ELSEVIER

Neurocomputing 38–40 (2001) 797–805

NEUROCOMPUTING

www.elsevier.com/locate/neucom

A modular neural network architecture for inverse kinematics model learning

Eimei Oyama[a,*], Arvin Agah[b], Karl F. MacDorman[c], Taro Maeda[d], Susumu Tachi[d]

[a]*Intelligent System Institute, National Institute of Advanced Industrial Science and Technology, 1-2 Namiki, Tsukuba Science City, Ibaraki 305-8564, Japan*

[b]*Department of Electrical Engineering and Computer Science., The University of Kansas, USA*

[c]*Artificial Intelligence Laboratory, Wakayama University, Japan and Filonet Corporation, Canada*

[d]*School of Engineering, The University of Tokyo, Japan*

Abstract

In order to reach an object, we need to solve the inverse kinematics problem, i.e., the coordinate transformation from the visual coordinates to the joint angle vector of the arm. The learning of the inverse kinematics model for calculating every joint angle that would result in a specific hand position is important. However, the inverse kinematics function of the human arm is a multi-valued and discontinuous function. It is difficult for a well-known continuous neural network to approximate such a function. In order to overcome the discontinuity of the inverse kinematics function, a novel modular neural network architecture is proposed in this paper. © 2001 Published by Elsevier Science B.V.

Keywords: Inverse kinematics learning; Discontinuity of inverse kinematics; Modular neural network; Online learning

1. Introduction

In order to reach an object, we need to solve the inverse kinematics problem, i.e., the coordinate transformation from the visual coordinates to the joint angle or muscle length vector coordinates of the arm. The inverse kinematics problem refers to the process of calculating all the joint angles of a robotic arm that would result in

* Corresponding author. Tel.: + 81-298-61-7298; fax: + 81-298-61-7201.
E-mail address: eimei.oyama@aist.go.jp (E. Oyama).

0925-2312/01/$ - see front matter © 2001 Published by Elsevier Science B.V.
PII: S0925-2312(01)00416-7

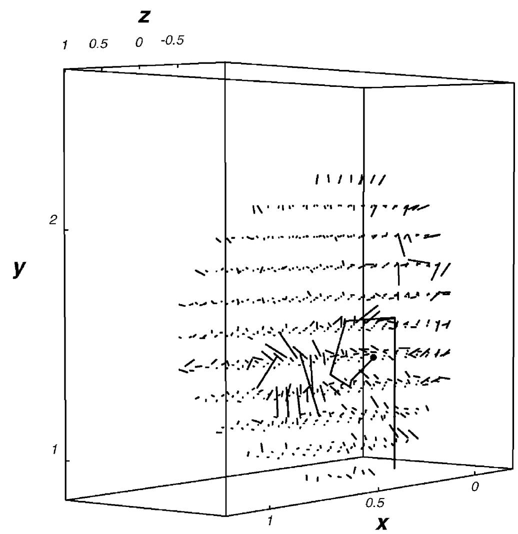

Fig. 1. Position error vectors of inverse kinematics model consisting a single neural network.

a specific position/orientation of the arm's end-effector (hand). Although numerous researchers have used artificial neural networks for learning the inverse kinematics model of a human arm [7,6], they have not fully considered the discontinuity nature of the inverse kinematics function of the human arm. The inverse kinematics function of the arm is a multi-valued and discontinuous function, and it is difficult for continuous artificial neural networks to approximate such a function. The continuous neural network has a number of problems as a model of human neural networks. However, it is plausible that the human nervous system utilizes the local continuity of the inverse kinematics function according to a number of psycho-physical experiments [4]. In this paper, a new methodology is proposed for inverse kinematics learning, using continuous neural networks.

We will consider the learning of the inverse kinematics model of 7 degrees-of-freedom (DOF) human arm. Fig. 1 shows the position error vectors of the inverse kinematics model consisting of a single neural network learned by forward and inverse modeling, as proposed by Jordan [6]. The arrows in the figure represent the hand position errors by the inverse kinematics model at each desired hand position. Although in most regions, the inverse kinematics model is precise, there are some regions where it is far from precise. These are caused by the discontinuity of the inverse kinematics function, with the maximum error values being larger than 0.4 m. Therefore, a novel neural network architecture for learning the inverse kinematics model is necessary.

Jacobs et al. proposed a modular neural network architecture that consisted of a number of expert networks, and a gating network which synthesized the outputs of the expert networks appropriately [5]. Gomi and Kawato applied the modular neural network architecture to object recognition, in order to manipulate a variety of objects and to learn the inverse dynamics [3]. Wolpert and Kawato proposed multiple pairs of forward and inverse models as a computational model of the cerebellum [11]. Ghahramani et al. showed that the human inverse kinematics model consists of modular neural networks, by performing psycho-physics experiments using virtual reality technology [2]. However, the input–output relation of their proposed networks is continuous and the learning method is not sufficient for handling the non-linearity of the kinematics system of a human arm. Therefore, their architecture is not suitable for learning the inverse kinematics model.

Another methodology is based on the concept that the inverse kinematics function can be decomposed into a finite number of solution branches. DeMers et al. proposed an inverse kinematics learning method where a neural network learns each solution branch calculated by the global searches in the joint space [1]. However, the method is a purely off-line learning method and is not applicable for on-line learning, i.e. simultaneous or alternate execution of the control and the inverse model learning. We believe that the human nervous system has an on-line learning capability. Furthermore, DeMers's method is not goal-directed. In order to overcome the drawbacks of DeMers's method, a novel modular neural network system for the inverse kinematics model learning is proposed in this paper.

2. Proposed modular neural network architecture

In order to learn a discontinuous inverse kinematics function, selecting one expert can yield better results than mixing all experts. We have proposed a novel modular neural network architecture for inverse kinematics learning based on DeMers' method [9].

In this paper, $\boldsymbol{\theta}$ denotes the $m \times 1$ joint angle vector and $\boldsymbol{x}$ denotes the $n \times 1$ position/orientation vector of a robotic arm. The relationship between $\boldsymbol{\theta}$ and $\boldsymbol{x}$ is described by $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\theta})$, where $\boldsymbol{f}$ is a C^1 class function. The Jacobian of the robotic arm is denoted by $\boldsymbol{J}(\boldsymbol{\theta})$, and is defined as $\boldsymbol{J}(\boldsymbol{\theta}) = \partial \boldsymbol{f}(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$. When a desired hand position/orientation vector $\boldsymbol{x}_d$ is given, an inverse kinematics problem that calculates the joint angle vector $\boldsymbol{\theta}_{\mathrm{d}}$ satisfying the equation $\boldsymbol{x}_{\mathrm{d}} = \boldsymbol{f}(\boldsymbol{\theta}_{\mathrm{d}})$ is considered.

Fig. 2 illustrates the conceptual diagram of the modular neural network architecture for inverse kinematics learning. The system consists of a number of experts, an expert selector, an expert generator, and a feedback controller extended to the non-linearity of the kinematics system. Each expert network approximates the continuous region of the inverse kinematics function. The expert selector in the proposed system selects an appropriate expert whose output minimizes the expected hand position error. The neural network calculating the predicted errors of each experts is called the performance prediction network. The extended feedback controller is a model of the human inverse kinematics computation system that has a kind of global search function of the joint angle vector space. The expert generator produces

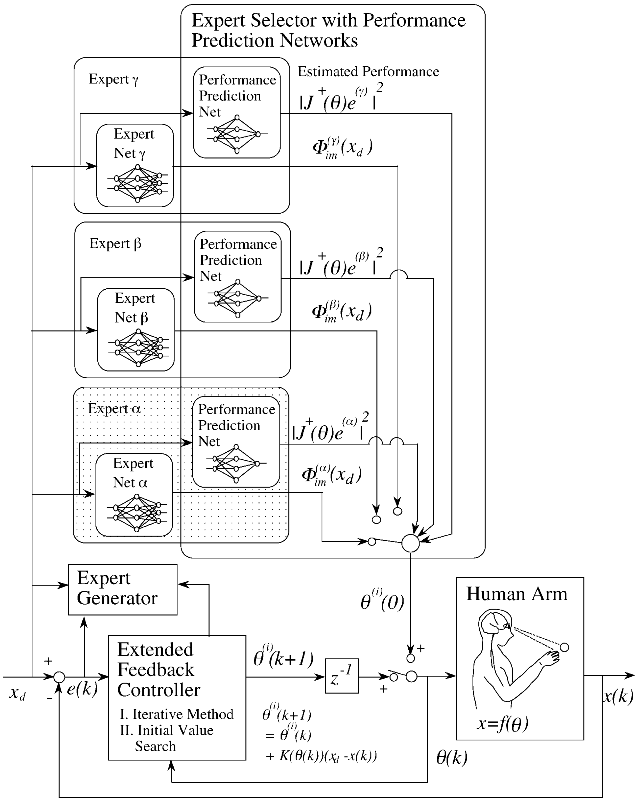

Fig. 2. Inverse kinematics computation system with modular neural networks.

a new expert network based on the inverse kinematics solution obtained by the global search.

Let N_e be the number of experts. Let $\boldsymbol{\Phi}_{\mathrm{im}}^{(i)}(\boldsymbol{x}_{\mathrm{d}})$ $(i = 1, 2, \ldots, N_e)$ be the output of the ith expert and let $\boldsymbol{\Phi}_{\mathrm{pp}}^{(i)}(\boldsymbol{x}_{\mathrm{d}})$ be the output of the performance prediction network which estimates the error of the ith expert. The learning of the performance prediction network will be described in Section 3.

In order to cover the overall work space, each expert has its own representative posture. The representative posture is the inverse kinematics solution obtained in the global searches by the extended feedback controller when the expert is generated. Let $\boldsymbol{\theta}_{\mathrm{r}}^{(i)}$ be the representative posture of the ith expert and $\boldsymbol{x}_{\mathrm{r}}^{(i)}$ be the hand position/orientation that corresponds to $\boldsymbol{\theta}_{\mathrm{r}}^{(i)}$. Let $\boldsymbol{\Phi}_{\mathrm{im}}^{(i)}(\boldsymbol{x})$ be the output of the ith expert when the input of the expert is $\boldsymbol{x}$. Each expert is trained to satisfy the following equation: $\boldsymbol{x}_{\mathrm{r}}^{(i)} = \boldsymbol{f}(\boldsymbol{\Phi}_{\mathrm{im}}^{(i)}(\boldsymbol{x}_{\mathrm{r}}^{(i)}))$. Each expert approximates the continuous region of the inverse kinematics function in which the reaching motion can move the hand smoothly from its representative posture.

The proposed inverse kinematics computation system calculates an inverse kinematics solution according to the following procedural steps:

(1) When a desired hand position $\boldsymbol{x}_{\mathrm{d}}$ is given, the performance prediction networks calculate $\boldsymbol{\Phi}^{(i)}_{\mathrm{pp}}(\boldsymbol{x}_{\mathrm{d}})$ $(i = 1, 2, \ldots, N_{\mathrm{e}})$. The expert selector then selects the expert with the minimum predicted error.
(2) If the predicted error of the selected expert is lower than a specified threshold r_{eim}, the controller moves the arm to the posture that corresponds to the output of the selected expert and then moves the hand to $\boldsymbol{x}_{\mathrm{d}}$ by using the hand position error feedback, as described in Section 3. In case no precise inverse kinematics solution is obtained, the reaching motion from the representative posture of the expert to $\boldsymbol{x}_{\mathrm{d}}$ is conducted. If the predicted error of the selected expert is larger than r_{eim}, the reaching motion from the representative posture of the expert to $\boldsymbol{x}_{\mathrm{d}}$ is conducted.
(3) When no precise inverse kinematics solution is obtained in Step (2), another expert is selected in increasing order of the predicted error, and the reaching motion as described in Step (2) is conducted. This procedure is repeated until a precise solution is found or all the experts are tested.
(4) When no solution is obtained in the above procedural steps, the controller starts a type of global search. The controller repeats the initial joint angle vector generation by using a neural system that produces a uniform random signal along with the reaching motion from the generated posture, until a precise solution is obtained. When a precise solution is obtained, a new expert is generated and the solution is used as the representative posture $\boldsymbol{\theta}_{\mathrm{r}}$ of the expert.

3. Reaching motion and expert learning

Let $\boldsymbol{\theta}(0)$ be the initial posture of the iterative procedure, which is the output of the selected expert $\boldsymbol{\Phi}^{(i)}(\boldsymbol{x}_{\mathrm{d}})$; the representative posture of the selected expert $\boldsymbol{\theta}^{(i)}_{\mathrm{r}}$; or the randomly generated posture. Let $\boldsymbol{x}_{\mathrm{s}}$ be the initial hand position which is defined as $\boldsymbol{x}_{\mathrm{s}} = \boldsymbol{f}(\boldsymbol{\theta}(0))$. The extended feedback controller conducts a reaching motion from $\boldsymbol{x}_{\mathrm{s}}$ to $\boldsymbol{x}_{\mathrm{d}}$ by using resolved motion rate control (RMRC) [10]. The desired trajectory $\boldsymbol{x}_{\mathrm{d}}(k)(k = 0, 1, \ldots, T+1)$ is a straight line from $\boldsymbol{x}_{\mathrm{s}}$ to $\boldsymbol{x}_{\mathrm{d}}$, which is generated to satisfy $\|\boldsymbol{x}_{\mathrm{d}}(k+1) - \boldsymbol{x}_{\mathrm{d}}(k)\| < r_{\mathrm{st}}$.

We assume that a precise hand position feedback controller is already obtained through learning [8]. Let $\boldsymbol{J}^{+}(\boldsymbol{\theta})$ be the pseudo-inverse matrix (Moore–Penrose generalized inverse matrix) of $\boldsymbol{J}(\boldsymbol{\theta})$ which is calculated as $\boldsymbol{J}^{+}(\boldsymbol{\theta}) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{\theta})(\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{\theta}))^{-1}$. $\boldsymbol{J}^{+}(\boldsymbol{\theta})$ is used as the coordinate transformation gain of the output error feedback. Let $\boldsymbol{\theta}(k)$ be an approximate inverse kinematics solution at step k. When r_{st} is small enough, $\boldsymbol{\theta}(k)$ can be calculated as follows:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \boldsymbol{J}^{+}(\boldsymbol{\theta}(k))(\boldsymbol{x}_{\mathrm{d}}(k+1) - \boldsymbol{f}(\boldsymbol{\theta}(k))). \tag{1}$$

Let $\boldsymbol{\Phi}'^{(i)}_{\mathrm{im}}(\boldsymbol{x}_{\mathrm{d}})$ be the desired output signal for the ith expert and $\boldsymbol{\Phi}'^{(i)}_{\mathrm{pp}}(\boldsymbol{x}_{\mathrm{d}})$ be the desired output signal for the performance prediction network of the ith expert. If a precise solution $\boldsymbol{\theta}(k)$, whose hand position error norm $\|\boldsymbol{x}_{\mathrm{d}}(k) - \boldsymbol{f}(\boldsymbol{\theta}(k))\|$ is lower than

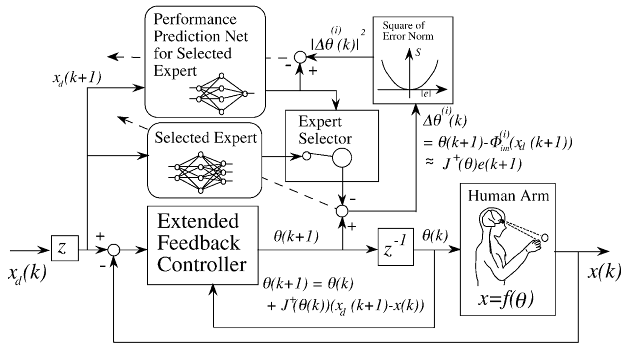

Fig. 3. Learning of expert network and performance prediction network.

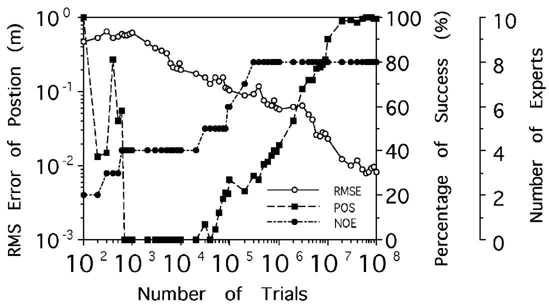

Fig. 4. Progress of learning.

an appropriate threshold r_e, is obtained, the solution can be used for the selected expert learning as $\boldsymbol{\Phi}'^{(i)}_{\mathrm{im}}(\boldsymbol{x}_{\mathrm{d}}(k)) = \boldsymbol{\theta}(k)$. The learning of the performance prediction network is conducted as $\boldsymbol{\Phi}'_{\mathrm{pp}}(\boldsymbol{x}_{\mathrm{d}}(k)) = \|\Delta\boldsymbol{\theta}^{(i)}(k)\|^2 = \|\boldsymbol{\theta}(k) - \boldsymbol{\Phi}^{(i)}_{\mathrm{im}}(\boldsymbol{x}_{\mathrm{d}}(k))\|^2 \approx \|\boldsymbol{J}^+(\boldsymbol{\theta})\ \boldsymbol{e}(k)\|^2$. This value is not the hand position error of the expert but directly corresponds to it. The learning of the selected expert network and the corresponding performance prediction network are illustrated in Fig. 3. When the controller cannot find a precise solution because of the singularity of Jacobian or the joint limits, the reaching motion is regarded as a failure.

4. Numerical experiments

Numerical experiments of the inverse kinematics model learning of a 7 DOF arm were performed. Four-layered neural networks were used for the simulations. The first layer and the fourth layer consisted of linear units. The second and third layers of the experts had 25 units each. The second and third layers of the performance prediction networks had 10 units each. The back-propagation method was utilized for the learning. Fig. 4 shows the progress of the inverse kinematics model learning. The line

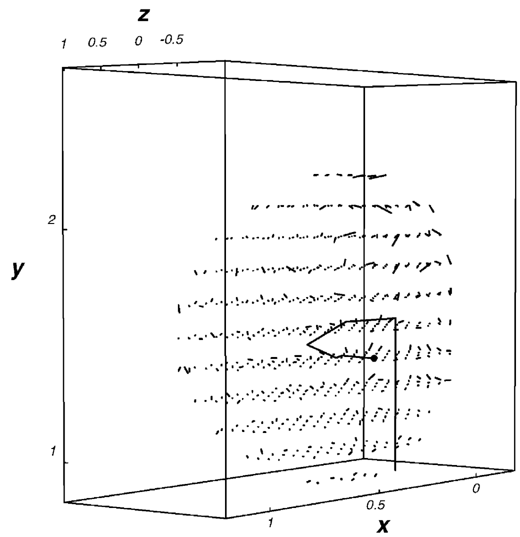

Fig. 5. Error vectors of proposed inverse kinematics model.

marked with white circles shows the RMS error of the hand position $\sqrt{E[e^{\mathrm{T}}e]}$ (RMSE). The dashed line with black boxes shows the percentage of successful trials (POS) in which the posture generated by the first selected expert can successfully reach the desired position. The dashed line with black circles shows the number of experts (NOE).

After 30 million learning trials, the RMS hand position error became lower than 0.01 m and the expert selection was always appropriate. Fig. 5 shows the position error vector of an inverse kinematics model that consists of the proposed modular neural networks. The proposed architecture can approximate the discontinuous inverse kinematics function precisely.

5. Conclusion

In this paper, a novel modular neural network architecture was proposed for the inverse kinematics model learning. The effectiveness of the proposed approach was illustrated through numerical experiments. Although the proposed architecture has a number of limitations (for instance, the inverse kinematics computation procedure is very complex and the learning speed is low), we believe that the proposed architecture can be potentially used as a prototype for the human inverse kinematics model.

References

[1] D. DeMers, K. Kreutz-Delgado, Solving inverse kinematics for redundant manipulators, in: O. Omidvar, P.V.D. Smagt (Eds.), Neural Systems for Robotics, Academic Press, New York, 1997.

[2] Z. Ghahramani, D.M. Wolpert, Modular decomposition in visuomotor learning, Nature 386 (1997) 392–395.
[3] H. Gomi, M. Kawato, Recognition of manipulated objects by motor learning with modular architecture networks, Neural Networks 6 (1993) 485–497.
[4] H. Imamizu, Y. Uno and M. Kawato, Internal representations of the motor apparatus: implications from generalization in visuomotor learning, J. Exp. Psychol. Hum. Percept. Perform. 21 (5) (1995) 1174–1198.
[5] R.A. Jacobs, M.I. Jordan, Learning piecewise control strategies in a modular neural network architecture, IEEE Trans. Systems, Man, Cybernet. 23 (1993) 337–345.
[6] M.I. Jordan, Supervised learning and systems with excess degrees of freedom, COINS Technical Report, Vol. 88–27, 1988, pp. 1–41.
[7] M. Kuperstein, Neural model of adaptive hand-eye coordination for single postures, Science 239 (1988) 1308–1311.
[8] E. Oyama, S. Tachi, Coordinate transformation learning of hand position feedback controller by using change of position error norm, in: M.S. Kearns, S.A. Solla, D.A. Cohn (Eds.), Advances in Neural Information Processing Systems, Vol. 11, MIT Press, Cambridge, MA, 1999, pp. 1038–1044.
[9] E. Oyama, S. Tachi, Modular neural net system for inverse kinematics learning, Proceedings of the International Conference on Robotics and Automation, 2000.
[10] D.E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Trans. Man-Machine System 10 (2) (1969) 47–53.
[11] D. Wolpert, M. Kawato, Multiple paired forward and inverse models for motor control, Neural Networks 11 (1998) 1317–1329.

Arvin Agah

Karl F. MacDorman

Susumu Tachi

Elmei Oyama

Taro Maeda