Correspondence

Online Learning of Virtual Impedance Parameters in Non-Contact Impedance Control Using Neural Networks

Toshio Tsuji, Mutsuhiro Terauchi, and Yoshiyuki Tanaka

***Abstract*—Impedance control is one of the most effective methods for controlling the interaction between a manipulator and a task environment. In conventional impedance control methods, however, the manipulator cannot be controlled until the end-effector contacts task environments. A noncontact impedance control method has been proposed to resolve such a problem. This method on only can regulate the end-point impedance, but also the virtual impedance that works between the manipulator and the environment by using visual information. This paper proposes a learning method using neural networks to regulate the virtual impedance parameters according to a given task. The validity of the proposed method was verified through computer simulations and experiments with a multijoint robotic manipulator.**

***Index Terms*—Impact control, impedance control, noncontact impedance, neural networks (NN), robot manipulator.**

I. Introduction

A human can perform a variety of contact tasks in daily activities while regulating his/her own dynamic properties according to time-varying environmental conditions. In the ball-catching task, for example, a player should take action for catching the ball before contacting with the approaching ball. If action is not taken, it would be too late to prepare for the task and the player would fail to catch the ball. Besides, the human player has to regulate dynamic properties of his arm according to ball velocity as well as its physical properties, i.e., the player should make his arm compliant before catching a ball and move his hand to a suitable catching point as smooth as possible. He should also stiffen his arm to cope with the impact force in the moment of catching.

To realize such human skillful strategies for dynamic tasks by a robot, it is naturally required to change dynamic characteristics of the robot according to time-varying circumstances during a target task. The main purpose of this paper is to bring such human skillful strategies for dynamic tasks into robot motion control in the framework of the impedance control.

Impedance control [1], [2] is one of the most important methods of controlling the interaction between a manipulator and an environment. This method can regulate response properties of the manipulator to external disturbances by modifying the mechanical impedance parameters; i.e., inertia, viscosity, and stiffness. However, since no external force is exerted until the end-effector contacts task environments, the conventional method is not applicable in situations where the manipulator must reduce the end-effector velocity before it contacts the object.

To cope with such problems of the conventional method, some studies have employed visual information of the task space in impedance control of robot manipulators [3]–[5]. Castano and Hutchinson [3] proposed a concept of visual compliance using vision-based control. However, their proposal does not discuss regulating the value of visual compliance of the end-effector. Tsuji *et al.* [4] and Nakabo *et al.* [5] proposed a virtual impedance concept using visual information. In particular, Tsuji *et al.* have developed a non-contact impedance control method [4], where the virtual impedance is defined between the end-effector and the object when the object enters a virtual sphere established around the tip of the manipulator. Therefore, a virtual force for controlling the end-effector motion can be generated before the end-effector contacts the object by using local visual information inside the virtual sphere. This method can also regulate dynamic properties of the end-effector by the virtual impedance parameters according to tasks in the same way as the conventional impedance method. In these methods based on visual information, however, it is extremely difficult to regulate the virtual impedance parameters according to the time-varying characteristics of moving objects and task environments. For such an awkward problem, the learning technique using a neural network (NN) is applied in this study to design the desired impedance parameters.

There have been numerous studies on force and hybrid control and impedance control using NNs [6]–[17]. However, most of these methods aim to realize the desired impedance parameters, given in advance, through learning of NNs in consideration of the model uncertainties of manipulator dynamics and task environments, and the external disturbances. In contrast, some methods using NNs try to find the desired impedance according to tasks and environmental conditions [18]–[23]. For example, Yang and Asada [20] proposed a progressive learning method using NNs that can obtain the target impedance parameters by modifying the desired velocity trajectory. However, it does not clearly discuss how to regulate the operational velocity during learning of the NNs. Also, many iterative trials (200–300 times) are required to complete the learning. Tsuji *et al.* [21]–[23] thus proposed iterative learning methods using NNs that can regulate all impedance parameters as well as a desired end-point trajectory at the same time. These methods by Tsuji *et al.* can provide a smooth transition of the end-effector from free to contact movements, but cannot be applied to noncontact tasks in which the manipulator may not touch its environment. In addition, its application is limited to only cyclical tasks because the learning is performed in offline.

In this paper, we propose an online learning method using NNs for regulating the virtual impedance parameters in the noncontact impedance control by expanding the previous methods [21]–[23]. The present method can design the desired virtual impedance through learning of NNs with an energy function depending on a given task, and can also adapt the relative velocity during free movements and the interaction force during contact movements. This paper is organized as follows. First, the noncontact impedance control method is explained in Section II. The proposed online learning method is described in detail in Section III. In Section IV, validity of the proposed method is investigated through a series of computer simulation and experiments with a real robot in a ball-catching task by the proposed online learning method.

II. Non-Contact Impedance Control

A. Impedance Control

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

$$M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) = \tau + J^T(\theta)F_{\rm int} \tag{1}$$

Manuscript received March 21, 2003; revised November 28, 2003. This paper was rescommended by Associate Editor D. Y. Lee.

Toshio Tsuji and Yoshiyuki Tanaka are with the Department of Artificial Complex Systems Engineering, Hiroshima University, Hiroshima 739-8527, Japan (e-mail: tsuji@bsys.hiroshima-u.ac.jp).

Mutsuhiro Terauchi is with the Department of Kansei Information, Hiroshima International University, Hiroshima 724-0695, Japan.

Digital Object Identifier 10.1109/TSMCB.2004.829133

where $\theta \in \Re^m$ is the joint angle vector; $M(\theta) \in \Re^{m\times m}$ is the nonsingular inertia matrix (hereafter, denoted by M_θ); $h(\theta,\dot\theta) \in \Re^m$ is the nonlinear term including the joint torque due to the centrifugal, Coriolis, gravity, and friction forces; $\tau \in \Re^m$ is the joint torque vector; $F_{\rm int} \in \Re^l$ is the external force exerted on the end-effector; and $J(\theta) \in \Re^{l\times m}$ is the Jacobian matrix (hereafter, denoted by J).

The desired impedance properties of the end-effector can be expressed as

$$M_e d\ddot{X} + B_e d\dot{X} + K_e dX = F_{\rm int} \tag{2}$$

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; and $dX = X_e - X_d \in \Re^l$ is the displacement vector between the current position of the end-effector X_e and the desired one X_d. The impedance control law does not use an inverse of the Jacobian matrix and is given [2] as follows:

$$\tau = \tau_{\rm effector} + \tau_{\rm comp} \tag{3}$$

$$\tau_{\rm effector} = J^T\left\{M_x(\theta)\left[M_e^{-1}(-K_e dX - B_e d\dot{X}) + \ddot{X}_d - \dot{J}\dot\theta\right] - \left[I - M_x(\theta)M_e^{-1}\right]F_{\rm int}\right\} \tag{4}$$

$$\tau_{\rm comp} = (M^{-1}J^T M_x(\theta)J)^T\hat{h}(\theta,\dot\theta), \tag{5}$$

where $M_x(\theta) = (J\hat{M}^{-1}J^T)^{-1} \in \Re^{l\times l}$ indicates the operational space kinetic energy matrix that is proper as long as the joint configuration θ is not singular; $\tau_{\rm effector} \in \Re^m$ in (4) is the joint torque vector necessary to realize the desired end-effector impedance; $\tau_{\rm comp} \in \Re^m$ in (5) is the joint torque vector for nonlinear compensation; $\hat{h}(\theta,\dot\theta)$ and $\hat{M}$ denote the estimated values of $h(\theta,\dot\theta)$ and M_θ, respectively, and I is the $l\times l$ unit matrix.

Impedance properties of the end-effector can be regulated by the designed controller in (3).

B. Non-Contact Impedance Control

Fig. 1 schematically represents the noncontact impedance control. Let us consider the case in which an object approaches a manipulator, and set a virtual sphere with radius r at the center of the end-effector. When the object enters the virtual sphere, the normal vector from the surface of the sphere to the object $dX_o \in \Re^l$ can be represented as

$$dX_o = X_r - rn \tag{6}$$

where $X_r = X_o - X_e$ is the displacement vector from the center of the sphere (namely, the end-point position) $X_e = [x_e^1, x_e^2, \ldots, x_e^l]^{\rm T} \in \Re^l$ to the object $X_o = [x_o^1, x_o^2, \ldots, x_o^l]^{\rm T} \in \Re^l$; and the vector $n \in \Re^l$ is given by

$$n = \begin{cases} \dfrac{X_r}{|X_r|} & (|X_r| \neq 0) \\ 0 & (|X_r| = 0) \end{cases}. \tag{7}$$

When the object is in the virtual sphere ($|X_r| < r$), the virtual impedance works between the end-effector and the object so that the virtual external force $F_o \in \Re^l$ is exerted on the end-effector by

$$F_o = \begin{cases} M_o d\ddot{X}_o + B_o d\dot{X}_o + K_o dX_o & (|X_r| < r) \\ 0 & (|X_r| \geq r) \end{cases} \tag{8}$$

where M_o, B_o, and $K_o \in \Re^{l\times l}$ represent the virtual inertia, viscosity and stiffness matrices. It should be noted that F_o becomes zero when the object is outside the virtual sphere or at the center of the sphere. Thus, the dynamic equation of the end-effector for noncontact impedance control can be expressed with (2) as

$$M_e d\ddot{X} + B_e d\dot{X} + K_e dX = F_{\rm int} + F_o. \tag{9}$$

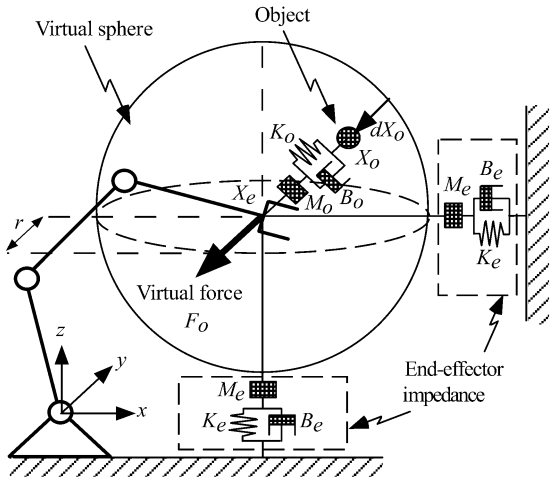

Fig. 1. Schematic representation of a noncontact impedance control [4], [5].

Substituting (8) into (9), the motion equation of the end-effector for the external forces depending on the object position X_o and the desired end-effector position X_d yields

$$M\ddot{X}_e + B\dot{X}_e + KX_e = \bar{F} \tag{10}$$

$$\bar{F} = F_{\rm int} + M_e\ddot{X}_d + B_e\dot{X}_d + K_eX_d + M_o\ddot{\bar{X}}_o + B_o\dot{\bar{X}}_o + K_o\bar{X}_o \tag{11}$$

where $M = M_o + M_e, B = B_o + B_e, K = K_o + K_e$, and $\bar{X}_o = X_o - rn$. For simplicity, we argue the stability of the proposed linear time-invariant system given by (10) under $\bar{F} = 0$. The candidate of a Lyapunov function for the proposed control system can be designed as

$$V_e = \frac{1}{2}\left\{X_e^T(M^{-1}K)X_e + \dot{X}_e^T\dot{X}_e\right\} \tag{12}$$

assuming that the matrix $M^{-1}K$ is a positive definite one. Time-derivation of the Lyapunov function is then derived by using (10) as follows:

$$\dot{V}_e = -\ddot{X}_e^T(M^{-1}B)\ddot{X}_e < 0. \tag{13}$$

Therefore, the proposed control system is stable as far as $M^{-1}K$ and $M^{-1}B$ are positive definite, and this condition is equivalent to that M, B, K are positive definite matrices. It should be noted that the virtual impedance parameters can take negative values within the derived stable condition.

Fig. 2 depicts a block diagram of the noncontact impedance control. In the noncontact impedance control, the relative motion between the end-effector and the object can be regulated by the virtual impedance parameters during noncontact movements. In addition, the end-effector impedance can be modified according to the target task in the same way as in the conventional method. The main purpose of this paper is to develop an online learning method using NNs for regulating the virtual impedance parameters M_o, B_o, K_o according to a target task including contacts with environments under the stable conditions.

III. Learning of Virtual Impedance by NNs

A. Structure of Control System

In the proposed control system, the virtual impedance part in Fig. 2 is composed of three multilayered NNs as shown in Fig. 3: *a virtual stiffness network* (VSN) at K_o, *a virtual viscosity network* (VVN) at B_o, and *a virtual inertia network* (VIN) at M_o. The detailed structure

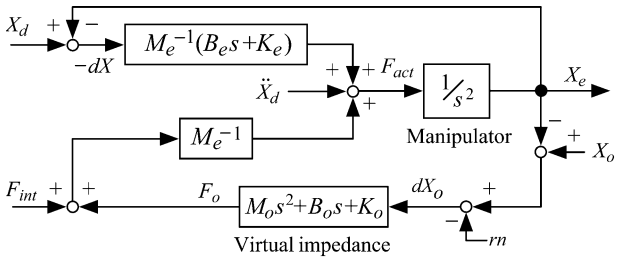

Fig. 2. Block diagram of the noncontact impedance control.

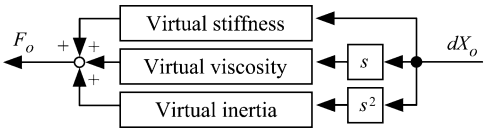

Fig. 3. Virtual impedance composed of three components.

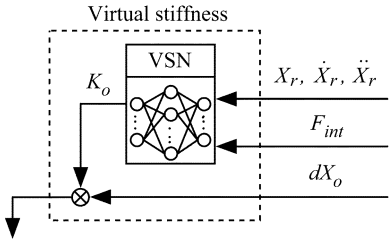

Fig. 4. Virtual stiffness realized by a neural network.

of the VSN is shown in Fig. 4, and VVN and VIN have the same structure; an input layer, a set of hidden layers, and an output layer. The number of hidden layers and units are determined according to target tasks. The NNs input the relative motion between end-effector and object ($X_r, \dot{X}_r$, and $\ddot{X}_r$) and the interaction force $F_{\rm int}$, while each NN outputs the corresponding impedance parameter: K_o from the VSN, B_o from the VVN, and M_o from the VIN.

The NNs utilize a linear function in the input units and a sigmoid function in the hidden and output units. Therefore, the input and output of each unit in the i-th layer, x_i and y_i, can be derived as

$$x_i = \begin{cases} I_i & \text{(input layer)} \\ \sum w_{ij} y_i & \text{(middle and output layers)}, \end{cases} \tag{14}$$

$$y_i = \begin{cases} x_i & \text{(input layer)} \\ \frac{1}{1+e^{-x_i}} & \text{(middle layer)}, \\ \frac{U}{2}\left(\frac{1-e^{-x_i+\psi}}{1+e^{-x_i+\psi}}\right) & \text{(output layer)} \end{cases} \tag{15}$$

where w_{ij} indicates the weight coefficient from the unit j to i; and U and ψ are positive constants for the maximum output and the threshold of NN, respectively. The outputs of NNs are represented by the following vectors:

$$O_S = \left(o_{S1}^T, o_{S2}^T, \ldots, o_{Sl}^T\right)^T \in \Re^{l^2} \tag{16}$$

$$O_V = \left(o_{V1}^T, o_{V2}^T, \ldots, o_{Vl}^T\right)^T \in \Re^{l^2} \tag{17}$$

$$O_I = \left(o_{I1}^T, o_{I2}^T, \ldots, o_{Il}^T\right)^T \in \Re^{l^2} \tag{18}$$

where o_{Si}, o_{Vi}, and $o_{Ii} \in \Re^l$ are the vectors which consist of the output values of the VSN, VVN, and VIN corresponding to the i-th row of the matrices K_o, B_o, and M_o, respectively.

B. Learning of NN

The learning of NNs is performed by modifying synaptic weights in the NNs so as to minimize an energy function $E(t)$ depending on tasks under the stable conditions derived in Section III-B. The synaptic weights in the VSN, $w_{ij}^{(S)}$, the VVN, $w_{ij}^{(V)}$, and the VIN, $w_{ij}^{(I)}$, are modified in the direction of the gradient descent to reduce the energy function $E(t)$ as

$$\Delta w_{ij}^{(S)} = -\eta_S \frac{\partial E(t)}{\partial w_{ij}^{(S)}} \tag{19}$$

$$\Delta w_{ij}^{(V)} = -\eta_V \frac{\partial E(t)}{\partial w_{ij}^{(V)}} \tag{20}$$

$$\Delta w_{ij}^{(I)} = -\eta_I \frac{\partial E(t)}{\partial w_{ij}^{(I)}} \tag{21}$$

$$\frac{\partial E(t)}{\partial w_{ij}^{(S)}} = \frac{\partial E(t)}{\partial F_{\rm act}(t)} \frac{\partial F_{\rm act}(t)}{\partial O_S(t)} \frac{\partial O_S(t)}{\partial w_{ij}^{(S)}} \tag{22}$$

$$\frac{\partial E(t)}{\partial w_{ij}^{(V)}} = \frac{\partial E(t)}{\partial F_{\rm act}(t)} \frac{\partial F_{\rm act}(t)}{\partial O_V(t)} \frac{\partial O_V(t)}{\partial w_{ij}^{(V)}} \tag{23}$$

$$\frac{\partial E(t)}{\partial w_{ij}^{(I)}} = \frac{\partial E(t)}{\partial F_{\rm act}(t)} \frac{\partial F_{\rm act}(t)}{\partial O_I(t)} \frac{\partial O_I(t)}{\partial w_{ij}^{(I)}} \tag{24}$$

where η_C $(C = S, V, I)$ is the learning rate of each NN, $F_{\rm act}(t)$ is the control input, and $O_C(t) \in R^{l\times l}$ is the NN output; i.e., K_o, B_o, M_o.

The term $(\partial F_{\rm act}(t))/(\partial O_C(t))$ can be computed from Fig. 2 and (8), and $(\partial O_C(t))/(\partial w_{ij}^{(C)})$ can be obtained by the error back-propagation learning method. However, the term $(\partial E(t))/(\partial F_{\rm act}(t))$ cannot be computed directly because of the nonlinear dynamics of the manipulator. In the online method, the term $(\partial E(t))/(\partial F_{\rm act}(t))$ is approximated in the discrete-time system so that $\Delta w_{ij}^{(C)}$ can be calculated in real time by using the change of $E(t)$ for a slight variation of $F_{\rm act}(t)$.

Defining the energy function $E(t)$ depending on end-point position and velocity, $X_e(t)$ and $\dot{X}_e(t)$, the term $(\partial E(t))/(\partial F_{\rm act}(t))$ can be expanded from Fig. 2 as

$$\frac{\partial E(t)}{\partial F_{\rm act}(t)} = \frac{\partial E(t)}{\partial X_e(t)} \frac{\partial X_e(t)}{\partial F_{\rm act}(t)} + \frac{\partial E(t)}{\partial \dot{X}_e(t)} \frac{\partial \dot{X}_e(t)}{\partial F_{\rm act}(t)}. \tag{25}$$

Applying the nonlinear compensation technique with

$$\tau = \{\hat{M}^{-1} J^T M_x J\}^T \hat{h}(\theta, \dot{\theta}) - J^T F_{\rm int} + J^T M_x \{F_{\rm act} - \dot{J}\dot{\theta}\} \tag{26}$$

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

$$\ddot{X} = F_{\rm act}. \tag{27}$$

The slight change of control input $\Delta F_{\rm act}(t)$ within short time yields the following approximations:

$$\Delta X_e(t) \approx \Delta F_{\rm act}(t) \Delta t_s^2 \tag{28}$$

$$\Delta \dot{X}_e(t) \approx \Delta F_{\rm act}(t) \Delta t_s \tag{29}$$

so that $(\partial X(t))/(\partial F_{\rm act}(t))$ and $(\partial \dot{X}(t))/(\partial F_{\rm act}(t))$ can be expressed [23] as follows:

$$\frac{\partial X_e(t)}{\partial F_{\rm act}(t)} \approx \frac{\Delta X_e(t)}{\Delta F_{\rm act}(t)} = \Delta t_s^2 I \tag{30}$$

$$\frac{\partial \dot{X}_e(t)}{\partial F_{\rm act}(t)} \approx \frac{\Delta \dot{X}_e(t)}{\Delta F_{\rm act}(t)} = \Delta t_s I \tag{31}$$

where Δt_s is a sampling interval. Consequently the term $(\partial E(t))/(\partial F_{\rm act}(t))$ can be approximately computed by

$$\frac{\partial E(t)}{\partial F_{\rm act}(t)} = \frac{\partial E(t)}{\partial X_e(t)}\Delta t_s^2 + \frac{\partial E(t)}{\partial \dot{X}_e(t)}\Delta t_s. \tag{32}$$

With the designed learning rules in (19)–(32), an online learning can be performed so that the output of NNs, $O_C(t)$, will be regulated to the optimal virtual impedance parameters for tasks.

IV. Application to Contact Tasks

Many researchers have discussed and analyzed theoretical conditions for the control parameters and stabilities in contact tasks by the manipulator [24], [25]. In contrast, this paper investigates the effectiveness of the proposed online learning method using NNs for regulating the virtual impedance according to contact tasks of the manipulator.

An example of contact tasks is a catching task in which a manipulator contacts an approaching object by the end-effector and tries to make the relative movements as smooth as possible. Computer simulations and experiments with a real robot were conducted by means of *the proposed online learning method*.

A. Energy Function for Catching Task

In contact tasks by a robotic manipulator, the interaction force between the end-effector and its environment should converge to the desired value without overshooting to avoid exerting a large interaction force on the manipulator and the environment. To this end, the relative velocity between the end-effector and the environment should be reduced before contact, and the end-point force after contact with the environment should be controlled. Accordingly, an energy function for the learning of NNs can be defined as

$$E(t) = E_v(t) + \mu E_f(t) \tag{33}$$

$$E_v(t) = \frac{1}{2}(\alpha(X_r)\dot{X}_r(t_i) - \dot{X}_r(t))^2 \tag{34}$$

$$E_f(t) = \frac{1}{2}\int_0^t (F_d(u) - F_{\rm int}(u))^2\, du \tag{35}$$

where t_i is the time when the virtual sphere just contacts the environment; and $\alpha(X_r)$ is the time-varying gain function that should be designed according to contact tasks so as to avoid generating an excessive interaction force while performing the stable learning of NNs immediately after the environment enters the virtual sphere. The terms $E_v(t)$ and $E_f(t)$ in (33) evaluate the relative velocity and the force error.

B. Computer Simulations

Fig. 5 illustrates a catching task by a manipulator with one degree-of-freedom ($l = 1$). A ball is represented by a viscoelastic model as shown in Fig. 5(b) and hung from the ceiling by a pendulum with length $L = 2.1$ [m], where the base of the pendulum is located at $X_f = [0.15, 2.1]^{\rm T}$ [m] in the operational task coordinate system which origin is set at the initial hand position. The virtual impedance works between the end-effector and the ball when the ball is in the virtual sphere with radius $r = 0.2$ [m]. Both the initial and target positions of the end-effector are established at the origin of the task space, and the initial angle of the pendulum at $\theta_0 = -\pi/18$ [rad]. Also, the impedance parameters of the end-effector are set as $M_e = 25$ [kg], $B_e = 200$ [Ns/m], $K_e = 400$ [N/m], and those of the ball as $B_b = 40$ [Ns/m], $K_b = 500$ [N/m] with weight $M_b = 0.5$ [kg] and radius $R_b = 0.03$ [m].

The NNs utilized in the computer experiments were four-layered networks with four input units, two hidden layers with ten units, and one output unit; the initial values of the synaptic weights w_{ij} were randomly chosen under $|w_{ij}| < 0.05$; the learning rates of NNs were $\eta_S = 0.04$ for VSN, $\eta_V = 0.08$ for VVN, and $\eta_I = 1.0 \times 10^{-4}$ for VIN; the sigmoid functions in the output units were adjusted so that the output values of NNs were within -1000–1000, that is, $U = 2000$ in (12); the synaptic weights were modified five times in every sampling interval; and the constant parameter μ of $E(t)$ in (33) was set to $\mu = 5.0 \times 10^{-3}$.

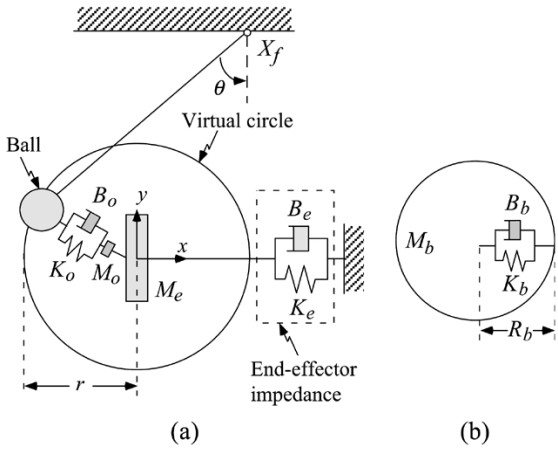

Fig. 5. Example of a catching task.

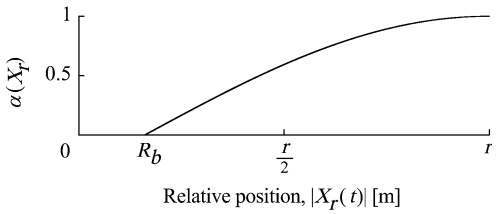

Fig. 6. Gain function used in an energy function.

The term $(\partial E(t))/(\partial F_{\rm act}(t))$ in (22), (23), and (24) was calculated online by using the energy functions in (33)–(35) based on the approximation techniques given in (30), (31) as follows:

$$\frac{\partial E(t)}{\partial F_{\rm act}(t)} = \frac{\partial E_v(t)}{\partial F_{\rm act}(t)} + \mu\frac{\partial E_f(t)}{\partial F_{\rm act}(t)} \tag{36}$$

$$\frac{\partial E_v(t)}{\partial F_{\rm act}(t)} = \Delta t_s \frac{\partial E_v(t)}{\partial \dot{X}_r(t)} \tag{37}$$

$$\frac{\partial E_f(t)}{\partial F_{\rm act}(t)} = \Delta t_s^2 \frac{\partial E_f(t)}{\partial X_r(t)} + \Delta t_s \frac{\partial E_f(t)}{\partial \dot{X}_r(t)}. \tag{38}$$

In the ball-catching task, the end-effector should move with the same direction as the approaching direction of the ball at the first phase and then slow down gradually in order to catch the ball. Consequently the gain function $\alpha(X_r)$ in (34) was designed as shown in Fig. 6 by

$$\alpha(X_r) = \begin{cases} \sin\dfrac{(|X_r| - R_b)\pi}{2(r - R_b)} & (|X_r| \geq R_b) \\ 0 & (|X_r| < R_b) \end{cases}. \tag{39}$$

In the simulation experiments, the virtual force exerted on the end-effector is invalidated ($F_o = 0$ [N]) after the end-effector contacts with the ball, and the learning of NNs is terminated when the relative velocity between the end-effector and the ball becomes under $\dot{X}_r < 5.0 \times 10^{-3}$ [m/s] in contact movements.

Figs. 7 and 8 present the simulation results with and without the proposed real-time learning under the desired interaction force $F_d = 2$ [N], where the figure (a) depicts the trajectories of the end-effector

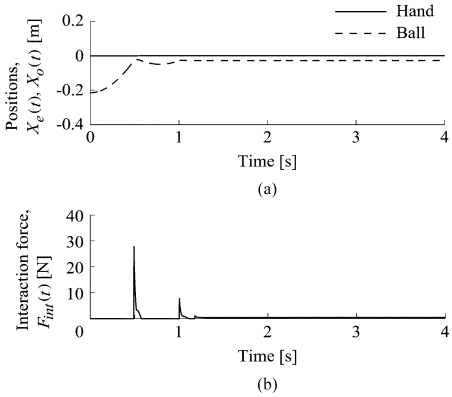

Fig. 7. Simulation results of the catching-a-ball task without learning. (a) Hand and ball motions. (b) Interaction force.

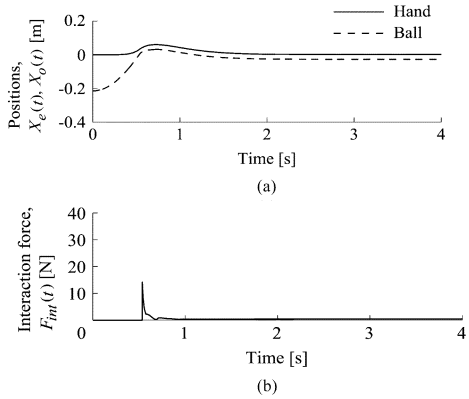

Fig. 8. Simulation results of the catching-a-ball task with real-time learning. (a) Hand and ball motions. (b) Interaction force.

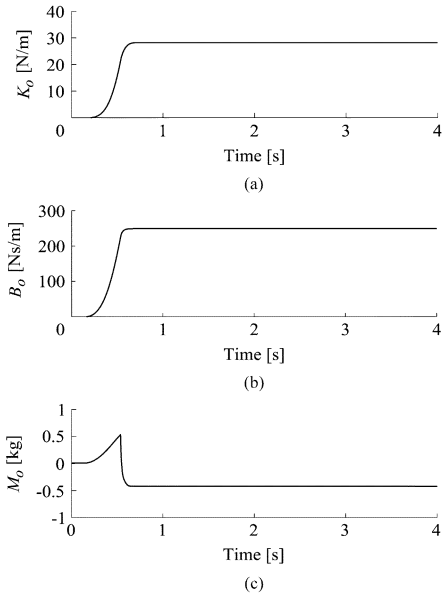

Fig. 9. Change of the virtual impedance parameters during the catching-a-ball task. (a) Virtual stiffness K_0. (b) Virtual viscosity B_0. (c) Virtual inertia M_0.

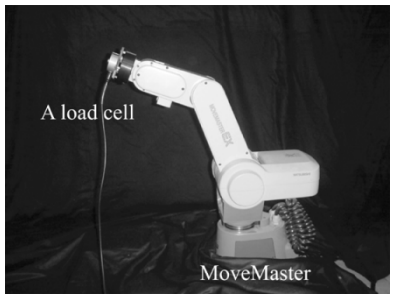

Fig. 10. Six-DOF multijoint manipulator.

(solid line) and the ball (broken line) while the figure (b) describes the time history of the interaction force. Fig. 8 shows that the end-effector takes avoidance actions before contacting with the approaching ball. Consequently, the manipulator catches the ball smoothly, and it's end-effector force after catching the ball is almost equal to the desired interaction force.

Fig. 9 shows the time histories of the virtual impedance parameters K_o, B_o, and M_o during the catching-a-ball task. Both of K_o and B_o increase just after the ball enters the virtual sphere, so that the manipulator can reduce the impact force by shifting the end-effector in the opposite direction of the approaching ball. On the other hand, M_o increases during free movements to control the end-effector with stable movements, and then reduces after contacting with the ball to reduce the interaction force. It can be seen that the proposed learning method can regulate the virtual impedance parameters effectively according to situations in the target task. It should be noted that the stable conditions are fulfilled in all cases: both $M^{-1}K$ and $M^{-1}B$ are positive definite.

C. Experiments With the Robotic Manipulator

To show the effectiveness of the proposed method, we performed catching-task experiments with a six-DOF multijoint robotic manipulator (MoveMaster RM-501: Mitsubishi Electric, Corp.) as shown in Fig. 10. A load cell (Kyowa Co., Ltd.) is attached at the end-effector of the manipulator to measure the contact force (see Fig. 11), where the object is a wooden cubic with 0.1 [m] each side. The object is hung from the ceiling by an aluminum stick with 1.7 [m] and swung like a pendulum, while its position is measured by a three-dimensional (3-D) position sensor (ISOTRACKII: POLHEMUS, Ltd.). Note that motions of the robotic arm and the object are restricted on the vertical two–dimensional (2-D) plane in the experiments.

The experiments were performed by changing the weight of object M_b, where the initial distance between the object and the end-effector was set as $D = 0.25$ [m]. The desired hand impedance of the end-effector was set at $K_e = \mathrm{diag.}[400, 400, 400]$ [N/m], $B_e = \mathrm{diag.}[200, 200, 200]$ [Ns/m], $M_e = \mathrm{diag.}[10, 10, 10]$ [kg]; and the radius of a virtual sphere was at $r = 0.2$ [m]. The sampling time for controlling the robotic manipulator was set at 1 [ms] that is short enough to catch the impact force at the time of the collision between the end-effector and the object.

The NNs utilized were of four-layered networks with four input units, two hidden layers with twenty units, and one output unit. The initial values of the synaptic weights w_{ij} were randomly chosen under $|w_{ij}| < 0.05$; and the learning rates were $\eta_S = 0.01$ for VSN, $\eta_V = 0.015$ for VVN, and $\eta_I = 1.0 \times 10^{-5}$ for VIN; and the output of NNs, M_o, B_o, K_o, were limited within -1000–1000. The constant parameter μ in the defined energy function $E(t)$ was set at $\mu = 1.0 \times 10^{-3}$; and the desired interaction force was at $F_{\mathrm{int}} = \mathrm{diag.}[0.0, -2.0, 0.0]$ [N]. Note that the learning of NNs were terminated after contacting the object with the end-effector by setting the virtual force $F_o = 0$ [N].

Fig. 12 shows typical experimental results with and without online learning under $M_b = 0.2$ [kg] and $D = 0.25$ [m], respectively. The figure (a) illustrates the time histories of the end-effector position along the x-axis (solid line) and the object (broken line), while the figures (b) and (c) show the time histories of the interaction force with and without

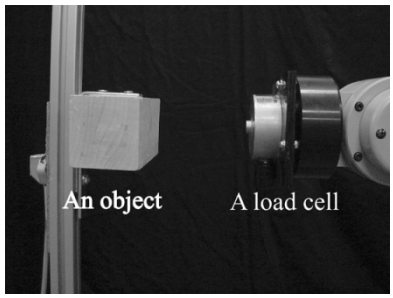

Fig. 11. Object and a load cell used in the experiments.

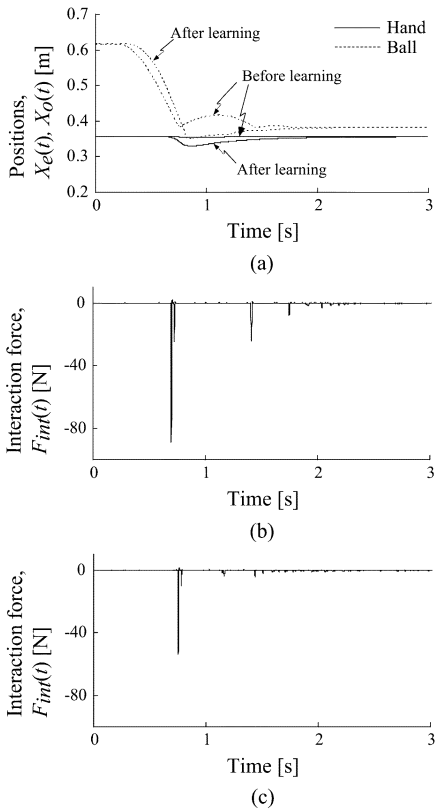

Fig. 12. Experimental results of the multi-joint manipulator ($M_b = 0.20$ [kg], $D = 0.25$ [m]). (a) Position before and after leranimg. (b) Interaction before after leranimg. (c) Interaction force after leranimg.

real-time learning. t_c indicates the time when the object comes into the virtual sphere. It can be seen that the manipulator moves its end-effector according to the ball movements after the learning of NNs, so that the robot cathes the ball smoothly by reducing the impact force between the hand and the ball.

The time course of the virtual impedance parameters along the x-axis during the online learning is displayed in Fig. 13. Since the manipulator had to avoid generating a large impact force by shifting the end-effector toward the opposite direction of the approaching ball, all of virtual impedance parameters, K_o^x, B_o^x, M_o^x, were increased by the learning of NNs. Here the learning rate for virtual inertia M_o^x was set at a small value because the object were swung with large acceleration in the experiment, and thus M_o^x was not dynamically changed compared with K_o^x, B_o^x. The experimental result shows that the robot did perform a ball-catching task by regulating virtual viscosity B_o^x actively on the basis of the energy function $E_v(t)$.

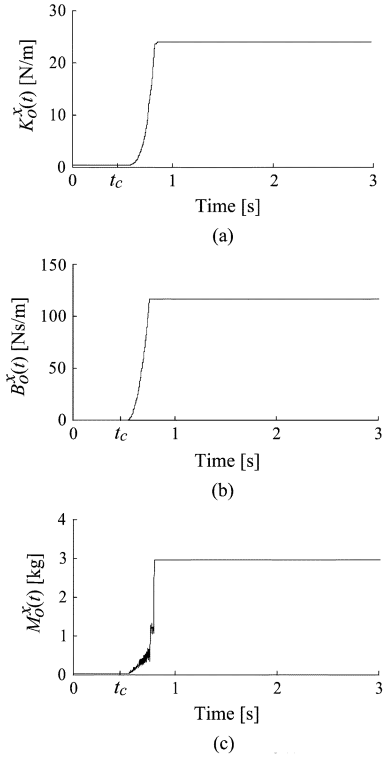

Fig. 13. Time history of the virtual impedance parameters during the catching-a-ball task by the real robot ($M_b = 0.20$ [kg], $D = 0.25$ [m]).

V. Conclusion

The present paper has proposed an online learning method using NNs to regulate the virtual impedance parameters in the noncontact impedance control of manipulators. The proposed method can obtain the desired virtual impedance by minimizing an energy function according to the given task through the learning of NNs. Experimental results with the multi-joint robotic manipulator have proven that the proposed method can design the desired virtual impedance for the end-effector to realize the target end-point force without exerting an excessive interaction force on its environment.

Because the proposed learning method adopts a back propagation learning approach, it can obtain the desired virtual impedance parameters that may minimize the energy function even if some impedance parameters are fixed in the learning. In other words, it does not always need to optimize all virtual impedance parameters to perform the given task. For instance, the stiffness and viscosity can be regulated under the fixed inertia parameter by the learning.

Although the present paper has focused on regulating virtual impedance parameters, the end-effector should be controlled by the conventional impedance method that regulates the end-point impedance during contact movements. 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. We also plan to develop an online learning method of the virtual impedance in parallel with the end-effector impedance at the same time.

References

[1] N. Hogan, "Impedance control: An approach to manipulation, parts I, II, III," *Trans. ASME, J. Dyn. Syst., Meas. Contr.*, vol. 107, no. 1, pp. 1–24, 1985.

[2] ——, "Stable execution of contact tasks using impedance control," in *Proc. IEEE Int. Conf. Robotics Automation*, 1987, pp. 1047–1054.

[3] A. Castano and S. Hutchinson, "Visual compliance: Task-directed visual servo control," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 334–342, June 1994.
[4] T. Tsuji and M. Kaneko, "Non-contact impedance control for redundant manipulator," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, pp. 184–193, Mar. 1999.
[5] Y. Nakabo and M. Ishikawa, "Visual impedance using 1 ms visual feedback system," in *Proc. IEEE Int. Conf. Robotics Automation*, 1998, pp. 2333–2338.
[6] M. Tokita, T. Mitsuoka, T. Fukuda, T. Shibata, and F. Arai, "Position and force hybrid control of a robotic manipulator by application of a neural network," *J. Math. Model. Sci. Comput.*, vol. 1, no. 3–4, pp. 235–246, 1993.
[7] H.-L. Pei, Q.-J. Zhou, and T. P. Leung, "A neural network robot force controller," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 1992, pp. 1974–1979.
[8] T. H. Connolly and F. Pfeiffer, "Neural network hybrid position/force control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 1993, pp. 240–244.
[9] J. M. Tao and J. Y. S. Luh, "Application of neural network with real-time training to robust position/force control of multiple robots," in *Proc. IEEE Int. Conf. Robotics Automation*, 1993, pp. 142–148.
[10] K. Kiguchi and T. Fukuda, "Position/force control of robot manipulators for geometrically unknown objects using fuzzy neural networks," *IEEE Trans. Ind. Electron.*, vol. 47, pp. 641–649, June 2000.
[11] G. Gomi and M. Kawato, "Neural network control for a closed loop system using feedback-error-learning," *Neural Networks*, vol. 6, no. 7, pp. 933–946, 1993.
[12] D. Katiæ and M. Vukobratoviæ, "Learning impedance control of manipulation robots by feedforward connectionist structures," in *Proc. IEEE Int. Conf. Robotics Automation*, 1994, pp. 45–50.
[13] C. C. Cheah and D. Wang, "Learning impedance control for robotic manipulators," *IEEE Trans. Robot. Automat.*, vol. 14, no. 3, pp. 452–465, June 1998.
[14] S. Jung and T. C. Hsia, "Neural network impedance force control of robot manipulator," *IEEE Trans. Ind. Electron.*, vol. 45, no. 3, pp. 452–465, June 1998.
[15] S.-T. Lin and H.-C. Tsai, "Impedance control with online neural network compensator for dual-arm robots," *J. Intell. Robot. Syst.*, vol. 18, pp. 87–104, 1997.
[16] S. Jung and T. C. Hsia, "Robust neural force control scheme under uncertainties in robot dynamics and unknown environment," *IEEE Trans. Ind. Electron.*, vol. 47, no. 2, pp. 403–412, Apr. 2000.
[17] S. Jung, S. B. Yim, and T. C. Hsia, "Experimental studies of neural network impedance force control for robot manipulators," in *Proc. IEEE Int. Conf. Robotics Automation*, 2001, pp. 3453–3458.
[18] H. Asada, "Teaching and learning of compliance using NeuralNets: Representation and generation of nonlinear compliance," in *Proc. IEEE Int. Conf. Robotics Automation*, 1990, pp. 1237–1244.
[19] M. Cohen and T. Flash, "Learning impedance parameters for robot control using an associative search networks," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 382–390, June 1991.
[20] B.-H. Yang and H. Asada, "Progressive learning and its application to robot impedance learning," *IEEE Trans. Neural Networks*, vol. 7, pp. 941–952, May 1996.
[21] T. Tsuji, M. Nishida, and K. Ito, "Iterative learning of impedance parameters for manipulator control using neural networks," *Trans. Soc. Instrum. Contr. Eng.*, vol. 28, no. 12, pp. 1461–1468, 1992.
[22] T. Tsuji, K. Ito, and P. G. Morasso, "Neural network learning of robot arm impedance in operational space," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 290–298, Apr. 1996.
[23] T. Tsuji, K. Harada, H. Akamatsu, and M. Kaneko, "On-line learning of robot arm impedance using neural networks," *J. Robot. Soc. Jpn.*, vol. 17, no. 2, pp. 234–241, 1999.
[24] K. Kitagaki and M. Uchiyama, "Optimal approach velocity of an end-effector to the environment," *Adv. Robot.*, vol. 8, no. 2, pp. 123–137, 1994.
[25] Y. Shoji, M. Inaba, and T. Fukuda, "Stable force control of 1 degree-of-freedom manipulator in contact tasks," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 1993, pp. 1511–1515.