Research on Robot Accuracy Compensation Method Based on PSO-ELM (Postprint)

Authors: Feng Yuming, Dong Xiucheng, Jin Tao

Date: 2018-06-19T00:00:00+00:00

Abstract

To improve robot positioning accuracy, traditional neural network-based robot accuracy compensation methods are enhanced. Two accuracy compensation methods based on Particle Swarm Optimization-Extreme Learning Machine (PSO-ELM) models are adopted to compensate robot joint coordinates and Cartesian coordinates, respectively. Simulation case analyses and comparisons are performed for both methods, and they are further contrasted with the Genetic Algorithm-Extreme Learning Machine (GA-ELM) model. The simulation results indicate that the PSO-ELM robot accuracy compensation method for Cartesian coordinate compensation is superior to other compensation methods and achieves higher prediction accuracy.

Full Text

Preamble

Title: Research on Methods of Robot Accuracy Compensation Based on PSO-ELM

Authors: Feng Yuming, Dong Xiucheng, Jin Tao
(Signal & Information Processing Laboratory, Xihua University, Chengdu 610039, China)

Abstract: To improve robot positioning accuracy, this paper enhances traditional neural network-based robot accuracy compensation methods. Two accuracy compensation approaches based on Particle Swarm Optimization-Extreme Learning Machine (PSO-ELM) models are employed to compensate for robot joint coordinates and Cartesian coordinates. Both methods are analyzed and compared through simulation examples, and contrasted with a Genetic Algorithm-optimized Extreme Learning Machine (GA-ELM) model.

Simulation results demonstrate that the PSO-ELM robot accuracy compensation method for Cartesian coordinates outperforms other compensation methods and achieves higher prediction accuracy.

Keywords: robot; accuracy compensation; extreme learning machine; PSO-ELM

0 Introduction

In recent years, robots have been widely employed in high-precision applications such as offline programming and robot-assisted surgery. However, due to manufacturing tolerances, assembly errors, structural deformation, and other factors, the actual kinematic model parameters of robots deviate from their nominal values, significantly reducing positioning accuracy. Therefore, necessary calibration procedures must be performed before robot deployment to enhance positioning precision.

Error sources in robot kinematic models can be categorized into two types: geometric parameter errors (e.g., link length and link twist errors) and non-geometric errors (e.g., gear backlash, link and joint compliance). Some studies have focused on model establishment and geometric parameter identification while neglecting non-geometric errors, assuming their impact on robot positioning error is minimal. However, non-geometric errors still affect robot positioning accuracy. Consequently, some researchers have proposed real-time error compensation methods based on neural networks, which can be broadly classified into two categories: one uses joint angles and corresponding joint angle errors as BP neural network inputs and outputs to obtain error values at arbitrary joint angles, achieving position error compensation by correcting joint angles; the other uses joint angles and corresponding actual coordinate position deviations as BP neural network inputs and outputs to improve robot positioning accuracy through joint angle compensation.

Although traditional BP neural networks demonstrate some effectiveness in real-time robot compensation, they suffer from inherent model limitations, including susceptibility to local optima and long training times, which compromise real-time compensation efficiency. Extreme Learning Machine (ELM), proposed by Huang et al. to address the inherent drawbacks of conventional neural network algorithms, is a single-hidden-layer feedforward neural network learning algorithm. Unlike traditional BP algorithms, ELM randomly generates connection weights between input and hidden layers and thresholds of hidden layer neurons. Due to its extremely fast learning speed and excellent generalization performance, ELM has been widely applied in pattern recognition, computer vision, data mining, signal processing, and control systems.

This paper improves upon the two aforementioned neural network-based robot accuracy compensation methods by employing a PSO-ELM model that combines Particle Swarm Optimization (PSO) with Extreme Learning Machine (ELM). Additionally, a GA-ELM model combining Genetic Algorithm (GA) with ELM is introduced for comparison, followed by a comparative analysis of the improved methods' effectiveness.

---

1 Extreme Learning Machine Model

The ELM network training adopts a single hidden-layer feedforward structure. Assuming there are n input layer neurons and o output layer neurons representing n input variables and o output variables, respectively, with h hidden layer neurons, and given N distinct samples $\{(x_i, t_i) | x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^o, i = 1, 2, ..., N\}$, the ELM training model is shown in [Figure 1: see original paper].

The ELM network model can be expressed as:

$$\sum_{i=1}^{h} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad j = 1, 2, ..., N$$

where $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is the weight vector connecting all input layer neurons to the i-th hidden layer neuron, b_i is the threshold of the i-th hidden layer neuron, $w_i \cdot x_j$ denotes the inner product of w_i and x_j, $g(\cdot)$ is the activation function, and $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{io}]^T$ is the weight vector connecting the i-th hidden layer neuron to all output neurons. The cost function can be expressed as:

$$C(W, b, \beta) = \sum_{j=1}^{N} \left(\sum_{i=1}^{h} \beta_i g(w_i \cdot x_j + b_i) - t_j \right)^2$$

where $W = [w_1, w_2, ..., w_h]$ and $b = [b_1, b_2, ..., b_h]$. The goal of ELM training is to find optimal W, b, and β that minimize the error between output values and actual values. Therefore, W, b, and β take the following form:

$$H\beta = T$$

where H is the hidden layer output matrix of samples, and T is the target matrix of the sample set. Equation (3) is equivalent to finding the least squares solution of $H\beta = T$, i.e., finding the optimal weight $\hat{\beta}$ that minimizes the cost function, where $\hat{\beta} = H^+ T$ and H^+ is the Moore-Penrose generalized inverse matrix of H.

According to the basic principle of ELM, the input layer weight matrix W and hidden layer threshold matrix b are randomly generated. Therefore, with a fixed ELM model structure, this randomness can lead to large prediction errors. Since

PSO exhibits excellent global optimization capability, it can find optimal initial W and b for the ELM model, thereby obtaining an optimal ELM model.

---

2 PSO-Optimized ELM Algorithm

The Particle Swarm Optimization algorithm was proposed by Eberhart and Kennedy based on social behavior theory in populations. As an emerging evolutionary algorithm originating from bird flock foraging simulation, it features simple principles and easy implementation for optimal identification of multiple parameters.

The main steps of the PSO-ELM algorithm are as follows:

a) **Experimental data preprocessing**: Divide experimental data into training and testing sets, and perform normalization.

b) **Initialization**: Initialize particle swarm parameters and ELM model network structure parameters. The position of the j-th particle X_j represents the connection weights between input layer neurons and hidden layer neurons, as well as the thresholds of hidden layer neurons.

c) **Finding initial extrema**: Substitute particle positions X_j and training sample data into the ELM model to obtain output predictions, thereby calculating particle fitness values. Identify individual and global extrema while recording their positions and fitness values. The particle fitness function is defined as:

$$f = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

where $\hat{Y}_i$ is the ELM output prediction for training samples, Y_i is the true value, and N is the number of training samples.

d) **Inertia weight selection**: After several iterations, each particle' s position gradually converges to the optimal position. To enhance global search capability in early iterations and local optimization capability in later iterations, the inertia weight ω decreases linearly with iteration number k:

$$\omega(k) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{J_{\max}} \times k$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum inertia weights, and $J_{\max}$ is the maximum number of iterations.

e) **Iterative optimization**: During each iteration, particles update their velocities and positions based on individual and global extrema, ultimately obtaining the optimal individual's fitness value and corresponding particle position.

f) **PSO-ELM model completion**: The particle position corresponding to the optimal individual represents the optimal initial weights and thresholds for ELM. Substituting these into the ELM model, the testing set samples are used for model testing and performance evaluation.

The PSO-ELM algorithm flowchart is shown in [Figure 2: see original paper].

3.1 ABB Robot Geometric Model

This paper employs the DH method to establish the kinematic model of the ABB robot. The structural diagram of the ABB robot is shown in [Figure 3: see original paper]. After establishing coordinate system 0, coordinate system i is defined using coordinate system $i-1$ starting from coordinate system 1. The homogeneous transformation matrix between two adjacent links' DH coordinate systems is:

$$^{i-1}T_i = \mathrm{Rot}(z, \theta_i) \cdot \mathrm{Trans}(0, 0, d_i) \cdot \mathrm{Trans}(a_i, 0, 0) \cdot \mathrm{Rot}(x, \alpha_i)$$

which expands to:

$$^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where a_i, α_i, d_i, and θ_i represent the link length, link twist, link offset, and joint angle of the i-th axis, respectively. The standard D-H parameter values for the ABB robot are given in , where a_i and d_i are in mm, and α_i and θ_i are in rad.

Multiplying the individual link transformation matrices yields the transformation matrix of the manipulator end-effector relative to the base coordinate system:

$$^0T_6 =^0 T_1 \cdot^1 T_2 \cdot^2 T_3 \cdot^3 T_4 \cdot^4 T_5 \cdot^5 T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The end-effector position vector can be expressed as $P = [p_x, p_y, p_z]^T$. The end-effector position error can be represented as $\Delta X = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$.

If superscript letters N denote theoretical values and letters R denote actual values, the total position error can be expressed as $\Delta X = \sqrt{(p_x^N - p_x^R)^2 + (p_y^N - p_y^R)^2 + (p_z^N - p_z^R)^2}$.

3.2 Experimental Data Acquisition

Assuming the geometric errors of the standard D-H parameter values for the ABB robot in are as shown in , where Δa_i and Δd_i are in mm, and $\Delta \alpha_i$ and $\Delta \theta_i$ are in rad, the actual D-H parameters of the ABB robot can be obtained as: $a_i^R = a_i^N + \Delta a_i$, $\alpha_i^R = \alpha_i^N + \Delta \alpha_i$, $d_i^R = d_i^N + \Delta d_i$, $\theta_i^R = \theta_i^N + \Delta \theta_i$.

Based on the variation ranges of each robot joint angle and actual operational requirements, 5,400 different joint angle combinations are generated, with 5,000 groups serving as the training sample set for the ELM model and the remaining 400 groups as test data.

3.3 PSO-ELM Robot Accuracy Compensation Method for Joint Coordinates

The PSO-ELM robot accuracy compensation model for joint coordinates uses the robot' s theoretical joint angles as inputs to the PSO-ELM model and joint angle errors as outputs for training. The training process block diagram is shown in [Figure 4: see original paper]. The theoretical joint angles θ^N are calculated through forward kinematics (using theoretical D-H parameters) to obtain the theoretical end-effector pose X^N. This theoretical pose is then processed through inverse kinematics (using actual D-H parameters) to solve for actual joint angles θ^R, yielding joint angle errors $\Delta\theta = \theta^N - \theta^R$. Therefore, the PSO-ELM model has 6 input layer nodes and 6 output layer nodes.

The positioning process for the robot end-effector can be performed using the steps described in [Figure 5: see original paper]. Here, θ_{mod} represents the corrected joint angles after compensation, and X_{mod} is the predicted position. The goal is to move the robot end-effector to the theoretical position X^N in its workspace, where ε represents the upper limit of robot accuracy.

To demonstrate the model' s accuracy and superiority, it is compared with the GA-ELM model, which uses genetic algorithms to find optimal W and b for ELM (see reference [14] for detailed algorithm flow). After extensive experimentation, the PSO-ELM parameters are determined as: 68 hidden layer nodes, population size of 40, 300 iterations, $C_1 = C_2 = 2$, $V_{\max} = 1$, $V_{\min} = -1$, maximum inertia weight $\omega_{\max} = 0.9$, and minimum inertia weight $\omega_{\min} = 0.4$. The GA-ELM parameters are: 58 hidden layer nodes, population size of 100, 350 iterations, crossover probability of 0.6, and mutation probability of 0.01. Both models use the "sig" activation function for the hidden layer. Using 5,000 training samples and 400 test samples, the position errors after robot inverse kinematics accuracy compensation are shown in [Figure 6: see original paper].

To verify the generality of input data, 200 random data points are generated within each joint angle' s variation range based on actual operational requirements. These are fed into the trained PSO-ELM model for simulation, with position errors shown in [Figure 7: see original paper].

3.4 PSO-ELM Robot Accuracy Compensation Method for Cartesian Coordinates

The PSO-ELM robot accuracy compensation model for Cartesian coordinates uses the robot' s theoretical joint angles as inputs to the PSO-ELM model and the position error between actual and theoretical positions as outputs for network training. The training process block diagram is shown in [Figure 8: see original paper]. The theoretical joint angles θ^N are calculated through forward kinematics (using theoretical D-H parameters) to obtain the theoretical end position X^N, and through forward kinematics (using actual D-H parameters) to obtain the actual end position X^R, yielding position error $\Delta X = X^N - X^R$. Thus, the PSO-ELM model has 6 input layer nodes and 3 output layer nodes.

The trained PSO-ELM model is then used to compensate for robot position errors. The detailed steps for robot end-effector positioning are shown in [Figure 9: see original paper]. Since the ABB robot' s position depends only on the first three joint angles θ_1, θ_2, θ_3, position compensation only targets these three joints. $J(\theta)$ is the Jacobian matrix of the robot joint variables, and θ_{mod} represents the compensated joint angles. The experimental objective is to move the robot end-effector to the theoretical position X^N in its workspace, where ε is the upper limit of robot accuracy.

To demonstrate accuracy and superiority, comparison with GA-ELM is again performed. After extensive experimentation, PSO-ELM parameters are set as: 65 hidden layer nodes, population size of 50, 300 iterations, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$ (other parameters same as above). GA-ELM parameters are: 73 hidden layer nodes, population size of 80, 400 iterations, crossover probability of 0.6, and mutation probability of 0.01. Both use the sig activation function. Using 5,000 training samples and 400 test samples, position errors after robot inverse kinematics accuracy compensation are shown in [Figure 10: see original paper].

To verify input data generality, the same 200 random data points from Section 3.3 are fed into the trained PSO-ELM model, with position errors shown in [Figure 11: see original paper].

3.5 Results Analysis and Comparison of Two Accuracy Compensation Methods

Comparative analysis of the two compensation methods yields and based on [Figure 6: see original paper], [Figure 7: see original paper], [Figure 10: see original paper], and [Figure 11: see original paper].

TABLE:3 Comparison of Position Accuracy for Two Methods (Test Data Comparison, 400 Groups)

Method	Max Error (mm)	Min Error (mm)	Avg Error (mm)
Joint Coordinate GA-ELM			
Joint Coordinate PSO-ELM			
Cartesian Coordinate GA-ELM			
Cartesian Coordinate PSO-ELM			

TABLE:4 Comparison of Position Accuracy for Two Methods (Random Data Comparison, 200 Groups)

Method	Max Error (mm)	Min Error (mm)	Avg Error (mm)
Joint Coordinate GA-ELM			
Joint Coordinate PSO-ELM			
Cartesian Coordinate GA-ELM			
Cartesian Coordinate PSO-ELM			

Analysis of both test and random data reveals that compensated robot position errors are significantly lower than uncompensated errors, with Cartesian coordinate compensation producing lower errors than joint coordinate compensation. In both methods, PSO-ELM outperforms GA-ELM. Compared to uncompensated data, joint coordinate GA-ELM reduces average position error by 69.48%, joint coordinate PSO-ELM by 72.57%, Cartesian coordinate GA-ELM by 92.60%, and Cartesian coordinate PSO-ELM by 95.06%. For random data, the corresponding reductions are 69.17%, 71.58%, 91.69%, and 94.88%, respectively.

---

4 Conclusion

To improve the absolute positioning accuracy of industrial robots in practical applications, this paper proposes two methods: robot joint coordinate compensation and Cartesian coordinate compensation. The Particle Swarm Optimization-Extreme Learning Machine (PSO-ELM) model is employed to predict joint coordinate errors and Cartesian coordinate errors, and compared with the Genetic Algorithm-Extreme Learning Machine (GA-ELM) model. In both simulation experiments, PSO-ELM demonstrates superior prediction performance over GA-ELM. Moreover, the Cartesian coordinate compensation method significantly outperforms the joint coordinate compensation method. The PSO-ELM accuracy compensation method for Cartesian coordinates achieves the highest prediction accuracy and exhibits excellent robustness.

References

[1] Ma Le, Bazzoli P, Sammons P, et al. Modeling and calibration of high-order joint-dependent kinematic errors for industrial robots [J]. Robot Comput-Integr Manuf, 2017, 0: 1–15.

[2] Liu Zhi, Zhao Zhengda, Xie Ying, et al. Kinematic calibration and compensation for a robot with structural deformation [J]. Robot, 2015, 37 (3): 376-384.

[3] Palmieri G, Palpacelli M, Carbonari L, et al. Vision-based kinematic calibration of a small-scale spherical parallel kinematic machine [J]. Robot Comput-Integr Manuf, 2018, 49: 162–169.

[4] Fang Lijin, Dang Pengfei. Kinematic calibration method of robots based on quantum-behaved particle swarm optimization [J]. Journal of Mechanical Engineering, 2016, 52 (7): 23-30.

[5] Zhang Xu, Zheng Zelong, Qi Yong. Parameter Identification and calibration of D-H model for 6-DOF serial robots [J]. Robot, 2016, 38 (3): 360-370.

[6] Švaco M, Šekoranja B, Šuligoj F. Calibration of an industrial robot using a stereo vision system [J]. Procedia Engineering. 2014, 69: 459-463.

[7] Park I W, Lee B J, Cho S H, et al. Laser-based kinematic calibration of robot manipulator using differential kinematics [J]. IEEE-ASME T MECH. 2012, 17 (6): 1059-1067.

[8] Wang Dongshu, Chi Jiannan. Survey on robot kinematics calibration [J]. Application Research of Computers, 2007, 24 (9): 8-11.

[9] Aoyagi S, Kohama A, Nakata Ya, et al. Improvement of robot accuracy by calibrating kinematic model using a laser tracking system, compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2010: 5660–5665.

[10] Wang Dali, Bai Yanhua, Zhao Jiying. Robot manipulator calibration using neural network and a camera-based measurement system [J]. Trans Inst Meas Control. 2010, 32 (4): 105-121.

[11] Zhong X L, Lewis J M, Rea H. Neuro-accuracy compensator for industrial robots [C]// Proc of IEEE International Conference on Neural Networks. 1994:

2797-2802.

[12] Nguyen H N, Zhou Jian. A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network [J]. Neurocomputing, 2015, 151: 996–1005.

[13] Huang Guangbin, Zhu Qinyu, Siew C K. Extreme learning machine: theory and applications [J]. Neurocomputing, 2006, 70 (1): 489-501.

[14] Mei Yi, Sun Quanlong, Yu Lihua, et al. Grain size prediction of aluminum alloy die castings based on GA-ELM [J]. Acta Metallurgica Sinica, 2017, 53 (9): 1125-1132.

[15] Liu Dong, Liu Chunlei, Fu Qiang, et al. ELM evaluation model of regional groundwater quality based on the crow search algorithm [J]. Ecological Indicators, 2017, 81: 302-314.

[16] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. 1995: 1942-1948.

Note: Figure translations are in progress. See original paper for figures.