

# A Human System Learning Model for Solving the Inverse Kinematics Problem by Direct Inverse Modeling

Eimei Oyama

Mechanical Engineering Laboratory, Tsukuba, Japan 305

Taro Maeda and Susumu Tachi

Faculty of Engineering, The University of Tokyo, Bunkyo, Tokyo, Japan 113

## SUMMARY

The problem of computing a human arm posture that will take the hand to a desired hand position given by vision is called an inverse kinematics problem. To solve this problem, the human nervous system has a system for solving the inverse kinematics problem computing the proper joint angles from the desired hand position.

Although the direct inverse modeling method is popular for the acquisition of an inverse kinematics model, a sufficient inverse model cannot be obtained for such systems with many-to-one input-output correspondence as a human arm system.

This paper, proposes a new model of a human inverse kinematics solver which uses the learned inverse model of the linearized model of the human arm. This inverse model transforms the hand position error to the update vector of the joint angles. The solver inverse kinematics problems by using the hand position error feedback. The performance of the solver is shown using numerical simulations.

**Key words:** Human hand position; control learning; inverse kinematics neural networks; direct inverse modeling; output feedback inverse model.

## 1. Introduction

The problem of computing a human arm posture that will take the human hand to a desired position given by vision is a kind of inverse problem. In order to solve the problem, an inverse kinematics solver, i.e., a system for solving an inverse kinematics problem, computing the proper joint angles from the desired hand position is necessary. A human forms the solver in the human nervous system. How the human nervous system acquires the inverse kinematics solver is one of the important problems in neuroscience. Although some theoretical models were proposed, they have a number of drawbacks.

A representative model of the solver solves inverse kinematics problems by using a learned inverse kinematics model which calculates a joint angle vector that corresponds to the hand position.

A representative inverse kinematics model learning method employed by many researchers uses a target systems output as input to the inverse model can uses the target systems input as the teaching signal of the inverse kinematics model that is composed of a learning element. This method called *Direct Inverse Modeling* (DIM) by Jordan [1]. Kuperstein [2] and Sakaguchi [3] proposed a model of a human inverse

kinematics solver that uses an inverse kinematics model learned by DIM. However, a sufficient inverse model cannot be obtained by DIM when the correspondence between the input and the output of the target system is not one-to-one and is nonlinear [1]. There are many joint angle vectors that correspond to one desired hand position. The inverse kinematics problem of a human arm is an ill-posed problem and cannot be solved by an inverse kinematics model learned by DIM.

In this paper, we propose a new learning model of the inverse kinematics solver that avoids the ill-posedness of the inverse kinematics problem by a modification of the target system. The relationship between the infinitesimal change of the joint angle vector and that of the hand position vector is linear. A kind of inverse kinematics model which transforms the desired change of the hand position to the update vector of the joint angles can be obtained by using DIM. The proposed inverse kinematics solver uses the acquired inverse model of the linearized model of the human arm as a hand position error feedback system and finds an inverse kinematics solution through iterative improvement.

The performance of the proposed method is shown by numerical simulations. At this stage, the proposed learning model still has some problems. However, the model can solve inverse kinematics problems of redundant arms. It can be a basic model of human inverse kinematics solver for discussion.

## 2. A Model of Human Hand Position Control System

### 2.1. Formulation of problem

Let  $x$  be a  $n \times 1$  hand position vector given by the vision system and  $\theta$  be a  $m \times 1$  vector of joint angle vector. The relationship between  $x$  and  $\theta$  is expressed as

$$x = f(\theta) \quad (1)$$

where  $f$  is a  $C^1$  class function. When a desired hand position vector  $x_d$  is given, consider an inverse kinematics problem that calculates the joint angle vector  $\theta_d$  that satisfies

$$x_d = f(\theta_d) \quad (2)$$

$f$  is  $C^1$  class function. Assume that an ideal learning element can approximate any continuous function. When the input of the learning element is  $t$ , let  $\Phi(t)$  be

the output of the element and  $\Phi'(t)$  be the teaching signal for the learning element. After teaching signals  $s_j$  ( $j = 1, 2, 3, \dots$ ) are given for the learning element as

$$\Phi'(t) = s_j \quad (3)$$

then

$$\Phi(t) \doteq E(s_j) \quad (4)$$

is established;  $E(s_j)$  is the mean of  $s_j$ ; and  $\Phi(t)$  is a  $C^1$  class function. By the generalization function of the learning element,  $\Phi(t)$  learned at representative points  $t_i$  ( $i = 1, 2, 3, \dots$ ) in the input space  $t$  can generate correct outputs at the whole input space  $t$ .

Representative learning elements, for example, the multilayer neural networks learned through back-propagation learning [4], Albus' CMAC [5], and Kohonen's topographic mapping networks [6] satisfy Eq. (4) approximately.

### 2.2. Assumption about human inverse kinematics solver

Kawato et al. proposed the hierarchical information processing model of the human voluntary movement [7]. The model consists of the following three information processings: (I) trajectory planning in the visual coordinates; (II) transformation of the trajectory from visual coordinates to body coordinates; and (III) control that generates motor commands to realize the desired trajectory.

In this paper, (II) coordinate transformation is considered. When a target point in the visual place is given, a human can move the hand to the point precisely by use of visual feedback. On the other hand, the precision of the hand position is greatly deteriorated without the viewing of the hand [8]. Considering these facts, the possibility is only slight that the human inverse kinematics solver depends mainly on the relationship between the absolute hand position in the visual coordinates and the absolute kinesthetic information. Since the measurement precision of the relative position and the change of the position is high [9-11], we assume that the human inverse kinematics solver is based on the relationship between the change of the hand position vector in the visual coordinates and the change of the joint angle vector in the body coordinates. Kawato proposed a learning model that acquires the Moore-Penrose Inverse of the Jacobian of the human arm for coordinate transformation by using the relationship between the vector of the change of

the hand position and that of the joint angle vector [12]. However, since this learning model uses a complex adaptive rule based on the steepest descent method, it has a drawback as a model of a human nervous system.

Georgopoulos et al. found neurons in the motor cortex of a rhesus monkey that become active at high firing rate when the monkey moves its hand in a certain direction [13-15]. A population vector is defined as the weighted sum of the peculiar direction of the neuron. The weight is calculated based on the difference between the current firing rate of the corresponding neuron and the firing rate at rest. The population vector predicted accurately the direction of the hand movement. The length of the vector has a strong relationship to the hand velocities. It is found that the firing rates of these direction-sensitive neurons in the motor cortex changes as the starting point of the hand changes [16].

Anderson et al. found the neurons which encode the position of the visual target on the head coordinates in the parietal association cortex [17]. Gentilucci et al. found target-sensitive neurons in inferior premotor cortex [18]. There are some developed models and objections regarding Georgopoulos' model [19-20]. However, there are neurons that have strong relationships to the direction of the hand movement in the parietal association cortex, the frontal association cortex, and the primary motor cortex. The primary motor cortex generates motor command based on the body coordinates [21]. The coordinate transformation system from the visual coordinates to the body coordinates must exist in these areas.

The vector of the change of the hand position or that of the velocity of the hand position has the same direction as the direction of the hand movement. From the physiological viewpoint, the change of the hand position is important in the human hand position control. Since the active area of the motor cortex changes according to the initial posture of the arm, the joint angle vector also is an important information.

Let  $\theta$  be the joint angle vector,  $x_d$  be the desired hand position vector,  $\Delta x_d$  be the desired change of the hand position vector, and  $\Delta \theta_d$  be the desired change of the joint angle vector. In this paper, a learning model of the human inverse kinematics solver that transforms  $\Delta x_d$  to  $\Delta \theta_d$  and generates the position command of the joint angles sequentially is proposed.

It is clear that a human has an inverse kinematics

model that is based on the relationship of the absolute hand position and the absolute joint angle vector and transforms the desired hand position to the joint angles command directly. This type of inverse kinematics model is not considered in this paper. After acquisition of the visual feedback system stated in section 2.3, the inverse kinematics model can be obtained by feedback error learning scheme proposed by Kawato [7].

### 2.3. Output feedback inverse model

Bullock and Grossberg proposed a computational model of the human arm control system that can explain Georgopoulos' experimental results [22]. By using the error vector between the final goal of the hand position  $x_{de}$  and the hand position  $x$  at the time  $t$  is defined as

$$e(t) = x_{de} - x(t), \quad (5)$$

the desired hand position  $x_d(t)$  is updated sequentially. If  $x(t)$  is controlled successfully to track the desired hand position  $x_d(t)$ , the direction of  $e(t)$  is constant in the reaching motion. Thus, Bullock and Grossberg asserted that Georgopoulos' results are explained.

We assume also that the desired trajectory is calculated in the visual coordinates and the temporal desired hand position is calculated sequentially at each time of the motion.

Hereafter, we use discrete descriptions for modeling of the human motion control system.

Let  $\Delta t$  be the sampling interval;  $t_k$  refers to the time  $k\Delta t$ . Hereafter, vector  $y(k)$  refers to the value of the vector  $y$  at time  $t_k$ . The sampling interval  $\Delta t$  is set to 0.1-0.3 s that corresponds to the time delay between the visual stimulus and the start of the visually guided motion.

The change of the desired hand position is calculated as

$$\Delta x_d(k) = x_d(k+1) - x_d(k) \quad (6)$$

The hand position error is calculated as

$$e(k) = x_d(k) - x(k) \quad (7)$$

The desired change of the hand position including the output error feedback is calculated as

$$\Delta x_d'(k) = \Delta x_d(k) + Ke(k) \quad (8)$$

where  $K$  is an appropriate coefficient.

Let  $\Delta\theta_d(k)$  be the update vector of the joint angle vector that will correspond to the desired change of the hand position  $\Delta x_d'(k)$ . By updating the joint angle vector as

$$\theta(k+1) = \theta(k) + \Delta\theta_d(k) \quad (9)$$

tracking control of the desired hand position  $x_d(k)$  is possible. Let  $J(\theta)$  be the Jacobian of the human arm expressed as

$$J(\theta) = \frac{\partial f(\theta)}{\partial \theta} \quad (10)$$

$\Delta\theta_d(k)$  can be calculated as follows:

$$\Delta\theta_d(k) = J^*(\theta(k)) \Delta x_d'(k) \quad (11)$$

where  $J^*(\theta)$  is a generalized inverse of  $J(\theta)$ .

We assume that the human nervous system acquires a coordinate transformation system as Eq. (11) by learning.

Let  $\Delta\theta$  be the infinitesimal change of the joint angle vector and  $\Delta x$  be the corresponding change of the hand position vector calculated as

$$\Delta x = f(\theta + \Delta\theta) - f(\theta) \quad (12)$$

When the joint angle vector changes smoothly,

$$\Delta x = J(\theta) \Delta\theta \quad (13)$$

is established approximately. Consider a learning element for the coordinate transformation, the input of which consists of  $\theta$  and  $\Delta x$ . Let  $\Phi(\theta, \Delta x)$  be the output of the learning element  $\Phi'(\theta, \Delta x)$  be the teaching signal for the learning element. By applying DIM to the system described in Eq. (13), the teaching signal

$$\Phi'(\theta, \Delta x) = \Delta\theta \quad (14)$$

is obtained. After off-line learning is conducted,

$$\Phi(\theta, \Delta x) \approx J^*(\theta(k)) \Delta x \quad (15)$$

is obtained. The details of the learning will be illustrated in section 3. This learning element  $\Phi(\theta, \Delta x)$  can be used as the coordinates system. By using the above equations, the following hand position control system

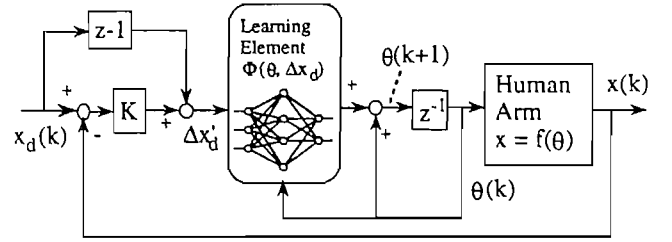


Fig. 1. Configuration of output feedback inverse model.

can be obtained:

$$\begin{aligned} \theta(k+1) &= \theta(k) + \Delta\theta_d(k) \\ \Delta\theta_d(k) &= \Phi(\theta, \Delta x_d') \\ &\approx J^*(\theta(k)) \Delta x_d'(k) \\ \Delta x_d'(k) &= \Delta x_d(k) + Ke(k) \end{aligned} \quad (16)$$

If  $K$  equals 1,

$$\Delta x_d'(k) = x_d(k+1) - x(k) \quad (17)$$

The hand position control system is expressed by

$$\theta(k+1) \approx \theta(k) + J^*(\theta(k))(x_d(k+1) - f(\theta(k))) \quad (18)$$

The above equation approximates the quasi-Newton method.

The proposed inverse kinematics solver expressed in Eq. (16) is not an ordinary inverse kinematics model. However, since it can calculate an output similar to the inverse kinematics model as the result of iterative improvements, we have named it the *output feedback inverse model* [23]. We have also referred to the same solver as the *output error feedback inverse model*. Hereafter, OFIM refers to the proposed inverse kinematics solver.

Figure 1 shows the configuration of OFIM, where  $z^{-1}$  indicates the control element that calculates a previous value of a signal at  $\Delta t$  time units in the past;  $z - 1$  indicates the control element that calculates a change of a signal. Figure 2 shows the teaching signal for the learning element in Eq. (14).

From the engineering viewpoint, the output feed-

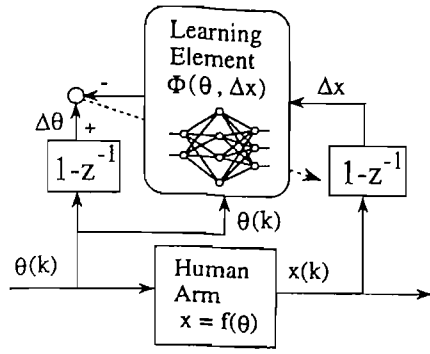


Fig. 2. Teaching signal for learning element.

back inverse model does not have sufficient precision and usually requires a long learning time. Kawato's learning method of a nonlinear feedback gain or our improved model for engineering application can be useful for practical application [7, 24].

#### 2.4. Coordinate transformation system paying attention to velocity and acceleration

The relationship between the velocity vector of the joint angles and that of the hand position

$$\dot{x} = J(\theta)\dot{\theta} \quad (19)$$

is linear as well as Eq. (13). An inverse model of the system as described in Eq. (19) can also be obtained by DIM.

The relationship between the acceleration vector of the joint angles and that of the hand position

$$\ddot{x} = J(\theta)\ddot{\theta} + \left[ \frac{\partial J(\theta)}{\partial \theta} \dot{\theta} \right] \dot{\theta} \quad (20)$$

also is linear paying attention to  $\ddot{\theta}$  and  $\dot{\theta}$ . An inverse model of the system as described in Eq. (20) can also be obtained by using DIM. Inverse kinematics solvers based on the above inverse models are presented in Appendix 2.

#### 2.5. Solution for static inverse kinematics problems

This section considers a static inverse problem in which a desired trajectory of the hand position and an initial posture of the arm are not given, but a fixed desired hand position is given.

In order to calculate the joint angle vector that corresponds to the fixed desired hand position, first, an initial joint angle vector is generated and a trajectory that continuously connects the initial hand position and the fixed desired hand position is calculated. Then, by tracking the trajectory, OFIM can move the hand to the fixed desired hand position and finally calculate the joint angles vector that corresponds to the fixed desired hand position.

When any joint reaches its stroke end or the Jacobian of the arm becomes singular, OFIM cannot calculate the joint angle vector. In such a case, a new initial joint angle vector is generated by a uniform random number generator and the above procedure is repeated until the error becomes smaller than the desired precision.

### 3. Learning of Coordinate Transformation System

In this section, the learning characteristics of the neural networks  $\Phi(\theta, \Delta x)$  for the coordinate transformation is presented.

#### 3.1. Off-line learning by direct inverse modeling

In this paper, the term *off-line learning* refers to learning which is conducted in motions generated when a human does not intend to control the hand position.

When  $|\Delta\theta|$  is sufficiently small, the approximate equation (13) is established. Assume that  $\Phi(\theta, \Delta x)$  is an ideal learning element. Let  $\Delta\theta^*$  be the change of joint angle vector that satisfies Eq. (13). After the learning based on Eq. (14),

$$\Phi(\theta, \Delta x) \approx E(\Delta\theta^*) \quad (21)$$

is established according to Eq. (4). When  $n$  is equal to  $m$  and  $J(\theta)$  is a full rank matrix,  $\Delta\theta$  that corresponds  $\Delta x$  is unique:

$$\Phi(\theta, \Delta x) \approx E(\Delta\theta^*) = J^{-1}(\theta)\Delta x \quad (22)$$

is obtained.

When  $n$  is smaller than  $m$ , there are numerous numbers of  $\Delta\theta^*$  that satisfy Eq. (13);  $\Delta\theta^*$  can be expressed by using  $\Delta x$  and  $m \times 1$  vector  $u$  that has no correlation to  $\Delta x$  as follows:

$$\begin{aligned}\Delta\theta^* &= \Delta\theta^*(\Delta x, u) \\ &= J^*(\theta)\Delta x + (I - J^+(\theta)J(\theta))u\end{aligned}\quad (23)$$

where  $J^*(\theta)$  is a generalized inverse of  $J(\theta)$  and

$$J^+(\theta) = J^T(\theta)(J(\theta)J^T(\theta))^{-1}\quad (24)$$

is the Moore-Penrose inverse of  $J(\theta)$ ;  $J^*(\theta)$  can be expressed as

$$J^*(\theta) = J^+(\theta) + (I - J^+(\theta)J(\theta))G(\theta)\quad (25)$$

where  $G(\theta)$  is a  $m \times m$  matrix. From Eq. (4),

$$\begin{aligned}\Phi(\theta, \Delta x) &\approx E(\Delta\theta^*(\Delta x, u)) \\ &= J^*(\theta)\Delta x + (I - J^+(\theta)J(\theta))E(u)\end{aligned}\quad (26)$$

is obtained;

$$J(\theta)\Phi(\theta, \Delta x) \approx \Delta x\quad (27)$$

is established for any value of  $E(u)$ ;  $\Phi(\theta, \Delta x)$  is an approximate inverse model of the system described in Eq. (13). A coordinate transformation system can be acquired by the teaching signal described in Eq. (14).

When  $\text{rank}(J(\theta))$  is  $n$ ,  $\text{rank}(I - J^+(\theta)J(\theta))$  is  $m - n$ . From Eq. (26),  $\Delta\theta^*$  exists in the  $m - n$  dimensional space on the  $\Delta\theta$  space. Let  $dv$  be a volume element of  $m - n$ -dimensional space where  $\Delta\theta^*$  exists, and  $p(\Delta\theta)$  be the probability density function that describes the distribution of  $\Delta\theta$ . Equation (21) can be expressed as

$$\begin{aligned}\Phi(\theta, \Delta x) &\approx E(\Delta\theta^*(\Delta x, u)) \\ &= \frac{\int_Q p(\Delta\theta^*(\Delta x, u))\Delta\theta^*(\Delta x, u)dv}{\int_Q p(\Delta\theta^*(\Delta x, u))dv}\end{aligned}\quad (28)$$

The region of integral  $Q$  is the whole space where  $\Delta\theta^*(\Delta x, u)$  distributes;  $p(\Delta\theta)$  is 0 where  $|\Delta\theta|$  is large. In human usual movements without the intention controlling the human hand position,

$$p(\Delta\theta) = p(-\Delta\theta)\quad (29)$$

is established. Therefore

$$\Phi(\theta, 0) \approx \frac{\int_Q p(\Delta\theta^*(0, u))\Delta\theta^*(0, u)dv}{\int_Q p(\Delta\theta^*(0, u))dv} = 0\quad (30)$$

is obtained. Since  $\Phi(\theta, \Delta x)$  can be developed as

$$\begin{aligned}\Phi(\theta, \Delta x) &= \\ \Phi(\theta, 0) &+ \left[\frac{\partial\Phi(\theta, \Delta x)}{\partial\Delta x}\right]_{\Delta x=0}\Delta x + O(\Delta x^2)\end{aligned}\quad (31)$$

$O(\Delta x^2)$  is the sum of higher-order terms of  $\Delta x$  in each equation. From Eqs. (26), (30), and (31), Eq. (15) is established at least where  $|\Delta x|$  is sufficiently small.

In the human usual movements,  $p(\Delta\theta)$  is a constant on a curved surface of an appropriate ellipsoid. By using an appropriate  $m \times m$  positive-definite symmetric matrix  $P$  and an appropriate scalar function  $g$ ,  $p(\Delta\theta)$  can be expressed as

$$p(\Delta\theta) = g(\Delta\theta^T P^{-1} \Delta\theta)\quad (32)$$

It should be noted that if Eq. (32) is established and the distribution of  $\Delta\theta$  is normal, the covariance matrix of  $\Delta\theta$  can be expressed as

$$E(\Delta\theta\Delta\theta^T) = \lambda P\quad (33)$$

where  $\lambda$  is an appropriate scalar. According to Appendix 1,

$$\Phi(\theta, \Delta x) \approx P J^T (J P J^T)^{-1} \Delta x\quad (34)$$

is established on the  $\Delta x$  region, where  $\Delta x$  used for the learning is distributed.

The right-hand of Eq. (34) is the solution of Eq. (13) that will minimize the performance index  $S$  defined by

$$S = \frac{1}{2} \Delta\theta^T P^{-1} \Delta\theta\quad (35)$$

If  $P$  is a diagonal matrix, the performance index can be expressed as

$$S = \frac{1}{2} \sum_{i=1}^m \frac{\Delta\theta_i^2}{P_{ii}}\quad (36)$$

From the fact that the above performance index is minimized it can be interpreted that a joint that moves well in usual motions, moves well to satisfy Eq. (13). If the desired hand position  $x_d(k)$  changes smoothly, it is guaranteed that the joint angle vector  $\theta(k)$  calculated by the coordinate transformation in Eq. (34) also changes smoothly. Whitney proposed the coordinate transformation method that uses the Moore-Penrose inverse of the Jacobian in order to control a redundant arm [25]. Kawato proposed an approximate learning method for the Moore-Penrose

inverse of the Jacobian for the coordinate transformation [12]. The result of the proposed learning rule approximates those coordinate transformation methods.

### 3.2. On-line learning

In this paper, *on-line learning* refers to a learning that is conducted when a human intends to control the hand position and controls the hand by using Eq. (16);  $\Delta\theta_d$  is calculated by the following equation:

$$\Delta\theta_d = \Phi(\theta, \Delta x'_d) \quad (37)$$

On-line learning is conducted according to the following equations:

$$\Phi'(\theta, \Delta x) = \Delta\theta_d = \Phi(\theta, \Delta x'_d) \quad (38)$$

$$\begin{aligned} \Delta x &= f(\theta + \Delta\theta_d) - f(\theta) \\ &\approx J(\theta)\Delta\theta_d \end{aligned} \quad (39)$$

When the target system is a one-to-one system and the inverse model starts learning from its rather precise initial status, goal directed learning is possible by on-line direct inverse modeling [26, 27]. The term *goal directed learning* refers to learning that can improve the precision of the learning element regarding the desired output of the system. The forward and inverse modeling proposed by Jordan [1] and the feedback error learning scheme proposed by Kawato [7] can conduct goal-directed learning. Off-line direct inverse modeling cannot conduct goal directed learning.

In the case of the many-to-one target system, goal directed learning is sometimes possible by on-line direct inverse modeling if the rather precise initial status of the inverse model has already been acquired. However, it is difficult for DIM to acquire the rather precise initial status of the inverse model.

Furthermore, in on-line learning, Eq. (27) is established but Eq. (15) is not established. If the input of the inverse model  $\Delta x'_d$  becomes 0,  $\Phi(\theta, \Delta x'_d)$  does not always become 0 and the joint angles still move. These characteristics are not good for arm control.

### 3.3. Characteristics of on-line and off-line learning

The learning result of off-line learning does not depend on the initial status of the learning element but

it does depend on what teaching signals for the learning element are given.

Since the reaching signal of on-line learning is determined by the output of the learning element, the learning result of on-line learning depends on the initial status of the learning element. The learning sometimes fails because of the bad initial status of the learning element. It should be noted that the forward and inverse modeling and the feedback error learning scheme have the same characteristics. By using only on-line learning, if the initial state of the learning element is bad, a sufficiently precise inverse model cannot be obtained. Even if the initial state of the learning element is good,  $\Phi(\theta, 0)$  does not always become 0.

It is difficult for the off-line learning to conduct goal-directed learning regarding the specified trajectory on which a human wants to move.

Therefore, we consider that both off-line and on-line learning are conducted in human sensory motor learning. We call this way of learning hybrid learning. If both off-line and on-line learning are conducted, the final learning result does not depend on the initial status but depends mainly on the teaching signals used in off-line learning.

Before a human becomes able to reach and grasp objects, he/she starts many reflex motions and repeats a variety of motions in his/her infancy. In this period, a large amount of off-line learning is conducted and the inverse kinematics solver is formed. The adaptation of the inverse kinematics solver after the acquisition is conducted by both on-line and off-line learning.

## 4. Simulations

Numerical experiments were performed in order to evaluate the performance of the proposed model. The inverse kinematics model of the 3 DOF arm moving on the 2 DOF plane is considered. The relationship between the joint angle vector  $(\theta_1, \theta_2, \theta_3)^T$  and the hand position vector  $(x, y)^T$  is as follows:

$$\begin{aligned} x &= x_0 + L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ &\quad + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= y_0 + L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \\ &\quad + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{aligned} \quad (40)$$

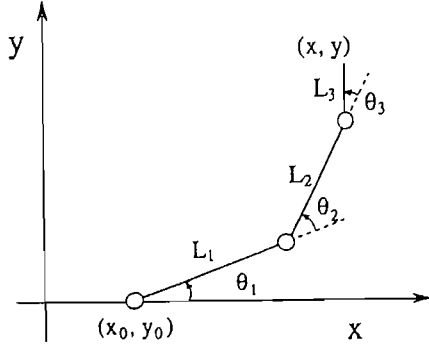


Fig. 3. Configuration of arm.

This is a simplified model of the human arm moving on a vertical/horizontal plane. Figure 3 shows the configuration of the arm. The range for  $\theta_1$ , which corresponds to the shoulder joint, is  $(-30^\circ, 120^\circ)$ , the range for  $\theta_2$ , which corresponds to the elbow joint, is  $(0^\circ, 120^\circ)$ , and the range for  $\theta_3$ , which corresponds to the wrist joint, is  $(-60^\circ, 60^\circ)$ .  $L_1$  is 0.30 m,  $L_2$  is 0.25 m and  $L_3$  is 0.15 m.

Consider the problem of calculating the joint angles  $(\theta_1, \theta_2, \theta_3)^T$  which brings the hand to the desired position  $(x, y)^T$ .

For simplicity,  $K$  is set to 1 in Eq. (16). An inverse kinematics solver that calculates the right side of Eq. (18) numerically is used for comparison. Numerical methods (NM) refers to this solver.

Trajectories generated as follows are used for the evaluation of the inverse kinematics solver.

First, an initial joint angle vector  $\theta(0)$  is generated by using a uniform random number generator. The hand position  $x(0)$  that corresponds to the joint angle vector is regarded as the initial desired hand position  $x_{ds}$ . Second, the desired total change of the hand position  $\Delta x_{se}$  is generated by using a normal random number generator. The  $i$ -th component of  $\Delta x_{se}$  is calculated by:

$$\Delta x_{sei} = 0.2\omega(m) \quad (41)$$

The final desired hand position  $x_{de}$  is calculated as

$$x_{de} = x_{ds} + \Delta x_{se} \quad (42)$$

A trajectory  $x_d(k)$  ( $0 \leq k \leq 2T$ ) is defined as

$$x_d(k) = \begin{cases} (1 - k/T)x_{ds} + k/Tx_{de} & (0 \leq k \leq T) \\ x_{de} & (T < k \leq 2T) \end{cases} \quad (43)$$

If  $x_d(k)$  is not reachable kinematically, a new  $\Delta x_{se}$  is generated again. Finally, a trial in which this trajectory is tracked by the inverse kinematics solver is conducted.

$T$  is set to 5. The sampling period  $\Delta t$  is 0.2 s. The standard deviation of the velocity of the desired hand position is 0.2 m/s.

In order to evaluate the performance of the solver, 10,000 tracking trials are conducted for the estimation of root mean square (RMS) of  $e(k)$ . A trial in which any joint angle vector reaches its stroke-end in computation is regarded as a failure.

Four layered neural networks are used for the simulations. The 1st layer, i.e., the input layer, has 5 neurons. The 2nd and the 3rd layers have 15 neurons each. The 4th layer, i.e., the output layer, has 3 neurons. The 1st layer and the 4th layer consist of linear neurons. The 2nd layer and the 3rd layer consist of nonlinear neurons which input-output relationship is defined by a sigmoid function  $y = \tanh(x)$ .

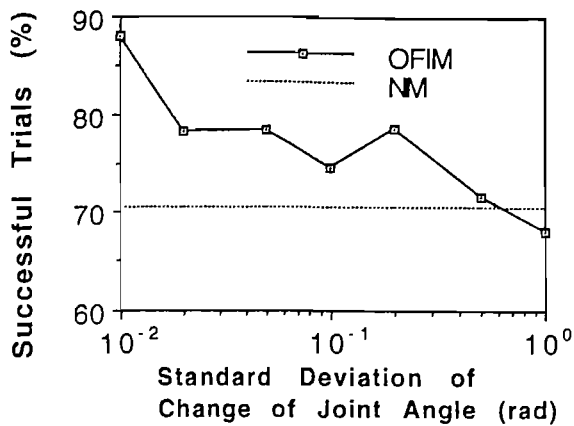
The backpropagation method is used for the learning of the neural networks. The possibility is only slight that the human nervous system utilizes the backpropagation method. However, since any learning element that satisfies Eqs. (3) and (4) can be used for the evaluation and the backpropagation method is popular and not complex, this method is used in this simulation. The more realistic learning elements such as Kohonen's topographic mapping will be used in the future.

Let  $\Phi(i)$  be the output of the neural network at  $i$ -th learning trial,  $W(i)$  be the weight vector of the connection of the neural network, and  $\Phi'(i)$  be the teaching signal. The backpropagation method [4] used for the learning is expressed as:

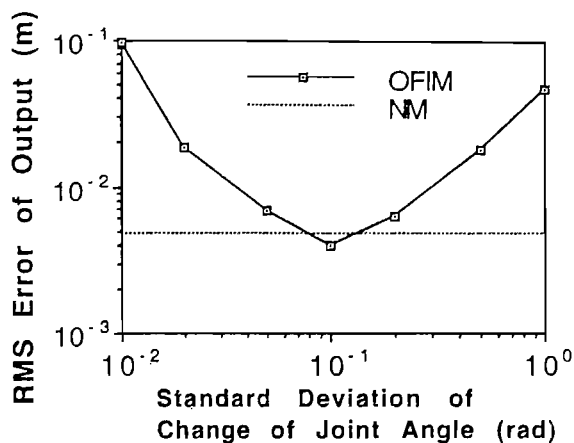
$$\begin{aligned} W(i+1) &= W(i) + \Delta W(i) \\ \Delta W(i) &= -\eta \frac{\partial S(i)}{\partial W} + \alpha \Delta W(i-1) \\ S(i) &= |\Phi'(i) - \Phi(i)|^2 \end{aligned} \quad (44)$$

where  $\eta$  is the learning rate and  $\alpha$  is the inertial factor.





(a) Percentage of successful trials (%)



(b) RMS error of output (m)

Fig. 4. Performance of OFIM after off-line learning.

The initial values of  $W(i)$  are generated by using uniform random number which range is  $(-0.5, 0.5)$ .

#### 4.1. Off-line learning

A numerical simulation of off-line learning is conducted in order of the relationship between the characteristics of the teaching signal and the precision of the hand position control of OFIM.

A learning trial is conducted as follows: First, an initial joint angle vector is generated by using uniform random number. Second, the joint angle vector is changed by the small step  $\Delta\theta$  generated by using normal random number generator. Finally, the change of the hand position  $\Delta x$  is measured and the learning is conducted according to Eq. (14).

The  $i$ -th component of  $\Delta\theta$  is calculated by using

a normal random number  $\omega$  as follows:

$$\Delta\theta_i = \sigma\omega \quad (45)$$

where  $\sigma$  is the standard deviation of the change of the joint angle vector.

In this simulation, the sampling period  $\Delta t$  is set to 0.2 s and  $\sigma = 0.1$  corresponds 0.5 rad/s.  $\eta$  is 0.001 and  $\alpha$  is 0.5. The 10,000,000 learning trials were carried out before the evaluation.

These 10,000 trajectories were generated, in order to estimate the percentage that the hand succeeds in tracking the desired trajectory within the range of the joint angle vector and the RMS error of the hand position. It should be noted that, since the arm used in the simulation is a redundant arm, some joint angles reach their stroke end at certain times. These trials are regarded as failure trials.

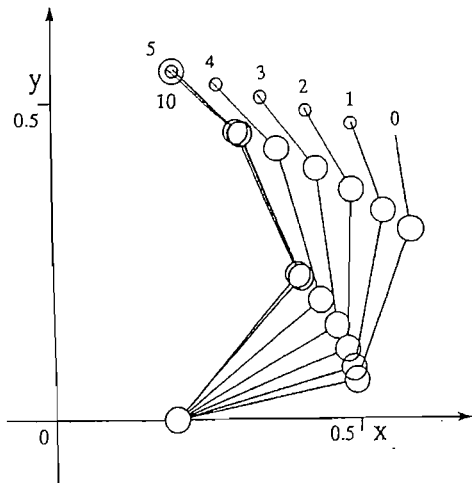
Figure 4 shows the relationship between the precision of the inverse kinematics solver and  $\sigma$ . Figure 4(a) shows the percentage of the successful tracking. Figure 4(b) shows the RMS error of the output. NM in the figures indicates the simulation results of the numerical method. If the parameter  $\sigma$  is appropriate, the precision of the proposed inverse kinematics solver is nearly equal to that of the numerical method.

It is understood that the precision of the proposed output feedback inverse model is almost the same as that of the numerical method when the learning parameter  $\sigma$  is appropriate. In the future, the real value of  $\sigma$  that is used in human learning will be considered.

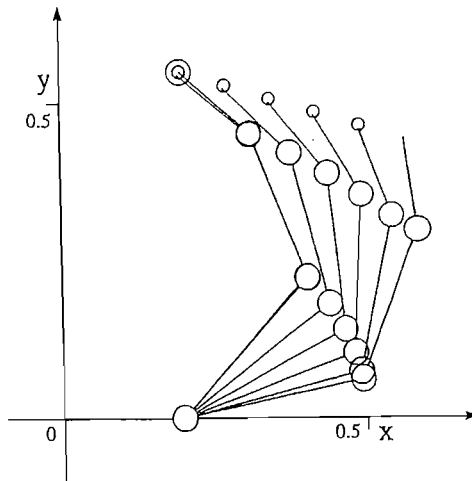
Figure 5 shows one example of the arm postures generated for tracking a trajectory of the desired hand position. The number near from the end point of the arm indicates the value of  $k$ . The center of a small circle in the figure indicates the desired hand position at  $k$ -th update. The radius of the circle is 0.01 m. The center of a large circle in the figure indicates the final desired hand position  $x_{de}$ . The radius of the circle is 0.02 m. The arm postures generated by OFIM is almost similar to those generated by NM.

#### 4.2. On-line learning

Tracking trials in which the hand position is controlled to track the desired trajectory generated by Eq. (43) and the inverse model is learned simultaneously are conducted in the simulation of on-line learn-



(a) Output feedback inverse model



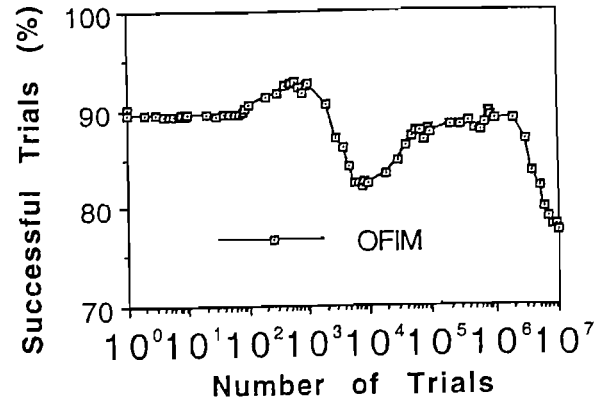
(b) Numerical method

Fig. 5. Path of hand (off-line learning).

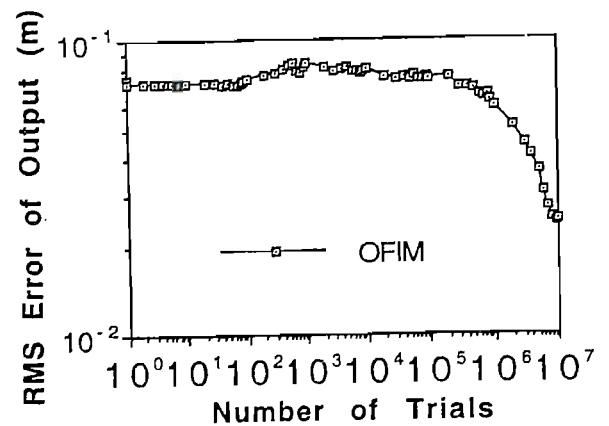
ing. The update of the weights of the neural networks is conducted at each update of Eq. (16) according to Eq. (14). Ten times update of the weight is conducted in one trial.

By on-line learning from 100 initial states of the neural networks, no sufficiently precise inverse model can be obtained.

This inverse model learning is almost impossible by only on-line learning. After 100,000 times off-line learning trials as described in section 4.1, the simulation of on-line learning was conducted. In the off-line learning,  $\rho$  is 0.5 rad. The learning rate  $\eta$  in the Eq. (44) for on-line learning is  $2.0 \times 10^{-1}$  and the in-



(a) Percentage of successful trials (%)



(b) RMS error of output (m)

Fig. 6. Simulation result of on-line learning.

ertial factor  $\alpha$  is 0.5. The learning rate for the on-line learning is smaller than that for the off-line learning in order to stabilize the learning.

Figure 6 shows the progress of on-line learning. The RMS error of the hand position decreases as the number of learning trials increases. The performance of the inverse kinematics solver increases by on-line learning. However, after  $10^7$  tracking trials, the RMS error stops decreasing and starts increasing.

Figure 7 shows the hand position control by the inverse kinematics solver learned by  $10^7$  learning trials. Although the on-line learning started from rather good initial state of the learning element, the precision of the solver learned by the on-line learning is lower than that of the solver learned by the off-line learning shown in section 4.1. The solver learned by on-line learning becomes unstable easily and the RMS error of the solver sometimes increases easily. As stated in section

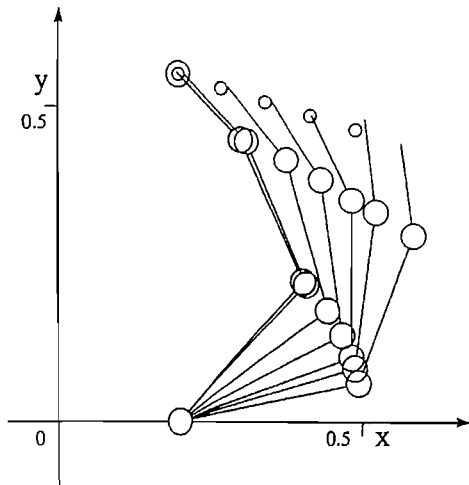


Fig. 7. Path of hand (OFIM, on-line learning).

3.3, the possibility that a human forms the inverse kinematics solver by using only on-line learning is low.

#### 4.3. Hybrid learning

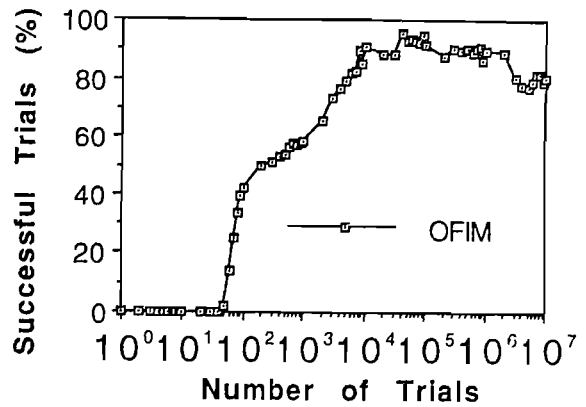
The simulation of the hybrid learning that consists of both on-line and off-line is conducted.

The off-line learning is conducted when the arm moves from the end point of one tracking trial to the start point of the next tracking trial. When the change of the joint angle vector reaches

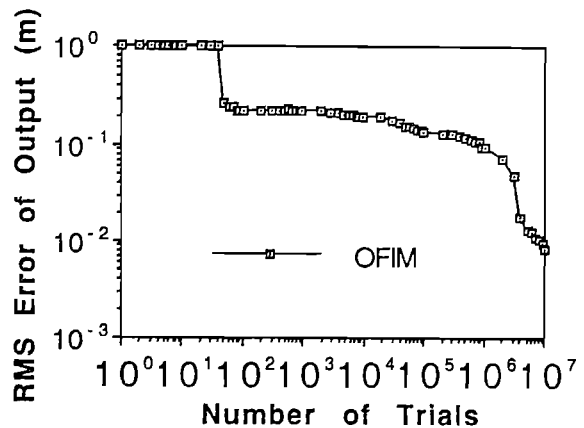
$$|\Delta\theta| = 0.25(\text{rad}) \quad (46)$$

a trial of learning is conducted. 0.25 rad corresponds to 1.25 rad/s when the sampling period is 0.2 s. The learning rate  $\eta$  is  $5.0 \times 10^{-4}$  and the inertial factor  $\alpha$  is 0.5.

Figure 8 shows the progress of hybrid learning. It is understood that the RMS error decreases and the precision of the solver becomes higher as the number of trials increases. After  $10^7$  tracking trials, the RMS error still keeps decreasing. The RMS error is  $5.03 \times 10^{-3}$  at  $10^8$  tracking trials. The mean number of learning in one tracking trial is 16.3. Figure 9 shows the hand position control by the learned solver. As the off-line learning in section 4.1, the path of the hand is almost similar to that of the numerical method. It is understood that the solver learned by the hybrid learning has sufficient precision of the hand position control.



(a) Percentage of successful trials (%)



(b) RMS error of output (m)

Fig. 8. Simulation result of hybrid learning.

#### 4.4. Solution for static inverse kinematics problems

The simulation solving static inverse kinematics problems by using the solver learned in section 4.3 is conducted. A trial for solving a static inverse kinematics problem is conducted as follows. First, a joint angle vector is generated by using uniform random number. Then, an inverse kinematics problem, which regards the hand position corresponding the joint angle vector as a desired position, is solved. The RMS error of the hand positions generated by the solver is measured for the evaluation of the solver. The 10,000 desired hand positions were generated for the evaluation.

When a desired hand position is given, an initial joint angle vector  $\Delta\theta(0)$  is generated by using a uniform random generator. A desired hand trajectory is

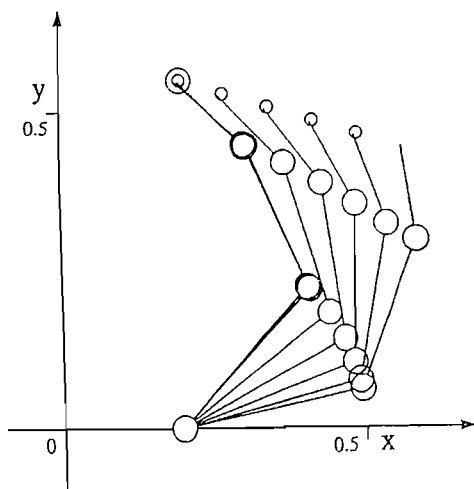


Fig. 9. Path of hand (OFIM, hybrid learning).

defined by Eq. (43) and the tracking procedure of the trajectory is started as stated in section 2.5. If the norm of the error between the hand position and the desired position does not become smaller than the desired position, a new initial joint angle vector is generated and the tracking procedure is repeated.

The upper limit of the generation of the initial joint angle vector is 20 for one desired hand position. If no solution which satisfies the desired precision is obtained, the solver outputs the joint angle vector which generates the smallest hand position error in all tracking procedures as a solution of the inverse kinematics problem.

For comparison, the precision of the solver that uses the inverse kinematics model learned by DIM is estimated. In this chapter, DIM refers to the solver learned by DIM. A 4-layered neural network is used for DIM. The 1st layer of the networks has two neurons. The 2nd and 3rd layers have 15 neurons each. The 4th layer has three neurons. The learning by DIM is conducted from 10 initial states of the neural networks and the best learning result is selected for the evaluation. The RMS error of the OFIM is  $6.39 \times 10^{-3}$  m and the mean number of the initial value changes is 0.692 when the desired precision is 0.01 m. The RMS error is  $5.48 \times 10^{-3}$  m and the mean number of the initial value changes is 4.29 when the desired precision is 0.005 cm. The RMS error of DIM is  $1.62 \times 10^{-2}$ . The precision of the output feedback inverse model is higher than that of DIM in this simulation.

## 5. Conclusions

In this paper, we proposed a new learning model which avoids the ill-posedness of the of the inverse kinematics problem and obtains an inverse kinematics solver of a human arm and corroborated its learning capability through numerical experiments.

DIM has another problem as a human learning model. The large-scale connection change is necessary before the inverse model is used for control after learning. When controlling the hand position, the input of the learning element is the desired change of the joint angles.

When learning, the input of the inverse model is the observed change of the joint angles. Though the desired change and the observed change should coincide, the difference of the characteristics of these two signals is large. The solution of this problem will be reported in the future.

**Acknowledgement.** The authors thank Dr. Mitsuo Kawato, the director of Department 3 of ATR Human Information Processing Research Laboratories, for providing papers and helpful comments. They also thank Dr. Arvin Agah for his help in the translation of this paper.

## REFERENCES

1. M. I. Jordan. Supervised learning and systems with excess degrees of freedom. COINS Technical Report, 88-27, 1-41 (1988).
2. M. Kuperstein. Neural model of adaptive hand-eye coordination for single postures. *Nature*, **239**, pp. 1308-1311 (1988).
3. Y. Sakaguchi, M. Zama, T. Maeda, K. Nakano and T. Omori. Coordination System of Sensor System and Motor System I. Proc. 26-th Conference of Society of Instrument and Control Engineering, pp. 73-74 (1988).
4. D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning Internal Representation by Error Propagation. *Parallel Distributed Processing*, D. E. Rumelhart and J. L. MacClelland and the PDP Research Group (ed.), pp. 318-326, MIT Press (1986).
5. J. S. Albus. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, **97**, pp. 220-227 (1975).
6. T. Kohonen. Formation of topographic maps and

- columnar microstructures in nerve fields. *Biological Cybernetics*, **35**, pp. 63-72 (1979).
7. M. Kawato, K. Furukawa and R. Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, **57**, pp. 169-185 (1987).
  8. T. Maeda and S. Tachi. Sensory integration of binocular visual space and kinesthetic space in visual reaching experiment. *Trans. Society of Instrument and Control Engineers*, **29**, 2, pp. 201-210 (1993).
  9. H. V. Helmholtz. *Handbuch der Physiologischen Optik*, **2**, 3rd ed. Hamburg und Leipzig, Voss (1911).
  10. S. Shlaer. The relation between visual acuity and illumination. *Journal of General Physiology*, **21**, pp. 165-188 (1937).
  11. H. W. Leibowitz. The relation between the rate threshold for the perception of movement and luminance for various durations of exposure. *Jour. Experimental Psychology*, **49**, pp. 829-830 (1955).
  12. M. Kawato. An optimization and learning in neural networks for formation and control of coordinated movement. ATR Technical Report, TA-A-0086 (1990).
  13. A. P. Georgopoulos, J. F. Kalaska, R. Caminiti and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Jour. Neuroscience*, **2**, pp. 1527-1537 (1982).
  14. A. P. Georgopoulos, J. F. Kalaska, J. D. Crutcher, R. Caminiti and J. T. Massey. The representation of movement direction in the motor cortex: Single cell and population studies. *Dynamical Aspects of Neocortical Function*, G. M. Edelman, W. E. Gall, and W. M. Cowan (ed.), Neuroscience Research Foundation, pp. 501-524 (1984).
  15. A. P. Georgopoulos, A. B. Schwartz and R. E. Kettner. Neuronal population coding of movement direction. *Science*, **233**, pp. 1416-1419 (1986).
  16. R. Caminiti, P. B. Johnson and A. Urbano. Making arm movements within different parts of space, dynamic aspects in the primate motor cortex. *Jour. Neuroscience*, **10**, p. 2039 (1990).
  17. R. A. Anderson, G. K. Essick and R. M. Siegel. Encoding of spatial location by posterior parietal neurons. *Science*, **230**, pp. 456-458 (1985).
  18. M. Gentilucci, L. Fogassi, G. Luppino, M. Matelli, R. Camarda and G. Rizzolatti. Functional organization of inferior area 6 in the macaque monkey. *Experimental Brain Research*, **71**, pp. 475-490 (1988).
  19. M. A. Arbib and B. Hoff. Trends in neural modeling for reach grasp. *Insights into the Reach to Grasp Movement*, K. M. B. Bennett, and U. Castiello ed., North-Holland, pp. 311-344 (1994).
  20. K. Kurata. Role of cortical motor areas in voluntary movements. *Brain and Nerve*, **47**, 12, pp. 1135-1142 (1995).
  21. E. V. Everts, Y. Shinoda and P. Wise. *Neurophysical Approaches to Higher Brain Functions*. New York, Wiley (1984).
  22. D. Bullock and S. Grossberg. Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, **95**, pp. 49-90 (1988).
  23. E. Oyama and S. Tachi. A study of human hand position control learning—output feedback inverse model. *Proc. 1991 International Joint Conference on Neural Networks (IJCNN'91 Singapore)*, pp. 1434-1443 (1991).
  24. E. Oyama and S. Tachi. A learning method for solving inverse problems of static systems. *Proc. 1993 International Joint Conference on Neural Networks (IJCNN'93 Nagoya)*, pp. 2843-2849 (1993).
  25. D. E. Whitney. The mathematics of coordinated control of prosthetic arms and manipulators. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, **93**, pp. 303-309 (1972).
  26. J. S. Albus. Data storage in the cerebellar model articulation controller (CMAC). *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, **97**, pp. 228-233 (1975).
  27. S. I. Colombano, M. Compton and M. Baulat. Goal Directed Model Inversion. *Proc. of International Joint Conference on Neural Networks (IJCNN'91 Singapore)*, pp. 2422-2427 (1991).

## APPENDIX

### Appendix 1. Off-line Learning

From Eqs. (23) and (25),

$$\begin{aligned} \Delta\theta(\Delta x, u) &= J^+(\theta)\Delta x \\ &+ (I - J^+(\theta)J(\theta))(G(\theta) + u) \end{aligned} \quad (47)$$

is obtained. Assuming that  $\text{rank}(J(\theta))$  is  $n$ ,  $\text{rank}(I - J^+(\theta)J(\theta))$  is  $m - n$ . Let  $u^*$  be an  $m \times 1$  vector  $n$ , components of which are appropriately selected and set 0. For a given  $u$ ,  $u^*$  that satisfies the following equation exists:

$$\begin{aligned} (I - J^+(\theta)J(\theta))(G(\theta)\Delta x + u) \\ = (I - J^+(\theta)J(\theta))u^* \end{aligned} \quad (48)$$

We can assume the following equations with generality:

$$u^* = (u'_1, u'_2, \dots, u'_{m-n}, 0, \dots, 0)^T \quad (49)$$

$$u' = (u'_1, u'_2, u'_3, \dots, u'_{m-n})^T \quad (50)$$

The correspondence between  $u'$  and  $\Delta\theta(\Delta x, u)$  is one-to-one. Hereafter,  $J^+(\theta)$  is designated as  $J^+$ . Because Eq. (28) contains a scaling term, the volume element  $dv$  can be identified with  $du'$ . Therefore,

$$\begin{aligned} \Phi(\theta, \Delta x) &\approx E(\Delta\theta^*(\Delta x, u)) \\ &= \frac{\int_U p(\Delta\theta^*(\Delta x, u)) \Delta\theta^*(\Delta x, u) du'}{\int_U p(\Delta\theta^*(\Delta x, u)) du'} \\ &= J^+ \Delta x + \frac{\int_U p(\Delta\theta^*)(I - J^+J)u^* du'}{\int_U p(\Delta\theta^*(\Delta x, u)) du'} \end{aligned} \quad (51)$$

The region of the integration  $U$  in the forementioned equation is the whole space of  $(m - n) \times 1$  vector  $u'$  expressed as

$$\int_U h(u') du' = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h(u') du'_1 \dots du'_{m-n} \quad (52)$$

Consider the distribution of  $\Delta\theta$  to be spherical symmetry. In this case, the probability density function  $p(\Delta\theta)$  is the function of the square of the norm of  $\Delta\theta$  as

$$p(\Delta\theta) = g(\Delta\theta^T \Delta\theta) \quad (53)$$

Let  $B$  be a  $m \times m$  matrix defined as

$$B = I - J^+J \quad (54)$$

Since

$$J^{+T} B = 0 \quad (55)$$

is established,

$$p(\Delta\theta^*) = g(\Delta x J^{+T} J^+ \Delta x + u^{*T} B^T B u^*) \quad (56)$$

is obtained. Because  $p(\Delta\theta^*)$  is an even function of each element of  $u'$ , and  $Bu^*$  is an odd function of each element of  $u'$ , the integral of the product of  $p(\Delta\theta^*)$  and  $Bu^*$  is 0. Consequently,

$$\Phi(\theta, \Delta x) \approx E(\Delta\theta^*) = J^+ \Delta x \quad (57)$$

is obtained. The learning element calculates the minimum norm solution of Eq. (13).

Consider that  $p(\Delta\theta)$  can be expressed as Eq. (32). Because  $P$  is a positive symmetric matrix, there exists an appropriate  $R$  that satisfies

$$P = RR^T \quad (58)$$

By the transformation of the variables as:

$$\begin{aligned} \Delta\theta' &= R^{-1} \Delta\theta \\ J' &= JR \\ \Delta\theta'^* &= J'^T \Delta x + (I - J'^+ J') u \\ J'^+ &= J'^T (J' J'^T)^{-1}, \end{aligned} \quad (59)$$

the same calculation as in the preceding equations can be used. Consequently,

$$\begin{aligned} \Phi(\theta, \Delta x) &\approx E(\Delta\theta^*) \\ &= RE(\Delta\theta'^*) \\ &= RJ'^T (J' J'^T)^{-1} \Delta x \end{aligned}$$

$$= PJ^T(JPJ^T)^{-1} \Delta x \quad (60)$$

is obtained.

## Appendix 2. Coordinate Transformation System Paying Attention to Velocity and Acceleration

Coordinate transformation systems can be obtained by paying attention to the relationship between the joint angles velocity and the hand position velocity. Let  $\Phi(\theta, \dot{x})$  be the output of neural networks, the output of which is the velocity command of the joint angles and  $\Phi'(\theta, \dot{x})$  be the teaching signal of  $\Phi(\theta, \dot{x})$ . We propose the learning method of  $\Phi(\theta, \dot{x})$  as follows:

$$\Phi'(\theta, \dot{x}) = \dot{\theta} \quad (61)$$

As the coordinate transformation system based on the change of the joint angle vector described in section 3.1,

$$\begin{aligned} \dot{\theta} &= \Phi(\theta, \dot{x}) \\ &\approx E(\dot{\theta}^*) = J^*(\theta)\dot{x} \end{aligned} \quad (62)$$

is obtained.  $\Phi(\theta, \dot{x})$  is the inverse model of the system described in Eq. (19). By inputting the desired hand velocity, including the position error feedback

$$\dot{x}'_d = \dot{x}_d + K(x_d - x) \quad (63)$$

to  $\Phi(\theta, \dot{x})$ , the velocity command of the joint angles that makes the hand tracking the desired hand position can be calculated.

As the coordinate transformation system calculating the velocity command of the joint angles, the coordinate transformation system  $\Phi(\theta, \dot{\theta}, \ddot{x})$  calculating the acceleration command of the joint angles can be obtained by the teaching signal defined as

$$\Phi'(\theta, \dot{\theta}, \ddot{x}) = \ddot{\theta} \quad (64)$$

$\ddot{\theta}^*$  that satisfies Eq. (20) can be expressed as

$$\ddot{\theta}^* = J^*(\theta)(\ddot{x} - [\frac{\partial J(\theta)}{\partial \theta} \dot{\theta}] \dot{\theta}) + (I - J^+(\theta)J(\theta))u \quad (65)$$

According to the learning shown in Eq. (64),

$$\Phi(\theta, \dot{\theta}, \ddot{x}) \approx E(\ddot{\theta}^*) = J^*(\theta)(\ddot{x} - [\frac{\partial J(\theta)}{\partial \theta} \dot{\theta}] \dot{\theta}) \quad (66)$$

is obtained.  $\Phi(\theta, \dot{\theta}, \ddot{x})$  is the inverse model of the system described in Eq. (20) that can transform the acceleration of the hand to the acceleration of the joint angles.

By inputting the desired hand acceleration including the position and velocity error feedback

$$\ddot{x}'_d = \ddot{x}_d + K_D(\dot{x}_d - \dot{x}) + K(x_d - x) \quad (67)$$

to  $\Phi(\theta, \dot{\theta}, \ddot{x})$ , the acceleration command of the joint angles that makes the hand tracking the desired hand position can be calculated.  $K_D$  is an appropriate velocity feedback gain.

## AUTHORS



**Eimei Oyama** received his B.E. degree in 1985 and his M.E. degree in 1987 in Aeronautics from the University of Tokyo. He joined MEL (Mechanical Engineering Laboratory), AIST, MITI, Japan, in 1987 and was a Research Associate in the Man-Machine System Division, Robotics Department from 1987 to 1994. He has been a Senior Researcher at Bio-Robotics Division since 1994. His research interests include human motion control system, neuroscience, computer vision, nonlinear optimization, man-machine interface and teleexistence. He is a member of IEICE (the Institute of Electronics, Information and Communication Engineers); RSJ (Robotics Society Japan); SICE (the Society of Instrument and Control Engineers); and JSME (the Japan Society of Mechanical Engineers).

## AUTHORS (continued, from left to right)



**Taro Maeda** received his B.E. degree in 1987 and Dr. of Eng. degree in 1994 in Mathematical Engineering and Information Physics from the University of Tokyo. He joined Mechanical Engineering Laboratory, AIST in 1987 and was a Research Associate in the Man-Machine System Division, Robotics Department from 1987 to 1992. He was a Research Associate at RCAST (Research Center of Advanced Science and Technology), the University of Tokyo from 1992 to 1994. He has been a Research Associate at the University of Tokyo since 1994. His research interests include human perception system, human motion control system, neuroscience, neural computation, man-machine interface, and teleexistence. He is a member of IEICE; RSJ; SICE; JSME; JNNS (Japanese Neural Network Society).

**Susumu Tachi** received his B.E. degree in 1968 and his Dr. of Eng. degree in 1973 in Mathematical Engineering and Information Physics from the University of Tokyo. He was a Research Associate at the University of Tokyo; a Researcher at Mechanical Engineering Laboratory, AIST, a Senior Researcher, Director of Man-Machine System Division, Director of Bio-Robotics Division and an Assistant Professor at the University of Tokyo. He has been a Professor at the University of Tokyo since 1992. He was a Visiting Researcher at MIT from 1979 to 1980. His research interests include signal processing using spectrum analysis, guide dog robot, virtual reality and teleexistence. He is a member of RSJ; SICE; JSME; IEEE; and IMECO. He was the chair of IMECO TC 17 (Robotics) and is an SICE Fellow.