2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)

A Calibration Method of Industrial Robots Based on ELM

Ying Cai[1], Peijiang Yuan[1], Dongdong Chen[1], Doudou Gao[1], Xulei Wu[1], Lei Xue[2], Tianmiao Wang[1]

1. Beihang University, Beijing 100191
E-mail: itr@buaa.edu.cn

2. Shanghai Aircraft Manufacturing Co., Ltd., Shanghai 201324
E-mail: xuelei@comac.cc

Abstract: A calibration method for enhancing absolute position errors of industrial robots is presented by the paper. In this study, the ELM algorithm is utilized for error prediction. Firstly, the relationship between absolute position errors and robot joint angles is built by training samples. Then joint angles of positions to be predicted are provided to predict position errors based on the relationship. Besides, this paper compensates the position coordinates of the predicted positions. All processes implemented in the paper needn't to build the kinematic model. Finally, the experiment is carried out on the KUKA KR210 R2700 industrial robot. Experimental results indicate that the maximum of the spatial position errors is from 1.6866 mm to 0.3565 mm with the reduction of 78.86%, which verifies the method's effectiveness.

Key Words: Industrial Robots, Absolute Position Errors, ELM, Error Prediction, Error Compensation

1 INTRODUCTION

Industrial robots have good extensive applications which require precisely locating, like wielding, cutting, drilling, surgery, et al. Repeatability and absolute accuracy, as the measurement index of industrial robots' positioning accuracy, are the basis of all processes. Nowadays, repeatability can reach ±0.05mm, while absolute accuracy is beyond the range of ±0.5mm, which is the criteria in aviation field [1]. More and more industrial robots are operated by offline programming at present [2]. Generally, industrial robots with only high repeatability are not active enough in high-end manufacturing that also needs high absolute accuracy [3]. So high absolute accuracy has been feverishly pursued by scholars.

Sources causing the loss of absolute accuracy are usually concurrent. Errors caused by link length, non-parallel axis, base misalignment, manufacturing error and assembly error are classified as geometric errors making up ninety percent of all the errors. The other ten percent of errors caused by payload, thermal deflection, gear backlash, servo error, et al, are called non-geometric errors. Therefore, in some researches, geometric errors are in consideration while non-geometric errors are ignored when calibration is taken [4-10].

So far two kinds of calibration method have been developed: model-based calibration and model-free calibration [11]. As for model-based calibration method, Denavit-Hartenberg (DH) model develops a convenience method to find out the deviation of kinematic parameters. Nevertheless, when it comes that adjacent axis are parallel, the solution will be singular. In order to solve the problem, Modified Denavit-Hartenberg (MDH) model is proposed. Compared to DH model, MDH model adds an additional parameter representing the rotation amount around y axis [12]. S model that has six parameters is also a method to overcome the singular problem of DH model [13]. In the model-based calibration, in addition to modeling, parameter identification is another difficulties. The linear least square method [14, 15], non-linear optimization procedure [15], Levenberg-Marquardt algorithm [15], kalman filter [15] and artificial neural networks [16, 17] are all the algorithms for parameter identification. The model-based methods generally ignore small errors when modelling, which results in the non-ideal calibration accuracy. Besides, the model-based methods are not of universality for all kinds of industrial robots.

Another kind of calibration method, model-free method, holds that position errors and robot joint values or robot positions have associations. Approaches, e.g. Fourier polynomials [18], inverse distance weighting [1] and artificial neural network (ANN) [19], have been utilized to predict the position errors. However, Fourier polynomials has limitations because of its low accuracy and complexity. Moreover, inverse distance weighting is only applicable when the movement of the robot space is in a small range. In comparison to the methods mentioned above, ANN, which owns high learning ability and high adaptability, can constantly adjust the weight of the node associated, so that the output can approach the desired ones. So ANN has been developed rapidly to provide higher precision for industrial robots.

Single hidden layer forward neural network (SLFN) has been widely used, which is a traditional ANN [20]. Despite this, SLFN is limited accounted for its slow training, global optimal solution not easy to obtain, and learning rate difficult to choose. Then, extreme learning machine (ELM) is proposed by Huang [21, 22] to solve the problems. Compared to the ordinary SLFN, the ELM algorithm, whose

This research is partially supported by the National Natural Science Foundation of China (No.61375085) and Foundation of Shanghai Aircraft Manufacturing Co., Ltd. (No. 32284001)

generalization is improved, simplifies the training without iteration.

This paper presents a calibration method of robot based on the ELM. As for the remaining part, it consists of three other sections: Section 2 applies the ELM algorithm for predicting the position errors followed by an error compensation. Section 3 provides the experiments and analysis of results. Finally, some conclusions are shown in Section 4.

2 ERROR CALIBRATION BY ELM

The traditional methods of robot positioning calibration calibrates robot's DH parameters through establishing the complex kinematic model, which modifies kinematic geometric parameters of the robot, but does not take into account the non-geometric errors. .In order to avoid the loss of accuracy due to establishing the mathematical model, ELM, a special kind of SLFN that can approximate any non-linear functions depending on its own learning ability, high adaptability and optimal solution, is used for error prediction and compensation of robots in this study.

2.1 ELM: a special kind of SLFN

The structure of ELM that is the same as SLFN, can be expressed as follows (see Figure 1):

Given N different learning sample pair $\boldsymbol{x_j}$ and $\boldsymbol{t_j}$ ($\boldsymbol{x_j}$ belongs to R^n and $\boldsymbol{t_j}$ belongs to R^m) the equation(1) describes the output of ELM (SLFN).

$$f(\boldsymbol{x_j}) = \sum_{i=1}^{k} \boldsymbol{\beta_i} G_i(\boldsymbol{\alpha_i}, b_i, \boldsymbol{x_j}), j = 1, 2, \ldots, N \quad (1)$$

Where k is the number of hidden nodes, $\boldsymbol{\alpha_i}$ and b_i are respectively the input weights and the threshold of the ith hidden node, and the output weights connecting the ith hidden node and the output layer is presented as $\boldsymbol{\beta_i}$. $\boldsymbol{x_j}$, $\boldsymbol{t_j}$, $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$ are defined as:

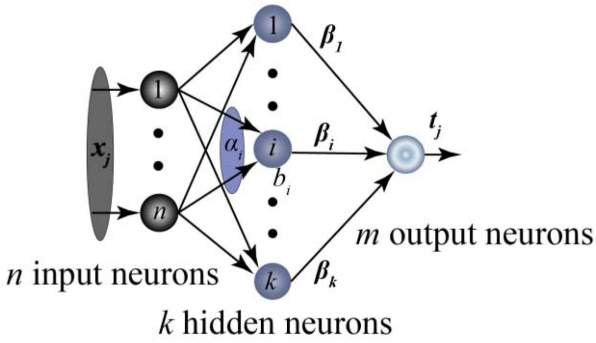

Fig 1. ELM structure

$$\boldsymbol{x_j} = [x_{j1}, x_{j2}, \ldots, x_{jn}]^T \quad (2)$$

$$\boldsymbol{t_j} = [x_{j1}, x_{j2}, \ldots, x_{jm}]^T \quad (3)$$

$$\boldsymbol{\alpha_i} = [\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in}]^T \quad (4)$$

$$\boldsymbol{\beta_i} = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T \quad (5)$$

Besides, G_i ($\boldsymbol{\alpha_i}$, b_i, $\boldsymbol{x_j}$) represents the output of the ith hidden node that associated with $\boldsymbol{x_j}$. Because ELM (SLFN) can be divided into two types: sum function as the hidden function and radical basis function (RBF) as the hidden function, G_i ($\boldsymbol{\alpha_i}$, b_i, $\boldsymbol{x_j}$) has the following calculation formula in order:

$$G(\boldsymbol{\alpha_i}, b_i, \boldsymbol{x_j}) = g(\boldsymbol{\alpha_i x_j} + b_i) \quad (6)$$

$$G(\boldsymbol{\alpha_i}, b_i, \boldsymbol{x_j}) = g(b_i \| \boldsymbol{x_j} - \boldsymbol{\alpha_i} \|) \quad (7)$$

Where g is hidden nodes' activated function.

The alternative form of the matrix form in equation(1) is:

$$\boldsymbol{H\beta} = \boldsymbol{T} \quad (8)$$

Where matrix $\boldsymbol{H}$ is defined as equation(9), matrix $\boldsymbol{\beta}$ and $\boldsymbol{T}$ are presented in the order of equation(10) and equation(11).

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{h}(x_1) \\ \boldsymbol{h}(x_2) \\ \vdots \\ \boldsymbol{h}(x_N) \end{bmatrix} = \begin{bmatrix} G(\boldsymbol{\alpha_1}, b_1, \boldsymbol{x_1}) & \cdots & G(\boldsymbol{\alpha_k}, b_k, \boldsymbol{x_1}) \\ G(\boldsymbol{\alpha_1}, b_1, \boldsymbol{x_2}) & \cdots & G(\boldsymbol{\alpha_k}, b_k, \boldsymbol{x_2}) \\ \vdots & \cdots & \vdots \\ G(\boldsymbol{\alpha_1}, b_1, \boldsymbol{x_N}) & \cdots & G(\boldsymbol{\alpha_k}, b_k, \boldsymbol{x_N}) \end{bmatrix}_{N \times k} \quad (9)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T & \boldsymbol{\beta}_2^T & \cdots & \boldsymbol{\beta}_k^T \end{bmatrix}_{k \times m}^T \quad (10)$$

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T & \boldsymbol{t}_2^T & \cdots & \boldsymbol{t}_N^T \end{bmatrix}_{N \times m}^T \quad (11)$$

$\boldsymbol{H}$ shows the output of the hidden layer, whose per column shows the output of all the N input samples associated with the per hidden node.

Through iterating the parameters step by step, SLFN aims to search for the optimal pairs $\boldsymbol{W}$ (i.e. $\boldsymbol{\alpha}^*$, $\boldsymbol{b}^*$, $\boldsymbol{\beta}^*$) making $E(\boldsymbol{W})$ in equation(12) (i.e. the sum of the squares of errors between the expected and actual values) the smallest.

$$E(\boldsymbol{W}) = \sum_{j=1}^{N} (f(\boldsymbol{x_j}) - \boldsymbol{t_j})^2 \quad (12)$$

Then the ELM algorithm, proposed by Huang [21, 22], simplifies the training sessions of SLFN based on the following two theorems:

Theory 1 [23]: Given SLFN with k hidden nodes and activation function g (mapping from R to R) infinitely differentiable in any region, for N different samples $\boldsymbol{x_j}$ and $\boldsymbol{t_j}$, if the parameter $\boldsymbol{\alpha_i}$ and b_i of hidden layer are selected randomly, according to the continuous probability distribution, there exist $\boldsymbol{\alpha}^*$ and $\boldsymbol{b}^*$ that can make the output matrix $\boldsymbol{H}$ reversible and have the relationship as

$$\left\| \boldsymbol{H}_{N \times k} \boldsymbol{\beta}_{k \times m} - \boldsymbol{T}_{N \times m} \right\| = 0 \quad (13)$$

Theory 2 [23]: Given an arbitrary positive number ε and SLFN with k hidden nodes and activation function g

(mapping from R to R) infinitely differentiable in any region, for N different samples $\boldsymbol{x}_j$ and $\boldsymbol{t}_j$, if the parameter $\boldsymbol{\alpha}_i$ and b_i of hidden layer are selected randomly, according to the continuous probability distribution, there exist $\boldsymbol{\alpha}^*$ and $\boldsymbol{b}^*$ that can make the output matrix $\boldsymbol{H}$ reversible and have the relationship as

$$\left\|\boldsymbol{H}_{N\times k}\boldsymbol{\beta}_{k\times m}-\boldsymbol{T}_{N\times m}\right\|<\varepsilon \tag{14}$$

ELM specifies the number of the hidden nodes (i.e. k) and assigning the learning parameter pairs of the hidden nodes (i.e. $\boldsymbol{\alpha}_i$ and b_i) randomly. It indicates that the connection weights of the hidden nodes (i.e. $\boldsymbol{\beta}$) can be directly solved without iterations by equation(15):

$$\boldsymbol{\beta}=\boldsymbol{H}^{+}\boldsymbol{T} \tag{15}$$

Where $\boldsymbol{H}^+$ is Moore - Penrose generalized inverse matrix.

On the whole, ELM is not only quicker but also more generalized [21, 22].

2.2 Error prediction

It is unnecessary to build the robot kinematic model for the prediction model in this paper. Without calibrating the complex kinematic geometric parameters of the robot, the prediction of the robots' position errors based on ELM treats the robot as a black box system. The position errors of the robot in the desired positions can be obtained by the ELM algorithm, and the absolute positioning errors of the robot can be eliminated by correcting the instruction coordinates of the robot according to the predicted errors.

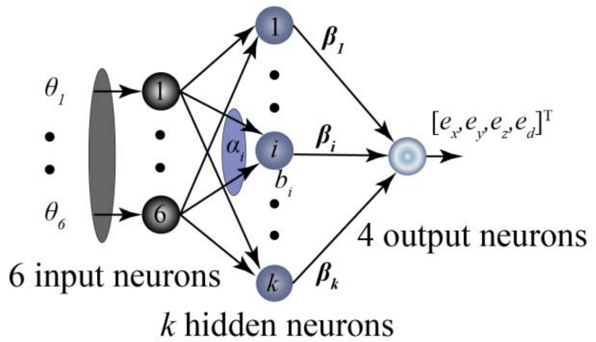

Fig 2. ELM for the error prediction

Before forecast, the relationship between the joint angles (i.e. θ_1, θ_2, θ_3, θ_4, θ_5, θ_6) and position errors (i.e. e_x, e_y, e_z, e_d, where e_x, e_y and e_z is the position error along x, y and z axis, and e_d is the spatial position error) between actual positions and nominal positions should be found by training the sample pairs. For training, the method in this study takes the joint angles as the input of ELM and position errors as the expected output (see Figure 2). During training, the proper k hidden nodes needs to be repeatedly specified until the output of the ELM approaches the desired output. And then through importing the joint values of the positions to be predicted, the position errors can be gained with the relationship.

In the prediction model, multiple solutions of inverse kinematics of the robot manipulator are taken into account. So the joint angles is input not the nominal position coordinates. Besides, the hidden nodes should be assigned appropriately during the repeated training.

2.3 Error compensation

As for error compensation, because the error prediction above is not rely on the kinematic model, so the kinematic parameters need not been modified. In this study, error compensation is performed by compensating the positions coordinates gave to the robot manipulator. Position compensations after error identification are presented as:

$$p_x'(P_0)=p_x(P_0)-e_x(P_0) \tag{16}$$

$$p_y'(P_0)=p_y(P_0)-e_y(P_0) \tag{17}$$

$$p_z'(P_0)=p_z(P_0)-e_z(P_0) \tag{18}$$

Where P_0 means the predicted point, $e_x(P_0)$, $e_y(P_0)$ and $e_z(P_0)$ present the position error of the predicted point along x, y and z axis, $p_x(P_0)$, $p_y(P_0)$ and $p_z(P_0)$ are the nominal position coordinates of the x, y and z axis respectively, and $p_x'(P_0)$, $p_y'(P_0)$ and $p_z'(P_0)$ are the position coordinates after compensation along the x, y, z axis respectively. In the main, in order to reach the specified location, the controller would give the robot manipulator $p_x'(P_0)$, $p_y'(P_0)$ and $p_z'(P_0)$, not the nominal position coordinates.

3 THE EXPERIMENT AND RESULTS

The calibration method proposed above is applied in experiment. And the experiment results verify the accuracy enhanced by the method. This section will show the experiment and the result.

3.1 Experimental system

Figure 3 shows the experimental system. The KUKA KR210 R2700 industrial robot (six degrees of freedom) is selected in the experiment with the ±0.06mm repeatability, which has the open-chain structure. And the Leica AT901 laser tracker whose measurement accuracy is 7.5μm + 3μm/m is also used to measure the actual position of the TCP that is defined as the center of the spherical mounted reflector (SMA) installed at an appointed location of the robot flange. Before the start of the formal experiment, the coordinate systems are established and unified, and the calibration of TCP is also done.

Through the offline programming, the experiment plans out a track in the work space with a total of 2541 measuring points. And then, 1000 points selected randomly as the learning samples of the error model are trained to build the relationship between the input and output. After completing the error model training, another 150 points also picked out at random is inputted in the model for verifying and predicting the position deviation.

978-1-5386-3260-4/17/$31.00 ©2017 IEEE

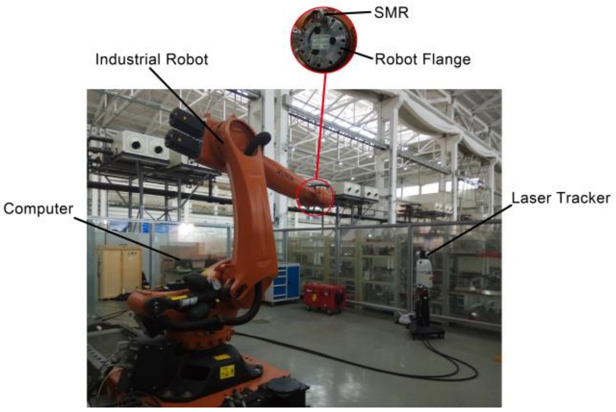

Fig 3. Experimental system

3.2 Experimental results

Figure 4, Figure 5 and Figure 6 show the 150 verified points' position errors of the robot TCP along *x*, *y* and *z* axis before and after the compensation.

It can be seen from Figure 4, Figure 5 and Figure 6 that the position errors are around zero and become smaller after compensation. Compared to the position errors before compensation, the compensation method in the study limits the range of the errors in ±0.3mm. It should be noted that the position accuracy on *x* and *y* axis are improved more greatly than that on *z* axis because the accuracy of *z* axis itself approximates the accuracy after compensation.

Figure 7 is the spatial position errors (i.e. the absolute position errors) of the robot TCP before and after compensation. It is calculated by the equation(19):

$$e_d(P_0) = \sqrt{e_x(P_0)^2 + e_y(P_0)^2 + e_z(P_0)^2} \tag{19}$$

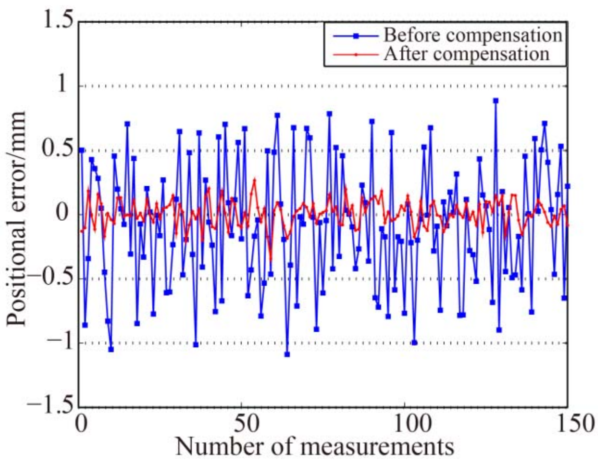

Fig 4. Position errors on *x* axis

Figure 7 shows that on the whole, the absolute position accuracy is improved a lot. The method decreases the amplitude of the absolute position errors obviously.

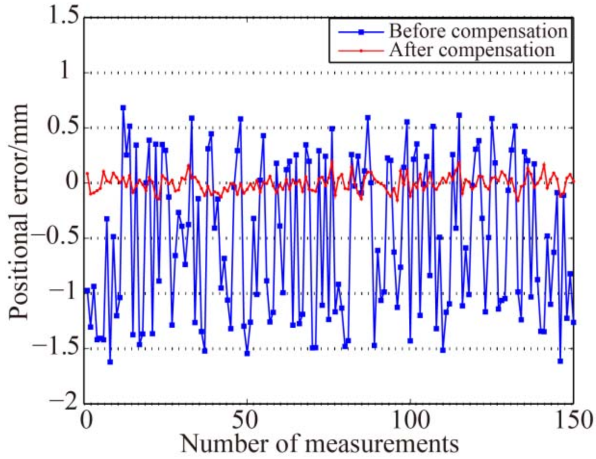

Fig 5. Position errors on *y* axis

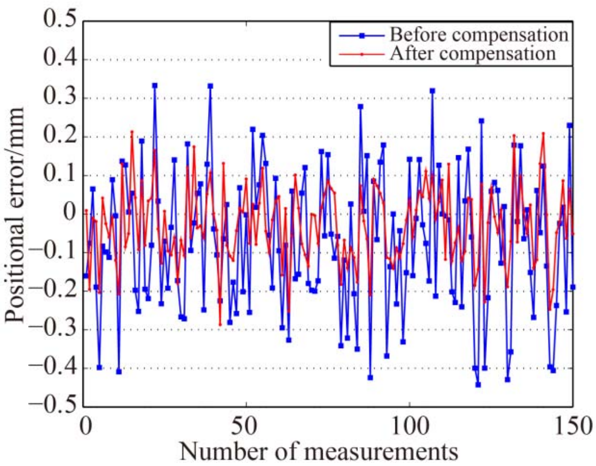

Fig 6. Position errors on *z* axis

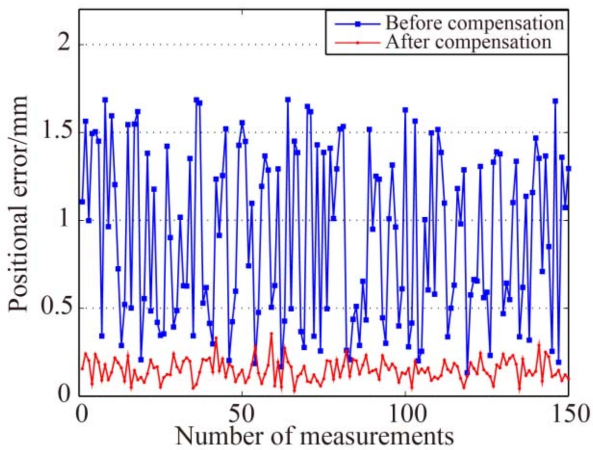

Fig 7. Absolute position errors

Figure 8 shows the absolute position errors after compensation and their projections in XY plane, YZ plane and XZ plane. It's obviously that most of the points in Figure 8 gather in the neighborhood of the zero point of the coordinate system.

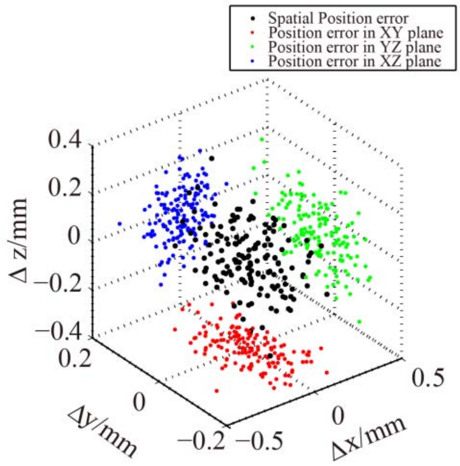

Fig 8. Absolute position errors after compensation and their projections

The experimental statistical results are presented in Table 1. The table shows that the means positional errors along x, y and z axis are all in the negative direction, which may be caused by kinematics parameters. According to Table 1, the maximum of the spatial position errors is from 1.6866 mm to 0.3565 mm with the reduction of 78.86%. Besides, the mean of the absolute position errors is reduced by 83.31% from 0.9024 mm to 0.1506 mm and the standard deviation increases by 87.62%. Overall, the experimental results prove the effectiveness of the method in this paper.

Table1. Experimental Statistical Results

Direction	Maximum/mm		Mean/mm		Std Dev/mm	
	Before	After	Before	After	Before	After
x	0.8866	0.2659	-0.0848	0.0081	0.4811	0.1015
y	0.6828	0.1889	-0.5291	-0.0032	0.7067	0.0730
z	0.3330	0.2130	-0.0708	-0.0288	0.1805	0.0998
d	1.6866	0.3565	0.9024	0.1506	0.4876	0.0604

4 CONCLUSIONS

In this paper, the calibration method proposed is based on ELM, an algorithm that needn't establish precise kinematic model for predicting and compensating position errors, which can eliminate both non-geometric and geometric errors. Experimental results verify that relationship between absolute position errors and joint angles can be built by the method. And the relationship found by the method applies to positional error prediction effectively, which guides the error compensation.

The method has good generality and high speed. Different from the model-based method (i.e. DH model, MDH model and S model), the method proposed in this study is not for individual robot, that is to say, the method suits the most robot. And the method considers the multiplicity of the inverse kinematics, which embodies on that the inputs are the robot joint angles not the position coordination. The presented method has the wide application prospect in automatic assembly, aeronautic drilling, welding, surgery aided by robots, et al. The method can provide high absolute position accuracy for the guarantee of efficiency.

Also, the method is limited because it needs a multiplicity of training samples. That is to say, sampling precise can influence the prediction and compensation. In addition to that, the number of hidden nodes is also the influence factor, which should be selected appropriately during training. In consequence, how to improve the sampling accuracy and decide the proper hidden nodes' number are problems that should be solved in future works.

REFERENCES

[1] P. Hong, W. Tian, D. Q. Mei, Robots' accuracy compensation technique based on variable parameters using space grids, Robot, Vol.37, No.3, 327-335, 2015.

[2] Z. Pan, J. Polden, N. Larkin, et al, Recent progress on programming methods for industrial robots, Robotics and Computer-Integrated Manufacturing, Vol.28, No.2, 87-94, 2012.

[3] D. S. Wang, J. N. Chi, A Survey of Kinematic Calibration of Robot, Application Research of Computers, Vol.24, No.9, 8-11, 2007.

[4] J. Denavit, R. S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, ASME J. Appl. Mech., 215-221, 1955.

[5] J.J. Craig, Introduction to robotics: mechanics and control, Upper Saddle River, Pearson Prentice Hall, 2005.

[6] J. M. Renders, E. Rossignol, M. Becquet, & R. Hanus, Kinematic calibration and geometrical parameter identification for robots. IEEE Transactions on robotics and automation, Vol.7, No.6, 721-732, 1991.

[7] Ikits, Milan and J. M. Hollerbach, Kinematic calibration using a plane constraint, Robotics and Automation, 1997 Proceedings., 1997 IEEE International Conference on. Vol. 4. IEEE, 1997.

[8] R. He, Y. Zhao, S. Yang, Kinematic-parameter identification for serial-robot calibration based on POE formula. IEEE Transactions on Robotics, Vol.26, No.3, 411-423, 2010.

[9] H. Wang, S. Shen, X. Lu, A screw axis identification method for serial robot calibration based on the POE model, Industrial Robot: An International Journal, Vol.39, No.2 146-153, 2012.

[10] G. Antonelli, S. Chiaverini, & G. Fusco, A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation, IEEE Transactions on Robotics, Vol.21, No.5, 994-1004, 2005.

[11] Y. Zeng, W. Tian, W. Liao, Positional error similarity analysis for error compensation of industrial robots, Robotics and Computer-Integrated Manufacturing, Vol.42, 113-120, 2016.

[12] G. Gatti, G. Danieli, A practical approach to compensate for geometric errors in measuring arms: application to a six-degree-of-freedom kinematic structure, Measurement science and technology, Vol.19, No.1, 015107, 2007.

[13] H. W. Stone, A. C. Sanderson, Statistical performance evaluation of the S-model arm signature identification technique, Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on. IEEE, 939-946, 1988.

[14] Y. Wu, A. Klimchik, S. Caro, B. Furet, & A. Pashkevich, Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments, Robotics and Computer-Integrated Manufacturing, Vol.35, 151-168, 2015.

[15] A. Y. Elatta, L. P. Gen, F. L. Zhi, et al. An overview of robot calibration, Information Technology Journal, 2004, Vol.3, No.1, 74-78, 2004.

[16] H. Wu, W. Tizzano, T.T. Andersen, et al. Hand-eye calibration and inverse kinematics of robot arm using neural network, Robot Intelligence Technology and Applications 2. Springer International Publishing, 581-591, 2014.

[17] J. Q. Xuan, S. H. Xu, Review on kinematics calibration technology of serial robots, International journal of precision engineering and manufacturing, Vol.15, No.8, 1759-1774, 2014.

[18] Y. Takeda, G. Shen, & H. Funabashi, A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a Fourier series. Journal of Mechanical Design, Vol.126, No.5, 856-865, 2004.

[19] J. H. Jang, S. H. Kim, Y. K. Kwak, Calibration of geometric and non-geometric errors of an industrial robot, Robotica, Vol.19, No.3, 311-321, 2001.

[20] R. MA, Y.S. Liu, Y. H. Du, An intrusion detection method based on neural network expert system, Computer Engineering and Applications, Vol.40, No.2, 151-153, 2004.

[21] G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. IEEE, Vol.2, 985-990, 2004.

[22] G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: theory and applications, Neurocomputing, Vol.70, No.1, 489-501, 2006.

[23] C. Cortes, V. Vapnik, Support vector networks, Machine Learning, Vol.20, No.3, 273-297, 1995.

978-1-5386-3260-4/17/$31.00 ©2017 IEEE