Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

Inverse Kinematics Learning by Modular Architecture Neural Networks with Performance Prediction Networks

Eimei OYAMA and Nak Young Chong
Intelligent Systems Laboratory
National Institute of Advanced Industrial Science and Technology
Namiki 1-2, Tsukuba-shi, Ibaraki 305-8564 Japan
eimei@mel.go.jp

Arvin Agah
Department of Electrical Engineering and Computer Science
The University of Kansas
Lawrence, KS 66045 USA

Taro MAEDA and Susumu TACHI
School of Engineering, The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656 Japan

Abstract

Inverse kinematics computation using an artificial neural network that learns the inverse kinematics of a robot arm has been employed by many researchers. However, the inverse kinematics system of typical robot arms with joint limits is a multi-valued and discontinuous function. Since it is difficult for a well-known multi-layer neural network to approximate such a function, a correct inverse kinematics model cannot be obtained by using a single neural network. In order to overcome the discontinuity of the inverse kinematics function, we proposed a novel modular neural network system that consists of a number of expert neural networks. Each expert approximates the continuous part of the inverse kinematics function. The proposed system uses the forward kinematics model for selection of experts. When the number of the experts increases, the computation time for calculating the inverse kinematics solution also increases without using the parallel computing system. In order to reduce the computation time, we propose a novel expert selection by using the performance prediction networks which directly calculate the performances of the experts.

1 Introduction

The task of calculating all of the joint angles that would result in a specific position/orientation of an end-effector of a robot arm is called the inverse kinematics problem. An inverse kinematics solver using an artificial neural network that learns the inverse kinematics system of a robot arm has been used in many researches [1][2]; however, many researchers do not pay enough attention to the discontinuity of the inverse kinematics function of typical robot arms with joint limits. The inverse kinematics function of the robot arms, including a human arm with a wrist joint, is a multi-valued and discontinuous function. It is difficult for a well-known multi-layer neural network to approximate such a function. Therefore a novel modular neural network architecture for the inverse kinematics model learning is necessary.

A modular neural network architecture was proposed by Jacobs et al. and has been used by many researches [3][4][5]. However, the input-output relation of their networks is continuous and the learning method of them is not sufficient for the non-linearity of the kinematics system of the robot arm.

In order to learn a discontinuous inverse kinematics function, selecting one expert [6] has better performance than mixing all experts. The inverse kinematics function decomposes into a finite number of inverse kinematics solution branches [7][8][9][10]. DeMers et al. proposed the inverse kinematics learning method that a neural network learns each solution branch calculated by the global searches in the joint space [8][9][10]. However, the method is a purely offline learning method and is not applicable for on-line learning, i.e. simultaneous or alternate execution of the robot control and the inverse model learning. Furthermore, the method is not goal-directed. There is no direct way to find an joint angle vector that corre-

sponds to a desired hand position.

We proposed a novel modular neural network architecture for inverse kinematics learning based on DeMers' method [11][12]. The proposed modular neural network system consists of a number of experts, implemented by using artificial neural networks. Each expert approximates the continuous region of the inverse kinematics function. The proposed modular neural net system selects one appropriate expert whose output minimizes the expected position/orientation error of the end-effector of the arm calculated by using a forward kinematics model. The proposed system can learn a precise inverse kinematics model.

Since the proposed system uses the forward kinematics model of the robot arm for the calculation of the expected position/orientation error, the system requires the calculation of the outputs of all the experts and the calculation of the predicted position/orientation of the end-effector by using the forward kinematics model. When the number of the experts increases, the computation time for the calculation of the predicted errors of the experts also increases without using the parallel computing system.

In order to reduce the computation time, we propose a novel expert selection by using the performance prediction networks which directly calculate the performances of the experts. In order to evaluate the proposed architecture, numerical experiments of the inverse kinematics model learning were performed.

2 Modular Neural Net System with the Performance Prediction Networks

Let $\boldsymbol{\theta}$ be the $m \times 1$ joint angle vector and $\boldsymbol{x}$ be the $n \times 1$ position/orientation vector of a robot arm. The relationship between $\boldsymbol{\theta}$ and $\boldsymbol{x}$ is described by $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\theta})$. $\boldsymbol{f}$ is a C^1 class function. Let $\boldsymbol{J}(\boldsymbol{\theta})$ be the Jacobian of the robot arm, 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 $\boldsymbol{\theta}_d$ satisfying the equation $\boldsymbol{x}_d = \boldsymbol{f}(\boldsymbol{\theta}_d)$ is considered.

In this paper, a function $\boldsymbol{g}(\boldsymbol{x})$ that satisfies $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x}))$ is called an inverse kinematics function of $\boldsymbol{f}(\boldsymbol{\theta})$. The acquired model of the inverse kinematics system $\boldsymbol{g}(\boldsymbol{x})$ in the robot controller is called an inverse kinematics model. Let $\boldsymbol{\Phi}_{im}(\boldsymbol{x})$ be the output of the inverse kinematics model. Although $\boldsymbol{g}(\boldsymbol{x})$ is usually a multi-valued and discontinuous function, the inverse kinematics function can be constructed by the appropriate synthesis of continuous functions [10][11][12].

2.1 Configuration of Proposed Inverse Kinematics Solver

Fig. 1 shows the configuration of the improved inverse kinematics solver with the modular architecture networks for inverse kinematics learning. Each expert network in Fig. 1 approximates the continuous region of the inverse kinematics function. The performance prediction network learns the performance of each expert. The expert selector selects one appropriate expert based on the outputs of the performance prediction networks as described in Section 2.2. The extended feedback controller calculates the inverse kinematics solution based on the output of the selected expert by using the output error feedback. When no precise solution is obtained, the controller performs a kind of global search, as shown in Section 2.3. The expert generator generates a new expert network based on the inverse kinematics solution.

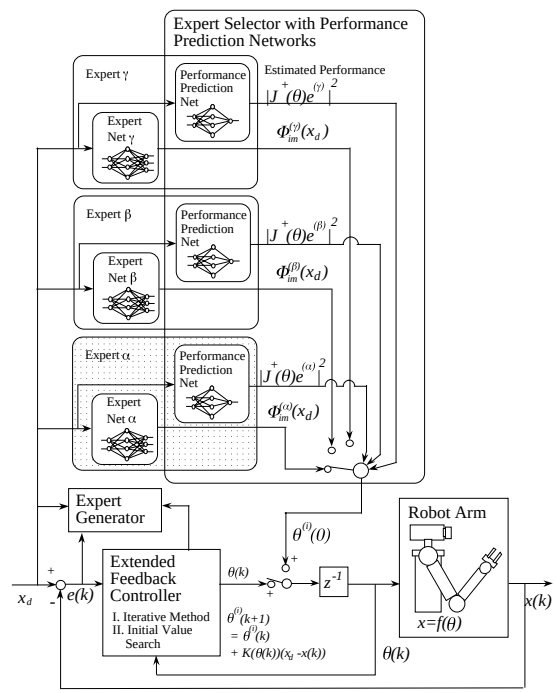


Fig. 1: Modular Neural Net System with Performance Prediction Networks

2.2 Configuration of the Expert and Selection by the Performance Prediction Networks

In order to cover the overall work space, each expert has its 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 $\boldsymbol{\theta}_r^{(i)}$ be the representative posture of the *i-th* expert and $\boldsymbol{x}_r^{(i)}$ be the end-effector position/orientation that corresponds to $\boldsymbol{\theta}_r^{(i)}$. Let $\boldsymbol{\Phi}_{im}^{(i)}(\boldsymbol{x})$ be the output of *i-th* expert when the input of the expert is $\boldsymbol{x}$. Each expert is trained to satisfy the following equation:

$$\boldsymbol{x}_r^{(i)} = \boldsymbol{f}(\boldsymbol{\Phi}_{im}^{(i)}(\boldsymbol{x}_r^{(i)}))). \tag{1}$$

By changing the bias parameters of the output layer of the neural network, the above equation can easily be satisfied. Each expert approximates the continuous region of the inverse kinematics function in which the reaching motion can move the end-effector smoothly from its representative posture.

As stated in Section 1, the previously proposed modular net system with the forward kinematics model requires relatively large computation time. In order to reduce the computation time, we propose the use of the performance prediction networks that directly calculates the values which corresponds to the predicted end-effector position/orientation errors of the experts. The expert selector selects an expert with the best predicted performance among all the experts. If the performance prediction networks are accurate, the calculation of the output of only one selected expert instead of the outputs of all the experts is necessary for the inverse kinematics computation.

The idea of the performance prediction networks is based on a primitive reinforcement learning technique [13]. However, since the properties of each expert changes by learning, the careful construction of the learning algorithm of the proposed architecture is necessary. The learning of the performance prediction network will be described Section 2.4.

Let N_e be the number of the experts. Let $\boldsymbol{\Phi}_{im}^{(i)}(\boldsymbol{x}_d)(i = 1, 2, \ldots, N_e)$ be the output of the *i-th* expert and let $\boldsymbol{\Phi}_{pp}^{(i)}(\boldsymbol{x}_d)$ be the output of the performance prediction network which estimates the error of the *i-th* expert. When the desired end-effector position $\boldsymbol{x}_d$ is given, the performance prediction networks calculates the expected performances of all the experts $\Phi_{pp}^{(i)}(\boldsymbol{x}_d)(i = 1, 2, \ldots, N_e)$.

2.3 Extended Feedback Controller

The conventional on-line inverse model learning methods, such as Forward and Inverse Modeling proposed by Jordan [2] and Feedback Error Learning proposed by Kawato [14], are based on the local information of the forward system near the output of the inverse model. The desired output signal provided by these methods is not always in the direction that finally reaches the correct solution of the inverse problem [15]. An extended feedback controller avoids that drawback by employing a global search technique based on the multiple starts of the iterative procedure [16][15].

When a desired end-effector position $\boldsymbol{x}_d$ is given, the expert selector selects the expert with the minimum predicted error among all the experts. The extended feedback controller moves the arm to the posture that corresponds to the output of the expert and then improves the end-effector position/orientation by using the output error feedback, as described in Section 2.4. When no precise inverse kinematics solution is obtained, the other expert which predicted error is lower than an appropriate threshold r_{eim} is selected in increasing order of the predicted error and the iterative improvement procedure by the output error feedback is conducted. When no solution is obtained by the reaching motions from all the output of the selected experts, an expert is randomly selected and a reaching motion from the representative posture of the selected expert is conducted. an repeated until the reaching motion is successfully conducted or all the experts are tested. If a precise solution is obtained in the above procedural steps, the solution is used as the desired output signal for the expert, as shown in 2.4.

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 uniform random number generator and the reaching motion from the generated posture, as described in 2.4, 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 $\boldsymbol{\theta}_r$ of the expert.

2.4 Reaching Motion and Expert Learning

An illustration of the reaching motion, which is a kind of iterative improvement procedure, follows.

Let $\boldsymbol{\theta}(0)$ be the initial posture of the iterative procedure, which is the output of the selected expert $\boldsymbol{\Phi}^{(i)}(\boldsymbol{x}_d)$; the representative posture of the selected expert $\boldsymbol{\theta}_r^{(i)}$; or the randomly generated posture.

Let $\boldsymbol{x}_s$ be the initial end-effector position which is defined as $\boldsymbol{x}_s = \boldsymbol{f}(\boldsymbol{\theta}(0))$. The extended feedback controller conducts a reaching motion from $\boldsymbol{x}_s$ to $\boldsymbol{x}_d$ by using Resolved Motion Rate Control (RMRC) [17]. The reaching motion is conducted as the tracking control to the following desired trajectory of the end-

effector $\boldsymbol{x}_d(k)(k = 0, 1, \ldots, T+1)$ described as follows.

Let T be an integer that satisfies $T - 1 \leq ||\boldsymbol{x}_d - \boldsymbol{x}_s||/r_{st} < T$. The desired trajectory $\boldsymbol{x}_d(k)$ is a straight line from $\boldsymbol{x}_s = \boldsymbol{f}(\boldsymbol{\theta}(0))$ to $\boldsymbol{x}_d$ which is calculated as follows:

$$\boldsymbol{x}_d(k) = \begin{cases} (1 - \frac{k}{T})\boldsymbol{x_s} + \frac{k}{T}\boldsymbol{x}_d & (0 \leq k < T) \\ \boldsymbol{x}_d & (k \geq T) \end{cases} \quad (2)$$

When the orientation is represented by the Direction Cosine Matrix or the Quaternion, the components of $\boldsymbol{x}_d(k)$ must be normalized.

We assume that a precise end-effector position feedback controller is already obtained by learning [18][19][20]. Otherwise, we assume that the controller can accurately estimate the coordinate transformation by the observation of the robot arm movement and the numerical differentiation technique [15].

Let $\boldsymbol{J}^+(\boldsymbol{\theta})$ be the pseudo-inverse matrix (Moore-Penrose generalized inverse matrix) of $\boldsymbol{J}(\boldsymbol{\theta})$ which is calculated as $\boldsymbol{J}^+(\boldsymbol{\theta}) = \boldsymbol{J}^T(\boldsymbol{\theta})(\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}^T(\boldsymbol{\theta}))^{-1}$. $\boldsymbol{J}^+(\boldsymbol{\theta})$ 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:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \boldsymbol{J}^+(\boldsymbol{\theta}(k))(\boldsymbol{x}_d(k+1) - \boldsymbol{f}(\boldsymbol{\theta}(k))). \quad (3)$$

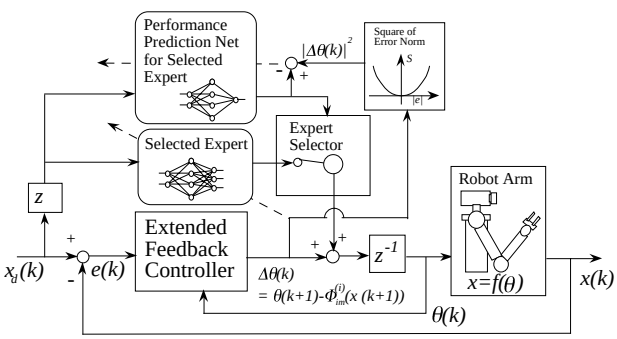


Fig. 2: Learning of Expert Network and Performance Prediction Network

Let $\boldsymbol{\Phi}'^{(i)}_{im}(\boldsymbol{x}_d)$ be the desired output signal for the *i-th* expert and $\boldsymbol{\Phi}'^{(i)}_{pp}(\boldsymbol{x}_d)$ be the desired output signal for the performance prediction network of the *i-th* expert. If a precise solution $\boldsymbol{\theta}(k)$, which end-effector position error norm $||\boldsymbol{x}_d(k) - \boldsymbol{f}(\boldsymbol{\theta}(k))||$ is lower than an appropriate threshold r_e, is obtained, the solution can be used for the selected expert learning as follows:

$$\boldsymbol{\Phi}'^{(i)}_{im}(\boldsymbol{x}_d(k)) = \boldsymbol{\theta}(k). \quad (4)$$

The learning of the performance prediction network is conducted as follows:

$$\boldsymbol{\Phi}'_{pp}(\boldsymbol{x}_d(k)) = ||\boldsymbol{\theta}(k) - \boldsymbol{\Phi}^{(i)}_{im}(\boldsymbol{x}_d(k))||^2. \quad (5)$$

The above 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. 2. 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.

3 Simulations

We performed simulations of the inverse kinematics model learning of a 7-DOF arm as shown in Figure 3. This arm is called TELESAR II (TELE-existence Slave Arm II), the original of which was developed for the experimental study on the remote robot control using the virtual reality [21]. The configuration of the arm is illustrated in Figure 3. The parameters $L_i(i = 1, 2, 3, 4)$ in Figure 3 is defined as $L_1 = 0.305m$, $L_2 = 0.260m$, $L_3 = 0.04m$ and $L_4 = 0.150m$.

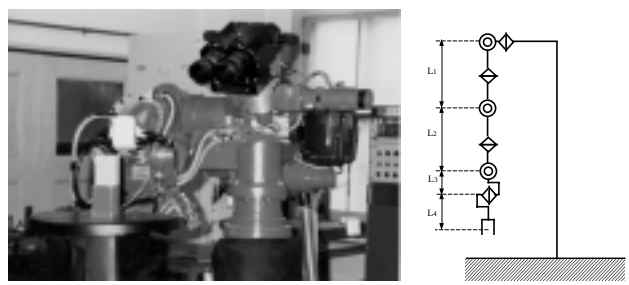
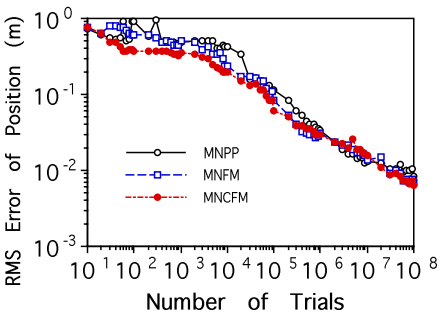


Fig. 3: TELE-existence Slave ARm II (TELESAR II)

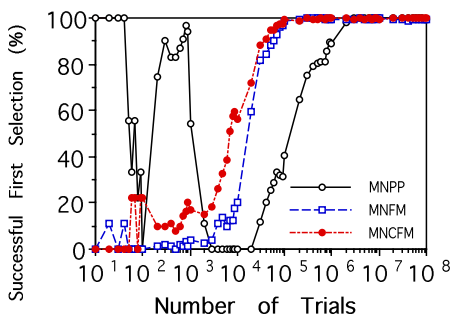
For comparison, the modular neural network system which selects the expert with the minimum error among all the experts by using the complete forward model $\boldsymbol{f}(\boldsymbol{\theta})$ without error were tested. Furthermore, the system which uses the forward model consisting of a neural networks were also tested. Hereafter, MNPP indicates the Modular Neural network system with the performance prediction networks that consist of neural networks with no previous learning. MNCFM indicates the Modular Neural network system with a Complete Forward Model. MNFM indicates the Modular Neural network system with a learning Forward Model.

In the simulations, joint angle vectors were generated by using a uniform random number generator, and the end-effector positions that correspond to

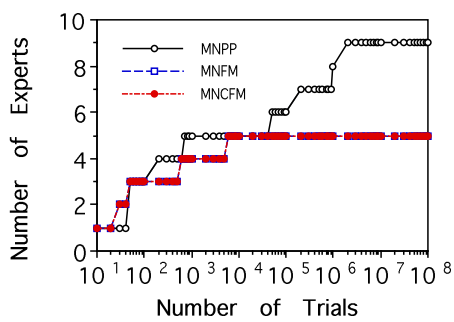
the generated vectors were used as the desired end-effector positions. In order to evaluate the performance of the solver, 16,384 desired end-effector positions were generated for the estimation of the root mean square (RMS) error of the end-effector position $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{f}(\boldsymbol{\Phi}_{im}(\boldsymbol{x}_d))$. r_{eim} was $0.2m$, r_e was $0.0025m$, r_{st} was $0.02m$, and r_{jix} was 10^2.



(a)RMS position error



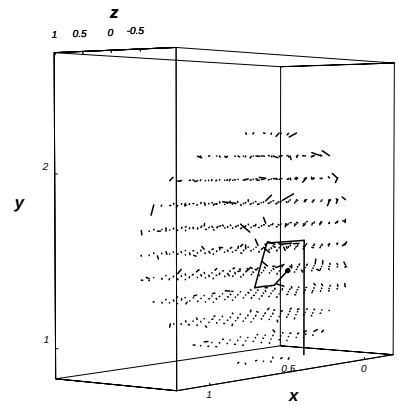
(b)Percentage of successful first selection



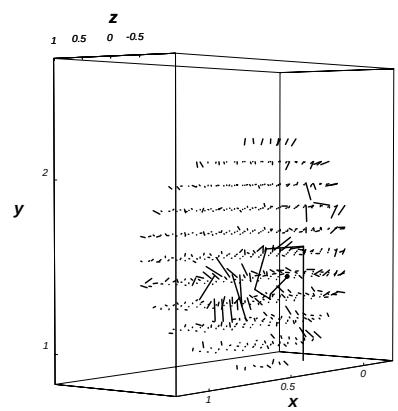
(c) Number of experts

Fig. 4: Performance Change of Proposed Inverse Kinematics Solver

A 4-layered neural network was used for the simulations. The 1st layer, i.e., the input layer, and the 4th layer, i.e., the output layer, consisted of linear neurons. The 2nd and the 3rd layers of the experts and the forward kinematics model had 25 neurons each. The 2nd and the 3rd layers of the performance prediction networks had 10 neurons each. The back-propagation method was utilized for the learning. The learning rate for the experts was set at 0.05. That for the performance prediction networks was set at 0.005. The momentum parameter was set at 0.5.



(a) Proposed Modular Neural Networks



(b) Single Neural Network

Fig. 5: Position Error Vector of Inverse Kinematics Model of 7-DOF Arm

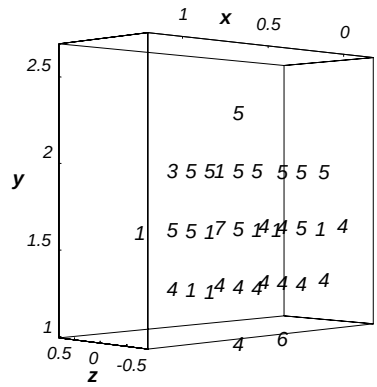
Fig. 4(a) shows the change of the RMS error of the end-effector position. It can be seen that the RMS error decreases and the precision of the inverse model becomes higher as the number of trials increases. The RMS end-effector position error of MNCFM, MNFM, and MNPP became lower than $1.5 \times 10^{-2}m$ after 10^7 learning trials. The precision of MNCFM is better than that of MNFM. The precision of MNFM is better than that of MNPP. However, there is not so much difference between them. We concluded that the proposed method succeeded in the inverse kinematics model learning of a 7-DOF arm.

Fig. 4(b) shows the percentage of the trials in which the posture generated by the first selected expert can successfully reach the desired position. The percentage of the successful first selection of MNCFM and MNFM was larger than 99% before 2×10^5 learning trials. On the other hand, that of MNPP was larger than 99% after 2×10^6 learning trials. The expert selection of MNCFM and MNFM was more precise than that of MNPP. Fig. 4(c) shows the number of the experts that constructs the inverse kinematics model. After 10^7 learning trials, MNFM and MNCFM generated 5 experts. MNPP generated 9 experts. The number of the experts of MNPP is larger than that of MNFM or MNCFM.

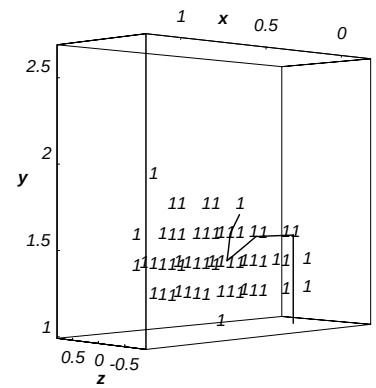
Simulations of the inverse kinematics model learning, consisting of a single neural network, using Forward and Inverse Modeling [2], were also performed. The learning was performed from 10 different initial states of the neural network. The minimum RMS error of the inverse kinematics model using a neural networks after 10^9 learning trials was $1.50 \times 10^{-2} m$. Fig. 5(a) shows the position error vectors of the proposed modular inverse kinematics model. Fig. 5(b) shows those of the inverse kinematics model which consists of single neural network learned by forward and inverse modeling. There are some regions where the inverse kinematics model is far from precise, which are caused by the discontinuity of the inverse kinematics function. The maximum error was larger than $0.4m$. An inverse kinematics model which consists of a single neural networks cannot obtain a precise inverse kinematics model. We also tested the mixture of experts used in [5]. However, we could not obtain a more precise inverse kinematics model by using the mixture of experts than by using a single neural network.

The selection of the expert is illustrated in Fig. 6(a). Each number in the figure indicates which expert is selected at each desired hand position. Fig. 6(b) shows the region where the predicted output error of the first expert is lower than $0.02m$. The graphics of the robot arm in Fig. 6(b) show the representative posture of the first expert. Fig. 6(c) shows the region where the predicted output error of the fifth expert is lower than $0.02m$.

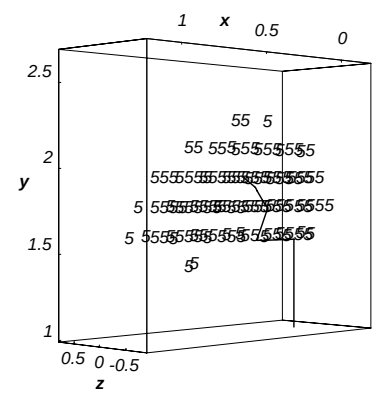
The computation time of one kinematics solution of MNCFM with 5 experts was $0.43ms$ (Intel Pentium II 450 MHz) after 10^7 times learning trials, that of MNFM with 5 experts was $0.73ms$, and that of MNPP with 9 experts was $0.23ms$. The computation time improved using the performance prediction networks. However, the learning speed of the performance prediction networks was much slower than that of the forward kinematics model. The improvement of the learning method is necessary.



(a) Selected experts



(b) Active region of first expert



(c) Active region of fifth expert

Fig. 6: Configuration of Inverse Kinematics Model

4 Conclusions

In this paper, we proposed a novel modular neural network architecture with the performance predic-

tion networks for the inverse kinematics model learning and confirmed the performance of the proposed system by numerical experiments. The computation time for calculating the inverse kinematics solution is reduced by the performance prediction networks. Although the proposed architecture has a number of limitations (for instance, the learning speed is very slow), we believe that the architecture can be used as a prototype of the inverse kinematics solver with learning function. The improvement for faster learning, the elimination of useless experts, and the utilization of the redundant degrees of freedom [22] will be reported in near future.

References

[1] M. Kuperstein,"Neural Model of Adaptive Hand-Eye Coordination for Single Postures," *Science*, Vol.239, pp.1308-1311, 1988.

[2] M. I. Jordan, "Supervised Learning and Systems with Excess Degrees of Freedom," *COINS Technical Report, 88-27*,pp.1-41,1988.

[3] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton,"Adaptive Mixtures of Local Experts," *Neural Computation*,Vol.3, pp.79-87,1991.

[4] R. A. Jacobs and M. I. Jordan," Learning Piecewise Control Strategies in a Modular Neural Network Architecture," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.23, pp.337-345, 1993.

[5] H. Gomi and M. Kawato,"Recognition of Manipulated Objects by Motor Learning with Modular Architecture Networks," *Neural Networks*, Vol.6, pp.485-497, 1993.

[6] K. S. Narendra and J. Balakrishnan, "Adaptation and Learning Using Multiple Models, Switching and Tuning," *IEEE Control Systems Magazine*, Vol. 15, pp.37-51, 1994.

[7] Burdick, J., "Kinematics and Design of Redundant Robot Manipulators," Ph. D. Thesis, Dept. of Mechanical Engineering, Stanford University, 1988.

[8] D. DeMers and K. Kreutz-Delgado, "Learning Global Direct Inverse Kinematics," *Advances in Neural Information Processing Systems 4*, pp. 589-594, 1992.

[9] D. DeMers and K.Kreutz-Delgado, "Global Regularization of Inverse Kinematics for Redundant Manipulators," *Advances in Neural Information Processing Systems 5*, pp.255-262, 1993.

[10] D. DeMers and K. Kreutz-Delgado, "Solving Inverse Kinematics for Redundant Manipulators," in *Neural Systems for Robotics*, O. Omidvar and P. v. d. Smagt ed., Academic Press, 1997.

[11] E. Oyama and S. Tachi, "Inverse Kinematics Model Learning by Modular Architecture Neural Networks," *Proceedings of International Joint Conference on Neural Networks '99*, 1999.

[12] E. Oyama and S. Tachi, "Modular Neural Net System for Inverse Kinematics Learning," *Proceedings of International Conference on Robotics and Automation 2000*, 2000.

[13] A. G. Barto, "Reinforcement Learning," *Handbook of Brain Theory and Neural Networks*, M. A. Arbib ed., The MIT Press, pp.804-809, 1995.

[14] M. Kawato, K. Furukawa and R. Suzuki, "A Hierarchical Neural-network Model for Control and Learning of Voluntary Movement," *Biological Cybernetics*,Vol.57, pp.169-185, 1987

[15] E. Oyama and S. Tachi, "Inverse Model Learning by Using Extended Feedback System," *Proceedings of the fifth International Symposium on Measurement and Control in Robotics(ISMCR'95)*, pp.291-296, 1995.

[16] A. W. Moore,"Fast, Robust Adaptive Control by Learning only Forward Models," *Advances in Neural Information Processing Systems 4*, Morgan Kaufmann Publishers, pp.571-578, 1992.

[17] D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Trans. on Man-Machine System*, Vol.10, No.2, pp.47-53, 1969.

[18] F. H. Guenther and D. M. Barreca, "Neural Models for Flexible Control of Redundant Systems," in P. Morasso and V. Sanguineti (Eds.), *Self-organization, Computational Maps, and Motor Control*, Elsevier, pp. 383-421, 1997.

[19] E. Oyama and S. Tachi, "Coordinate Transformation Learning of Hand Position Feedback Controller by Using Change of Position Error Norm," *Advances in Neural Information Processing Systems 11*, pp. 1038-1044, 1999.

[20] E. Oyama, A. Agah, T. Maeda, S. Tachi and K. F. MacDorman, "Coordinate Transformation Learning of Human Visual Feedback Controller based on Disturbance Noise and Feedback Error Signal," *Proceedings of International Conference on Robotics and Automation 2001*, 2001.

[21] E. Oyama, N. Tsunemoto, Y. Inouse and S. Tachi, "Experimental Study on Remote Manipulation Using Virtual Reality," *PRESENCE*, Vol.2, No.2, pp.112-124, 1993.

[22] Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *The International Journal of Robotics Research*, Vol.6, No.2, pp.3-15, 1987.