International Journal of Advanced Robotic Systems

OPEN ACCESS
ARTICLE

An Adaptive Neural Network Learning-based Solution for the Inverse Kinematics of Humanoid Fingers

Regular Paper

Byoung-Ho Kim[1,*]

1 Biomimetics and Intelligent Robotics Lab., Dept. of Mechatronics Eng., Kyungsung University, Busan, Korea
* Corresponding author E-mail: kimbh@ks.ac.kr

Received 27 May 2013; Accepted 28 Nov 2013

DOI: 10.5772/57472

Abstract This paper presents an adaptive neural network learning-based solution for the inverse kinematics of humanoid fingers. For the purpose, we specify an effective finger model by considering the interphalangeal joint coordination inherent in human fingers. In order to find a proper joint combination for any fingertip trajectory, we propose an adaptive learning scheme by using a multi-layered neural network. It is interesting to use an adaptive learning rate algorithm that leads the neural network to get the inverse kinematic solution quickly. The usefulness of the proposed approach is verified by exemplary simulations for the general motion of humanoid fingers.

Keywords Adaptive Neural Network Learning, Inverse Kinematics, Humanoid Fingers

1. Introduction

In order to manipulate the fingers of a robotic hand, we need to know the combination of joints of each finger [1][2]. Actually, the joint configuration of a finger plays an important role for the dexterous manipulation of an object grasped by multiple fingers. Thus, a method is necessary to get a proper joint combination for the given fingertip trajectory.

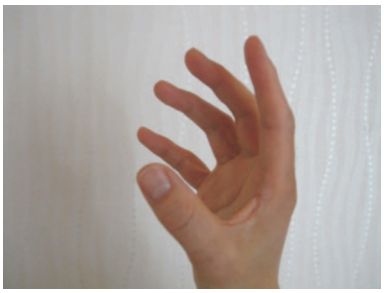
In practice, there exists a preferable configuration depending on the task in using a redundant manipulator or a finger with a coupling among joints. In fact it is not easy to obtain an effective joint configuration due to the redundancy or constraints. To solve this issue, some approaches have been proposed [3–5]. Yoshikawa [3] and Chiu [4] suggested a performance index-based algorithm using a manipulability criterion and a compatibility index, respectively, from the viewpoint of finding an effective posture of robot manipulators. These methods have an advantage with regard to resolving the singularity posture of a manipulator as well as avoiding obstacles. However, unbalancing of the joint configuration during grasp can be obtained by optimizing such a performance index. Secco, *et al.* [5] tried to solve the inverse kinematic problem for a prosthetic finger by adopting a physiological constraint among joints. Secco's method gives a simple closed-form solution, but it has a limitation with regard to implementing the realistic movement of human fingers. This is because the third joint of a humanoid finger should actuate identically with the second joint through his approach. According to the literature, the motion range of the joints in human fingers is not identical [6–8]. Moreover, since the method does not consider the phalangeal length parameters, this may lead to difficulties in making a consistent grasp configuration

in multi-fingered operations. Thus, the inverse kinematic solution for effective positioning of such a humanoid finger or a manipulator in manipulation tasks is still an interesting problem [9].

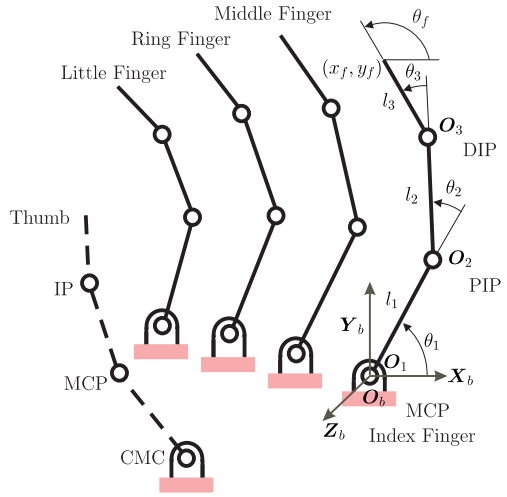
The objective of this paper is to provide an adaptive learning-based method to get the inverse kinematic solution of humanoid fingers with a coupling between the distal interphalangeal joint and the proximal interphalangeal joint. For this purpose, we use a multi-layered neural network learned at an adaptive learning rate. In Section 2, we specify a model of humanoid fingers based on the interphalangeal joint coordination of the human hand and reveal the issue of inverse kinematics for humanoid fingers. The adaptive neural network learning scheme for the inverse kinematics is described in Section 3. In Section 4, exemplary simulation results for the inverse kinematics of some humanoid fingers are shown and the usefulness of the proposed approach is also discussed. The concluding remarks are drawn in Section 5.

2. Modelling of Humanoid Fingers and Inverse Kinematics Issue

It is important to study the features of the human hand in order to develop a dexterous humanoid hand [10]. For a model of humanoid fingers, we consider the interphalangeal features of the human hand as shown in Fig. 1.



(a) The human hand



(b) A model of the human hand

Figure 1. Human hand and its structural model

In Fig. 1(a), for instance, the index finger is usually actuated by four joints, but its planar motion can be implemented by the combination of three revolute joints as shown in Fig. 1(b) [11]. The mechanism of the index finger is structurally the same as that of the middle, ring and little fingers. So, they can be modelled as a finger with three revolute joints in two-dimensional space. The thumb however differs from those fingers and it is not taken into consideration in this paper.

In particular, one of the interesting features inherent in human fingers is that interphalangeal joint coordination exists between the DIP (Distal InterPhalangeal) joint and the PIP(Proximal InterPhalangeal) joint of the human fingers except the thumb [7][8][12]. Thus, the motion of the third joint depends on the actuation of the second joint. Practically, each link $l_i (i = 1, 2, 3)$ in Fig. 1(b) corresponds to the proximal, middle and distal phalanges in the index finger, respectively, and the origin of the MCP(MetaCarpoPhalangeal) joint can be considered as the pivot connecting to the palm. In the previous research [12], we tried to find the interphalangeal joint coordination of the index, middle, ring and little fingers of the author. It has been confirmed that the joint coordination between the DIP joint and the PIP joint of those fingers can be modelled approximately as a linear relation as follows:

$$\theta_{3i} \approx \lambda_i \theta_{2i},\ i = 1, \cdots, 4 \tag{1}$$

where the phalangeal length parameters of the ith finger used in the experiment and the corresponding interphalangeal joint coordination parameter λ_i can be summarized as in Table 1.

i	λ_i	Length(m)			Remarks
		l_1	l_2	l_3	
1	0.6175	0.050	0.030	0.025	Index finger
2	0.4715	0.058	0.035	0.028	Middle finger
3	0.4390	0.055	0.032	0.025	Ring finger
4	0.4143	0.045	0.025	0.022	Little finger

Table 1. Phalangeal length and interphalangeal joint coordination parameters

By considering the interphalangeal joint coordination, the forward kinematic relations of the representative index finger in Fig. 1(b) can be described by

$$x_f = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + (1 + \lambda_1)\theta_2) \tag{2}$$

$$y_f = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + (1 + \lambda_1)\theta_2) \tag{3}$$

$$\theta_3 = \lambda_1 \theta_2 \tag{4}$$

$$\theta_f = \theta_1 + (1 + \lambda_1)\theta_2 \tag{5}$$

where x_f and y_f denote the x- and y-directional fingertip positions of the index finger, respectively, and λ_1 implies the coupling between the DIP joint and the PIP joint of the index finger. The parameter of l_j represents the length parameter of the jth link of the finger and θ_f represents its posture.

From (2) $\sim$ (5), it is natural that the fingertip position and its posture are definitely determined when the joint angles of the finger have been given. It is worth noting that reverse work is usually necessary in the object handling tasks by multi-fingered hands. That is, it is required to obtain the joint combination corresponding to the fingertip trajectory of each finger for the manipulating tasks. This is actually called the inverse kinematics problem which is the fundamental issue for general hand operations. For instance, the assembling performance of a stick manipulated by multiple fingers is basically dependant on the accuracy of the inverse kinematic solution of each finger. If such a finger has a coupling among joints, its inverse kinematic problem is usually not easily solved in a closed-form.

On the other hand, it is well-known that a multi-layered neural network enables us to get an effective solution for various identification and control tasks using a learning strategy [13–16].

Thus, this paper aims to provide an adaptive solution of the inverse kinematics of such a humanoid finger by utilizing the advantages of the neural network approach.

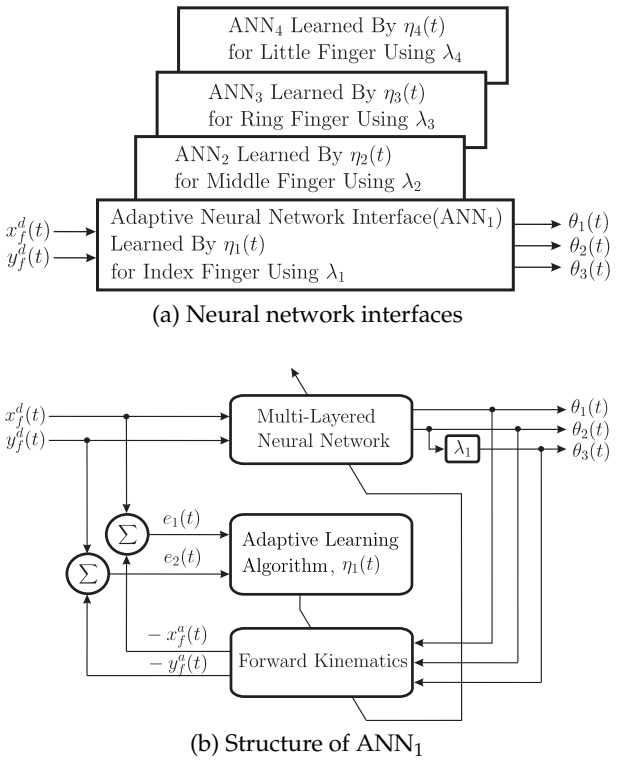
3. Neural Network Learning Scheme for Inverse Kinematics

This section describes a multiple neural network learning scheme for the inverse kinematic solutions of the humanoid fingers in Fig. 1(b). An adaptive learning algorithm to initialize and update the learning rate is also introduced for the neural network learning scheme.

3.1. Multiple Neural Network Scheme

Fig. 2(a) shows the multiple neural network learning scheme proposed for the inverse kinematics of the humanoid fingers. The structure of the adaptive neural network interface $\text{ANN}_i(i = 1, \cdots, 4)$ for the actual joint combination corresponding to the desired fingertip trajectory of each finger is shown in Fig. 2(b). In addition, the multi-layered neural network has been constructed as in Fig. 3. In fact, the positions, x_f^d and y_f^d, necessary in the fingertip space of each finger are assigned in the input port of each interface network. The output signals of the network are the actual joint angles, θ_1 and θ_2. Then the third joint θ_3 is determined by considering the interphalangeal joint coordination as shown in Fig. 3. The actual fingertip positions of each finger are determined by the forward kinematics using the resultant joint angles obtained by the learning scheme. By using the position error of each direction, the adaptive learning algorithm initializes a proper learning rate and updates it for the fast learning of the neural network. Thus, the neural network is recursively learned using the conventional error back propagation method. If the maximum position error assigned in the fingertip space is satisfied, the learning process is terminated, and the corresponding output of the learning scheme is finally accepted as the inverse kinematic solution of each finger.

The overall signal processing of the neural network in Fig. 3 is described as follows. The input of the neural network



(a) Neural network interfaces

(b) Structure of ANN_1

Figure 2. A neural network learning scheme for multiple inverse kinematic solutions

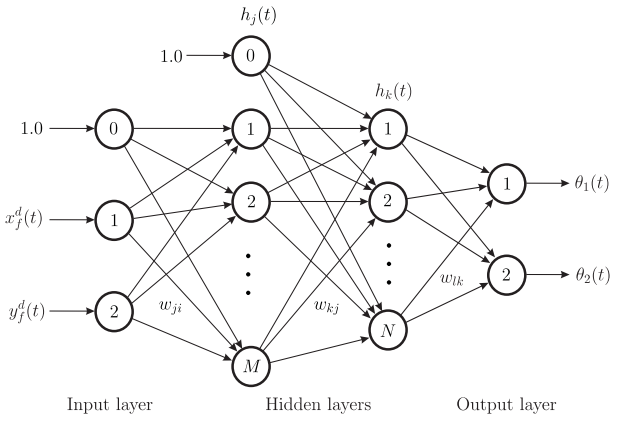


Figure 3. A multi-layered neural network

$z_i(t)$ is assigned by

$$z_i(t) = \begin{cases} 1.0, & i = 0 \\ x_f^d(t), & i = 1 \\ y_f^d(t), & i = 2 \end{cases} \tag{6}$$

where $x_f^d(t)$ and $y_f^d(t)$ denote the x- and y-directional positions necessary in the fingertip space, respectively. The term $z_0(t)$ represents a bias input that affects the output function of the first hidden layer. The function of the input layer is only to pass the input to the first hidden layer.

The output of the first hidden layer $h_j(t)$ is determined by

$$h_j(t) = \begin{cases} 1.0, & j = 0 \\ \frac{1}{1+e^{-s_j(I_j(t))}}, & j = 1, 2, \cdots, M \end{cases} \tag{7}$$

where

$$I_j(t) = \sum_{i=0}^{2} w_{ji}(t) z_i(t) \tag{8}$$

here $w_{ji}(t)$ denotes the weighting factor between the i-th input layer and the j-th first hidden layer. The parameter s_j implies the slope of the j-th sigmoid function at the first hidden layer and M indicates the number of neurons placed at the first hidden layer.

The output of the second hidden layer $h_k(t)$ is made by

$$h_k(t) = \frac{1}{1 + e^{-s_k(I_k(t))}},\ k = 1, 2, ..., N \tag{9}$$

where

$$I_k(t) = \sum_{j=0}^{M} w_{kj}(t) h_j(t) \tag{10}$$

here $w_{kj}(t)$ denotes the weighting factor between the j-th first hidden layer and the k-th second hidden layer. The parameter s_k implies the slope of the k-th sigmoid function at the second hidden layer.

The final output of the neural network $\theta_l(t)$ is determined by

$$\theta_l(t) = \begin{cases} \sum_{k=1}^{N} w_{lk}(t) h_k(t),\ l = 1 \\ \left(\sum_{k=1}^{N} w_{lk}(t) h_k(t) \right)^2,\ l = 2 \end{cases} \tag{11}$$

where $w_{lk}(t)$ denotes the weighting factor between the second hidden layer and the output layer. In particular, the second joint angle of each finger has been made to be at least positive and thus the third joint also has a positive angle. This is done in order to mimic the structural features observed in the general motion of the human fingers and it is easily experienced in real finger motions.

3.2. Adaptive Learning Algorithm

For the learning of the neural network in Fig. 3, we define an error function as follows:

$$J(t) = \frac{1}{2} \sum_{m=1}^{2} (e_m(t))^2 \tag{12}$$

where

$$e_1(t) = x_f^d(t) - x_f^a(t) \tag{13}$$

$$e_2(t) = y_f^d(t) - y_f^a(t). \tag{14}$$

In fact, the error function implies the sum of each position error at the fingertip space. For effective description of the learning algorithm, the case of the representative index finger has been considered in this section.

Through the conventional error back propagation [19], the error effect according to the change of the weighting factor between the second hidden layer and the output layer $\frac{\partial J(t)}{\partial w_{lk}(t)}$ is expressed by

$$\frac{\partial J(t)}{\partial w_{lk}(t)} = -e_1(t) \frac{\partial x_f^a(t)}{\partial w_{lk}(t)} - e_2(t) \frac{\partial y_f^a(t)}{\partial w_{lk}(t)}, \quad l = 1, 2, \text{and } k = 1, 2, \cdots, N \tag{15}$$

where $\frac{\partial x_f^a(t)}{\partial w_{lk}(t)}$ and $\frac{\partial y_f^a(t)}{\partial w_{lk}(t)}$ imply the x- and y-directional error effects by the change of the weight w_{lk}. In order to compute those terms, it is necessary for each angle term to be considered separately. When $l = 1$ in (15), for the first joint angle of the finger,

$$\frac{\partial x_f^a(t)}{\partial w_{lk}(t)} = \left(\sum_{n=1}^{3} a_n \right) h_k(t),\ k = 1, 2, \cdots, N \tag{16}$$

where

$$\begin{aligned} a_1 &= -l_1 \sin\theta_1(t) \\ a_2 &= -l_2 \sin(\theta_1(t) + \theta_2(t)) \\ a_3 &= -l_3 \sin(\theta_1(t) + (1 + \lambda_1)\theta_2(t)). \end{aligned}$$

And also

$$\frac{\partial y_f^a(t)}{\partial w_{lk}(t)} = \left(\sum_{n=1}^{3} b_n \right) h_k(t),\ k = 1, 2, \cdots, N \tag{17}$$

where

$$\begin{aligned} b_1 &= l_1 \cos\theta_1(t) \\ b_2 &= l_2 \cos(\theta_1(t) + \theta_2(t)) \\ b_3 &= l_3 \cos(\theta_1(t) + (1 + \lambda_1)\theta_2(t)). \end{aligned}$$

When $l = 2$, for the second joint angle of the finger,

$$\frac{\partial x_f^a(t)}{\partial w_{lk}(t)} = \left(\sum_{n=1}^{2} c_n \right) h_k(t),\ k = 1, 2, \cdots, N \tag{18}$$

where

$$\begin{aligned} c_1 &= -l_2 \sin(\theta_1(t) + \theta_2(t)) \left\{ 2 \left(\sum_{k=1}^{N} w_{lk}(t) h_k(t) \right) \right\} \\ c_2 &= -l_3 \sin(\theta_1(t) + (1 + \lambda_1)\theta_2(t)) \\ &\quad \cdot \left\{ 2(1 + \lambda_1) \left(\sum_{k=1}^{N} w_{lk}(t) h_k(t) \right) \right\}. \end{aligned}$$

And

$$\frac{\partial y_f^a(t)}{\partial w_{lk}(t)} = \left(\sum_{n=1}^{2} d_n \right) h_k(t),\ k = 1, 2, \cdots, N \tag{19}$$

where

$$\begin{aligned} d_1 &= l_2 \cos(\theta_1(t) + \theta_2(t)) \left\{ 2 \left(\sum_{k=1}^{N} w_{lk}(t) h_k(t) \right) \right\} \\ d_2 &= l_3 \cos(\theta_1(t) + (1 + \lambda_1)\theta_2(t)) \\ &\quad \cdot \left\{ 2(1 + \lambda_1) \left(\sum_{k=1}^{N} w_{lk}(t) h_k(t) \right) \right\}. \end{aligned}$$

By defining an output-layer error term $\delta_{ol}(t)$ for effective description, we have

$$\delta_{ol}(t) = \begin{cases} e_1(t) \sum_{n=1}^{3} a_n + e_2(t) \sum_{n=1}^{3} b_n,\ l = 1 \\ e_1(t) \sum_{n=1}^{2} c_n + e_2(t) \sum_{n=1}^{2} d_n,\ l = 2, \end{cases} \tag{20}$$

and the second hidden-layer error term at the k-th neuron $\delta_{h2k}(t)$ can be represented by

$$\delta_{h2k}(t) = \sum_{l=1}^{2} w_{lk}(t) \delta_{ol}(t) \frac{\partial h_k(t)}{\partial I_k(t)},\ k = 1, 2, \cdots, N. \tag{21}$$

In addition, the first hidden-layer error term at the j-th neuron $\delta_{h1j}(t)$ can be expressed by

$$\delta_{h1j}(t) = \sum_{k=1}^{N} w_{kj}(t)\delta_{h2k}(t)\frac{\partial h_j(t)}{\partial I_j(t)},\ j = 0, 1, \cdots, M. \quad (22)$$

Finally, all of the weighting factors at each layer for the index finger can be updated using the following rule:

$$w_{lk}(t+1) = w_{lk}(t) + \eta_1(t)\delta_{ol}(t)h_k(t) \quad (23)$$
$$w_{kj}(t+1) = w_{kj}(t) + \eta_1(t)\delta_{h2k}(t)h_j(t) \quad (24)$$
$$w_{ji}(t+1) = w_{ji}(t) + \eta_1(t)\delta_{h1j}(t)z_i(t) \quad (25)$$

where $\delta_{ol}(t)$, $\delta_{h2k}(t)$ and $\delta_{h1j}(t)$ denote the error terms propagated back from the fingertip space of the finger to the output and hidden layers. The learning rate $\eta_1(t)$ is determined dynamically according to the state of learning of the neural network, and the details of the procedure have been described in Section 7.1.

In addition, the same procedure considering the corresponding finger's coupling parameter is available for the middle, ring and little fingers.

4. Simulation Results: Inverse Kinematics

This section shows some representative simulation results for the inverse kinematics of the humanoid fingers in Fig. 1(b) by using the proposed neural network learning scheme. In particular, the simulation results for the index and middle fingers have been shown.

The specifications of the humanoid fingers for the simulation study have been specified in Table 1. Some test positions on the following curve according to the learning state of the neural network have been assigned for the desired fingertip trajectories in (6):

$$y_f = a_2 x_f^2 + a_1 x_f + a_0 \quad (26)$$

where a_0, a_1 and a_2 denote the coefficients of a second-order curve. Those parameters are specified in Table 2. The assigned trajectory implies that the index and middle fingers move along such a second-order function in a free grasp motion [12].

Finger	a_2	a_1	a_0
Index	-13.5273	0.8612	0.0622
Middle	-9.7340	0.5567	0.0806

Table 2. Parameters for the functions of the index and middle fingers

The multi-layered neural network used in this simulation has four layers as shown in Fig. 3. The number of neurons at the input layer, the first hidden layer, the second hidden layer, and the output layer has been empirically assigned by 2, 5, 3 and 2, respectively. All of the weights of the neural network have been initialized randomly in the range of -1.0 $\sim$ 1.0. The parameters for the adaptive learning algorithm, β, σ, P and Q, are assigned by 0.0015, 0.5, 3 and 5, respectively. The slope parameters of the sigmoid functions at the hidden layers have been

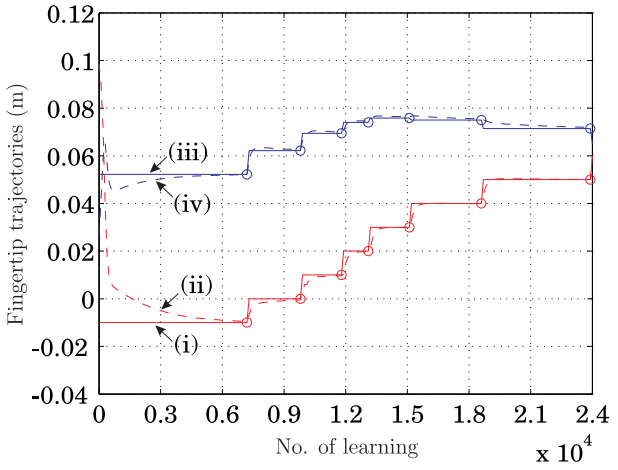


Figure 4. Fingertip trajectories of the index finger: (i) desired x-directional trajectory(x_f^d), (ii) actual x-directional trajectory followed by the proposed neural network learning(x_f^a), (iii) desired y-directional trajectory(y_f^d), and (iv) actual y-directional trajectory followed by the neural network learning(y_f^a). Note that the inverse kinematics process for each trajectory has been completed at the moment of each circle.

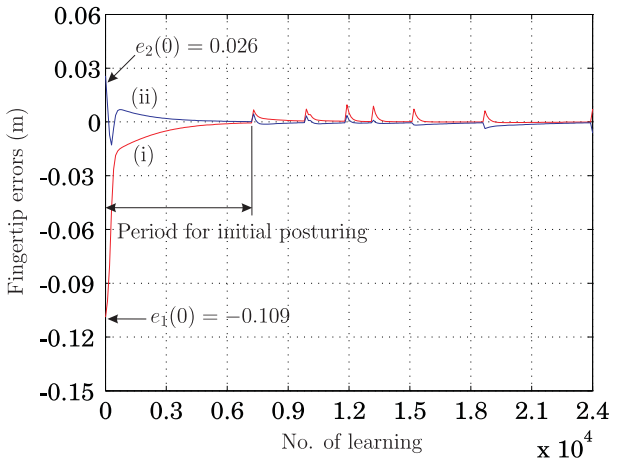


Figure 5. Fingertip error profiles of the index finger according to the learning process: (i) x-directional error, $e_1(t)$, and (ii) y-directional error, $e_2(t)$

assigned by 1.0. In fact, the desired accuracy of the given manipulation task is dependent on the range of the fingertip position error. In this simulation, if the maximum error of $e_1(t)$ and $e_2(t)$ in Fig. 2(b) is less than the desired range of each fingertip error, $\pm$0.5 mm, the inverse kinematic learning process of each trajectory has been completed and the current joint values have been determined as the inverse kinematic solution, and the inverse kinematic process is repeated for the next trajectory.

Fig. 4 and Fig. 8 show the x- and y-directional fingertip trajectories of the index and middle fingers, respectively, in the process of the neural network learning, where the desired fingertip trajectories have been assigned successively after getting the solution of the current inverse kinematics. The actual fingertip trajectories in Figs. 4 and 8 have been taken by the forward kinematics using the actual joint angles which are resultantly obtained by the neural network learning. Figs. 5 and

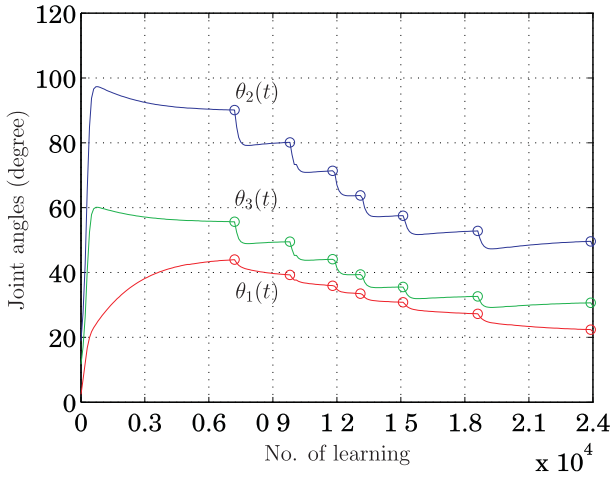


Figure 6. Joint angles of the index finger obtained by the proposed neural network learning-based inverse kinematics. Note that the joint angles at the moment of each circle have been accepted as the solution of the corresponding inverse kinematics.

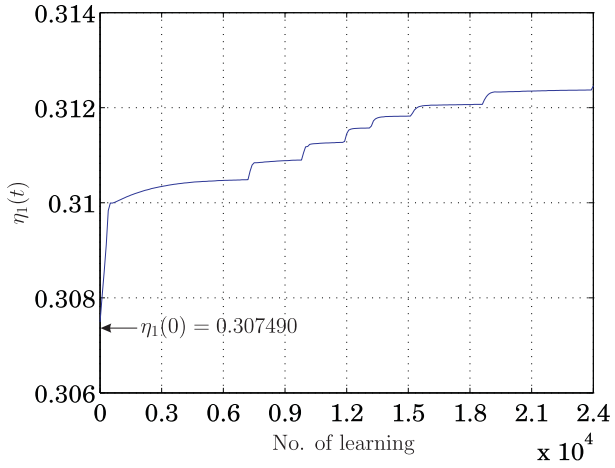


Figure 7. The learning rate adapted for the inverse kinematics of the index finger

9 show the fingertip trajectory errors according to the learning process, where we can confirm the satisfactory convergence of the neural network training algorithm under variable learning rate. In fact, the process of inverse kinematics is completed at the moment of each circle indicated in Figs. 6 and 10, and then the output angles of the neural network learning scheme have been set as the actual joint angles. Those actual joint angles for the desired fingertip positions have been plotted in Fig. 6 and Fig. 10, respectively. As a result, we can find that the actual fingertip positions have been approached satisfactorily to the desired positions through the adaptive learning process.

In particular, it should be noted from Figs. 4, 5, 6, 8, 9, and 10 that a rather long time is required for the learning of the initial position. The period actually means the initial posturing process of each finger and thus it is practically not related to the computing load in the manipulation process. Nevertheless, an effort to reduce the learning time is very important in the implementation aspect of neural networks [13–18]. So, it is remarkable that

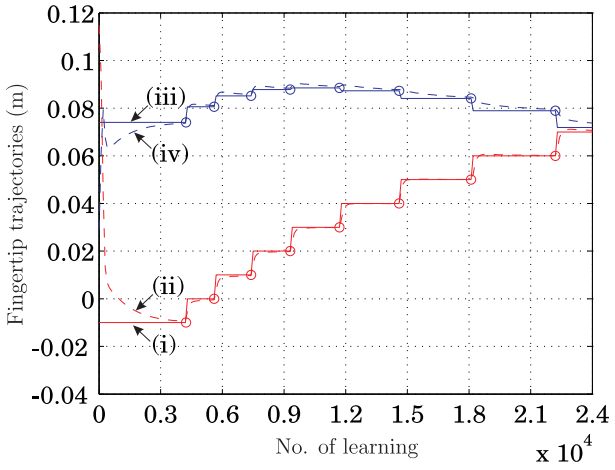


Figure 8. Fingertip trajectories of the middle finger: (i) desired x-directional trajectory(x_f^d), (ii) actual x-directional trajectory followed by the proposed neural network learning(x_f^a), (iii) desired y-directional trajectory(y_f^d), and (iv) actual y-directional trajectory followed by the neural network learning(y_f^a). Note that the inverse kinematics process for each trajectory has been completed at the moment of each circle.

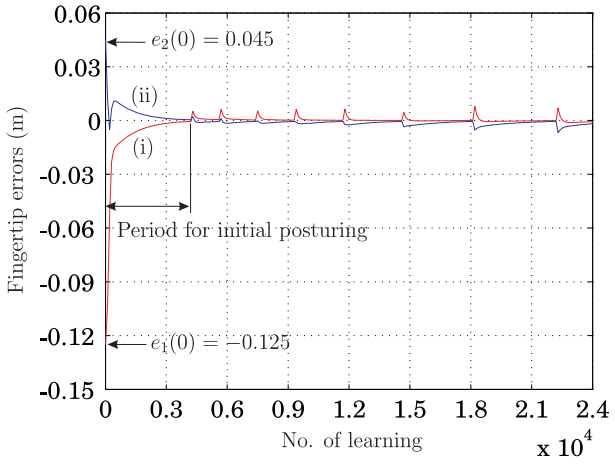


Figure 9. Fingertip error profiles of the middle finger according to the learning process: (i) x-directional error, $e_1(t)$, and (ii) y-directional error, $e_2(t)$

the adaptive learning rate used in this paper can contribute to reducing the learning time. The trace of the learning rate for the two fingers during the learning process is shown in Fig. 7 and Fig. 11, respectively. As you can see in Figs. 7 and 11, the learning rate for each finger has been initialized differently and the trace of updating is also different. This is because each learning rate is adjusted by the state of learning of the corresponding neural network. Indeed one can see via the results shown in Figs. 4 and 8 that those adaptive learning rates contribute to improving the speed of learning of the corresponding neural network.

In addition, the singularity issue of the humanoid fingers with regard to obtaining the inverse kinematic solution has been analysed in Section 7.2. The inverse kinematic solutions of the ring and little fingers can also be obtained from the same procedure considering the corresponding finger's coupling parameter.

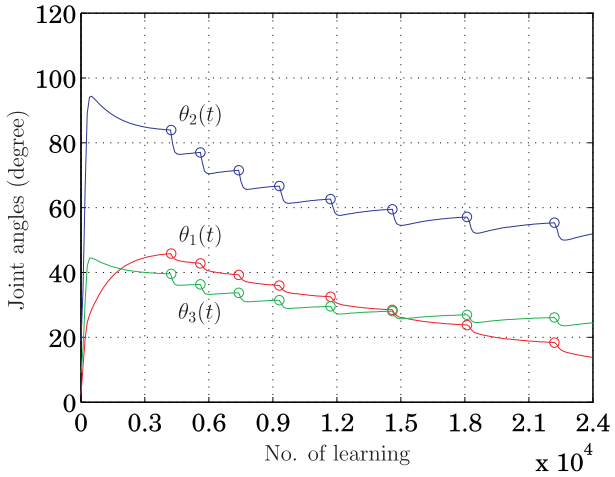


Figure 10. Joint angles of the middle finger obtained by the proposed approach of inverse kinematics. Note that the joint angles at the moment of each circle have been accepted as the solution of the corresponding inverse kinematics.

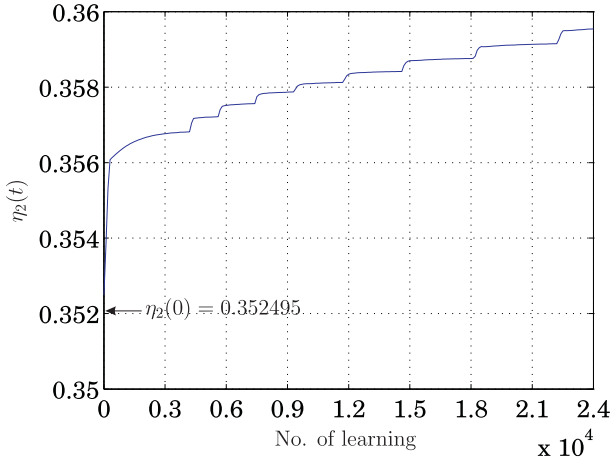


Figure 11. The learning rate adapted for the middle finger

As a result, it is concluded that the proposed neural network learning scheme is useful for the inverse kinematics of the humanoid fingers with a coupling. Hopefully, it is expected that the proposed approach can be applied to the effective motion control of humanoid fingers, prosthetic hands, and manipulators with such an interphalangeal coupling [20][21].

5. Concluding Remarks

An adaptive neural network learning-based solution for the inverse kinematics of the humanoid fingers with a coupling has been presented. In order to verify the usefulness of the adaptive neural network learning scheme, we utilized an effective model of the human fingers and performed exemplary simulations for the inverse kinematics of the index and middle fingers, where a four-layered neural network has been employed. Through the simulation study, it has been shown that the inverse kinematics of the humanoid fingers can be solved effectively by using the proposed neural network learning scheme, and its performance is satisfactory. In addition, the adaptive learning rate algorithm is practically useful for improving the learning speed of the neural network. Finally, it is concluded that the adaptive neural network learning scheme is applicable for manipulation tasks by humanoid robotic or prosthetic fingers. In our future work we intend to compare and improve the computational efficiency which is desirable for the practical applications of the proposed approach.

6. Acknowledgements

This research was supported by Kyungsung University Research Grants in 2013.

7. Appendix

7.1. Adaptive Learning Rate

In the field of applying a multi-layered neural network, a proper learning rate is used to update the weighting factors of the neural network, but in fact this is not easy to assign properly. So, a fixed small value has been utilized in many cases. In this case, the speed of learning of the neural network can be very slow. On the other hand, if a big learning rate is assigned in order to facilitate fast learning, the network's learning tends to be diverged. So, an adaptive learning rate is desirable for possible fast learning. In this sense, we studied an algorithm to initialize and change the learning rate for multi-layered neural networks as in the previous research [22].

In this paper, we employed the adaptive learning rate algorithm for the issue of the inverse kinematics of the humanoid fingers as shown in Fig. 1. The procedure to determine the learning rate has been summarized as follows. The initial learning rate $\eta_i(t)$ for the ANN$_i$ in Fig. 2 is firstly assigned by checking the maximum position error at the initialized state of the neural network:

$$\eta_i(t) = -\max|e_k(t)| \log_{10} \beta,\ k = 1, 2, \text{and } t = 0 \quad (27)$$

where $e_k(t)$ denotes the k-th error in Fig. 2 given by

$$e_1(t) = e_x(t)$$
$$e_2(t) = e_y(t).$$

The role of the parameter β $(0 < \beta < 1)$ adjusts the magnitude of the initial learning rate. This technique is originally based on simulated annealing [19].

Next, the initial learning rate is adjusted by checking the following steps repeatedly.

Step 1) Compute the sum of the absolute fingertip position errors:

$$S_e(r) = \sum_{k=1}^{2} |e_k(t)|, \quad (28)$$

$$\Delta_e = S_e(r+1) - S_e(r),\ r = 1, 2, \cdots, \mathrm{R} \quad (29)$$

where the initial error sum, $S_e(0) = 0$, and R denotes the number of learning.

Step 2) Check if the case of $\Delta_e \leq 0$ occurs P times continuously, and if the cases of $\Delta_e \leq 0$ and/or $\Delta_e > 0$ happen Q times irregularly. In these cases, select a random

value of ξ $(0 < \xi < 1)$ and then, if the following condition is satisfied

$$e^{-\frac{\eta_i(t)}{\Delta_e}} < \xi, \tag{30}$$

the learning rate for the next step is increased by

$$\eta_i(t+1) = (1+\alpha)\eta_i(t) \tag{31}$$

where

$$\alpha = \sigma \log_{10}(1+\Delta_e)$$

here σ also adjusts the learning rate minutely.

Or, check if the case of $\Delta_e \geq 0$ occurs P times continuously, and if the cases of $\Delta_e \geq 0$ and/or $\Delta_e < 0$ take place Q times irregularly. In these cases, also select a random value of ξ $(0 < \xi < 1)$ and then, if the following condition is satisfied

$$e^{\frac{\eta_i(t)}{\Delta_e}} < \xi, \tag{32}$$

the next learning rate is decreased by

$$\eta_i(t+1) = (1-\alpha)\eta_i(t). \tag{33}$$

Step 3) When the conditions in Step 2) do not occur, the current learning rate is used in the next learning process without any change.

7.2. Singularity Analysis

When the humanoid index finger shown in Fig. 1(b) is unfolded or folded maximally, its possible configurations can be drawn as in Fig. 12.

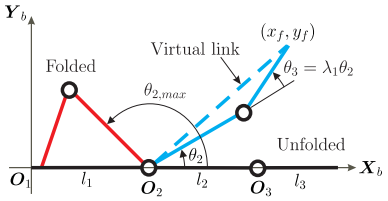


Figure 12. Possible configurations of the humanoid index finger

Basically, the posture of the humanoid finger can be formed within those possible configurations. The finger mechanism can be considered as a two-link mechanism with a virtual link because the motion of the third joint depends on the second joint. Therefore, when the fingertip trajectory is reasonably planned within the finger's workspace, its singular posture can be made structurally at the second joint angles, $\theta_2 = 0$ or $\theta_{2,max}$. The maximum angle of the second joint of the i-th humanoid finger can be obtained from the following auxiliary equation:

$$l_2 \sin(\theta_{2,max}) + l_3 \sin((1+\lambda_i)\theta_{2,max}) = 0.$$

In this paper, the maximum angles of the second joints of the index and middle fingers have been approximately founded as 141.5^o and 149.5^o, respectively. Since the actual trajectory of the second joint angle of each finger, as shown in Figs. 6 and 10, is satisfying the constraint range avoiding singularity, $0 < \theta_2 < \theta_{2,max}$, it is finally concluded that there is no problem in the singularity of the proposed inverse kinematic solution for the humanoid fingers.

8. References

[1] Pons J. L., Ceres R., Pfeiffer F (1999) Multifingered dexterous robotics hand design and control: a review. Robotica, 17: 661–674.

[2] Butterfass J., Grebenstein M., Liu H., Hirzinger G (2001) DLR-Hand II: Next generation of dextrous robot hand. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 109–114.

[3] Yoshikawa T (1984) Analysis and control of robot manipulators with redundancy. Robotics research: the first international symposium, Eds. M. Brady and R. Paul, Cambridge: MIT Press, pp. 735–747.

[4] Chiu S. L (1988) Task compatibility of manipulator postures. Int. Jour. of Robotics Research, 7(5): 13–21.

[5] Secco E. L., Visioli A., Magenes G (2004) Minimum jerk motion planning for a prosthetic finger. Jour. of Robotic Systems, 21(7): 361–368.

[6] Hahn P., Krimmer H., Hradetzky A., Lanz U (1995) Quantitative analysis of the linkage between the interphalangeal joints of the index finger. Jour. of Hand Surgery, 20B: 696–699.

[7] Nordin M., Frankel V. H (2001) Basic biomechanics of the musculoskeletal system. Lippincott Williams & Wilkins press, pp. 358–387.

[8] Kamper D. G., Cruz E. G., Siegel M. P (2003) Stereotypical fingertip trajectories during grasp. Jour. of Neurophysiology, 90: 3702–3710.

[9] Jiang L., Sun D., Liu H (2007) An inverse-kinematic table-based solution of a humanoid robot finger with nonlinearly coupled joints. IEEE/ASME Trans. on Mechatronics, 14(3): 273–281.

[10] Cutkosky M. R (1989) On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Trans. on Robotics and Automation, 5(3): 269–279.

[11] Iberall T (1997) Human prehension and dexterous robot hands. Int. Jour. of Robotics Research, 16(3): 285–299.

[12] Kim B.-H (2006) A bio-mimetic inter-articular coordination model for humanoid graspings. Proc. of IEEE Int. Conf. on Robotics and Mechatronics, pp. 211–216.

[13] Wojitara T., Nonami K (2004) Hand posture detection by neural network and grasp mapping for a master slave hand system. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 866–871.

[14] Afshar P., Mastuoka Y (2004) Neural-based control of a robotic hand: evidence for distinct muscle strategies. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 4633–4638.

[15] Kim S. S., Jung S (2004) Hardware implementation of a real time neural network controller with a DSP and an FPGA. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 4639–4644.

[16] Cong S., Liang Y (2009) PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. IEEE Trans. on Industrial Electronics, 56(10): 3872–3879.

[17] Choi W.-K., Ha S.-H., Kim S.-J., Kim Y.-T., Jeon H.-T (2006) The intelligent control system for biped robot using hierarchical mixture of experts. Jour. of Fuzzy Logic and Intelligent Systems, 16(4): 389–395.

[18] Ju J.-T., Kim D.-W., Sim K.-B (2006) Pattern classification algorithm of DNA chip image using ANN. Jour. of Fuzzy Logic and Intelligent Systems, 16(5): 556–561.
[19] Nielsen R. H (1990) Neurocomputing. Addison-Wesley, pp. 183–191.
[20] Kim B.-H (2006) A joint motion planning based on a bio-mimetic approach for human-like finger motion. Int. Jour. of Control, Automation, and Systems, 4(2): 217–226.
[21] Massa B., Roccella S., Carrozza M. C., Dario P (2002) Design and development of an underactuated prosthetic hand. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 3374–3379.
[22] Kim B.-H (2007) Characteristics modeling of dynamic systems using adaptive neural computation. Jour. of Control, Automation, and Systems, 13(4): 309–314.