Robotics Research Journal
Vol. 1, No. 1; 2018
Published by CanSRG
http://cansrg.com/journals/rrj/

Improve Calibration Accuracy of Modeless Robots by Using a Shallow Neural Network

Ying Bai[1,*] and Dali Wang[2]

[1] *Dept of Computer Science & Engineering, Johnson C. Smith University, Charlotte, NC 28216*
[2] *Dept of Physics, Comp Sci and Engineering, Christopher Newport University, Newport News, VA 23606*

Abstract

This paper describes a technique for the position error estimations and compensations of the modeless robots and manipulators calibration process based on a shallow neural network fitting function method. Unlike traditional model-based robots calibrations, the modeless robots calibrations do not need to perform any modeling and identification processes. Only two processes, measurements and compensations, are necessary for this kind of robots calibrations. The compensation of position error in modeless method is to move the robot's end-effector to a target position in the robot workspace, and to find the target position error based on the measured neighboring 4-point errors around the target position. A camera or other measurement device may be attached on the robot's end-effector to find or measure the neighboring position errors, and compensate the target positions with the interpolation results. By using the shallow neural network fitting technique, the accuracy of the position error compensation can be greatly improved, which is confirmed by the simulation results given in this paper. Also the comparisons among the popular traditional interpolation methods, such as bilinear and fuzzy interpolations, and this shallow neural network technique, are made via simulation studies. The simulation results show that more accurate compensation result can be achieved using the shallow neural network fitting technique compared with the bilinear and fuzzy interpolation methods.

Keywords: *robotics; modeless robots calibrations; fuzzy interpolations; artificial neural networks*

1. Introduction

The prerequisite requirement of the robotic modeless calibration is the successful self-calibration of the camera [1, 2] or other measurement device, such as laser tracking system [3]. Both internal and external parameters of the camera need to be calibrated accurately [4, 5]. Then the modeless robot calibration is divided into two steps [6].

The first step is to measure the position errors for all grid points on a standard calibration board, which is installed on the robot's workspace. A calibrated camera or other measurement device is used to find 4 neighboring position errors. This process can be considered as a measurement process, which is shown in Figure 1.

[*] Corresponding author

Submitted: August 7, 2018 Revised: December 10, 2018 Accepted: December 12, 2018

At each grid point, a calibrated camera is used to check the position errors of the end-effector of the robot. In Fig 1, the desired position of the grid point 0 is (x_0, y_0), and the actual position of the robot end-effector is (x_0', y_0'). The position errors for this grid point are $e_x = x_0 - x_0'$, and $e_y = y_0 - y_0'$. The robot will be moved to all these grid points on the standard calibration board, and all position errors on these grid points will be measured and stored in the memory for future usage.

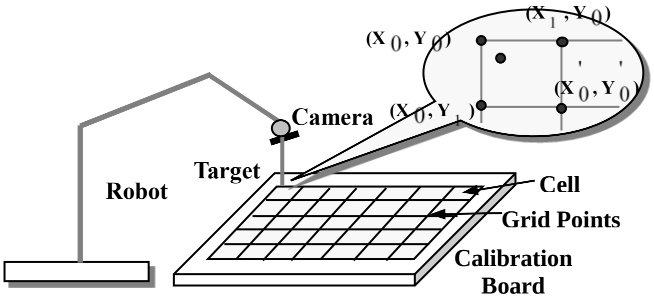


Figure 1. Setup of the modeless robots calibration.

In the second step, the robot's end-effector is moved to a target position that is located in the range of the workspace. The target position error could be found by an interpolation technique using the stored 4-neighboring grid position errors around the target position, which were obtained from the first step. Finally, the target position could be compensated with the interpolation results to obtain more accurate positions.

Triantafilis et al. reported approaches of using fuzzy interpolation methods to estimate the soil layer and geographical distributions for GIS database [7, 8]. Song et al described a fuzzy logic methodology for 4-dimensional (4D) systems with optimal global performance using enhanced cell state space [9]. The most popular interpolation techniques applied in the position compensations of the modeless robotic calibration include the bilinear interpolation and fuzzy interpolation methods; both methods can achieve satisfactory interpolation results for general calibration process [2, 10]. The bilinear interpolation technique assumes that the error of the target position is located on the surface that is constructed by the position errors of 4-neighboring grid points around the target position [9]. The fuzzy interpolation method assumes that the workspace can be divided into a group of smaller cells, and the target positions can be obtained by interpolating position errors on 4 neighboring grid points in cells via fuzzy inference system [10]. Consequently, the target point's errors are estimated according to the equations of the error surface or fuzzy inference techniques. Since the actual position errors are randomly distributed, and it is impossible to pinpoint a position on the error surface at any given moment, the traditional interpolation technique is unable to provide an accurate estimation of the position errors. Fuzzy error interpolation technique utilizes the fuzzy inference system to estimate the position errors, which is consistent with the random distributed nature of position errors. The position errors can be considered as a fuzzy set at any given moment of the time. The fuzzification process takes into account of a range of error rather only a crisp error value. Therefore, the fuzzy error interpolation technique has the fundamentals to improve error estimation results.

However, more and more robots calibration techniques developed by using artificial neural network (ANN) are reported in recent years [11-18]. Xu et al. reported to implement a feedforward neural network that has been trained to predict errors in the joint angles using a fast backpropagation learning rule and then implemented in the control system to correct the errors [11]. Wu et al. developed an effective hand-eye calibration method with an inverse kinematics of robot arms combined with a neural network. With a feedforward neural network and the network trained with error propagation algorithm, the cumbersome and time consuming calibration process due to the high nonlinearity existed in the models can be effectively resolved and improved [12]. Wells et al. presented a new technique used in vision-based robot positioning calibrations, which is based on neural learning and global image descriptors to overcome many limitations existed in the operational steps of this kind of calibration process, such as feature-matching algorithms, camera calibration, models of the camera geometry and object feature relationships. These steps are often computationally intensive and error-prone, and the complexity of the resulting formulations often limits the number of controllable degrees of freedom [13]. Nguyen et al. designed a new calibration method for enhancing robot position accuracy by using an extended Kalman filtering (EKF) algorithm and an artificial neural network (ANN) to effectively compensate those non-geometric error sources and un-modeled errors [14].

Xu et al. proposed a complete calibration method considering geometric errors, joint compliances and exterior load to improve poor accuracy of general industrial robots by using an integrated inverse kinematics algorithm and a neural network with analytical method [15]. Zouaoui and Mekki developed an online identification process by using a neural network (NN) to overcome some uncertainties existed in the relationship between the camera motion and the consequent changes on the visual features [16]. Tao and Yang reported to implement a Radial Basis Function (RBF) Neural Network (NN) augmented robot model together with a two stage calibration process for training the NN to improve the calibration accuracy of industrial robots [17]. Luo et al. presented a calibration method to resolve the absolute accuracy of robot TCP movement in a limited workspace by using the Rectified Linear Unit (ReLU) method based on deep neural network (DNN) [18]. All of these researches provide different ways to try to improve the robots' calibration accuracy and strengthen and simplify the calibration processes via artificial neural network techniques.

This study describes a technique for the position error estimations and compensations of the modeless robots and manipulators calibration process based on a shallow neural network (SNN) fitting function method. A feedforward neural network or called shallow neural network fitting technique is utilized to estimate the position errors based on errors on the 4 neighboring grind points. With this method, the calibration accuracy can be significantly improved and the calibration process can also be greatly simplified. A comparison among three different error interpolation methods, bilinear, fuzzy interpolation and SNN, are performed, and the simulation results indicate that the SNN method outperforms the other methods.

This paper is organized in 5 sections. After this introduction section, the principles of the two interpolation techniques, bilinear and fuzzy, are provided in sections 2 and 3. Section 4 discusses the SNN method. A simulation is given in section 5 to illustrate the effectiveness of the SNN technique. Section 6 presents the conclusion.

2. Bilinear Interpolations

The bilinear interpolation method is designed to construct a surface based on the known neighboring points' errors. Then the target position error is derived by using an error surface equation. The operation principles of the bilinear interpolation methods are discussed in this section.

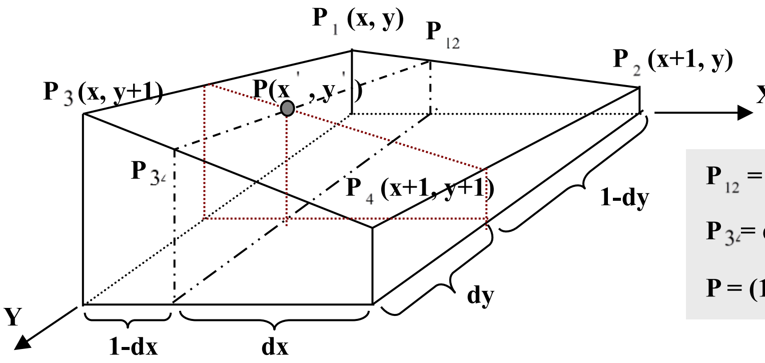
For easy illustration, we divide the position errors into two parts on a two-dimensional (2-D) plane: e_x, the error in *x* direction, and e_y, the error in *y* direction, respectively. We consider e_x and e_y are two variables that are a function of position *x* and *y* in a 2-D plane.

The bilinear interpolation method is based on a linear analysis method. The interpolated error is obtained from an error surface that is constructed based on 4 neighboring errors of the grid points, which is shown in Figure 2.

The interpolation error on the target position P(x', y') is

$$e_x(x', y') = dx\, dy\, e_x(x, y) + (1\text{-}dx)\, dy\, e_x(x+1, y) + (1\text{-}dy)\, dx\, e_x(x, y+1) + (1\text{-}dx)(1\text{-}dy)\, e_x(x+1, y+1)$$
$$e_y(x', y') = dx\, dy\, e_y(x, y) + (1\text{-}dx)\, dy\, e_y(x+1, y) + (1\text{-}dy)\, dx\, e_y(x, y+1) + (1\text{-}dx)(1\text{-}dy)\, e_y(x+1, y+1) \quad (1)$$

As illustrated in Figure 2, the bilinear interpolation technique is based on two assumptions. First, the position error of the target point P(x', y') must be located on the error surface, which is built based on errors of the four neighboring grid points P_1 to P_4. Secondly, the error surface has to be constructed prior to the application of bilinear interpolation technique. However, these assumptions have their drawback. All position errors on each cell are randomly distributed and the error curving surfaces, e_x and e_y, are also randomly distributed at any given moment. We can consider the e_x (x', y') as a third dimensional function value based on the position x' and y' inside each cell. The same consideration is applied to e_y (x', y'). The compensation accuracy of bilinear interpolation is limited by these assumptions.



$P_{12} = dx\, P_1 + (1\text{-}dx)\, P_2$

$P_{34} = dx\, P_3 + (1\text{-}dx)\, P_4$

$P = (1\text{-}dy)\, P_{34} + dy\, P_{12}$

Figure 2. Physical structure of the bilinear interpolation.

3. Fuzzy Error Interpolations

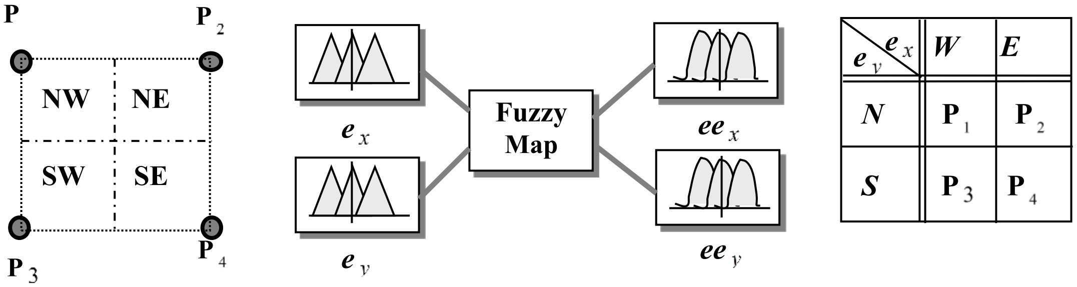
3.1 The on-line versus off-line fuzzy system

In order to improve the compensation accuracy, most time a dynamic on-line fuzzy error interpolation method is implemented. The off-line fuzzy inference system uses pre-defined membership functions and control rules to construct a lookup table; then a control output is selected from the lookup table. This type of system is called off-line fuzzy inference system because all inputs and outputs have been defined prior to the application process. The off-line fuzzy system cannot meet our requirement for the several reasons. First, the position error of the target point is estimated based on 4 errors of the neighboring grid points, and these 4 neighboring errors are randomly distributed. The off-line fuzzy output membership functions are defined based on the range of errors, which is the neighboring errors' range here. However, this range estimation is not as accurate as the real errors obtained on the grid points. Second, since each cell needs one lookup table for the off-line fuzzy system, it would require a huge memory space to save a large number of lookup tables. This results in demanding requirement in both space and time, and as a result, becomes not practical for real time processing. For example, in our study, 20×20 cells are utilized on the calibration board (each cell is 20×20 *mm*); this would require 400 lookup tables! By using an on-line dynamic fuzzy inference system, the target position error can be estimated by combining the output membership functions, which are defined based on the real errors on neighboring grid points and the control rules in real time. The output membership functions are not predetermined, and their definitions are based on the real errors on the grid points, not a range.

3.2 Overview of the fuzzy interpolation system

Figure 3 shows the definition of the fuzzy error interpolation inference system. Each square that is defined by four (4) grid points is called a cell; and each cell is divided into 4 equal smaller cells, which are NW, NE, SW and SE, respectively (Figure 3a). The position error at each grid point is defined as P1, P2, P3 and P4.

For the fuzzy inference system, we apply the fuzzy error interpolation method in two dimensions separately, so the inputs to the fuzzy inference system are e_x and e_y and the outputs are ee_x and ee_y (Figure 3b). The control rules are shown in Figure 3c, and will be discussed following the discussion of membership functions.



e_y \ e_x	W	E
N	P1	P2
S	P3	P4

(a) (b) (c)

Figure 3. Definition of the Fuzzy Error Interpolation System.

3.3 Membership functions

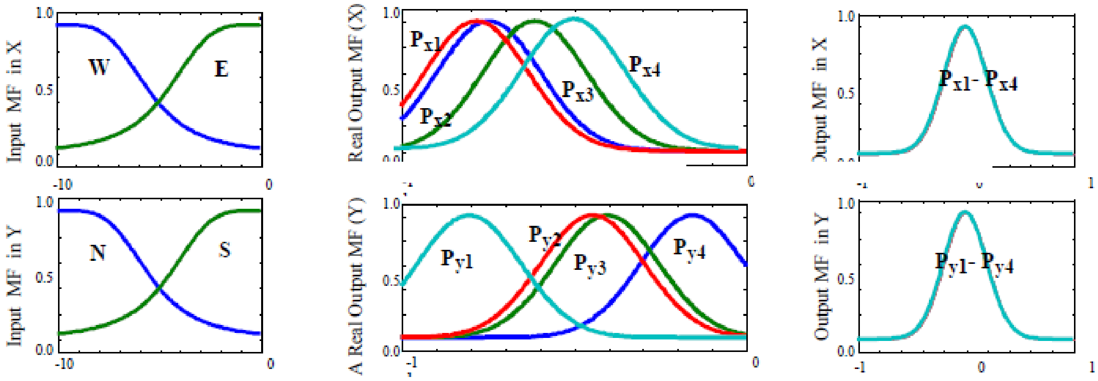
In this study, the distance between two neighboring grid points on the standard calibration board is 20 *mm* in both *x* and *y* directions, which is a standard value for a mid-size calibration workspace. The calibration board includes a total of 20 by 20 cells, which is equivalent to a 400 by 400 *mm* space.

The input membership functions for both *x* and *y* directions and the predefined output membership functions are shown in Figure 4. The predefined output membership functions are used as a default one, and the final output membership function will be obtained by shifting the default one by the actual error values on the grid points.

The Gaussian-bell waveforms are selected as the shape of the membership functions for both inputs (Figure 4a) in *x* and *y* directions. The ranges of inputs are between –10 mm and 10 *mm* (20 *mm* intervals). Zhuang and Wu reported a histogram method to estimate the optimal membership function distribution [19]. However, in our case, a Gaussian-bell shape is selected due to the fact that most errors in real world match this distribution. We use W and E to represent the location of inputs in *x* direction, N and S to represent the location of inputs in *y* direction.

Figure 4b shows an example of the output membership functions, which are related to the simulated random errors at neighboring grid points. Each P_{xi} and P_{yi} correspond to the position error at the *i*th grid point in *x* and *y* directions, respectively. During the design stage, all output membership functions are initialized to a Gaussian waveform with a mean of 0 and a range between –0.5 and 0.5 *mm*, which is a typical error range for this workspace in robotic calibration (Figure 4c). These output membership functions will be determined based on the errors of the neighboring grid points around the target in the workspace as mentioned above.

For example, during the compensation process if the input position in the *x* direction is in the NW area of a cell, the associated output membership function should be modified based on the position error in the NW grid point P_1. This modification is equivalent to shifting the P_{x1} Gaussian waveform (Figure 4b) and allowing the center of that waveform to be located at x_0 = the position error value of the P_1 in the *x* direction. A similar modification should be performed for the position error in the *y* direction. It can be seen from Figure 4b that for the position compensation process, the performance loss would be significant if the default membership function is utilized, which is shown in Figure 4c.



(a) (b) (c)

Figure 4. Input and output membership functions.

3.4 Control Rules

The control rules shown in Figure 3c can be interpreted as follows after the output membership functions are determined.

- If e_x is W and e_y is N, ee_x is P_{x1} and ee_y is P_{y1} (NW).
- If e_x is W and e_y is S, ee_x is P_{x3} and ee_y is P_{y3} (SW).
- If e_x is E and e_y is N, ee_x is P_{x2} and ee_y is P_{y2} (NE).
- If e_x is E and e_y is S, ee_x is P_{x4} and ee_y is P_{y4} (SE).

Each P_i should be considered as a combination of two error components, P_{xi} and P_{yi}, which are corresponding to errors in both *x* and *y* directions. The error on NW grid point should take more weight if the target position (input) is located inside the NW area on a cell. Similar conclusion can be derived for errors on SW, NE and SE grid points.

3.5 Fuzzy inference system

The fuzzy inference system implemented in this study is an on-line one. This means that output of the fuzzy system is not obtained from the pre-defined lookup table, but from a real time fuzzy inference calculation that utilizes the pre-defined input membership functions and the real time position errors. The input error variables can be expressed as a label set L(E), where E is a linguistic input variable.

$$L(E) = \{NW, NE, SW, SE\} \tag{2}$$

Assume that u_i is the membership function, U_i the universe of discourse and *m* the number of contributions, the traditional output of the fuzzy inference system can be represented as

$$u = \frac{\sum_{i=1}^{m}(u_i \times U_i)}{\sum_{i=1}^{m} u_i} \tag{3}$$

where *u* is the current crisp output of the fuzzy inference system. Equation (3) is obtained by using the Center-Of-Gravity method (COG). In this study, both u_i and U_i in the output membership functions are randomly distributed variables and the actual values of these variables depend upon the position errors of four neighboring grid points around the target position. These relationships can be expressed as

$$u_i = F_i(\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \mathrm{P}_4) \tag{4}$$

$$U_i = Q_i(\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \mathrm{P}_4) \tag{5}$$

where F_i and Q_i are randomly distributed functions. Substituting (4) and (5) into (3), we obtain

$$u = \frac{\sum_{i=1}^{m} F_i(P_1, P_2, P_3, P_4) \times Q_i(P_1, P_2, P_3, P_4)}{\sum_{i=1}^{m} F_i(P_1, P_2, P_3, P_4)} \tag{6}$$

In (6), both F_i and Q_i will not be determined until the fuzzy error interpolation technique is applied in an actual compensation process, which means that this fuzzy inference system is an on-line process. The final crisp output of the fuzzy error interpolation system is determined by the neighboring position errors of 4 grid points.

The advantage of using the on-line fuzzy inference system is that the control output has the real time control ability. On the other hand, it involves certain computational complexity. With the processing power of recent microprocessors, this should not become an obstacle for real applications.

4. Shallow neural network fitting function

4.1 Overview of Artificial Neural Networks

The basic unit or building block of an artificial neural network (ANN) is an artificial neuron, which can be considered as a node in a network [20]. The general neural neuron is composed of a set of inputs x_j, (j = 1, 2, … n) where the subscript j represents the jth input. Each input x_j has a definite weight factor w_j that is associated to the input x_j, exactly x_j is multiplied by the factor w_j to form a complete input prior to be input to the network. For an ANN with i (i = 1, 2, … m) nodes, an additional subscript i is needed to represent the ith node. Also it has a bias term w_0, a threshold value Θ, a nonlinear function $\boldsymbol{f}$ that acts on the produced signal $\boldsymbol{R}$, and an output $\boldsymbol{O}$. A basic model of a neuron i or node i is shown in Figure 5.

The input-output relationship of this neuron can be described by a transfer function as

$$O_i = F_i\left(\sum_{j=1}^{n} w_{ij}x_{ij} - \Theta_i\right) \tag{7}$$

and the neuron's firing condition is

$$\sum w_{ij}x_{ij} \geq \Theta_i \tag{8}$$

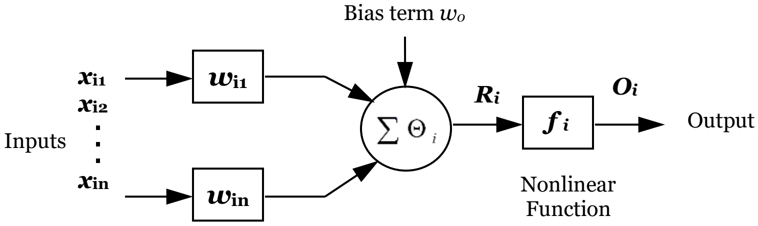


Figure 5. A basic model of a neuron i.

Regularly a complete ANN contains multiple nodes with multiple layers, including input layer, output layer and hidden layers. Figure 6 shows a model of multiple layers feedforward ANN. The dash lines mean that multilayer are included in this ANN and these layers cannot be observable. Exactly in Figure 6, on each feedforward arrow branch from one node to another, a weight factor w_{ij} should be multiplied as shown in Figure 5 to obtain a complete transfer signal.

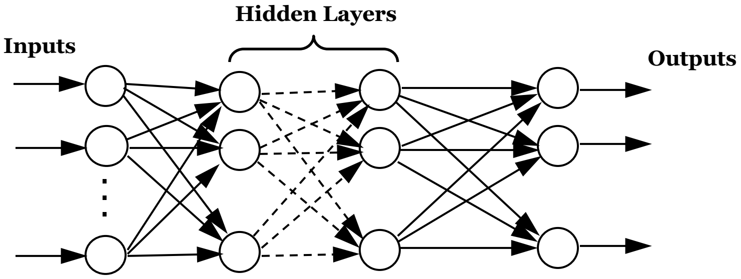


Figure 6. A model of multilayer feedforward ANN.

Overall, a feedforward ANN can be considered as a complex brain/machine system that is composed of multilayer with multi-neuron operating in parallel. In fact, the connections between nodes largely determine the network function. One can train an ANN to perform a specified function by adjusting the values of the connections (weight factors) between nodes via inputs and desired or target outputs. Figure 7 shows an illustration of this training process.

Generally, a neural network can be adjusted or trained, so that a particular input leads to a corresponding target output. In Figure 7, the network is adjusted, based on a comparison between the output and the target, until the network output matches the target. Typically, many such input-target pairs are needed to train a network.

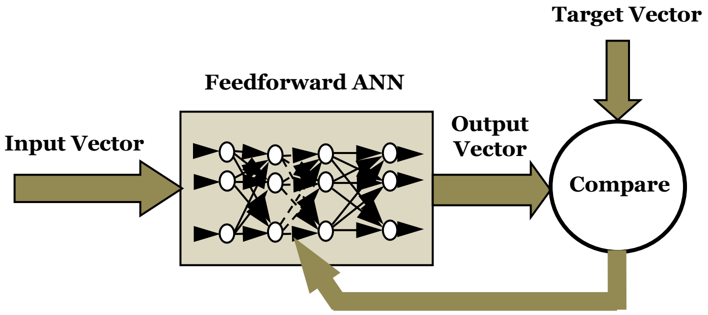


Figure 7. Training process of a feedforward ANN.

4.2 Neural network model and calibration process

To meet our requirements, a feedforward shallow neural network is selected. The model is trained by a set of input position pairs (input vector), x_i and y_i, respectively. The outputs are desired or target position error pairs (output vector), eex_i and eey_i. Four hidden neurons (nodes) with two hidden layers are adopted and its structure is shown in Figure 8. In this architecture, the neurons of the input layer apply the input signals to the neurons of the hidden layer. The output signals of a hidden layer are used as inputs to the next hidden layer and so on. Finally, the output layer produces the output results using the last hidden layer as its inputs. We use linear output nodes and a Sigmoid activation function in the hidden nodes.

The input vector consists of the target position vector x_i, y_i (i = 1 ~ 20), and output consists of the error vector. In this Figure, the notation of weights and biases follows the following convention: weights of connections between layer j and layer i are indicated by w_{ij}; the three sets of neurons are denoted as input set I, output set O, and hidden set H_1 and H_2 (for two hidden layers); the bias of layer i are indicated by Θ_i. Let Z_i be the output from any node i, hidden or output (for an input node, it is the received input signal).

$$\underset{i \in H_1}{Z_i} = Sigmoid(\sum_{j \in I} w_{ij} z_j - \Theta_i) \tag{9}$$

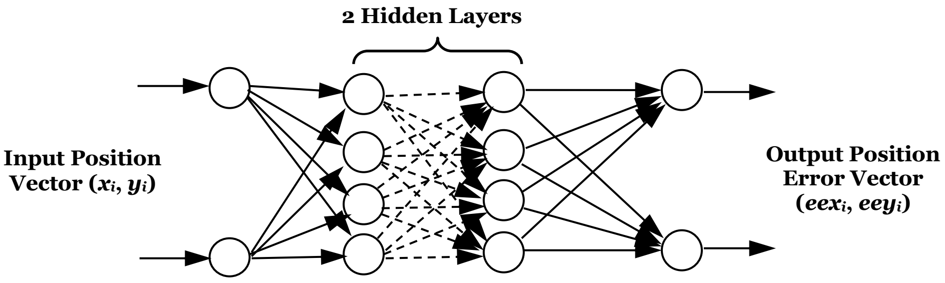

2 Hidden Layers
Input Position Vector (x_i, y_i)
Output Position Error Vector (eex_i, eey_i)

Figure 8. The shallow neural network structure.

$$\underset{i \in H_2}{Z_i} = Sigmoid(\sum_{j \in H_1} w_{ij} z_j - \Theta_i) \tag{10}$$

$$\underset{i \in O}{Z_i} = \sum_{j \in H_2} w_{ij} z_j - \Theta_i \tag{11}$$

The error signal at the output neuron $i \in$O at iteration k is defined by

$$err_i(k) = Z_i(k) - T_i(k) \tag{12}$$

where $T_i(k)$ is the target value for $Z_i(k)$. $T_i(k)$ is derived from x_i, y_i in our case. The error energy for all the neurons in the output layer becomes the L2 norm of error signal, which reflects the position offset.

$$\varepsilon(k) = \left\| err(k) \right\|_2 \tag{13}$$

The training objective is to minimize the mean-squared error of the shallow neural network over N_{all} sets of training data.

$$E = \frac{1}{N_{all}} \sum_{k=1}^{N_{all}} \varepsilon(k) \tag{14}$$

The goal in the calibration process is to build a shallow neural network model for each cell in the workspace. In the compensation stage, the errors at an arbitrary position are then estimated using neural network model built at this stage for the cell within which the position is located. Finally, these estimated or fitting errors are added into the target positions to obtain the compensated positions where the platform is commanded to. As the neural network models are ready to use at this stage for all the cells, the compensation involves very light computational load.

5. Simulation results

Extensive simulation study has been performed in order to illustrate the effectiveness of the proposed SNN technique in comparison to the bilinear and fuzzy interpolation methods. The uniform distributed random error is used for this simulation study because of its popularity.

We estimate the valid workspace of the robot to be 400 x 400 *mm*. We choose the size of each cell to be 20 x 20 *mm* after taking into consideration the repeatability of the robot. Consequently, the workspace consists of 20 cells in each direction. In our simulation, we simulate actual position based on in this format.

$$P_a = (x_i, y_i) + A\ rand(x, y) \tag{15}$$

where (x_i, y_i) is the nominal position, and A is the amplitude of the uniformly distributed noise in the interval (-0.5, 0.5) for the random component.

We implemented MATLAB® Neural Network Toolbox® to perform these simulation studies. Figures 9 and 10 show the structure view and training, validation and testing process of this fitting SNN. The Levenberg-Marquardt algorithm is used for this training.

After the first epoch training process, the mean square error (MSE) is reduced below 0.01, and the best validation result is in the first epoch (0.0096). The testing result is also good with all errors being about 0.01.

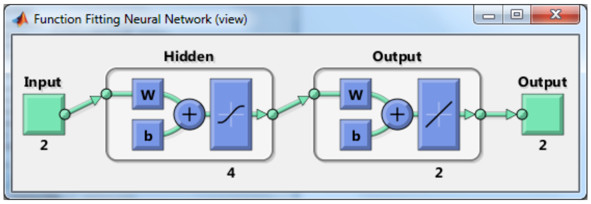


Figure 9. Structure view of the used SNN.

Figure 11 shows the training state (11a) and a comparison between the SNN outputs and the target positions (11b).

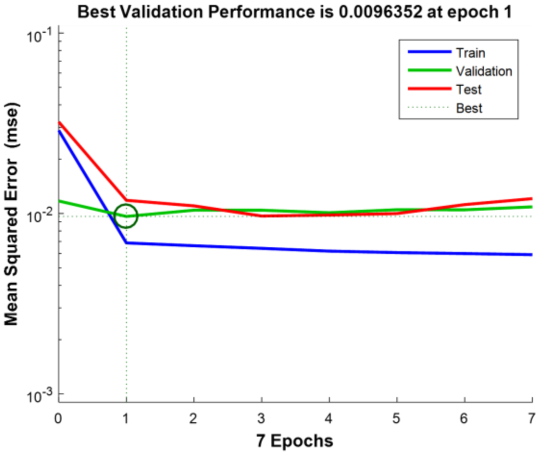
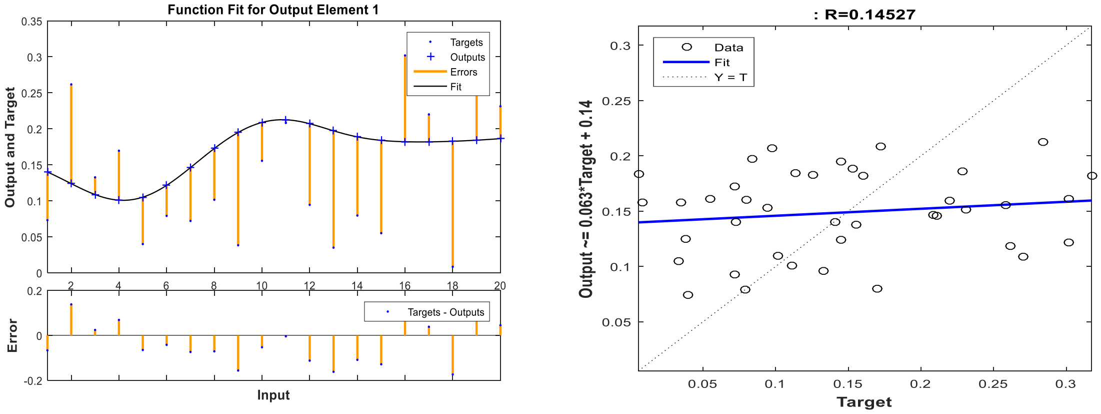


Figure 10. Performance of the used SNN (training, validation and testing).

Figure 12 shows the simulation results on position errors of bilinear, fuzzy interpolation and SNN fitting method. The simulated target (testing) positions on the standard calibration board are scanned from 1 *mm* to 20 *mm* within one cell with a step of 1 *mm*.



(a) (b)

Figure 11. The training state and comparison of outputs and targets.

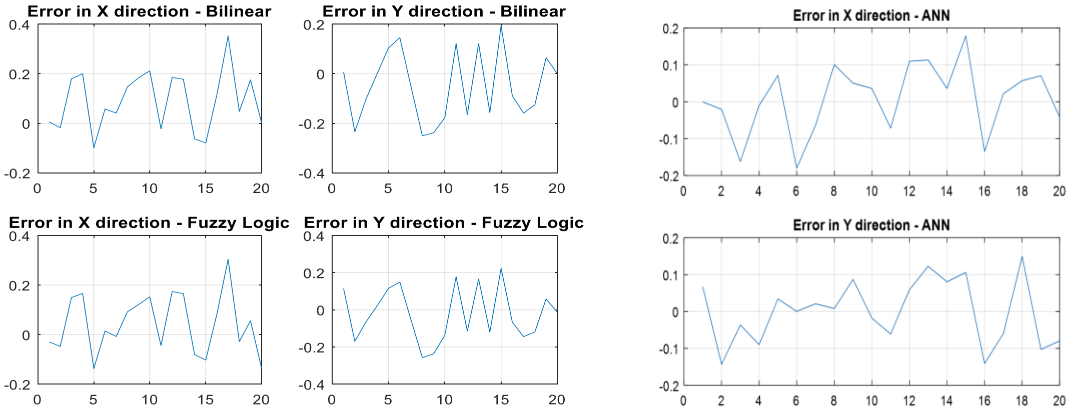


Figure 12. Simulated position errors for the uniform distributed noise.

Figure 13 shows comparisons in mean error, maximum error and STD values among bilinear, fuzzy interpolation and SNN technique in the histograms.

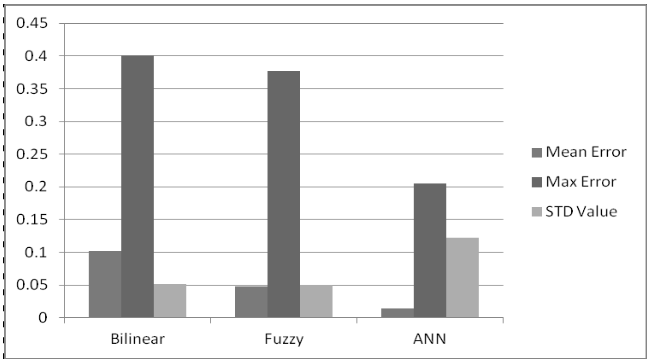


Figure 13. Comparison among bilinear, fuzzy interpolation and SNN.

It can be seen that both mean errors and maximum errors of SNN are much smaller than those of fuzzy interpolation and bilinear methods. For the uniform distributed error, the mean error of the SNN method is approximately 60% to 70% smaller compared with those of bilinear and fuzzy interpolation methods. The maximum errors of the SNN technique is about 40% to 50% smaller than those of bilinear and fuzzy interpolation techniques.

The simulated results show the effectiveness of the SNN fitting method in reducing the position errors in the modeless robot compensation process.

6. Conclusions

A shallow neural network fitting method is presented in this paper. The compensated position errors in a modeless robot calibration can be greatly reduced by the proposed technique. Simulation results demonstrate the effectiveness of the proposed shallow neural network (SNN) fitting method. One typical error model, uniform distributed error, is utilized for comparison and simulation study purpose. This SNN technique is ideal for the modeless robot position compensation, especially the high accuracy (<10 μm) robot calibration process.

References

[1] B. W. Mooring, Z. S. Roth and M. R. Driels, Fundamentals of Manipulator Calibration, John Wiley & Sons, Inc. 1991.

[2] M. A. Penna, Camera Calibration: A Quick and Easy Way to Determine the Scale Factor, IEEE Trans. Pattern Anal. Machine Intell. vol 13, pp. 1240-1245, 1991.

[3] Y. Bai, Design and Implementation of a Control System for a Laser Tracking Measurement System, Ph.D. Dissertation, Florida Atlantic University, Boca Raton, May, 2000.

[4] J. Weng et al, Camera Calibration with Distortion Models and Accuracy Evaluation, IEEE Trans, Pattern Anal. Machine Intell. vol 14, pp. 965-980, 1992.

[5] F. Du and M. Brady, Self Calibration of the Intrinsic Parameters of Cameras for Active Vision Systems, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, NY June, pp. 477-482, 1993.

[6] H. Zhuang and Z. S. Roth, Camera-Aided Robot Calibration, CRC Press, Inc. 1996.

[7] J. Triantafilis, W. T. Ward, I. O. A. Odeh and A. B. McBratney, Creation and Interpolation of Continuous Soil Layer Classes in the Lower Namoi Valley, Soil Science Society of America Journal, vol 65(2), pp. 403-414, 2001.

[8] D. Suzana, J. D. Marceau and M. Claude, Space, time, and dynamics modeling in historical GIS databases: A Fuzzy Logic Approch, Environment & Planning B: Planning & Design, vol 28(4), pp. 545-563, 2001.

[9] F. Song, S. M. Smith, and C. G. Rizk, A Fuzzy Logic Controller Design Methodology for 4D System with Optimal Global Performance Using Enhanced Cell State Space Based Best Estimate Directed Search Method, IEEE Int. Conf. On Systems, Man and Cybernetics, 1999.

[10] Y. Bai, On the Comparison of Model-Based and Modeless Robotic Calibration Based on a Fuzzy Interpolation Method, International Journal of Advanced Manufacturing Technology, vol 31(11-12), pp.1243-1250, 2007.

[11] W. L. Xu, K. H. Wurst, T. Watanabe and S. Q. Yang, Calibrating a modular robotic joint using neural network approach, Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), Orlando, FL, USA, pp. 2720-2725,1994.

[12] H. Wu, W. Tizzano, T. T. Andersen, N. A. Andersen and O. Ravn, Hand-Eye Calibration and Inverse Kinematics of Robot Arm Using Neural Network, Robot Intelligence Technology and Applications, vol 2, pp. 581-591, 2014.

[13] G. Wells, C. Venaille and C. Torras, Vision-based robot positioning using neural networks, Image and Vision Computing, vol 14(10), pp. 715-732, 1996.

[14] H. N. Nguyen, J. Zhou and H. J. Kang, A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network, Neurocomputing, vol 151(3), pp. 996-1005, 2015.

[15] W. Xu, L. Dongsheng and W. Mingming, Complete calibration of industrial robot with limited parameters and neural network, 2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Dec. 17-20, Tokyo, Japan, pp. 103-108, 2016.

[16] R. Zouaoui and H. Mekki, 2D visual servoïng of wheeled mobile robot by neural networks, 2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR), Sousse, Tunisia, pp. 130-133, 2013.

[17] P. Y. Tao and G. Yang, Calibration of industry robots with consideration of loading effects using

Product-Of-Exponential (POE) and Gaussian Process (GP), 2016 IEEE International Conference on Robotics and Automation (ICRA), May 26-30, Seattle, WA, USA, pp. 4380-4385, 2016.

[18] R. C. Luo, H. Wang and M. Kuo, Low cost solution for calibration in absolute accuracy of an industrial robot for ICPS applications, 2018 IEEE Industrial Cyber-Physical Systems (ICPS)*,* St. Petersburg, Russia, May 15-18, pp. 428-433, 2018.

[19] H. Zhuang and X. Wu, Membership Function Modification of Fuzzy Logic Controllers with Histogram Equalization, IEEE Trans. On System, Man, and Cybernetics – Part B: Cybernetics, vol 31(1), 2001.

[20] S. V. Kartalopoulos, Understanding Neural Networks and Fuzzy Logic, IEEE Press, 1996.

Biographical information

Dr. Ying Bai is a full professor in the Dept of Computer Science and Engineering at Johnson C. Smith University located in Charlotte, North Carolina in USA. His special interests include: soft-computing, artificial intelligence, mix-language programming, database programming, and robotic controls and robots calibrations. Since 2003, Dr. Bai has published 14 books with publishers such as Prentice Hall, CRC Press LLC, Springer, Cambridge University Press and Wiley IEEE Press. He also published about 60 peer reviewed international journal and conference papers.

Dr. Dali Wang received his Ph.D. degree in electrical engineering from Florida Atlantic University, Boca Raton, USA. He is currently a full professor of electrical and computer engineering at Christopher Newport University. His research interests include artificial intelligence, digital signal processing, soft computing, control and robotics.