IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 2, 2021, accepted September 17, 2021, date of publication September 23, 2021,
date of current version October 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115120

Robot Calibration Method Based on Extended Kalman Filter–Dual Quantum Behaved Particle Swarm Optimization and Adaptive Neuro-Fuzzy Inference System

HUNG QUANG CAO, HA XUAN NGUYEN, TY TRUNG NGUYEN, VINH QUANG NGUYEN, AND JAE WOOK JEON, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jae Wook Jeon (jwjeon@skku.edu)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korean Government [Ministry of Science and ICT (MSIT)] under Grant 2020R1A2C3011286.

ABSTRACT This paper combines the extended Kalman filter (EKF), dual quantum-behaved particle swarm optimization (DQPSO), and adaptive neuro-fuzzy inference system (ANFIS) to propose a novel robot calibration method. Robot precision is influenced by kinematic and non-kinematic error sources. The EKF algorithm is robust for the nonlinear system with Gaussian noise to identify kinematic parameter errors for kinematic calibration. However, if inappropriate covariance matrices are selected, the EKF algorithm may converge to an incorrect solution. To increase the effectiveness of the EKF algorithm, we propose a DQPSO algorithm that consists of the QPSO-1 and QPSO-2 algorithms. The QPSO-1 algorithm adapts the covariance matrices of measurement noise and process noise, while the QPSO-2 algorithm optimizes the kinematic parameter errors estimated by the EKF algorithm. In addition, the used ANFIS predicts and compensates the non-kinematic error for non-kinematic calibration. Experiments have been performed on a five-bar parallel robot to confirm the effectiveness of the proposed method. The experimental results demonstrate that the proposed method significantly improves the positional accuracy, and is better than the previous methods.

INDEX TERMS Adaptive neuro-fuzzy inference system, extended Kalman filter, dual quantum-behaved particle swarm optimization, five-bar parallel robot.

I. INTRODUCTION

Nowadays, robots are widely used in the industry for many applications that require high absolute positioning accuracy, such as the mounting, handling, and assembly of parts. However, kinematic errors resulting from existing manufacturing errors, assembly errors, and actuator errors, as well as non-kinematic errors, such as joint clearance, joint compliance, and temperature drift, cause the robot's actual position to deviate from the target position. Therefore, the robot may not achieve high accuracy. Hence, kinematic and non-kinematic calibrations are executed to improve the positional accuracy.

The main procedures for kinematic calibration include four steps: modeling, measurement, identification, and compensation. For modeling, the kinematic error model is established based on different mathematical tools, such as the derivative of the kinematic model [1], the product of exponential (PoE) [2], screw theory [3], bond graph theory [4], and the vector loop method [5]- [7]. In the case of the parallel robot, the vector loop method is widely utilized to establish the kinematic error model and is less complex than other methods. It is suitable for error modeling of the five-bar parallel robot. For measurement, data collection is carried out by various measurement instruments: laser tracker [8], coordinate measuring machine (CMM) [9], and FARO arm [10]. Although these precision measuring devices provide high accuracy, they are nevertheless high-cost, and relatively difficult to set up and operate. The identification process of kinematic calibration is executed by applying nonlinear optimization algorithms to find the optimum value of the kinematic parameter error. Wu [11] presented an overview of the identification

The associate editor coordinating the review of this manuscript and approving it for publication was Shun-Feng Su.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

of the relevant parameters. Many algorithms have been implemented to identify kinematic parameters, such as least-square estimation (LSE) [12], [13], Levenberg-Marquardt (LM) algorithm [14], maximum-likelihood estimation (MLE) [15], and Tikhonov regularization [16]. Among them, the LSE method is the most widely known and used. LSE usually presents fast convergence and lower computation costs. However, it is sensitive to measurement noise, and the convergence of this method is not always guaranteed. Intelligent algorithms, such as the particle swarm optimization (PSO) [17], genetic algorithm (GA) [18], and the quantum-behaved particle swarm optimization (QPSO) [19], can search for flexibility in multidimensional non-linear space. Nevertheless, when handling high-dimensional parameter identification problems, these algorithms would get stuck in a local optimum. The hybrid method based on EKF and particle filter (PF) [20] was used to identify the kinematic parameter of a 6-axis robot. Its effects are compared with those of the LSE algorithm; however, the PF algorithm requires complex computation. Park *et al.* [21] proposed the EKF to estimate the kinematic parameter error of robot manipulators. The EKF algorithm is the most popular nonlinear filter, and has been widely applied in system-state estimation for uncertain measurements. It is also suitable for kinematic parameter identification, because in a real robot, measurement and process noises are inevitable. However, for a highly nonlinear system with non-Gaussian noise, the EKF algorithm may converge to an incorrect solution.

Non-kinematic calibration compensates for the non-kinematic error source. The radial basis function neural network (RBFNN) [22] was used to predict and compensate for the positional error of the industrial robot. RBFNN is suitable for function approximation, but requires good coverage of the input space. Artificial neural network (ANN) [23]- [26], using the back-propagation algorithm to train the neural network, established a nonlinear relationship between the joint variables and the robot's positional error. The back-propagation neural network (BPNN) [24], trained using the gradient descent method, predicted the leg length error of the parallel robot. BPNN [26] was used to compensate for the effect of the non-kinematic error of a PUMA robot. However, the BPNN has disadvantages such as long training time, slow convergence, and it can fall into local minima. The genetic algorithm and deep neural network (GA-DNN) [27] was developed to achieve high accuracy positioning performance of the robot under any external payload. The teaching learning-based optimization neural network (TLBO-NN) [28] predicted the nonlinear positional error of the robot manipulator. While this method is easy to implement, it can become trapped in local minima.

To overcome the disadvantages of the existing research, we propose a new calibration method. The novelty of this paper is the proposal of a new method that is a combination of EKF, DQPSO, and ANFIS, to improve the absolute position accuracy of the robot's end-effector. The DQPSO algorithm includes the QPSO-1 and QPSO-2 algorithms. The main contribution of this paper can be summarized as follows:

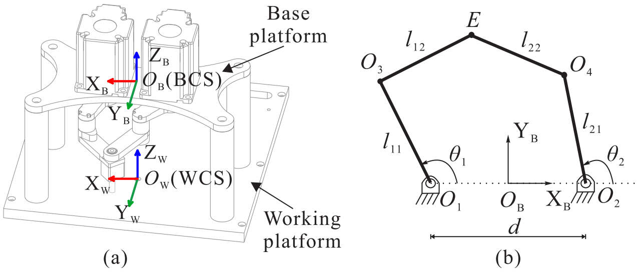

FIGURE 1. Schematic of the 2-DOF five-bar parallel robot.

1) The QPSO-1 algorithm was used to optimize the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the process noise and measurement noise of EKF algorithm. In addition, EKF algorithm estimates the preliminary kinematic parameter error of the robot for kinematic calibration.
2) The QPSO-2 algorithm was proposed to optimize the estimated kinematic parameter errors. The QPSO algorithm has a good search ability to solve the optimization problem.
3) ANFIS was proposed to compensate for the non-kinematic error for non-kinematic calibration. ANFIS establishes the nonlinear regression model that represents the relationship between the joint variables and the positional error after kinematic calibration.

The rest of the paper is organized as follows: Section II describes the kinematic model, the kinematic error model, and the proposed approach to robot calibration. Section III explains the EKF-DQPSO algorithm that determines the kinematic parameter errors for kinematic calibration. Section IV presents ANFIS to compensate for the non-kinematic error. Section V presents the experimental results. Finally, Section VI concludes the paper.

II. KINEMATIC ERROR MODEL AND THE PROPOSED APPROACH

A. KINEMATIC MODEL

Fig. 1(b) shows a schematic of the five-bar parallel robot, which has four links with link lengths l_{11}, l_{12}, l_{21}, and l_{22}. Fig. 1(a) introduces the two reference frames, the world-coordinate system (WCS) that is fixed on the working platform, and the base-coordinate system (BCS) that is fixed on the base platform. The original position of the BCS is located at the point that is the midpoint of O_1O_2. The distance between the axes of the actuated joint (O_1, O_2) is denoted by d. The position and the orientation of BCS with respect to WCS in the two-dimensional coordinate system XY are denoted by ${}^{WCS}\boldsymbol{P}_{BCS} = [x_0, y_0]^T$ and ${}^{WCS}\boldsymbol{R}_{BCS} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$, respectively, where α is the rotation angle of the BCS about the $O_W Z_W$ axis. The position of the end-effector E with respect to the WCS is calculated as:

$$\begin{aligned}\boldsymbol{P}_E &= {}^{WCS}\boldsymbol{P}_{BCS} + {}^{WCS}\boldsymbol{R}_{BCS}{}^{BCS}\boldsymbol{P}_E \\ &= \begin{bmatrix} x_0 + x_E\cos(\alpha) - y_E\sin(\alpha) \\ y_0 + x_E\sin(\alpha) + y_E\cos(\alpha)\end{bmatrix}\end{aligned} \tag{1}$$

where ${}^{BCS}\boldsymbol{P}_E = [x_E, y_E]^T$ is the position of the end-effector with respect to the BCS and is also presented by vector $\vec{\mathbf{u}}_{O_BE} = [x_E, y_E]^T$. The point E is the intersection of the vector $\vec{\mathbf{u}}_{O_3E}$ and $\vec{\mathbf{u}}_{O_4E}$:

$$\begin{cases} \left\|\vec{u}_{O_3E}\right\|^2 = (x_E - x_3)^2 + (y_E - y_3)^2 = l_{12}^2 \\ \left\|\vec{u}_{O_4E}\right\|^2 = (x_E - x_4)^2 + (y_E - y_4)^2 = l_{22}^2 \end{cases} \tag{2}$$

From Eq. (2), the end-effector position with respect to the BCS is represented as follows:

$$\begin{aligned}\vec{\mathbf{u}}_{O_BE} = &\left(\frac{l_{12}^2 - l_{22}^2}{2D^2} + \frac{2\beta\sigma}{D^2}\begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}\right)\left(\vec{\mathbf{u}}_{O_BO_4} - \vec{\mathbf{u}}_{O_BO_3}\right) \\ &+ \frac{1}{2}\left(\vec{\mathbf{u}}_{O_BO_4} + \vec{\mathbf{u}}_{O_BO_3}\right)\end{aligned} \tag{3}$$

where $\beta = \pm 1$, which is the index that defines the robot's assembly configuration; and

$$\vec{\mathbf{u}}_{O_BO_3} = [x_3, y_3]^T = \left[l_{11}\cos\theta_1 - d/2, l_{11}\sin\theta_1\right]^T \tag{4}$$

$$\vec{\mathbf{u}}_{O_BO_4} = [x_4, y_4]^T = \left[l_{21}\cos\theta_2 + d/2, l_{21}\sin\theta_2\right]^T \tag{5}$$

$$D = \left\|\vec{\mathbf{u}}_{O_3O_4}\right\| = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} \tag{6}$$

$$\sigma = \frac{1}{4}\sqrt{\left[(D + l_{12})^2 - l_{22}^2\right]\left[-(D - l_{12})^2 + l_{22}^2\right]} \tag{7}$$

From Eq. (1), there are 10 kinematic parameters $\left[l_{11}, l_{12}, l_{21}, l_{22}, d, x_0, y_0, \alpha, \theta_1, \theta_2\right]^T$ that affect the end-effector's position.

B. KINEMATIC ERROR MODEL

The objective of kinematic calibration is to determine the actual kinematic parameter $\mathbf{v}^*$ defined as follows:

$$\mathbf{v}^* = \mathbf{v} + \delta\mathbf{v} \tag{8}$$

where $\mathbf{v} = [l_{11}, l_{12}, l_{21}, l_{22}, d, x_0, y_0, \alpha, \theta_1, \theta_2]^T$ and $\delta\mathbf{v} = \left[\delta l_{11}, \delta l_{12}, \delta l_{21}, \delta l_{22}, \delta d, \delta x_0, \delta y_0, \delta\alpha, \delta\theta_1, \delta\theta_2\right]^T$ are the nominal parameter and kinematic parameter error, respectively.

The end-effector position can be written in the general form as follows:

$$\boldsymbol{P}_E = f\left(\mathbf{v}^*\right) \tag{9}$$

where $f\left(\mathbf{v}^*\right) = \left[f_x\left(\mathbf{v}^*\right), f_y\left(\mathbf{v}^*\right)\right]^T$ is determined by Eq. (1).

The kinematic model presented by Eq. (9) is linearized as the first-order terms of a Taylor expansion of a nonlinear function, as follows:

$$f\left(\mathbf{v}^*\right) = f\left(\mathbf{v}\right) + \frac{\partial f}{\partial\mathbf{v}}\left(\mathbf{v}^* - \mathbf{v}\right) = f\left(\mathbf{v}\right) + \frac{\partial f}{\partial\mathbf{v}}\delta\mathbf{v} \tag{10}$$

where

$$\begin{aligned}\frac{\partial f}{\partial\mathbf{v}}\delta\mathbf{v} = &\frac{\partial f}{\partial l_{11}}\delta l_{11} + \frac{\partial f}{\partial l_{12}}\delta l_{12} + \frac{\partial f}{\partial l_{21}}\delta l_{21} + \frac{\partial f}{\partial l_{22}}\delta l_{22} \\ &+ \frac{\partial f}{\partial d}\delta d + \frac{\partial f}{\partial x_0}\delta x_0 + \frac{\partial f}{\partial y_0}\delta y_0 + \frac{\partial f}{\partial\alpha}\delta\alpha \\ &+ \frac{\partial f}{\partial\theta_1}\delta\theta_1 + \frac{\partial f}{\partial\theta_2}\delta\theta_2\end{aligned}$$

The kinematic error model is presented as follows:

$$\boldsymbol{Y} = \boldsymbol{J}\boldsymbol{X} \tag{11}$$

$$\boldsymbol{Y} = \begin{bmatrix}\delta x \\ \delta y\end{bmatrix} = \begin{bmatrix}x_m - f_x(\mathbf{v}) \\ y_m - f_y(\mathbf{v})\end{bmatrix} \tag{12}$$

$$\boldsymbol{J} = \begin{bmatrix}\frac{\partial f_x}{\partial l_{11}} & \frac{\partial f_x}{\partial l_{12}} & \cdots & \frac{\partial f_x}{\partial\theta_1} & \frac{\partial f_x}{\partial\theta_2} \\ \frac{\partial f_y}{\partial l_{11}} & \frac{\partial f_y}{\partial l_{12}} & \cdots & \frac{\partial f_y}{\partial\theta_1} & \frac{\partial f_y}{\partial\theta_2}\end{bmatrix} \tag{13}$$

$$\boldsymbol{X} = \delta\mathbf{v} = \left[\delta l_{11}\ \delta l_{12} \cdots \delta\theta_1\ \delta\theta_2\right]^T \tag{14}$$

where $\boldsymbol{Y}$ is the positional error of the end-effector, $\boldsymbol{J}$ is a Jacobian matrix, and $\boldsymbol{X}$ is the kinematic parameter error. The target position $f(\mathbf{v}) = \left[f_x(\mathbf{v}), f_y(\mathbf{v})\right]^T$ can be obtained by Eq. (1), and $P_m = [x_m, y_m]^T$ is the measured position.

C. THE PROPOSED APPROACH

1) ROBOT ERROR ANALYSIS

The robot's positional error, which includes the kinematic error δP_k, non-kinematic error δP_{nk}, and measurement noise δP_{noise}, can be expressed as follows:

$$P_m = P_t + \delta P_k + \delta P_{nk} + \delta P_{noise} \tag{15}$$

where P_m and P_t are the measured position and target position of the robot's end-effector, respectively. The kinematic calibration is used to estimate the kinematic parameter error and compensate for the kinematic error. In addition, the non-kinematic calibration predicts and compensates for the non-kinematic error.

2) THE PROPOSED APPROACH

In kinematic calibration, the kinematic parameter error represented in Eq. (11) can be determined by many methods [12]-[16]. Among them, the LSE method is most often applied and can be expressed as follows:

$$\delta\mathbf{v} = \boldsymbol{X} = \left(\boldsymbol{J}^T\boldsymbol{J}\right)^{-1}\boldsymbol{J}^T\boldsymbol{Y} \tag{16}$$

where $\boldsymbol{Y} = P_m - P_t = \delta P_k + \delta P_{nk} + \delta P_{noise}$. However, it is sensitive to measurement noise and non-kinematic error in the positional error. The EKF algorithm has proven to be effective for a nonlinear system that has Gaussian noise. However, this algorithm may converge to an incorrect solution, because linearization of the observation model can induce errors in the estimation of state. The QPSO, that is an intelligence optimization algorithm, has been used to solve various optimization problems in robotics, including robot calibration and path planning [29]. However, the performance of the QPSO algorithm heavily depends on the initialization of the swarm, especially the initialization position. In non-kinematic calibration, BPNN was applied to approximate the complex mapping between pose error and robot's joint length and

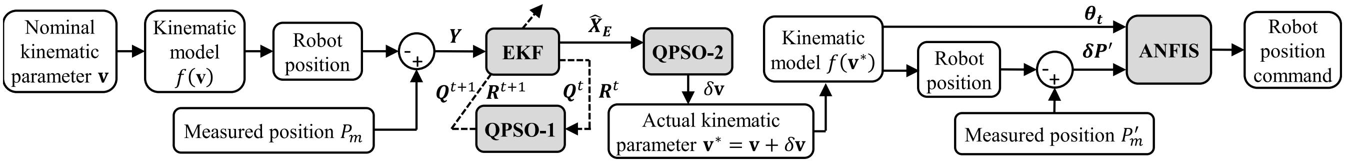

FIGURE 2. The identification procedure of the proposed method.

compensate for pose error. However, BPNN can fall into local minima [30]. This paper proposes a new calibration method to overcome these drawbacks and improve the calibration performance. The EKF-DQPSO, based on the combination of the EKF algorithm and DQPSO algorithm, is used for kinematic calibration, and the ANFIS model is used for non-kinematic calibration.

Figure 2 shows the calibration process of the proposed method. After collecting the positional error, the calibration procedure that includes two steps is carried out to compensate for the robot's positional error. First, the EKF-based QPSO-1 algorithm identifies the preliminary kinematic parameter errors, while the QPSO-2 algorithm optimizes the estimated kinematic parameter errors. Once the kinematic calibration is accomplished, the actual kinematic parameters are calculated by adding the parameter errors to the nominal kinematic parameters. Second, ANFIS establishes the complex relationship between the joint's rotational angle and the positional error. After that, the trained neural network compensates for the non-kinematic error.

The EKF algorithm is the robust optimization algorithm to solving the nonlinear system with measurement noise. It has some advantages such as reliability, fast convergence, and good estimation performance. In addition, the DQPSO algorithm has advantages such as few parameters, easy implementation, and good search ability to solve optimization problems. The ANFIS integrates the advantages of both artificial neural networks and fuzzy inference systems, to compensate for the non-kinematic error. As a result, ANFIS is an effective method to predict and modeling complex systems [31]–[33]. It presents a much better learning ability than the BPNN algorithm and can achieve highly nonlinear mapping.

III. KINEMATIC CALIBRATION

A. EXTENDED KALMAN FILTER

Because of the influence of measurement noise, EKF is used to estimate the kinematic parameter error by using the positional error and Jacobian matrix. To apply EKF, the nonlinear system is linearized, and presented as Eq. (11). The EKF algorithm includes time update step and measurement update step. In the time update step of EKF, the linear differential equation of the constant process $\hat{X}$ can be expressed as follows:

$$\hat{X}_{k|k-1} = \hat{X}_{k-1|k-1} \tag{17}$$

where $\hat{X}_{k|k-1} \in \Re^{10\times 1}$ is a vector representing the robot kinematic parameter errors, $k|k-1$ means a prior estimate at the k^{th} iteration. The predicted covariance matrix $\boldsymbol{P} \in \Re^{10\times 10}$ is calculated as follows:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1|k-1} + \boldsymbol{Q}_{k-1} \tag{18}$$

where $\boldsymbol{Q}_{k-1} \in \Re^{10\times 10}$ is the covariance matrix of the process noise.

The positional error can be presented by the measurement equation of EKF algorithm:

$$\boldsymbol{Y}_k = \boldsymbol{J}_k\hat{\boldsymbol{X}}_k + \boldsymbol{H}_k \tag{19}$$

where $\boldsymbol{Y}_k \in \Re^{2\times 1}$ is the positional error vector at the k^{th} iteration; $\boldsymbol{J}_k \in \Re^{2\times 10}$ is a Jacobian matrix; and $\boldsymbol{H}_k \in \Re^{2\times 1}$ is a vector that contains the sequence of the measurement noise with zero mean. $\boldsymbol{R}_k$ is the covariance matrix of the measurement noise, $\boldsymbol{R}_k = E\left(\boldsymbol{H}_k\boldsymbol{H}_k^T\right) \in \Re^{2\times 2}$.

In the measurement update step, the Kalman gain $\boldsymbol{K}_k$ is computed as follows:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{J}_k^T\left(\boldsymbol{J}_k\boldsymbol{P}_{k|k-1}\boldsymbol{J}_k^T + \boldsymbol{R}_k\right)^{-1} \tag{20}$$

The state vector $\hat{\boldsymbol{X}}_{k|k}$ and the updated covariance matrix $\boldsymbol{P}_{k|k}$ of the estimation error are calculated as follow:

$$\hat{\boldsymbol{X}}_{k|k} = \hat{\boldsymbol{X}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{Y}_k - \boldsymbol{J}_k\hat{\boldsymbol{X}}_{k|k-1}\right) \tag{21}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{J}_k)\,\boldsymbol{P}_{k|k-1} \tag{22}$$

where $\boldsymbol{I} \in \Re^{10\times 10}$ is an identity matrix.

An important issue of EKF is to select the correct matrices $\boldsymbol{Q}$, $\boldsymbol{R}$. Unsuitable choice of these matrices will create a large estimation error or even lead to divergence; consequently, tuning of $\boldsymbol{Q}$ and $\boldsymbol{R}$ is necessary. The covariance matrix can be obtained by experimental trial-and-error method. Although Bavdekar *et al.* [34] described a method to estimate $\boldsymbol{Q}$ and $\boldsymbol{R}$, it requires complex computation. The next subsection describes QPSO algorithm that optimizes the covariance matrices of EKF algorithm.

B. DUAL QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

The QPSO algorithm has many advantages on a few parameters, with good global search ability, to solve the optimization problems. QPSO algorithm consists of N particles moving in a m-dimension search space. Each particle is a potential solution and is characterized by its position vector:

$$X_i^t = \left[\mathbf{x}_{i1}^t\ \mathbf{x}_{i2}^t\ \cdots\ \mathbf{x}_{im}^t\right] \tag{23}$$

where $i = 1, 2, \ldots, N$. The particle's position is updated according to the following equation:

$$\mathbf{x}_{ij}^{t+1} = \boldsymbol{p}_{ij}^{t} \pm \alpha^{t} \left| \boldsymbol{mb}_{j}^{t} - \mathbf{x}_{ij}^{t} \right| \ln \left(\frac{1}{\mathbf{u}_{ij}^{t}} \right) \tag{24}$$

where $\mathbf{u}_{ij}^{t}$ is a uniformly distributed random number within the range (0,1), $j = 1, 2, \ldots, m$; and α^{t} is called the contraction-expansion factor, which can be used to control the convergence speed of the algorithm.

$\boldsymbol{p}_{ij}^{t}$ is the local attractor of i^{th} particle, and is presented as:

$$\boldsymbol{p}_{ij}^{t} = \varphi_{ij}^{t} \boldsymbol{P}_{ij}^{t} + \left(1 - \varphi_{ij}^{t}\right) \boldsymbol{Pg}_{j}^{t} \tag{25}$$

where φ_{ij}^{t} is a uniformly distributed random number within the range (0,1), $\boldsymbol{P}_{ij}^{t}$ is the j^{th} component of the personal best position $\boldsymbol{P}_{i}^{t}$ of the i^{th} particle, and $\boldsymbol{Pg}_{j}^{t}$ is the j^{th} component of the global best position $\boldsymbol{Pg}^{t}$ of all particles in the swarm.

The personal best position can be computed by:

$$\boldsymbol{P}_{i}^{t} = \begin{cases} \boldsymbol{P}_{i}^{t-1} & if\ f_{obj}\left(\boldsymbol{P}_{i}^{t-1}\right) < f_{obj}\left(\boldsymbol{X}_{i}^{t}\right) \\ \boldsymbol{X}_{i}^{t} & if\ f_{obj}\left(\boldsymbol{P}_{i}^{t-1}\right) \geq f_{obj}\left(\boldsymbol{X}_{i}^{t}\right) \end{cases} \tag{26}$$

where $f_{obj}\left(\boldsymbol{P}_{i}^{t-1}\right)$ and $f_{obj}\left(\boldsymbol{X}_{i}^{t}\right)$ are object functions.

The global best position $\boldsymbol{Pg}^{t}$ is expressed as:

$$\boldsymbol{Pg}^{t} = \min\left\{ \boldsymbol{P}_{1}^{t}\ \boldsymbol{P}_{2}^{t} \cdots \boldsymbol{P}_{N}^{t} \right\} \tag{27}$$

The mean best position ($\boldsymbol{mb}$) is defined as the center of the personal best position $\boldsymbol{P}_{i}^{t}$ and is updated by:

$$\boldsymbol{mb}^{t} = \frac{1}{N} \left[\sum_{i=1}^{N} \boldsymbol{P}_{i1}^{t}, \sum_{i=1}^{N} \boldsymbol{P}_{i2}^{t}, \ldots, \sum_{i=1}^{N} \boldsymbol{P}_{im}^{t} \right] \tag{28}$$

The optimization process is continuously updated according to Eq. (24). In this paper, DQPSO includes the QPSO-1 and QPSO-2 algorithm.

1) OPTIMIZING THE COVARIANCE MATRICES Q AND R

The object function of the QPSO-1 algorithm is designed as the root mean square error (RMSE) between the measured position $[x_m, y_m]^T$ and the estimated position $[x_{EKF}, y_{EKF}]^T$, and is represented as follows:

$$f_{obj}(Q, R) = \sqrt{\frac{\sum_{i=1}^{n} \left[\left(x_m^i - x_{EKF}^i\right)^2 + \left(y_m^i - y_{EKF}^i\right)^2 \right]}{n}} \tag{29}$$

where n is the number of measurement points. The estimated position is calculated as follows:

$$[x_{EKF}, y_{EKF}]^T = f\left(\mathbf{v} + \delta\mathbf{v}_{EKF(Q,R)}\right) \tag{30}$$

where f is the forward kinematic function, $\mathbf{v}$ is the nominal kinematic parameter, $\delta\mathbf{v}_{EKF(Q,R)}$ is the kinematic parameter error estimated by the EKF algorithm with covariance matrices Q and R.

The QPSO-1 algorithm optimizes the covariance matrices Q and R to minimize the object function $f_{obj}(Q, R)$.

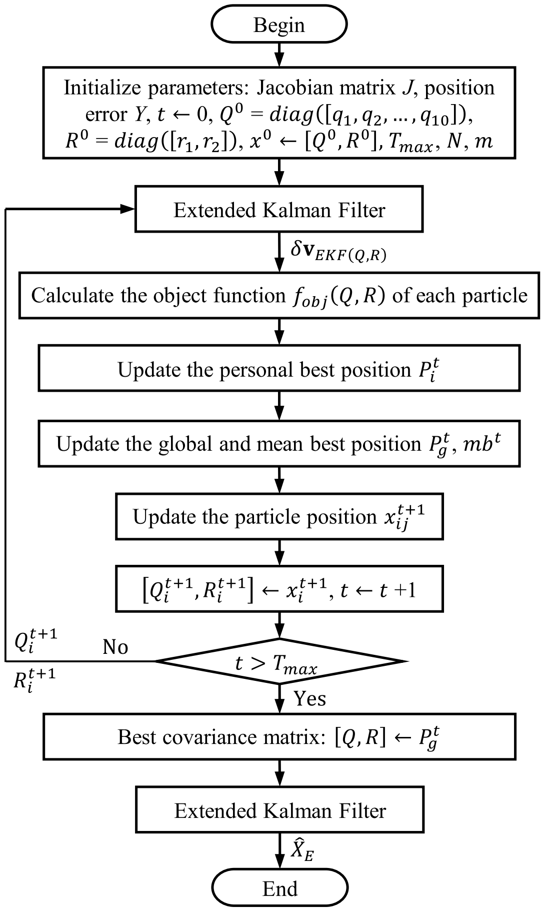
Begin
Initialize parameters: Jacobian matrix J, position error Y, $t \leftarrow 0$, $Q^0 = diag([q_1, q_2, \ldots, q_{10}])$, $R^0 = diag([r_1, r_2])$, $x^0 \leftarrow [Q^0, R^0]$, T_{max}, N, m
Extended Kalman Filter
$\delta\mathbf{v}_{EKF(Q,R)}$
Calculate the object function $f_{obj}(Q, R)$ of each particle
Update the personal best position P_i^t
Update the global and mean best position P_g^t, mb^t
Update the particle position x_{ij}^{t+1}
$[Q_i^{t+1}, R_i^{t+1}] \leftarrow x_i^{t+1}$, $t \leftarrow t+1$
Q_i^{t+1}
R_i^{t+1}
No
$t > T_{max}$
Yes
Best covariance matrix: $[Q, R] \leftarrow P_g^t$
Extended Kalman Filter
$\hat{X}_E$
End

FIGURE 3. Flow chart of the EKF-based QPSO-1 algorithm.

Furthermore, the EKF-based QPSO-1 algorithm identifies the preliminary kinematic parameter errors $\hat{\boldsymbol{X}}_E$ based on the positional error $\boldsymbol{Y}$ and Jacobian matrix $\boldsymbol{J}$. Fig. 3 shows the flow chart of the EKF-based QPSO-1 algorithm.

2) OPTIMIZING THE KINEMATIC PARAMETER ERRORS

The object function of the QPSO-2 algorithm is the RMSE between the measured position $[x_m, y_m]^T$ and the estimated position $[x_e, y_e]^T$, and is represented as follows:

$$f_{obj}\left(\delta\mathbf{v}_{QPSO-2}\right) = \sqrt{\frac{\sum_{i=1}^{n} \left[\left(x_m^i - x_e^i\right)^2 + \left(y_m^i - y_e^i\right)^2 \right]}{n}} \tag{31}$$

where n is the number of measurement points. The estimated position is calculated as follows:

$$[x_e, y_e]^T = f\left(\mathbf{v} + \delta\mathbf{v}_{QPSO-2}\right) \tag{32}$$

where f is the forward kinematic function, $\mathbf{v}$ is the nominal kinematic parameter, $\delta\mathbf{v}_{QPSO-2}$ is the kinematic parameter error estimated by the QPSO-2 algorithm.

The QPSO-2 algorithm optimizes the kinematic parameter error $\delta\mathbf{v}$ to minimize the object function $f_{obj}\left(\delta\mathbf{v}_{QPSO-2}\right)$. The initial position vector of the QPSO-2 algorithm is the kinematic parameter errors $\hat{\boldsymbol{X}}_E$ estimated by the EKF-based QPSO-1 algorithm.

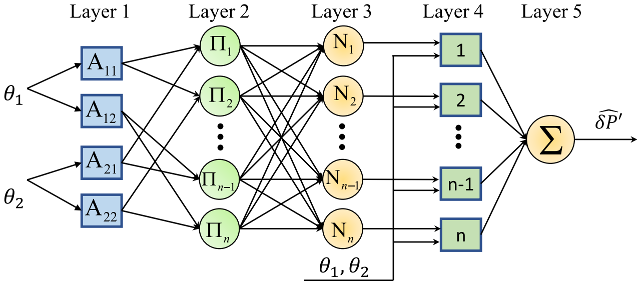

FIGURE 4. Structure of the ANFIS model.

IV. NON-KINEMATIC CALIBRATION

The positional error after kinematic calibration is represented as:

$$\delta P' = \delta P_{nk} + \delta P'_k + \delta P_{noise} = P'_m - P_t \tag{33}$$

where P'_m and P'_k are the measured position and remaining kinematic error after kinematic calibration The non-kinematic calibration uses the adaptive neuro-fuzzy inference systems to establish the complex nonlinear relationship between the joint angle and the positional error $\delta P'$, and compensate the non-kinematic error.

Jang proposed the adaptive neuro-fuzzy inference system (ANFIS), which is a combination of an artificial neural network and a fuzzy inference system [35]. The ANFIS has five layers. Fig. 4 shows the structure of the ANFIS model. The input of the ANFIS model in this paper is two rotation angles of two actuators. Layer 1 is the fuzzification layer and represents the fuzzy membership grade of the inputs, which is defined as:

$$O^1_m = \mu_{A_{jk}}\left(x_j\right) \tag{34}$$

where O^1_m and x are the output and input of node m, respectively, $m = \{1, 2, 3, 4\}, j = \{1, 2\}, k = \{1, 2\}$, and μ_{A_j} is the membership functions of the linguistic label A_j. The membership function can be a triangular, trapezoidal, sigmoidal, or Gaussian function. The parameters in layer 1 are called the premise parameters. Layer 2 is the rule layer. Each node is labeled Π and its output is the product of all the entry signals. The output of the i^{th} node represents the firing strength of a rule and is represented as:

$$O^2_i = w_i = \prod_{j=1}^{2} \mu_{A_{jk}}\left(x_j\right) \tag{35}$$

where $i = 1, 2, \ldots, n$. Layer 3 is a normalization layer. Each node in this layer is a fixed node labeled N. The output of the i^{th} node is the ratio between the i^{th} rules' firing strength and the sum of all the rule's firing strengths, and can be expressed as follows:

$$O^3_i = \bar{w}_i = \frac{w_i}{\sum\limits_{k=1}^{n} w_k} \tag{36}$$

Layer 4 is a defuzzification layer. In this layer, the output of each node is the product of a first order Sugeno model and the normalized firing strength, and is calculated as follows:

$$O^4_i = \bar{w}_i f_i = \bar{w}_i \left(\sum_{j=1}^{2} p_{ij} x_j + r_i \right) \tag{37}$$

where $\bar{w}_i$ is the output of layer 3, and $\{p_{ij}, r_i\}$ is the parameter set called consequent parameters. Layer 5 has a single node, which is a fixed node labeled Σ. The output computes the overall output as the summation of all incoming signals:

$$O^5_1 = \sum_{i=1}^{n} \bar{w}_i f_i \tag{38}$$

The output of ANFIS has only one node. Therefore, to train the positional error $(\Delta x, \Delta y)$, two ANFIS models have been implemented. The first and second ANFIS models are used to train the positional error Δx in the x direction and the positional error Δy in the y direction, respectively. The learning algorithm of the ANFIS model is a hybrid algorithm, which is a combination of least squares (LS) and back-propagation (BP) algorithms. The hybrid algorithm has high efficiency in training ANFIS with fast convergence [36]. LS and BP algorithm are used to optimize the consequent parameters and adjust the premise parameters, respectively.

V. EXPERIMENT

A. EXPERIMENTAL SETUP

Fig. 5(a) shows the experimental setup for robot calibration that consists of an EtherCAT-based five-bar parallel robot and a measurement system. Table 2 lists the nominal kinematic parameters of the robot. The robot controller is made based on Visual Studio and EtherCAT-WMX3 software. The calibration process was implemented using the Matlab program.

In this paper, we introduce a low-cost two-dimensional measurement system, using digital indicators to track the robot's absolute position. The measurement system includes an aluminum cuboid that was attached to the end-effector and 6 digital indicators that were fixed in a plane parallel to the working platform of the robot. The working platform plane $O_W X_W Y_W$ is calibrated to be parallel to the base platform plane $O_B X_B Y_B$. Fig. 5(b) shows the method to obtain the end-effector coordinate. At each location in the robot workspace, the displacements of only 3 digital indicators are used to calculate the end-effector's position in the XY plane. The end-effector's position is the center of the aluminum cuboid that has the same width and length. The end-effector coordinate is represented by the position of point E with respect to WCS, and is computed as follows:

$$\begin{aligned}\begin{bmatrix} x_E \\ y_E \end{bmatrix} &= \frac{l}{2}\begin{bmatrix} (\cos\gamma - \sin\gamma) \\ (\cos\gamma + \sin\gamma) \end{bmatrix} + \begin{bmatrix} \delta c - \delta d & -x_D + x_C \\ x_D - x_C & \delta c - \delta d \end{bmatrix}^{-1} \\ &\quad \times \begin{bmatrix} x_C\left(y_C - \delta d\right) - x_D\left(y_C - \delta c\right) \\ \left(x_B - \delta b\right)\left(x_D - x_C\right) + y_B\left(\delta c - \delta d\right) \end{bmatrix}\end{aligned} \tag{39}$$

where δb, δc, δd are the displacements of the indicators; and $\gamma = atan2\left(\delta c - \delta d, x_D - x_C\right)$; $[x_B, y_B]^T$, $[x_C, y_C]^T$,

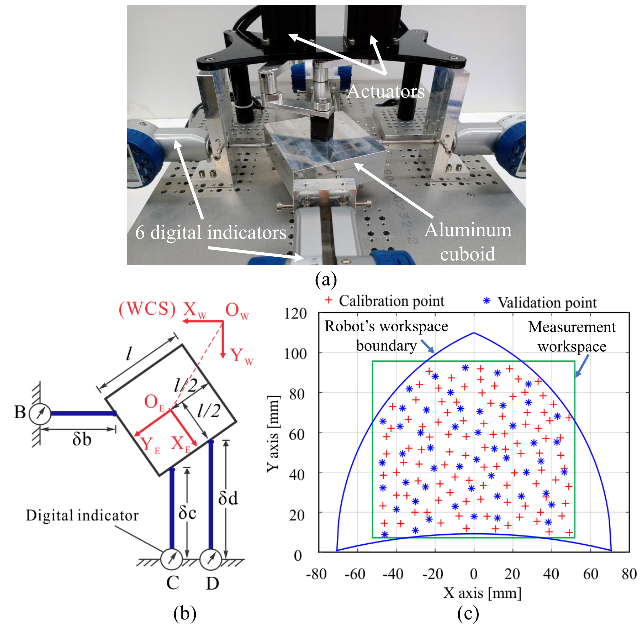

FIGURE 5. **(a) The experimental setup of the calibration system. (b) The measurement system in two dimensions *XY*. (c) The calibration and validation position in the workspace.**

and $[x_D, y_D]^T$ represent the coordinates with respect to WCS used to fix the indicators. The resolution and accuracy of the digital indicators are 0.001 mm and ±0.006 mm, respectively. The accuracy of the measurement system is 0.011 mm. The indicator data were transferred directly to the computer.

Two experiments of the calibration experiment and the validation experiment are performed to demonstrate the superiority of the proposed method. In the calibration experiment, a group of 100 positions is selected within the workspace, and is called group G_1. Fig. 6 shows a flow chart of the proposed method that includes the EKF-based QPSO-1 algorithm, the QPSO-2 algorithm, and the ANFIS model. In Fig. 6, f^{-1} () is the inverse kinematic function. After the positional errors are collected, the kinematic calibration is executed. The nominal position and nominal orientation of the BCS with respect to the WCS are $[x_0, y_0]^T = [0, 0]^T$ and $\alpha = 0$. The covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are initialized by $\boldsymbol{Q} = 10^{-4}\boldsymbol{I}_{10\times10}$ and $\boldsymbol{R} = 10^{-4}\boldsymbol{I}_{2\times2}$, respectively. The initial value of the kinematic parameter error for the EKF-based QPSO-1 algorithm is 0. The initial position vector of the QPSO-2 algorithm is the kinematic parameter error $\hat{\boldsymbol{X}}_E$ estimated by the EKF-based QPSO-1 algorithm. Table 1 lists the initial parameter of the QPSO-1 and QPSO-2 algorithm. The contraction expansion factor α of QPSO algorithm is initialized as 1 and reduces linearly to 0.5. Therefore, α^t can be calculated by the equation, $\alpha^t = 0.5\frac{T-t}{T} + 0.5$, with T being the maximum iteration. The object function of the kinematic parameter identification f_{obj} is the root mean square error (RMSE) between the measured position and the estimated position.

After finishing the kinematic calibration, the ANFIS is trained to establish a nonlinear relationship between the joint variables (θ_1, θ_2) and the robot's positional error $(\Delta x, \Delta y)$. In the ANFIS model, the Gaussian function is selected as a membership function and is represented as:

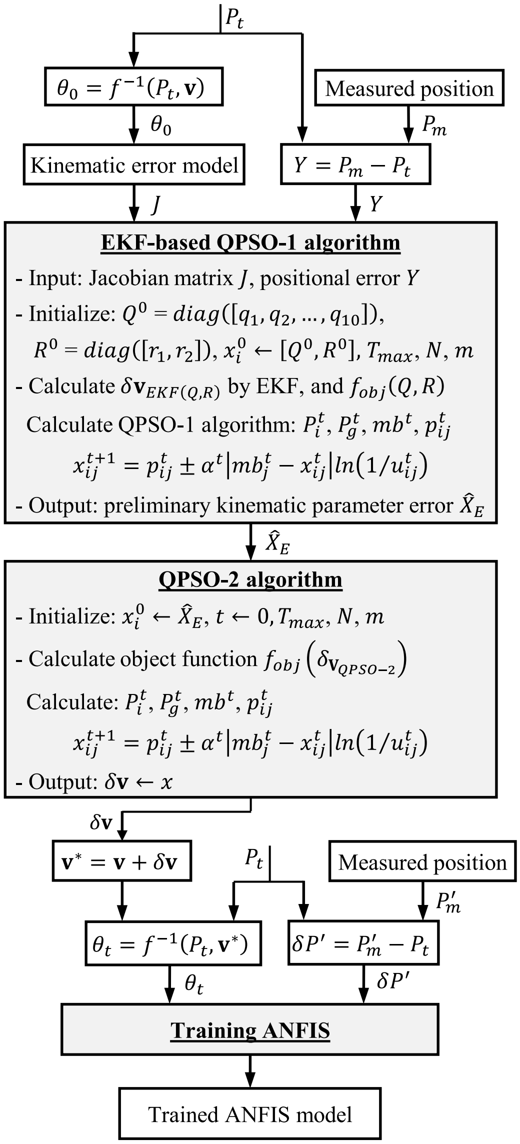
P_t
$\theta_0 = f^{-1}(P_t, \mathbf{v})$
Measured position
θ_0
P_m
Kinematic error model
$Y = P_m - P_t$
J
Y
EKF-based QPSO-1 algorithm
- Input: Jacobian matrix J, positional error Y
- Initialize: $Q^0 = diag([q_1, q_2, \dots, q_{10}])$,
$R^0 = diag([r_1, r_2])$, $x_i^0 \leftarrow [Q^0, R^0]$, T_{max}, N, m
- Calculate $\delta\mathbf{v}_{EKF(Q,R)}$ by EKF, and $f_{obj}(Q, R)$
Calculate QPSO-1 algorithm: $P_i^t, P_g^t, mb^t, p_{ij}^t$
$x_{ij}^{t+1} = p_{ij}^t \pm \alpha^t |mb_j^t - x_{ij}^t| ln(1/u_{ij}^t)$
- Output: preliminary kinematic parameter error $\hat{X}_E$
$\hat{X}_E$
QPSO-2 algorithm
- Initialize: $x_i^0 \leftarrow \hat{X}_E$, $t \leftarrow 0$, T_{max}, N, m
- Calculate object function $f_{obj}\left(\delta\mathbf{v}_{QPSO-2}\right)$
Calculate: $P_i^t, P_g^t, mb^t, p_{ij}^t$
$x_{ij}^{t+1} = p_{ij}^t \pm \alpha^t |mb_j^t - x_{ij}^t| ln(1/u_{ij}^t)$
- Output: $\delta\mathbf{v} \leftarrow x$
$\delta\mathbf{v}$
$\mathbf{v}^* = \mathbf{v} + \delta\mathbf{v}$
P_t
Measured position
P'_m
$\theta_t = f^{-1}(P_t, \mathbf{v}^*)$
$\delta P' = P'_m - P_t$
θ_t
$\delta P'$
Training ANFIS
Trained ANFIS model

FIGURE 6. **A flow chart of the proposed method.**

TABLE 1. **Initial parameters of the DQPSO algorithm.**

	QPSO-1 algorithm	QPSO-2 algorithm
Maximum iteration T_{max}	100	400
Contraction-expansion factor α^t	$0.5\frac{T-t}{T} + 0.5$	$0.5\frac{T-t}{T} + 0.5$
Number of particle N	50	50
Dimension search space m	12	10

$$\mu_{A_i}(x) = e^{-\left(\frac{x-c_i}{a_i}\right)^2} \tag{40}$$

The variables $\{c_i, a_i\}$ are the premise parameters of the ANFIS. The number of membership function and the maximum number of epochs are 6 and 500, respectively.

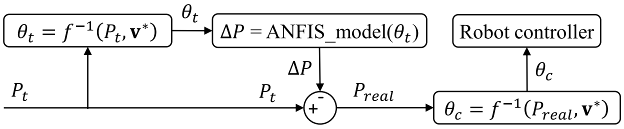
$\theta_t = f^{-1}(P_t, \mathbf{v}^*)$
θ_t
$\Delta P = \text{ANFIS\_model}(\theta_t)$
Robot controller
ΔP
θ_c
P_t
P_t
P_{real}
-
+
$\theta_c = f^{-1}(P_{real}, \mathbf{v}^*)$

FIGURE 7. The validation process.

The training object function of the output layer is the mean square error (MSE) between the actual output and the target output of the ANFIS. The target error of the training object function was set to 1e-5. ANFIS is implemented in Matlab program. The ANFIS and BPNN method used the positional error data of group G_1 to train the neural network model. By using the hold-out method, this training dataset that includes 100 samples of group G_1 was divided into a training set which contained 80% data and a testing set which contained 20% data. Therefore, the number of training and testing set was 80 and 20 samples, respectively. The trained neural network is used to compensate for the non-kinematic error. The BPNN model includes the input layer, hidden layer, and output layer. The activation function of the hidden layer and output layer are tan-sigmoid function and linear function, respectively. The number of hidden node in the hidden layer was calculated using the empirical formula [37]. It can be expressed as $h = a\sqrt{m+n}$, where n, m, h are the number of nodes in the input layer, output layer, and hidden layer, respectively, and $a \in [1, 20]$. In this paper, the hidden layer consists of 40 nodes.

In the validation experiment, the group of 50 positions, called group G_2, is used to validate the absolute positioning accuracy after calibration. Fig. 7 shows the validation process of the proposed method. θ_c is the actual joint angle that is used to control the robot. To demonstrate the superior performance of the proposed method, the proposed method is compared to EKF-DQPSO, LSE & BPNN [23], and EKF & BPNN [26]. Fig. 5(c) shows 100 calibration points of group G_1 and 50 validation points of group G_2 in the robot's workspace. The absolute positional error E_p is the Euclidean distance between the measured position $[x_m, y_m]^T$ and the target position $[x_t, y_t]^T$:

$$E_p = \sqrt{(x_m - x_t)^2 + (y_m - y_t)^2} \tag{41}$$

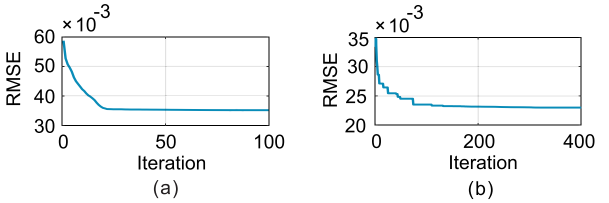

FIGURE 8. The evolution of the object function relative to (a) the EKF-based QPSO-1 algorithm and (b) the QPSO-2 algorithm.

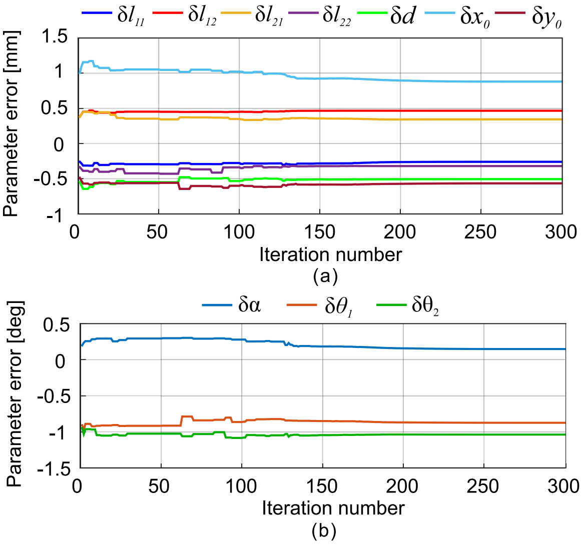

FIGURE 9. The kinematic parameter errors are identified through the EKF-DQPSO.

TABLE 2. Identified kinematic parameters of the five-bar parallel robot.

	Nominal parameter	Identified parameter
$l_{11}\,(mm)$	60	59.742
$l_{12}\,(mm)$	60	60.483
$l_{21}\,(mm)$	60	60.341
$l_{22}\,(mm)$	60	59.704
$d\,(mm)$	90	89.511
$x_0\,(mm)$	0	0.851
$y_0\,(mm)$	0	-0.556
$\alpha\,(deg)$	0	0.148
$\delta\theta_1\,(deg)$	0	-0.873
$\delta\theta_2\,(deg)$	0	-1.032

TABLE 3. Model parameters and training performance of the BPNN.

Activation function	Hidden nodes	Epochs	Object function	Target error	Training time (s)
Tan-sigmoid	40	40000	9.79e-5	1e-5	77.680

B. EXPERIMENTAL RESULTS

The convergence of the kinematic calibration method is demonstrated by the convergence of object function that is shown in Fig. 8. The RMSE of the EKF-based QPSO-1 algorithm and the QPSO-2 algorithm converges to 0.0349 after 52 iterations and 0.0233 after 261 iterations, respectively. Fig. 9 shows the convergence of the 10 kinematic parameter errors identified by using the EKF-DQPSO method. Table 2 summarizes the nominal and identified kinematic parameters after kinematic calibration. In order to show how well the proposed method outperforms the other methods, Fig. 10 compares the positional errors before and after the calibration by EKF-DQPSO, LSE & BPNN [23], EKF & BPNN [26], and the proposed method. Fig. 10 (a) and Fig. 10 (b) show the absolute positional error in the calibration experiment and validation experiment, respectively.

Tables 3 and 4 show the model parameters and training performance of the BPNN and ANFIS methods, respectively.

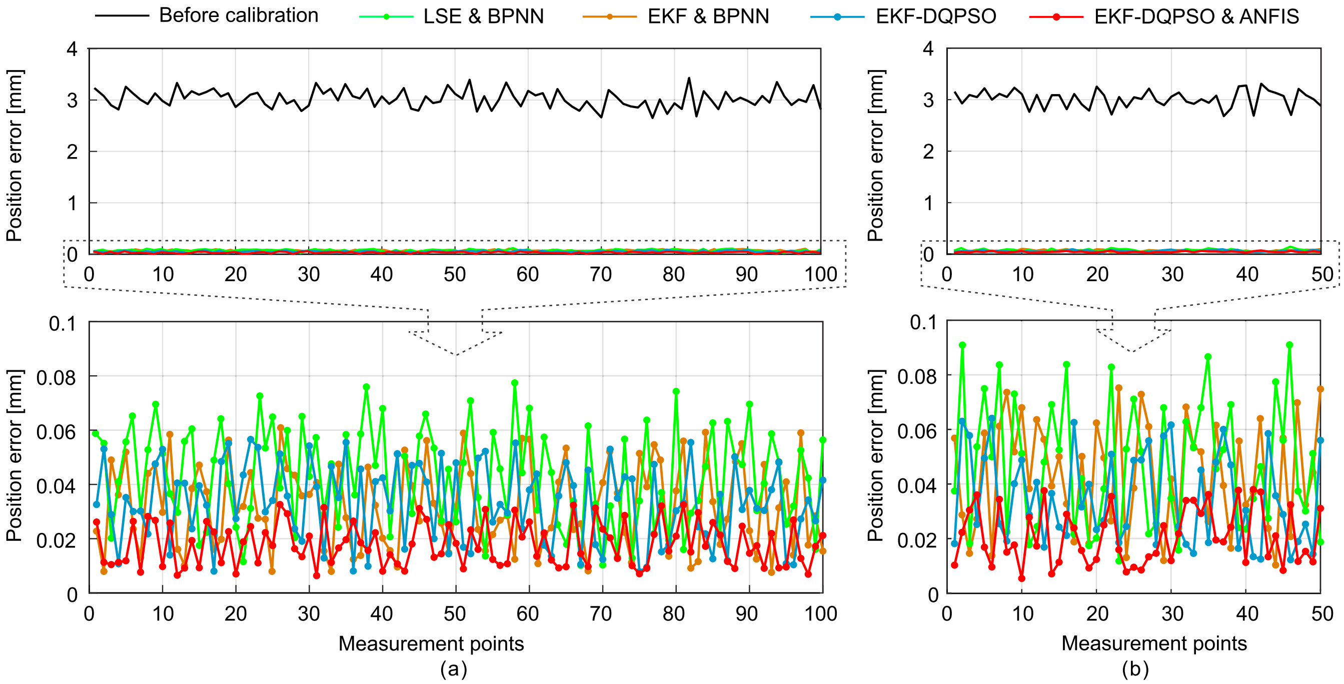

FIGURE 10. The absolute positional error before and after calibration by EKF-DQPSO, LSE & BPNN, EKF & BPNN, and EKF-DQPSO & ANFIS method in calibration experiment (a) and in validation experiment (b).

TABLE 4. Model parameters and training performance of the ANFIS.

Membership function	Number of rules	Epochs	Object function	Target error	Training time (s)
6	36	500	1e-5	1e-5	26.361

TABLE 5. The computational time of the proposed method.

	EKF-based QPSO-1	QPSO-2	ANFIS	Proposed method
Time (s)	11.756	7.840	26.361	45.957

The BPNN using the gradient descent algorithm fell into the local minimum and could not achieve the target training error. On the other hand, the learning capacity of the ANFIS is better than that of the BPNN. Therefore, the ANFIS could obtain the target training error and required a shorter training time than the BPNN method. Table 5 lists the computational time of the proposed method including the EKF-based QPSO-1 algorithm, QPSO-2 algorithm, and ANFIS method. Table 6 lists in detail the mean error, standard deviation, and maximum error of the position of the end-effector in the calibration and validation experiment. From Tables 5 and 6, the computational time and accuracy of the proposed method are suitable for the accuracy requirements of the offline robot calibration applications.

In the validation experiment, the mean and maximum positional errors after calibration by the proposed method are 0.022 mm and 0.038 mm, respectively. The positional error is reduced by 99 % by applying the proposed method (from 2.995 mm before calibration to 0.022 mm after calibration).

TABLE 6. Absolute position accuracy before and after calibration (in milimeter).

	Calibration			Validation		
	Mean	Std	Max	Mean	Std	Max
Before calibration	3.051	0.177	3.367	2.995	0.179	3.452
LSE & BPNN [23]	0.040	0.021	0.078	0.049	0.024	0.091
EKF & BPNN [26]	0.032	0.019	0.061	0.039	0.021	0.076
EKF-DQPSO	0.034	0.018	0.056	0.038	0.019	0.063
EKF-DQPSO & ANFIS	**0.018**	**0.012**	**0.032**	**0.022**	**0.013**	**0.038**

The mean positional error after calibration by LSE & BPNN, EKF & BPNN, EKF-DQPSO and the proposed method are 0.049 mm, 0.039 mm, 0.038 mm, and 0.022 mm, respectively. The positional error in the proposed method is decreased by 55 % compared to that of the LSE & BPNN method (from 0.049 mm to 0.022 mm), by 44 % compared to that of the EKF & BPNN method (from 0.039 mm to 0.022 mm), and by 42 % compared to that of the EKF-DQPSO method (from 0.038 mm to 0.022 mm). In addition, the proposed method achieves the lowest maximum error and the lowest standard deviation. Therefore, the validation results confirm that the proposed method outperforms the other calibration methods.

VI. CONCLUSION

In this paper, a robot calibration method based on EKF-DQPSO and ANFIS has been proposed. The results of the experiment performed on the five-bar parallel robot demonstrated the correctness and effectiveness of the proposed calibration method. Absolute positioning accuracy was significantly improved. The EKF algorithm showed the effect of solving the nonlinear system with measurement noise. The DQPSO algorithm optimized the performance of the EKF algorithm and the kinematic parameter errors. ANFIS established the relationship between the joint angle and the positional error, and compensated for the non-kinematic error. As a result, the mean error of the robot's position was reduced by 99 % (from 2.995 mm before calibration to 0.022 mm after calibration). In the future work, the proposed calibration algorithm will be applied to calibrate the other serial or parallel robots. In addition, the proposed method will be improved for online calibration.

REFERENCES

[1] J. Wu, Y. Gao, B. Zhang, and L. Wang, "Workspace and dynamic performance evaluation of the parallel manipulators in a spray-painting equipment," *Robot. Comput.-Integr. Manuf.*, vol. 44, pp. 199–207, Apr. 2017.

[2] R. He, Y. Zhao, S. Yang, and S. Yang, "Kinematic-parameter identification for serial-robot calibration based on POE formula," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 411–423, Jun. 2010.

[3] T. Sun, B. Lian, S. Yang, and Y. Song, "Kinematic calibration of serial and parallel robots based on finite and instantaneous screw theory," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 816–834, Jun. 2020.

[4] J. Wu, G. Yu, Y. Gao, and L. Wang, "Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool," *Mechanism Mach. Theory*, vol. 121, pp. 430–445, Mar. 2018.

[5] B. Mei, F. Xie, X.-J. Liu, and C. Yang, "Elasto-geometrical error modeling and compensation of a five-axis parallel machining robot," *Precis. Eng.*, vol. 69, pp. 48–61, May 2021.

[6] Y. Jiang, T. Li, L. Wang, and F. Chen, "Kinematic error modeling and identification of the over-constrained parallel kinematic machine," *Robot. Comput.-Integr. Manuf.*, vol. 49, pp. 105–119, Feb. 2018.

[7] J. Zhang, Q. Chen, C. Wu, and Q. Li, "Kinematic calibration of a 2-DOF translational parallel manipulator," *Adv. Robot.*, vol. 28, no. 10, pp. 707–714, Feb. 2014.

[8] T. Sun, Y. Zhai, Y. Song, and J. Zhang, "Kinematic calibration of a 3-DoF rotational parallel manipulator using laser tracker," *Robot. Comput.-Integr. Manuf.*, vol. 41, pp. 78–91, Oct. 2016.

[9] A. Joubair, M. Slamani, and I. A. Bonev, "Kinematic calibration of a 3-DOF planar parallel robot," *Ind. Robot: Int. J.*, vol. 39, no. 4, pp. 392–400, Jun. 2012.

[10] W. Wang, F. Liu, and C. Yun, "Calibration method of robot base frame using unit quaternion form," *Precis. Eng.*, vol. 41, pp. 47–54, Jul. 2015.

[11] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robot. Comput.-Integr. Manuf.*, vol. 26, no. 5, pp. 414–419, Oct. 2010.

[12] A. Joubair, M. Slamani, and I. A. Bonev, "Kinematic calibration of a five-bar planar parallel robot using all working modes," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 4, pp. 15–25, Aug. 2013.

[13] Y. Zheng and S. Peng, "A practical roadside camera calibration method based on least squares optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 831–843, Apr. 2014.

[14] J. Shawash and D. R. Selviah, "Real-time nonlinear parameter estimation using the Levenberg–Marquardt algorithm on field programmable gate arrays," *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 170–176, Jan. 2013.

[15] J.-M. Renders, E. Rossignol, M. Becquet, and R. Hanus, "Kinematic calibration and geometrical parameter identification for robots," *IEEE Trans. Robot. Automat.*, vol. 7, no. 6, pp. 721–732, Dec. 1991.

[16] T. Sun, P. Wang, B. Lian, S. Liu, and Y. Zhai, "Geometric accuracy design and error compensation of a one-translational and three-rotational parallel mechanism with articulated traveling plate," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 232, no. 12, pp. 2083–2097, Feb. 2017.

[17] G. Alici, R. Jagielski, Y. A. Şekercioğlu, and B. Shirinzadeh, "Prediction of geometric errors of robot manipulators with particle swarm optimisation method," *Robot. Auto. Syst.*, vol. 54, no. 12, pp. 956–966, Dec. 2006.

[18] N. Pitchandi and S. P. Subramanian, "GA-based camera calibration for vision-assisted robotic assembly system," *IET Comput. Vis.*, vol. 11, no. 1, pp. 50–59, 2017.

[19] L. Fang and P. Dang, "A step identification method of joint parameters of robots based on the measured pose of end-effector," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 229, no. 17, pp. 3218–3233, Dec. 2015.

[20] Z. H. Jiang, W. G. Zhou, H. Li, Y. Mo, W. C. Ni, and Q. Huang, "A new kind of accurate calibration method for robotic kinematic parameters based on the extended Kalman and particle filter algorithm," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3337–3345, Apr. 2018.

[21] I.-W. Park, B.-J. Lee, S.-H. Cho, Y.-D. Hong, and J.-H. Kim, "Laser-based kinematic calibration of robot manipulator using differential kinematics," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 6, pp. 1059–1067, Dec. 2012.

[22] D. Chen, T. Wang, P. Yuan, N. Sun, and H. Tang, "A positional error compensation method for industrial robots combining error similarity and radial basis function neural network," *Meas. Sci. Technol.*, vol. 30, no. 12, Sep. 2019, Art. no. 125010.

[23] H. N. Nguyen, P. N. Le, and H. J. Kang, "A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique," *Adv. Mech. Eng.*, vol. 11, no. 1, Jan. 2019, Art. no. 1687814018822935.

[24] D. Yu, "Parallel robots pose accuracy compensation using back propagation network," *Int. J. Phys. Sci.*, vol. 6, no. 21, pp. 5005–5011, Sep. 2011.

[25] Z. Wang, Z. Chen, Y. Wang, C. Mao, and Q. Hang, "A robot calibration method based on joint angle division and an artificial neural network," *Math. Problems Eng.*, vol. 2019, Mar. 2019, Art. no. 9293484.

[26] 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, pp. 996–1005, Mar. 2015.

[27] X. Chen, Q. Zhang, and Y. Sun, "Evolutionary robot calibration and nonlinear compensation methodology based on GA-DNN and an extra compliance error model," *Math. Problems Eng.*, vol. 2020, Jul. 2020, Art. no. 3981081.

[28] P.-N. Le and H.-J. Kang, "Robot manipulator calibration using a model based identification technique and a neural network with the teaching learning-based optimization," *IEEE Access*, vol. 8, pp. 105447–105454, 2020.

[29] Y. Shi, B. Liang, X. Wang, and W. Xu, "Cartesian non-holonomic path planning of space robot based on quantum-behaved particle swarm optimization algorithm," *J. Mech. Eng.*, vol. 47, no. 23, pp. 65–73, Dec. 2011.

[30] W. Bi, "Avoiding the local minima problem in backpropagation algorithm with modified error function," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vols. 88, no. 12, pp. 3645–3653, Dec. 2005.

[31] H. Azimi, S. Shabanlou, I. Ebtehaj, H. Bonakdari, and S. Kardar, "Combination of computational fluid dynamics, adaptive neuro-fuzzy inference system, and genetic algorithm for predicting discharge coefficient of rectangular side orifices," *J. Irrigation Drainage Eng.*, vol. 143, no. 7, Jul. 2017, Art. no. 04017015.

[32] I. Ebtehaj, H. Bonakdari, and B. Gharabaghi, "Development of more accurate discharge coefficient prediction equations for rectangular side weirs using adaptive neuro-fuzzy inference system and generalized group method of data handling," *Measurement*, vol. 116, pp. 473–482, Feb. 2018.

[33] H. Moeeni, H. Bonakdari, and I. Ebtehaj, "Integrated SARIMA with neuro-fuzzy systems and neural networks for monthly inflow prediction," *Water Resour. Manage.*, vol. 31, no. 7, pp. 2141–2156, 2017.

[34] V. A. Bavdekar, A. P. Deshpande, and S. C. Patwardhan, "Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter," *J. Process Control*, vol. 21, no. 4, pp. 585–601, 2011.

[35] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[36] A. Gholami, H. Bonakdari, I. Ebtehaj, and A. Akhtari, "Design of an adaptive neuro-fuzzy computing technique for predicting flow variables in a 90° sharp bend," *J. Hydroinform.*, vol. 19, no. 4, pp. 572–585, Jan. 2017.

[37] G. Gao, F. Liu, H. San, X. Wu, and W. Wang, "Hybrid optimal kinematic parameter identification for an industrial robot based on BPNN-PSO," *Complexity*, vol. 2018, Jul. 2018, Art. no. 4258676.

HUNG QUANG CAO received the B.S. degree in mechatronics engineering from Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

His research interests include robotics, motion control, and embedded systems.

HA XUAN NGUYEN received the B.S. degree in mechatronics engineering from Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2015, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2020.

He is currently a Postdoctoral Researcher with the College of Information and Computer Engineering, Sungkyunkwan University. His research interests include signal processing, motion control, robotics, and embedded systems.

TY TRUNG NGUYEN received the B.S. degree in automatic control engineering from Hanoi University, Hanoi, Vietnam, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

His research interests include signal processing, motion control, and robotics.

VINH QUANG NGUYEN received the B.S. degree in physics and engineering physics from Vietnam National University Ho Chi Minh City—University of Science, Ho Chi Minh City, Vietnam, in 2012, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2021.

Since 2021, he has been a Senior Engineer with Presto Solution Company, South Korea. His research interests include real-time networks, motion control, and embedded systems.

JAE WOOK JEON (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990.

From 1990 to 1994, he was a Senior Researcher with Samsung Electronics, Suwon, South Korea. Since 1994, he has been with Sungkyunkwan University, Suwon, where he was first an Assistant Professor with the School of Electrical and Computer Engineering and currently a Professor with the School of Information and Communication Engineering. His research interests include robotics, embedded systems, and factory automation.