore, the next section proposes the two-step learning strategy to regulate impedance parameters as well as the desired trajectory in which a motion controller part and a force controller part are separately trained through the learning of NNs.

4. On-line learning of end-effector impedance using NNs

4.1. Structure of control system

The proposed control system employs four NNs for regulating impedance parameters of the end-effector and for identifying task environment characteristics. Fig. 2 illustrates the structure of the proposed impedance control system including three multi-layered NNs: the Position Control Network (PCN) for controlling the end-effector position; the Velocity Control Network (VCN) for controlling the end-effector velocity, and the Force Control Network (FCN) for controlling the end-effector force. Inputs of these NNs are the end-point position and velocity and the tracking errors to the desired trajectory. Furthermore, the FCN takes the end-point force F_c . When learning is terminated, it can be expected that the trained NNs will output the optimal impedance parameters corresponding to gain matrices of the designed controller in (6)–(8); $M_e^{-1}K_e$ from the PCN, $M_e^{-1}B_e$ from the VCN, and M_e^{-1} from the FCN.

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

$$\sigma_i(x) = a_i \tan h(x), \quad (9)$$

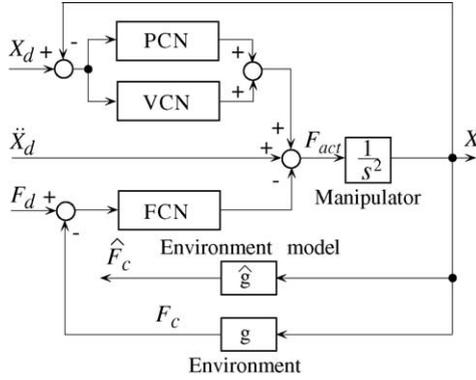


Fig. 2. The block diagram of the impedance control using three neural networks.

where a_i represents a positive constant for regulating the maximum output value. The following vectors represent outputs of NNs:

$$O_p = (o_{p1}^T, o_{p2}^T, \dots, o_{pl}^T)^T \in \mathfrak{N}^{l^2}, \quad (10)$$

$$O_v = (o_{v1}^T, o_{v2}^T, \dots, o_{vl}^T)^T \in \mathfrak{N}^{l^2}, \quad (11)$$

$$O_f = (o_{f1}^T, o_{f2}^T, \dots, o_{fl}^T)^T \in \mathfrak{N}^{l^2}, \quad (12)$$

where o_{pi} , o_{vi} and $o_{fi} \in \mathfrak{N}^l$ comprise the i -th row of matrices $M_e^{-1}K_e$, $M_e^{-1}B_e$, and M_e^{-1} , respectively.

On-line learning of the NNs is conducted in the following two-step procedure in which a motion controller part and a force controller part are separately trained using NNs.

- First: PCV and VCN in a motion control part is trained for improving the tracking control ability of the end-effector to follow the desired trajectory X_d during free movements through minimizing the tracking errors. This implies that the robot is able to prepare for contact by planning X_d adequately, if the environment is given.
- Second: FCN in a force control part is trained for realizing the target force F_d during contact movements through minimizing the force control error, while modifying the desired trajectory to reduce the end-point force error as much as possible.

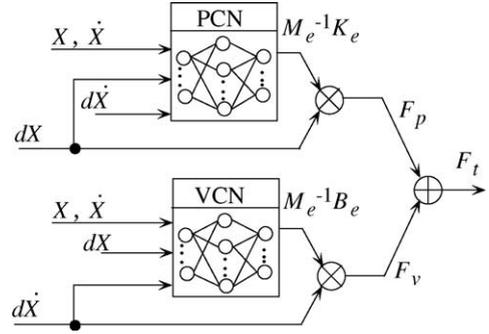


Fig. 3. The structure of the tracking control part using neural networks.

4.2. Learning during free movements

Fig. 3 shows the detailed structure of the tracking control part using PCN and VCN. In free movement, the force control input F_{act} is given as

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

where F_p and $F_v \in \mathfrak{N}^l$ are control vectors computed with outputs of PCN and VCN, respectively.

Learning of PCN and VCN is performed using the following energy function:

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

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

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

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

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

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

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), whereas $\frac{\partial O_p(t)}{\partial w_{ij}^{(p)}(t)}$ and $\frac{\partial O_v(t)}{\partial w_{ij}^{(v)}(t)}$ can be obtained by the error back-propagation method [10]. 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 the manipulator's dynamics. For such computational problems, $\frac{\partial X(t)}{\partial F_p(t)}$ and $\frac{\partial \dot{X}(t)}{\partial F_v(t)}$ are approximated in this paper by finite variations as $\Delta X(t) \approx \Delta F_p(t) \Delta t_s^2$ and $\Delta \dot{X}(t) \approx \Delta F_p(t) \Delta t_s$, respectively, and yield [11]:

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

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

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

When the learning of free movements has sufficiently progressed to reduce the energy function E_t , it can be expected that the trained PCN and VCN may output optimal impedance parameters $M_e^{-1} K_e$ and $M_e^{-1} B_e$, respectively.

4.3. Identification of environments by NN

In the proposed method, to reduce the burden of the FCN learning, the linear environmental model is introduced while modeling errors on the contact environment are estimated using another NN.

An environment model \hat{F}_c can be expressed as

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

According to the target task, it may be possible to give a robotic manipulator an environment model in advance. For that reason, the environment is expressed in this study with the following linear and time-invariant model as

$$\begin{aligned} F_{cm} &= g_m(dX_o, d\dot{X}_o, d\ddot{X}_o) \\ &= K_c dX_o + B_c d\dot{X}_o + M_c d\ddot{X}_o, \end{aligned} \quad (22)$$

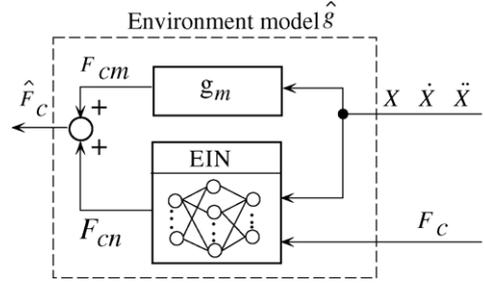


Fig. 4. Identification of the environment model using EIN.

where K_c , B_c , and $M_c \in \mathbb{R}^{l \times l}$ denote the stiffness, viscosity, and inertia property of the environment, respectively. However, some modeling errors are likely to exist between the real environment g in (2) and the given environment model g_m . The proposed control system improves modeling errors on the environment using the Environment Identification Network (EIN), which is put in parallel with the given environment model g_m , as shown in Fig. 4. The EIN receives the end-point force, position, velocity, and acceleration as the input data. Then it outputs the force vector F_{cn} , compensating for the force errors caused by modeling errors. Therefore, the estimated end-point force \hat{F}_c can be obtained as follows:

$$\hat{F}_c = F_{cm} + F_{cn}. \quad (23)$$

The energy function for the learning of EIN can be defined as

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

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

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

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

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), whereas $\frac{\partial F_{cn}(t)}{\partial w_{ij}^{(e)}(t)}$ by error back-propagation learning.

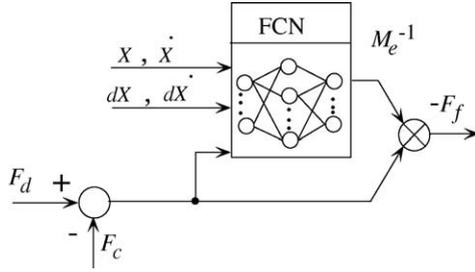


Fig. 5. The structure of the force control part using neural network.

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

4.4. Learning during contact movements

The FCN is trained to realize the desired end-point force F_d with the estimated end-point force \hat{F}_c calculated by the well-trained EIN under the condition that $\hat{F}_c = F_c$ during contact movements, in which the force control error is computed using the environmental model from the position control error of the end-effector.

Fig. 5 shows the FCN structure for learning during contact movements. Note that the synaptic weights of the PCN and VCN are fixed during the learning of FCN to maintain the tracking ability for the desired trajectory after leaving contacted environments.

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

$$F_{act} = F_t + F_f + \ddot{X}_d = F_t - \begin{bmatrix} o_{f1}^T \\ o_{f2}^T \\ \vdots \\ o_{fn}^T \end{bmatrix} (F_d - \hat{F}_c) + \ddot{X}_d. \quad (27)$$

Then, the energy function for the learning of FCN can be defined as

$$E_f(t) = \frac{1}{2} \{F_d(t) - \hat{F}_c(t)\}^T \{F_d(t) - \hat{F}_c(t)\}. \quad (28)$$

Synaptic weights in the FCN, $w_{ij}^{(l)}$, are modified in the direction of the gradient descent reducing E_f as

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

$$\begin{aligned} \frac{\partial E_f(t)}{\partial w_{ij}^{(l)}(t)} &= \frac{\partial E_f(t)}{\partial \hat{F}_c(t)} \left\{ \frac{\partial \hat{F}_c(t)}{\partial X(t)} \frac{\partial X(t)}{\partial F_f(t)} + \frac{\partial \hat{F}_c(t)}{\partial \dot{X}(t)} \frac{\partial \dot{X}(t)}{\partial F_f(t)} \right. \\ &\quad \left. + \frac{\partial \hat{F}_c(t)}{\partial \ddot{X}(t)} \frac{\partial \ddot{X}(t)}{\partial F_f(t)} \right\} \frac{\partial F_f(t)}{\partial O_f(t)} \frac{\partial O_f(t)}{\partial w_{ij}^{(l)}(t)}, \quad (30) \end{aligned}$$

where η_f is the learning rate for the FCN. The terms $\frac{\partial E_f(t)}{\partial \hat{F}_c(t)}$ and $\frac{\partial F_f(t)}{\partial O_f(t)}$ can be computed by (27) and (28), and $\frac{\partial O_f(t)}{\partial w_{ij}^{(l)}(t)}$ 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 similarly to the learning rules for free movements 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. The others, $\frac{\partial \hat{F}_c(t)}{\partial X(t)}$, $\frac{\partial \hat{F}_c(t)}{\partial \dot{X}(t)}$ and $\frac{\partial \hat{F}_c(t)}{\partial \ddot{X}(t)}$, are computed using the estimated end-point force \hat{F}_c to concern dynamic characteristics of the environment during contact movements with (22) and (23).

On the other hand, the desired trajectory is regulated to reduce learning burdens on the FCN as much as possible using the following modifying rule $\Delta X_d(t)$ as

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

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

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

The designed learning rules during contact movements can be utilized under the condition that the EIN has been trained sufficiently to establish the relationship $\hat{F}_c \approx F_c$. However, the estimated error of \hat{F}_c may be greatly increased because of unexpected environmental changes or the EIN learning error. To overcome this problem, the learning rates η_f and η_d during contact movements are determined with time-varying

functions with respect to $E_e(t)$ given in (24) as

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

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

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 is reasonable that the learning rates defined here become small automatically to avoid mislearning when the learning error $E_e(t)$ is large.

5. Application to contact tasks

Effectiveness of the proposed method is investigated through a series of computer simulations of two kinds of contact movements including transitions between free and contact movements. The employed robotic manipulator is of a four-joint planar manipulator with length of each link is 0.2 m, the mass 1.57 kg, and the moment of inertia 0.8 kg m². The impedance control law is designed by means of the multi-point impedance control method for a redundant manipulator [12,13]. The desired trajectory of the end-effector is generated using the fifth-order polynomial with respect to time t [14].

The PCN and VCN are of four-layered networks with eight input units, 2 hidden layers with 20 units, and four output units. The FCN and EIN are of four-layered networks with 10 and 8 input units, respectively, 2 hidden layers with 20 units, and four output units. The sampling interval of dynamics computations in the learning is set at $\Delta t_s = 0.001$ s.

5.1. Task 1: circular motion

The first target task for the manipulator is a circular motion of the end-effector, as shown in Fig. 6, in which the end-effector rotates counterclockwise in 8 s.

5.1.1. Learning during free movements

The learning of PCN and VCN during free movements is performed, in which the initial values of synaptic weights are generated randomly under $|w_{ij}^{(p)}|, |w_{ij}^{(v)}| < 0.01$. Learning rates are set at $\eta_p = 13,000$,

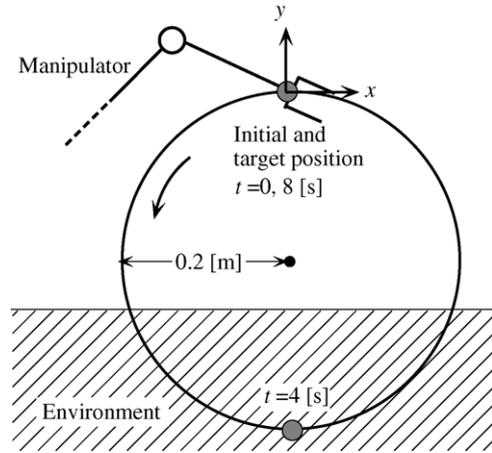


Fig. 6. An example of a contact task with the circular motion.

$\eta_v = 15$; the outputs of the NNs are within the limits of -100 to 100 .

Fig. 7 shows changes of the end-effector trajectory of the manipulator with progress of learning during free movements. The numbers represent rotation times. The generated trajectory by the end-effector does not agree with the desired circular trajectory before learning at all. The end-effector can roughly follow the desired trajectory in the first trial and almost agree with it in the second trial.

Fig. 8 shows the end-effector impedance parameters, $M_e^{-1}K_e^{-1}$ and $M_e^{-1}B_e^{-1}$, before and after learning of free movements: the output values of PCN and VCN. Diagonal elements after learning increase, whereas

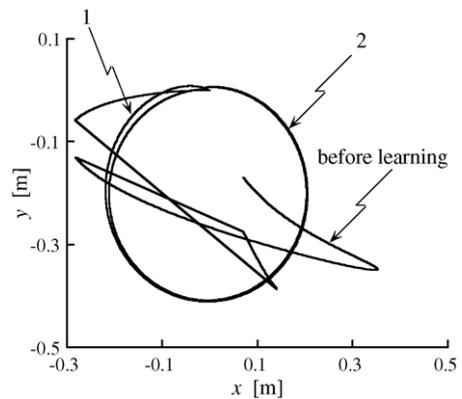


Fig. 7. End-point trajectories of the manipulator during the learning of free movements.

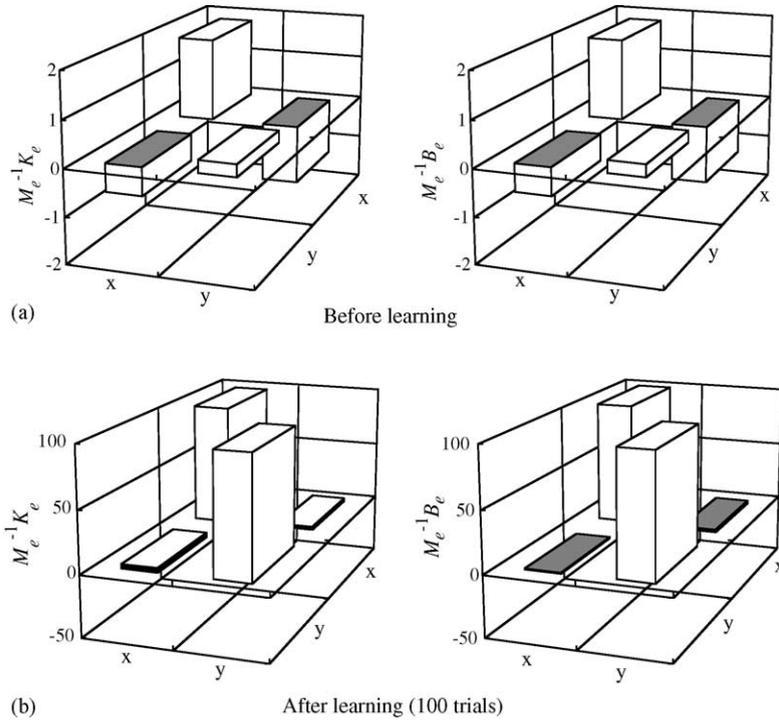


Fig. 8. Impedance parameters before and after the learning of free movements.

other elements converge upon very small values. Each output value is nearly constant and time-invariant during learning of free movements.

5.1.2. Learning during contact movements

Learning of FCN and EIN for contact movements is performed using trained PCN and VCN after learning free movements. The learning parameters are set as $\eta_f^{MAX} = 0.0001$, $\eta_d^{MAX} = 0.01$, $p = 10$, and $\eta_e = 0.001$. The desired end-point force is at $F_d = (0, 5)^T$ N. Note that no environmental positional information is given to the manipulator. The given environment model also contains modeling errors. The EIN starts identifying modeling errors just after the end-point contacts with the environment, whereas the FCN is trained and the desired trajectory is modified using the estimated end-point force \hat{F}_c given in (25).

Here, characteristics of the given environment model g_m in (22) agree with those of the real model g as $K_c = \text{diag}.[0, 10, 000]$ N/m, $B_c = \text{diag}.[0, 20]$ Ns/m, $M_c = \text{diag}.[0, 0.1]$ kg under $x < -0.1$ m, otherwise

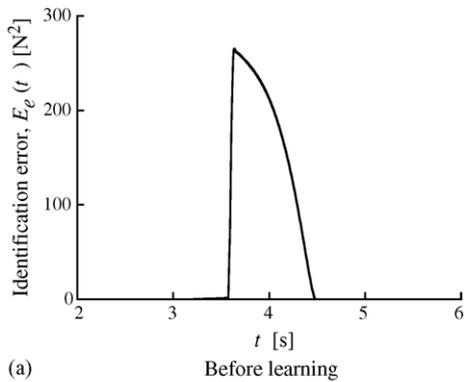
including some modeling errors expressed by the following nonlinear dynamics as

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

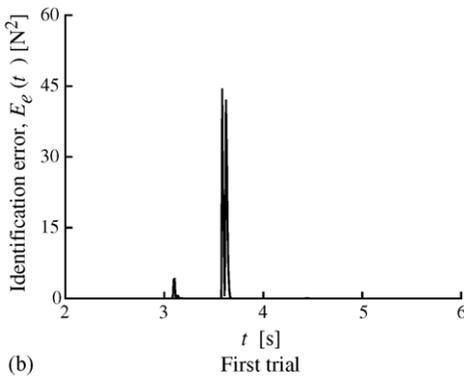
where F_{cy} is the normal interaction force from the environment to the end-effector, the tangential force $F_{cx} = 0$, and the impedance parameters are set as $K_{cy} = 1,000,000$ N/m², $B_{cy} = 2000$ N/m², $M_{cy} = 0.1$ kg, respectively.

Fig. 9 shows time changes of the force error $E_e(t)$ in (24) between the estimated force \hat{F}_c and the real force F_c during the learning of contact movements. The force error was considerably large before learning. It eventually decreased during learning. Some identification errors are evident at a moment when the end-effector contacts with environment and when environmental characteristics change discontinuously.

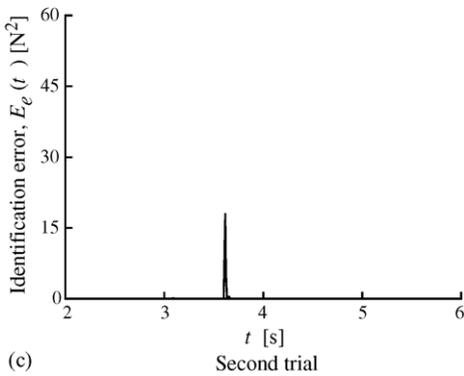
Fig. 10 shows the changes of arm postures and end-point forces F_c of the manipulator in the process of learning during contact movements. The large interaction force was generated until the learning of the



(a) Before learning



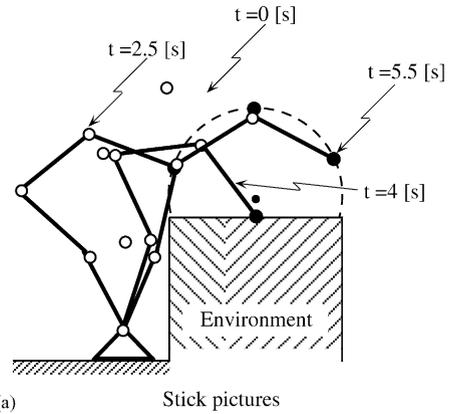
(b) First trial



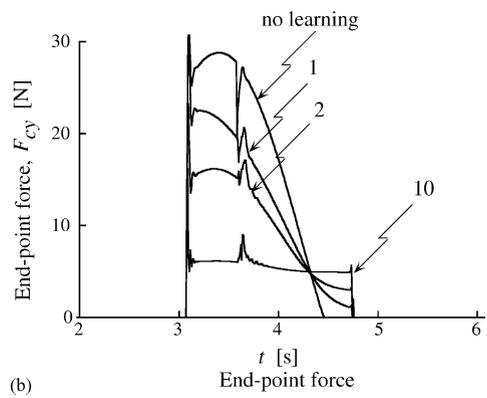
(c) Second trial

Fig. 9. Identification error of the environment model during the learning of contact movements.

FCN had progressed sufficiently because the manipulator tried to follow the initial desired trajectory with the trained PCN and VCN. The end-point force converges to the desired one (5 N) with progress of FCN learning. In particular, the desired force is realized just after contacts in the final trial. However, the force control



(a) Stick pictures



(b) End-point force

Fig. 10. End-point force of the manipulator during the learning of contact movements.

errors remain only in a moment when environmental characteristics change discontinuously. This fact indicates that the trained EIN did not completely identify the modeling errors.

Fig. 11 shows time profiles of the impedance parameter at the first and tenth trials, where (i, j) represents the matrix elements of M_e^{-1} . The elements converge to very small values, except for $(2, 2)$ which represents end-point mobility in the y direction. On the other hand, the time history of $(2, 2)$ after the learning that the inertia parameter for the y direction decreases only in contact movements. In addition, output values of the PCN and VCN, $M_e^{-1}K_e$ and $M_e^{-1}B_e$, remain almost constant during contact movements because both PCN and VCN are not trained during contact movements. Consequently, the stiffness K_e and viscosity B_e become small for the normal direction during contact movements. Therefore, the end-effector is

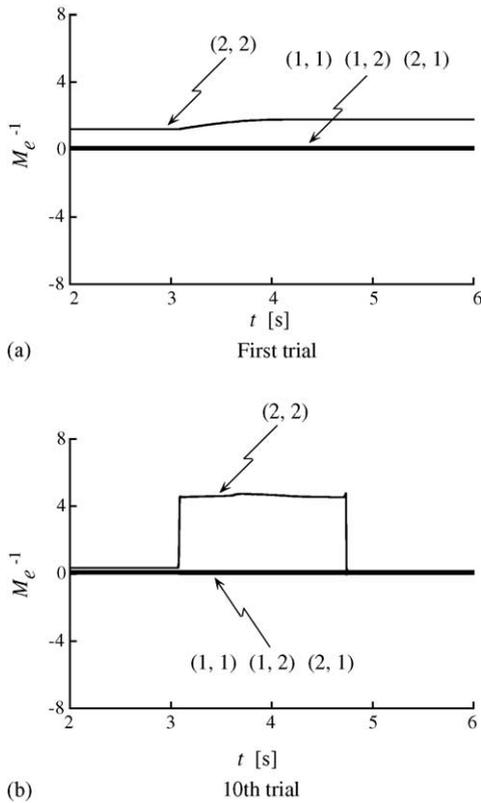


Fig. 11. Impedance parameters before and after the learning of contact movements.

compliant in the normal direction to the environment surface.

Fig. 12 shows the change of the desired trajectory. When the end-point force is larger than the desired end-point force, the desired trajectory is modified toward the environment surface. On the other hand, when the end-point force is smaller than the desired end-point force, the desired trajectory is modified to go away from the surface. Figs. 10 and 12 show that the desired trajectory is modified to realize the desired end-point force during contact movements.

5.2. Task 2: crank rotation

The proposed method is applied to a crank rotation task, as shown in Fig. 13, which is a more advanced constrained task than the previous contact tasks. In that figure, X_{oc} and $X_c \in \mathbb{R}^2$ denote the rotation center of the crank with radius r and the tip of the crank handle,

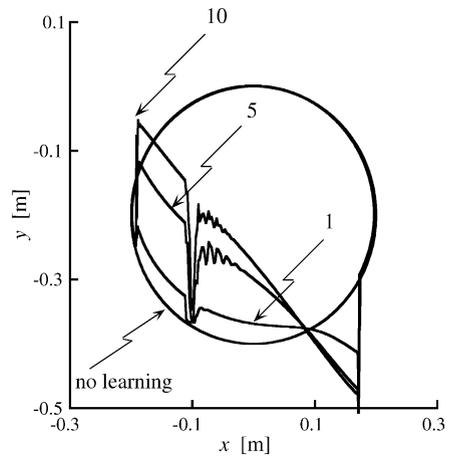


Fig. 12. Virtual trajectories of the manipulator during the learning of contact movements.

respectively. The origin of the task coordinate system is set at X_c . In addition, the end-effector of the manipulator is connected to the crank handle with viscoelastic properties.

Learning of the target task is carried out with a two-step algorithm to lighten the burden on NNs. First, X_{oc} and r are estimated to determine the task coordinate system that must generate the desired trajectory of the end-effector. Then, FCN and EIN are trained to regulate impedance parameters and to identify the environment modeling errors, respectively.

Computer simulations are executed using the same four-joint planar manipulator in the previous section, in where the manipulator has no information on crank parameters and the given crank model includes some modeling errors.

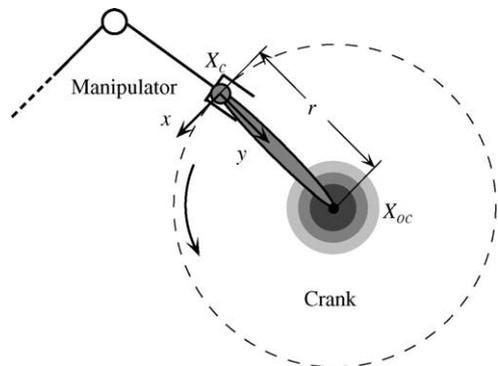


Fig. 13. An example of a crank rotation task.

5.2.1. Estimation of center position and radius of crank

The manipulator exerts a certain force F on the grasping crank handle at time $t = 0$, which starts the crank rotation. Estimation of X_{oc} and r is operated after the norm of the end-effector velocity $|\dot{X}|$ exceeds a certain value v_0 .

The following conditions are established at times t_a and t_b ($0 < t_a < t_b$) as

$$\dot{X}^T(t_a)N(t_a) = 0, \quad (36)$$

$$\dot{X}^T(t_b)N(t_b) = 0, \quad (37)$$

where $|\dot{X}(t_a)|, |\dot{X}(t_b)| > v_0$, and $|\dot{X}(t_b) - \dot{X}(t_a)| > v_1$; $N(t_a)$ and $N(t_b)$ represent the unit vectors perpendicular to $\dot{X}(t_a)$ and $\dot{X}(t_b)$, respectively. Therefore, a line passing on the end-point at each time can be determined with the corresponding norm vector, so that two lines can be obtained.

The intersection of these two lines can be given as

$$X(t_a) + pN(t_a) = X(t_b) + qN(t_b). \quad (38)$$

That equation yields the estimated center position of the crank \hat{X}_{oc} and the estimated radius \hat{r} as

$$\hat{r} = \frac{p + q}{2}, \quad (39)$$

$$\hat{X}_{oc} = X(t_a) + pN(t_a) = X(t_b) + qN(t_b). \quad (40)$$

On the other hand, the x axis of the task coordinate system is defined in parallel to the velocity vector of the end-effector, whereas the y axis points to the center of rotation. Moreover, the rotation angle of the task coordinate system with respect to the absolute coordinate system, $\alpha(t)$, can be obtained uniquely under $0 < \alpha(t) < 2\pi$ by the following equation:

$$\begin{pmatrix} \cos \alpha(t) & -\sin \alpha(t) \\ \sin \alpha(t) & \cos \alpha(t) \end{pmatrix} \begin{pmatrix} 0 \\ \hat{r} \end{pmatrix} = \begin{pmatrix} \hat{x}_{oc} - x(t) \\ \hat{y}_{oc} - y(t) \end{pmatrix}. \quad (41)$$

However, it is quite likely that \hat{X}_{oc} and \hat{r} estimated in the first trial may contain some errors. Therefore, the precision of \hat{X}_{oc} and \hat{r} is improved by the following iterative operations as

$$\hat{X}_{oc}(i) = \gamma \hat{X}_{oc}(i) + (1 - \gamma) \hat{X}_{oc}(i - 1), \quad (42)$$

$$\hat{r}(i) = \gamma \hat{r}(i) + (1 - \gamma) \hat{r}(i - 1), \quad (43)$$

where $\hat{X}_{oc}(i)$ and $\hat{r}(i)$ denote the computed values using (39) and (40) in the i -th trial, and $0 < \gamma < 1$.

5.2.2. Identification of crank model and learning of end-effector impedance

The end-point impedance, M_e , B_e , K_e , and the desired force F_d are expressed on the task coordinate system. Therefore, the desired impedance model of the end-effector in (3) is given by

$$R^T(\alpha(t))M_e R(\alpha(t))d\ddot{X} + R^T(\alpha(t))B_e d\dot{X}R(\alpha(t)) + R^T(\alpha(t))K_e R(\alpha(t))dX = R(\alpha(t))F_d - F_c, \quad (44)$$

where $R(\alpha(t)) \in \mathbb{R}^{2 \times 2}$ represents a rotation matrix on $\alpha(t)$. The FCN is trained to reduce the following energy function expressed on the task coordinate system as the following:

$$E_f(t) = \frac{1}{2} \{F_d(t) - R^T(\alpha(t))F_c(t)\}^T \times \{F_d(t) - R^T(\alpha(t))F_c(t)\}. \quad (45)$$

On the other hand, the environment model \hat{g} is constructed with the crank model and the EIN, as shown in Fig. 14. Therein, the EIN is put in a parallel situation with the given crank model expressed by

$$\hat{I}_c \hat{\phi} + \hat{B}_\phi \dot{\hat{\phi}} = \hat{r} F_c, \quad (46)$$

where $\hat{\phi}$ denotes the estimated rotation angle of the crank, \hat{I}_c the estimated moment of inertia, and \hat{B}_ϕ is the estimated coefficient of viscous friction on rotation. The position of the crank handle, X_{cm} , can be computed using estimated information \hat{r} and $\hat{\phi}$, but it may contain some errors. The EIN identifies the real crank model using the end-point force, position, velocity, and acceleration. It takes an active part in modifying the estimated position of the crank handle X_{cn} . Consequently, the estimated position of the crank handle can be obtained by

$$\hat{X}_c = X_{cm} + X_{cn} \quad (47)$$

and the estimated end-point force \hat{F}_c can be represented as

$$\hat{F}_c = K_c d\hat{X}_{oc} + B_c d\dot{\hat{X}}_{oc}, \quad (48)$$

where the matrices K_c and $B_c \in \mathbb{R}^{l \times l}$ represent viscoelastic properties for connecting the end-effector to the crank handle.

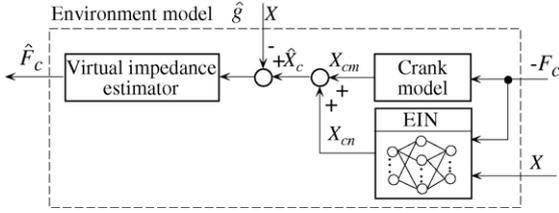


Fig. 14. Identification of the environment using EIN in the crank rotation task.

The energy function for the learning of EIN is defined by

$$E_e(t) = \frac{1}{2} \{ \hat{F}_c(t) - F_c(t) \}^T \{ \hat{F}_c(t) - F_c(t) \}. \quad (49)$$

Synaptic weights in the EIN, $w_{ij}^{(e)}$, are modified in the direction of the gradient descent as

$$\Delta w_{ij}^{(e)}(t) = -\eta_e \frac{\partial E_e(t)}{\partial w_{ij}^{(e)}(t)}, \quad (50)$$

$$\frac{\partial E_e(t)}{\partial w_{ij}^{(e)}(t)} = \frac{\partial E_e(t)}{\partial \hat{F}_c(t)} \frac{\partial \hat{F}_c(t)}{\partial X_{cn}(t)} \frac{\partial X_{cn}(t)}{\partial w_{ij}^{(e)}(t)}, \quad (51)$$

where η_e is the learning rate for the EIN. The terms $\frac{\partial E_e(t)}{\partial \hat{F}_c(t)}$ and $\frac{\partial \hat{F}_c(t)}{\partial X_{cn}(t)}$ can be computed by (48) and (49), whereas $\frac{\partial X_{cn}(t)}{\partial w_{ij}^{(e)}(t)}$ by back-propagation learning.

Furthermore, K_c and B_c should be modified to lighten the burden imposed on the FCN in learning during contact movements. Therefore, the modification variable of the stiffness ΔK_c is defined as

$$\Delta K_c(t) = -\eta_{kc} \frac{\partial E_e(t)}{\partial K_c(t)}, \quad (52)$$

$$\frac{\partial E_e(t)}{\partial K_c(t)} = \frac{\partial E_e(t)}{\partial \hat{F}_c(t)} \frac{\partial \hat{F}_c(t)}{\partial K_c(t)}, \quad (53)$$

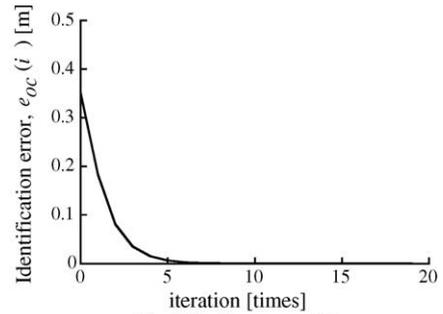
where η_{kc} is the modification rate. In the same manner, B_c is modified to reduce the energy function $E_e(t)$.

The relationship $\hat{F}_c = F_c$ is established when the energy function $E_e(t)$ is minimized by EIN learning such that \hat{F}_c can be used to the learning rule of contact movements.

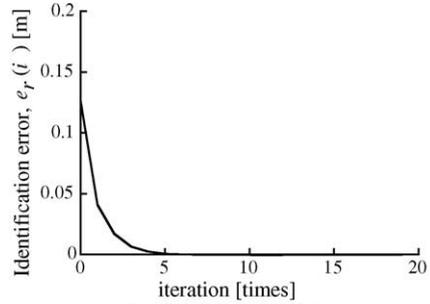
5.2.3. Learning during contact movements for crank rotation

Learning during contact movements is performed for the crank rotation using the trained PCN and VCN during the free movement in Section 5.1. Each structure of FCN and EIN is identical to those employed respectively in Section 5: four layered networks with 10 and 8 input units, respectively, 2 hidden layers with 20 units, and four output units. The desired trajectory of the end-effector is a counterclockwise circular rotation in 8 s, which is generated under estimated crank parameters using the fifth-order polynomial with respect to time [14].

Parameters for learning NNs were set as $\eta_f^{\text{MAX}} = 0.005$, $\eta_d^{\text{MAX}} = 0.01$, $p = 10$, $\eta_e = 0.0001$, $\eta_{kc} = 100$, and $\eta_{bc} = 50$, respectively. Characteristics of the real crank model in (46) were set as $I_c = 0.133 \text{ Nm}$, $B_\phi = 0.5 \text{ Nms/rad}$, $\dot{I}_c = 0.066 \text{ Nm}$, $\hat{B}_\phi = 0 \text{ Nms/rad}$, $X_{oc} = (0.2, 0)^T \text{ m}$, and $r = 0.2 \text{ m}$, respectively. Initial viscoelastic properties for connecting the end-effector to the crank handle were set as $K_c = \text{diag}[0, 10, 000] \text{ N/m}$, $B_c = \text{diag}[0, 20] \text{ Ns/m}$,



(a)



(b)

Fig. 15. Estimation errors of rotation center and radius of the crank.

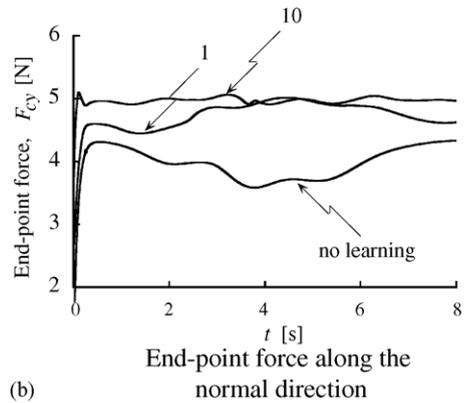
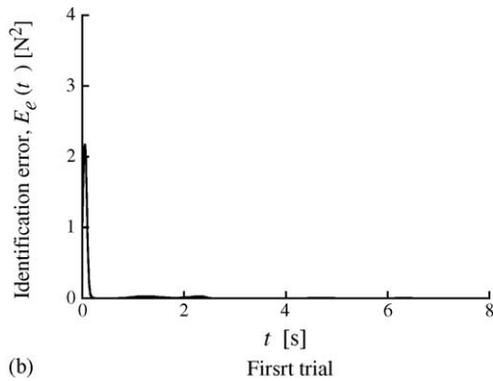
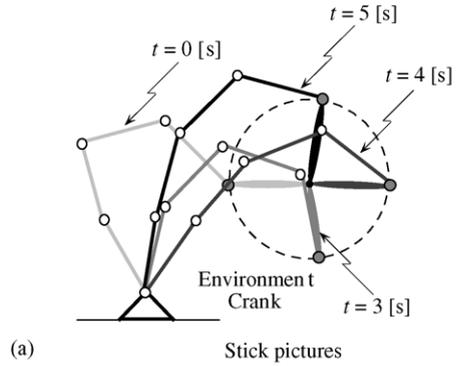
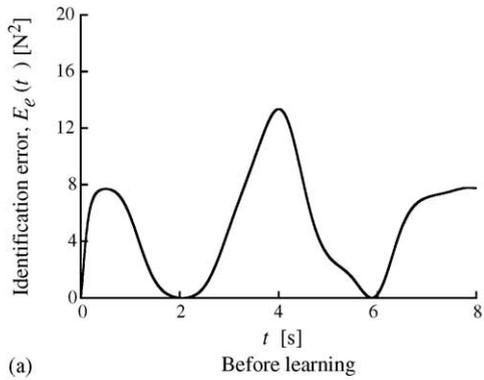


Fig. 16. Identification error of the environment during the learning of crank rotations.

Fig. 17. End-point force of the manipulator during the learning of crank rotations.

respectively. The desired end-point force was set as $F_d = (0, 5)^T$ N.

First, force $F = (5, -5)^T$ N is exerted on the crank handle by the manipulator end-effector at time $t = 0$ to estimate the center position and the crank radius. Fig. 15 shows changes of the estimation errors of rotation center $e_{oc}(i) = \|\hat{X}_{oc}(i) - X_{oc}\|$ and the radius of crank $e_r(i) = |\hat{r}(i) - r|$. Errors converge to almost zero through several iterative trials.

Next, the FCN and the desired trajectory are regulated using the estimated end-point force \hat{F}_c , whereas the EIN identifies the environment modeling error. Fig. 16 shows time changes of $E_e(t)$ defined in (49) before and after the learning of contact movements. The identification error is very large before learning, whereas the error decreases according to on-line learning progress.

Fig. 17 shows changes of arm postures and end-point force F_{cy} of the manipulator during the learning

of crank rotation movements, where F_{cy} is expressed on the task coordinate system. The numbers in the figure represent rotation times of the crank. It can be seen that the end-point force gradually converges to the desired one (5 N) with the progress of the learning of FCN.

Fig. 18 shows time changes of the impedance parameter M_e^{-1} , and output values of the FCN in the first trial and the tenth trial, where (i, j) in the figure represents elements of the matrix M_e^{-1} . The elements converge to very small values, except for (2, 2), which represents end-point mobility in the y direction on the task coordinate system. On the other hand, it can be found from the time history of (2, 2) after learning that the inertia parameter for the y direction decreases only in contact movements. The gain matrices $M_e^{-1}K_e$, $M_e^{-1}B_e$ remain almost constant because both PCN and VCN are not trained during contact movements. Consequently, stiffness of the end-effector K_e and viscosity B_e become small for the normal direction

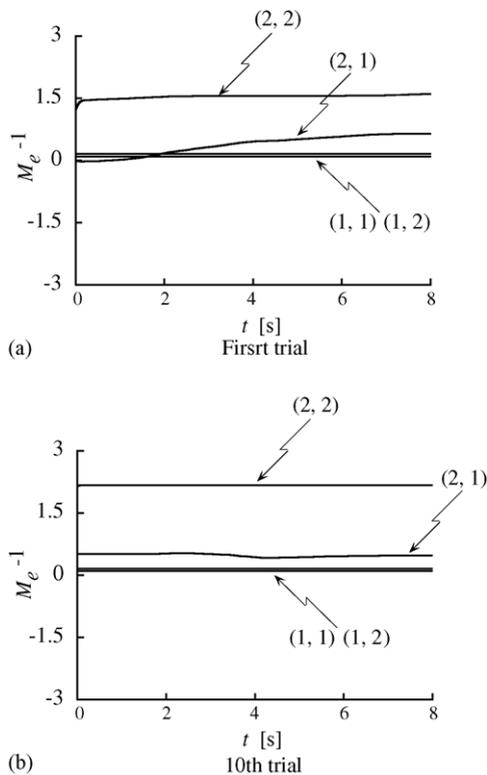


Fig. 18. Impedance parameters before and after the learning of crank rotations.

during contact movements. In addition, from the result of (2, 1) after learning, the NNs are actively trained using the tangential force to realize the desired end-point force in the direction of the rotation center of the crank.

6. Conclusions

This paper has presented an on-line learning method using NNs to regulate impedance parameters of manipulators' end-points. The proposed method achieves on-line learning by introducing the NN for identifying nonlinear characteristics of the environment in addition to the NNs for regulating impedance properties. The proposed method can regulate all impedance parameters: inertia, viscosity, and stiffness. It does so through learning and thereby reducing position and force control errors. In addition, the desired trajectory is modified actively to ensure steady convergence on the specified desired end-point force.

The proposed on-line learning method in this paper differs from the off-line learning methods in the literature in terms of whether not the iterative trial is needed. The off-line learning methods must execute the iterative trial because the gradient decent of error is calculated using results from the previous trial. In contrast, the proposed method can obtain that data on-line without using past results. Therefore, an iterative trial is not always needed for learning of NNs to progress efficiently. These considerations of the proposed method are evident from experimental results for contact tasks. For example, the manipulator can almost realize the desired circular trajectory from the first trial in free movement, as shown in Fig. 7. In contrast, about ten iterative trials were needed in the constrained movement, as shown in Figs. 9–12, because the initial condition changed considerably at each trial. For that reason, several iterative trials were required to complete the learning. The proposed on-line learning method is applicable even if the initial condition was changed at every trial. This fact represents a great advantage of the proposed method: it is an extremely stringent requirement that preserves the initial condition in the actual environment. To the contract, the off-line learning method can be conducted only under the fixed initial condition.

In addition, this study showed that learning for regulating the impedance parameters can be conducted effectively even with a conventional back propagation type NN by devising the control system and the learning laws. However, a suitable NN structure for learning robot impedance should be investigated to realize more effective learning with few iterations and to apply in experiments with real robots. In light of such considerations, future research will be directed to development of an effective method for determining NN learning rates and improving the proposed control method to allow for more complicated tasks.

References

- [1] N. Hogan, Impedance control: an approach to manipulation, Parts I–III, *Trans. ASME J. Dynam. Syst. Measure. Contr.* 107 (1) (1985) 1–24.
- [2] T. Tsuji, M. Nishida, K. Ito, Iterative learning of impedance parameters for manipulator control using neural networks, *Trans. Soc. Instrum. Contr. Eng.* 28 (12) (1992) 1461–1468 (in Japanese).

- [3] T. Tsuji, K. Ito, P.G. Morasso, Neural network learning of robot arm impedance in operational space, *IEEE Trans. Syst. Man Cybernet. Part B: Cybernet.* 26 (2) (1996) 290–298.
- [4] H. Asada, Teaching and learning of compliance using neural-nets: representation and generation of nonlinear compliance, in: *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 1237–1244.
- [5] M. Cohen, T. Flash, Learning impedance parameters for robot control using an associative search networks, *IEEE Trans. Robot. Automat.* 7 (3) (1991) 382–390.
- [6] B.-H. Yang, H. Asada, Progressive learning and its application to robot impedance learning, *IEEE Trans. Neural Netw.* 7 (4) (1996) 941–952.
- [7] N.F. Xiao, I. Todo, Neural network-based learning impedance control for a robot, *JSME Int. J. Ser. C: Mech. Syst. Machine Elem. Manuf.* 44 (3) (2001) 626–633.
- [8] S.T. Venkataraman, S. Gulati, J. Barhen, N. Toomarian, A neural network based identification of environments models for compliant control of space robots, *IEEE Trans. Robot. Automat.* 9 (5) (1993) 685–697.
- [9] Z.W. Luo, M. Ito, Control design of robot for compliant manipulation on dynamic environments, *IEEE Trans. Robot. Automat.* 9 (3) (1993) 286–296.
- [10] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Representations by Error Propagation, *Parallel Distributed Processing*, vol. 1, MIT Press, 1986, pp. 318–362.
- [11] T. Tsuji, B.H. Xu, M. Kaneko, Adaptive control and identification using one neural network for a class of plant with uncertainties, *IEEE Trans. Syst. Man Cybernet. Part A: Syst. Hum.* 28 (4) (1998) 496–505.
- [12] T. Tsuji, A. Jazidie, Impedance control for redundant manipulators: an approach to joint impedance regulation utilizing kinematic redundancy, *J. Robot. Soc. Jpn.* 12 (4) (1994) 609–615 (in Japanese).
- [13] T. Tsuji, A. Jazidie, M. Kaneko, Multi-point impedance control for redundant manipulators, *IEEE Trans. Syst. Man Cybernet. Part B: Cybernet.* 26 (5) (1996) 707–718.
- [14] T. Yoshikawa, Analysis and control of robot manipulators with redundancy, in: M. Brady, R. Paul (Eds.), *Proceedings of the First International Symposium on Robotic Research*, MIT Press, Cambridge, Mass, 1984, pp. 735–747.



Toshio Tsuji received his B.E. degree in industrial engineering in 1982, his M.E. and Doctor of Engineering degrees in systems engineering in 1985 and 1989, all from Hiroshima University. He was a Research Associate from 1985 to 1994, and an Associate Professor, from 1994 to 2002, in Faculty of Engineering at Hiroshima University. He was a Visiting Professor of University of Genova, Italy for 1 year from 1992 to 1993. He is currently a Professor of Department of Artificial Complex Systems Engineering at Hiroshima University. He won the Best Paper Award from The Society of Instrumentation and Control Engineers in 2002, and the K.S. Fu Memorial Best Transactions Paper Award of the IEEE Robotics and Automation Society in 2003. His current research interests have focused on Human–Machine Interface, and computational neural sciences, in particular, biological motor control. He is a member of the Institute of Electrical and Electronics Engineers, the Japan Society of Mechanical Engineers, the Robotics Society of Japan, and the Society of Instrumentation and Control Engineers in Japan.



Yoshiyuki Tanaka received his B.E. degree in Computer Science and Systems Engineering from Yamaguchi University in 1995, and his M.E. and Doctor of Engineering degrees in Information Engineering from Hiroshima University in 1997 and 2001, respectively. From 2001 to 2002, he was a Research Associate with the Faculty of Information Sciences, Hiroshima City University. He is currently a Research Associate in the Department of Artificial Complex Systems Engineering at Hiroshima University. His research interests include biological motor control, Human–Machine interaction. He is a member of the Robotics Society of Japan, Institute of Electrical Engineering of Japan, and the Society of Instrumentation and Control Engineers in Japan.