Optimal Kinematic Calibration of Parallel Manipulators With Pseudoerror Theory and Cooperative Coevolutionary Network

Dan Zhang, *Senior Member, IEEE*, and Zhen Gao

***Abstract*—Accuracy is one of the most crucial factors which affects the profound laboratory research and extensive industrial application of parallel robotic manipulators. Kinematic calibration is a necessary approach to make the nominal value approximately equivalent to the actual value for the pose of end-effector under different input of actuation variables. Since the error source of parallel manipulator is strong coupling, highly nonlinear, and uncontrollable, the pseudoerror theory is proposed by considering multiple errors, including manufacturing and assembly error, thermal error, and nonlinear stiffness error, as a single hypothetical error source, which only causes the deflection of joint variables. A novel cooperative coevolutionary neural network (CCNN) is designed to establish the complex nonlinear relationship between joint variables and the related deviation with respect to the measured pose of the end-effector. With CCNN, the pseudoerror in arbitrary joint configuration can be obtained, and thus, the control parameters can be adjusted accordingly. The results are validated through the case studies about a parallelogram-based 3-DOF parallel manipulator and a parallel robotic machine tool. This approach is generic and feasible for all types of robotic system.**

***Index Terms*—Cooperative coevolutionary neural network (CCNN), end-effector, error classification, optimal calibration, pseudoerror theory.**

I. Introduction

INSPIRED by the earliest parallel mechanism "Gough–Stewart" platform, nowadays, parallel manipulators have become the attractive key to many application fields, such as machine tool, motion simulator, microrobot, medical device, and physical sensor [1]–[9], owing to their intrinsic advantages in the factors of pay load, stiffness, accuracy, operational velocity, and acceleration compared with the traditional serial counterparts [10]–[17]. However, as is known, the structure of parallel robotic manipulator is considerably complicated since it is constituted with n $(n \geq 2)$ independent motion chains rather than one single serial link, thus increasing the error sources and error probability induced by the great number of links and passive joints. This has been a potential bottleneck for their practical development in manufacturing industry. Therefore, kinematic calibration is one of the most fundamental issues of parallel manipulators.

Generally, there exist three types of kinematic calibration for parallel robots [18]. The first type, named external calibration, uses an external device, i.e., laser tracker, optical system, and coordinate measuring machine, to acquire the information of platform poses or joint parameters. Maurine and Dombre [19] adopted the laser displacement device to identify the offsets on the three first joints and the absolute location of the Delta 4 robot base. Kim [20] studied the nonlinear least squares method of the Cartesian parallel manipulator with ball bar for length measurement. Vision devices have been widely applied for the kinematic calibration of parallel mechanisms [21]–[23]. Renaud *et al.* [21] used a camera to measure the pose of H4 parallel robot with a precision on the order of magnitude of 0.2 mm and 0.03° based on inverse kinematic model. However, it shows that the information related to the sensor installation cannot be well identified. The second type, called constrained calibration, depends on the physical constraint with a specific mechanical system when conducting the calibration process. Jeong *et al.* [24] considered the constraint torque equilibrium and the relationship of constraint torque to torsional deflection for redundantly actuated parallel mechanisms. Rauf and Ryu [25] explored the kinematic calibration of reconfigurable modular gantry-Tau robot by intersecting the workspace overlaying it with a grid of a desired number of measurement points. The third type, namely, self-calibration or autocalibration, utilizes the internal sensors to implement the measuring process. Daney and Emiris [26] proposed an approach relying on resultants and dialytic elimination which increased reliability by avoiding employing random configurations. Nahvi and Hollerbach [27] presented a least squares optimization method with multiple kinematic closed loops for self-calibration of the redundant shoulder joint. Mansour *et al.* [28] proposed a methodical way of self-calibrating for Hexaglide parallel robot that was performable only by measuring input joint variables and positioning errors. Last *et al.* [29] utilized the adaptronic swivel joint to achieve the required redundant information and executed the simulation on a RRRRR-structure. Hsu and Chen [30] found that, if higher accuracy was required, the nonlinear stiffness

Manuscript received January 12, 2010; revised March 4, 2011 and May 21, 2011; accepted August 8, 2011. Date of publication October 18, 2011; date of current version March 30, 2012. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), by Mathematics of Information Technology and Complex Systems-Networks of Centres of Excellence (MITACS-NCE) program, and by Early Researcher Awards (ERA) program. The work of D. Zhang was supported by the Canada Research Chairs program.

D. Zhang is with the Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, ON L1H 7K4, Canada (e-mail: Dan.Zhang@uoit.ca).

Z. Gao is with the Robotics and Automation Laboratory, Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, ON L1H 7K4, Canada (e-mail: Zhen.Gao@uoit.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2011.2166229

effect and the platform orientation error should be considered on the Cartesian-guided tripod.

Some researchers investigated on the improvement of specific calibration algorithms [31]–[36]. Vischer [32] solved the direct and inverse problem during identification based on implicit closure equations. David *et al.* [35] presented a kind of interval method that provided a numerically certified result of Gough platform. Savoure *et al.* [34] reported the static accuracy increment by identifying the whole geometrical structure of the parallel kinematic machine including the entire parallelogram. Gauss–Newton-method-based singular value decomposition and Levenberg–Marquardt (LM) method were proposed by Varziri and Notash [36] to identify the kinematic parameters of the central linkage of the wire-actuated parallel robot. In other related fields, general regression neural network (GRNN) was applied as the solution of inverse parameters for the sensitivity-calibration analysis [37]. Evolutionary neural network (ENN) was developed as a sensor self-calibration scheme that could adapt to the changing environmental conditions including temperature, pressure, and location [38]. Moreover, it should be necessary to mention that the most natural method as one part to perform the complete calibration is to minimize the error between the measured and nominal values of end-effector pose through the inverse kinematic model which usually can be derived rather easily.

The major contribution of this paper includes two aspects: 1) By integrating the multierror sources into one single hypothetical error source which induces the variation of actuator parameters, the pose compensation of end-effector can be settled in a natural and compact way, and 2) a novel cooperative coevolutionary neural network (CCNN) is developed to establish the complex nonlinear association between the information of pseudoerror source and measured parameters. Combined with the inverse kinematic model of parallel manipulator, the kinematic calibration can be implemented in an optimal and efficient manner.

The remainder of this paper is organized as follows. In Section II, the error classification of parallel robots and pseudoerror theory are presented, and the related error compensation scheme is described. The principle of the developed CCNN is illuminated in detail in Section III. In Section IV, two case studies are conducted to show the high performance of the proposed methods. Finally, the conclusions are given in Section V.

II. Pseudoerror Theory

Error directly affects the performance indices of parallel robots. According to its sources and characteristics, the error of parallel robots can be classified as shown in Fig. 1. For the error sources of parallel robots, it can be divided into two main types, namely, vibration error and quasi-static error. Vibration error is the operational inaccuracy induced by the vibration of tools, system axis, or motor. Quasi-static error includes the kinematic parameter error (or manufacturing and assembly error), thermal error, and nonlinear stiffness error. This differs from many studies reported in the past literature. The thermal error induced in the sliding motion (i.e., actuator leg) is a major error source in the operation of the end-effector, and it is distributed over the entire structure of a robotic mechanism. The nonlinear

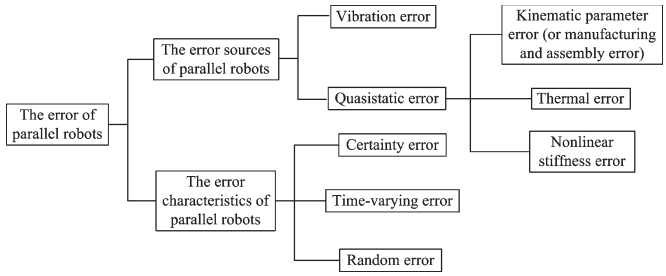


Fig. 1. Error classification of parallel robots.

stiffness error may be caused by materials of links and joints and/or external forces and/or moments which induce large or small deformation. The errors over an entire workspace can be visualized to effectively guide the design process. It can be expressed with the notion of an error map. The error map will allow an active participation of designers in determining strategies for error compensation and understanding sources of errors (i.e., manufacturing and assembly, thermal, or nonlinear stiffness) to determine a way to remove/reduce the errors.

In this paper, we will discuss how to conduct effective error compensation and calibration, thereby reducing its impact to the operational accuracy. The source of manipulator error is in a wide range. In addition to the main factors, such as the error induced by geometric size and thermal effect, there are many other factors, e.g., load deformation, gear clearance, voltage fluctuations, external environment, etc. Thus, it is very difficult to establish a general error model considering all the error sources.

In the field of measurement and signal processing, pseudoerror is usually considered as a kind of random noise generated by uncontrollable source. In [39], the measurement model was improved through adopting a concept of pseudo measurement error when considering the uncertainty in line parameters. In [40], the pseudorandom noises were injected to measure and calibrate the gain error and analog-to-digital converter error. In [41], the pseudorandom generator and the true joint probabilities were used to generate joint probability matrices to simulate replicate samples for inverse calibration. In [42], pseudoinvariant targets were developed to eliminate the unexplained errors associated with the internal calibration. However, the pseudoerror as a hypothetical error source for the optimal kinematic calibration of parallel manipulators has never been reported in other literature. In this scenario, based on the method of pose compensation of the end-effector, the multierror sources can be abstracted as a single hypothetical error source, namely, pseudoerror source, which only causes the deviation of joint variables. The relationship of the pseudoerror source and the pose deviation of end-effector is shown in Fig. 2.

Assuming the ideal pose, the end-effector of parallel robot with respect to the fixed reference frame can be defined as

$$Pose_{\text{ideal}} = \{x, y, z, \psi, \theta, \phi\} \tag{1}$$

where the first three elements represent the position values in x-, y-, and z-axes, respectively. The last three elements characterize the orientation values, in which ψ, θ, and φ mean rotations around x-, y-, and z-axes, respectively.

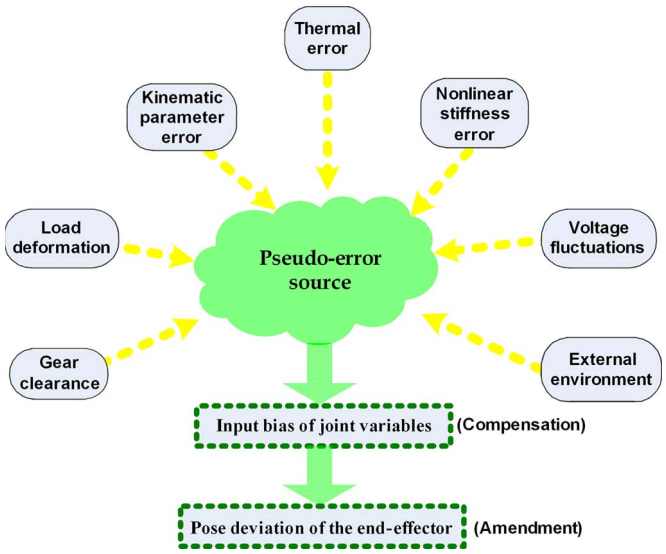


Fig. 2. Relationship of the pseudoerror source and the pose deviation of end-effector.

The homogeneous matrix of the end-effector with respect to the fixed reference frame can be expressed by $\boldsymbol{T}_P^O$, as shown in (2) at the bottom of the page.

When the real pose is not equal to the ideal value, it has

$$\begin{aligned} Pose_{\text{real}} &= \{x', y', z', \psi', \theta', \phi'\} \\ &= \{x + \Delta x, y + \Delta y, z + \Delta z, \psi + \Delta\psi, \\ &\quad \theta + \Delta\theta, \phi + \Delta\phi\} \end{aligned} \tag{3}$$

where $Pose_{\text{real}}$ is the real pose of the end-effector. Thus, when error happens, the homogeneous matrix is derived, as shown in (4) at the bottom of the page. The energy function of pose error of the end-effector is expressed as

$$E = E_{xyz} + \eta \cdot E_{\psi\theta\phi} \tag{5}$$

where the coefficient η denotes the influence degree of orientation error upon position error. Generally, the value η, whose normal range is [10, 40] mm, depends on the specific error conditions of the given parallel manipulators. This energy function reflects that the purpose of kinematic calibration and compensation of parallel manipulators is to minimize the position/orientation error of their end-effectors. The position error is defined by Euclidean distance, namely

$$E_{xyz} = \left(\Delta x^2 + \Delta y^2 + \Delta z^2\right)^{1/2}. \tag{6}$$

Similarly, orientation error can be expressed by cosine function

$$\begin{aligned} E_{\psi\theta\phi} = \arccos\big(&\cos\psi \cdot \cos(\psi + \Delta\psi) \\ &+ \cos\theta \cdot \cos(\theta + \Delta\theta) + \cos\phi \cdot \cos(\phi + \Delta\phi)\big). \end{aligned} \tag{7}$$

$$\boldsymbol{T}_P^O = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & x \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & y \\ -\sin\psi & \cos\theta\sin\phi & \cos\theta\cos\phi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\begin{aligned} \boldsymbol{T}_{P(Error)}^O &= \begin{bmatrix} \cos\psi'\cos\theta' & os\psi'\sin\theta'\sin\phi' - \sin\psi'\cos\phi' \\ \sin\psi'\cos\theta' & \sin\psi'\sin\theta'\sin\phi' + \cos\psi'\cos\phi' \\ -\sin\psi' & \cos\theta'\sin\phi' \\ 0 & 0 \end{bmatrix} \cdots \\ &\quad \begin{bmatrix} \cos\psi'\sin\theta'\cos\phi' + \sin\psi'\sin\phi' & x' \\ \sin\psi'\sin\theta'\cos\phi' - \cos\psi'\sin\phi' & y' \\ \cos\theta'\cos\phi' & z' \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\psi + \Delta\psi)\cos(\theta + \Delta\theta) \\ \sin(\psi + \Delta\psi)\cos(\theta + \Delta\theta) \\ -\sin(\psi + \Delta\psi) \\ 0 \end{bmatrix} \cdots \\ &\quad \begin{bmatrix} \cos(\psi + \Delta\psi)\sin(\theta + \Delta\theta)\sin(\phi + \Delta\phi) - \sin(\psi + \Delta\psi)\cos(\phi + \Delta\phi) \\ \sin(\psi + \Delta\psi)\sin(\theta + \Delta\theta)\sin(\phi + \Delta\phi) + \cos(\psi + \Delta\psi)\cos(\phi + \Delta\phi) \\ \cos(\theta + \Delta\theta)\sin(\phi + \Delta\phi) \\ 0 \end{bmatrix} \cdots \\ &\quad \begin{bmatrix} \cos(\psi + \Delta\psi)\sin(\theta + \Delta\theta)\cos(\phi + \Delta\phi) + \sin(\psi + \Delta\psi)\sin(\phi + \Delta\phi) & x' \\ \sin(\psi + \Delta\psi)\sin(\theta + \Delta\theta)\cos(\phi + \Delta\phi) - \cos(\psi + \Delta\psi)\sin(\phi + \Delta\phi) & y' \\ \cos(\theta + \Delta\theta)\cos(\phi + \Delta\phi) & z' \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{4}$$

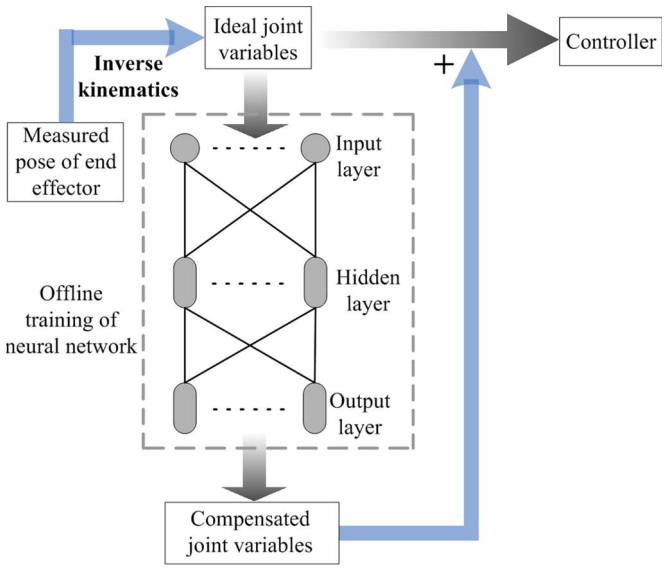


Fig. 3. Error compensation flow.

It is known that calibration is best performed in the least sensitive error region within an entire workspace. Furthermore, calibration should be conducted with the minimum number of joint configurations, as the calibration process is extremely time consuming. The system stiffness model is based on the kinetostatic model, with which the explicit expression of system stiffness can be generated at any pose. Thus, the critical components which have the largest effect on system stiffness can be identified. Due to the complexity of the error sources, it is difficult to develop the calibration model if all the errors will be considered. The neural network can be applied to describe the complex nonlinear relationship between joint variables (control parameters) and deviation of joint variables with respect to the measured pose of the end-effector. With neural network, the pseudoerror in arbitrary joint variable is acquired, and thus, the control parameters can be adjusted accordingly as shown in Fig. 3.

III. Cooperative Coevolutionary Network

The method for optimization of neural network based on cooperative coevolution algorithm is proposed aimed at the defects of previous genetic algorithm. The division of population, the selection of representation individuals, the method of fitness conformation, the design of evolution arithmetic operators, and the algorithm step are proposed. The credibility and the validity of this algorithm will be validated by idiographic examples. For the previous research, in [43], the cooperative coevolutionary algorithm with dual populations was developed for the training of radial basis function network to simultaneously implement the construction evolution and the feature selection via divide-and-cooperative mechanism. In [44], the modular structure of neural network was presented by means of a cooperative evolutionary process for pattern classification. In [45], the cooperative coevolutionary algorithm was combined with the mixture-of-experts model to overcome the defects of the original ME model. In [46], particle-swarm-optimization-based coevolutionary training techniques were applied for addressing the nonzero sum problem of the iterated prisoner's dilemma. However, the population segmentation method, encoding approach, selection strategy of representative individuals, and application potentials are still the main problems. Furthermore, to our knowledge, this is the first application of CCNN to establish the nonlinear relation between the pseudoerror source and joint parameters of parallel manipulators.

A. Overall Description

The genetic algorithms are based on the population-to-population rule; it can escape from local optima. Genetic algorithms have the advantages of robustness and good convergence properties by exploring all regions of state space and exponentially exploiting promising areas through mutation, crossover, and selection operations applied to individuals in the population. Cooperative coevolution, as an advanced genetic algorithm, models multispecie ecosystem for evolving interacting coadapted subcomponents [47]. Artificial neural networks contain nonlinear neurons which are intended to simulate some functionality of the human nervous system [48]. The whole neural network structure is organized with many neurons in order. The main function of neural networks is to establish the complex nonlinear relationship between inputs and outputs without deducing the mathematic expression. The primary properties for associative recall and function approximation have made them popular in real-world applications with a series of variations, such as multilayer perceptron [49], back propagation network [50], reduced size network [51], radial basis function network [52], wavelet neural network [53], and cascade neural network [54].

The simultaneous optimization for the structure and weights of neural network is a complex optimization problem, while the cooperative coevolutionary algorithm is the effective method to address this issue. This paper is focused on the pose compensation of end-effector of parallel robot based on CCNN.

The general idea of the algorithm is as follows: First, the entire neural network is divided into relatively simple multiple subnetworks; the divided subnetworks are encoded, respectively, to form multiple independent subpopulations as shown in Fig. 4. The genetic operation is conducted independently in the inner of each subpopulation, i.e., crossover and mutation. Only when conducting fitness assessment, will information be exchanged between subpopulations. Because individuals within a subpopulation only represent partial information of the neural network, direct evaluation of fitness cannot be accessed. In order to solve this problem, individuals to be assessed from different subpopulations should combine to form a complete network on which the fitness evaluation can be implemented. In this process, the selected individuals from other populations are called representative individuals. Some definition is given in the following fragments.

1) Population segmentation: It is through longitudinal splitting of the network to separate the connection between layers and its weight as independent populations, namely, subpopulations.
2) Encoding scheme: Perform binary encoding on the network topology structure and real-valued encoding on the connection weights, known as layered encoding.

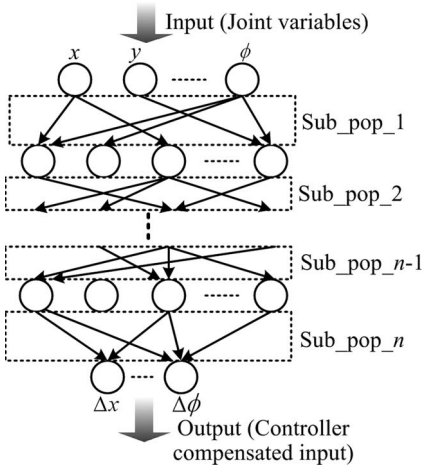


Fig. 4. Approach of segmentation population with CCNN for the error compensation of parallel robot.

3) Selection of representative individuals: As the coevolution adopts modular operator to carry out individual fitness assessment, the information from subpopulations is required. Thus, the representatives are chosen by the manner of "optimization + randomization."
4) Crossover operator: Conduct the multipoint crossover for binary code and the equivalent crossover for real code strings.
5) Mutation operator: Standard mutation operator is conducted on the binary code which indicates the network structure, and hybrid search method including the binary code and real code strings is performed to produce offspring.

B. Population Segmentation

The dashed frame in Fig. 4 is expressed by each subpopulation from Sub_pop_1 to Sub_pop_n, respectively, where Sub_pop_1 represents the connection relationship and weight coefficient between the input layer and the first hidden layer, Sub_pop_n donates the connection relationship and weight coefficient between the $(n-1)$th hidden layer and the output layer, and so forth.

C. Encoding Scheme

In order to simultaneously optimize the network structure and weights, the novel coding method is proposed which applies to all subpopulations. The basic idea is to encode the topology of the neural networks with binary string and encode weight coefficient with real-valued string.

1) Topology Encoding: For a neural network with $n-2$ hidden layers, $l_1, l_2 \ldots l_n$ represents each layer, respectively, and $n_1, n_2 \ldots n_n$ is the number of related nodes. $\boldsymbol{s}^{l_1 l_2}_{(n_2 \times n_1)}, \boldsymbol{s}^{l_2 l_3}_{(n_3 \times n_2)} \ldots \boldsymbol{s}^{l_{n-1} l_n}_{(n_n \times n_{n-1})}$ is the matrix denoting the connection relationship of every two adjacent layers. If $\boldsymbol{s}^{l_{m-1} l_m}_{(i,j)} = 1 (2 \le m \le n, 1 \le i \le n_m, 1 \le j \le n_{m-1})$, it means that the connection existed between the ith node of mth layer and jth node of $(m-1)$th layer. If $\boldsymbol{s}^{l_{m-1} l_m}_{(i,j)} = 0$, the connection does not exist. Taking Fig. 5 as an example, the structural encoding matrix of three subnetworks is

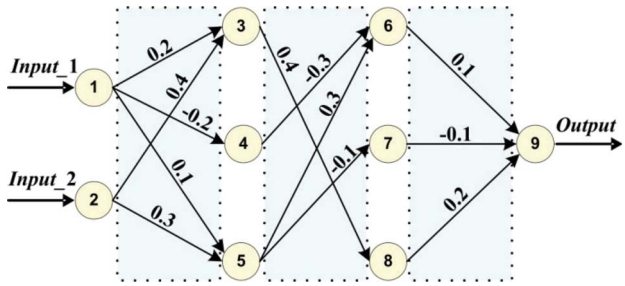


Fig. 5. Encoding of CCNN.

$$\boldsymbol{s}^{l_1 l_2}_{(3\times 2)} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}_{3\times 2} \tag{8}$$

which is equivalent to structural code string $\underline{11}\ \underline{10}\ \underline{11}$

$$\boldsymbol{s}^{l_2 l_3}_{(3\times 3)} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}_{3\times 3} \tag{9}$$

which is equivalent to structural code string $\underline{011}\ \underline{001}\ \underline{100}$, and

$$s^{l_3 l_4}_{(1\times 3)} = [1 \quad 1 \quad 1]_{1\times 3} \tag{10}$$

which is equivalent to structural code string $\underline{111}$.

2) Weight Coefficient Encoding: Real-valued encoding is utilized for weight coefficient. If the connection between nodes in adjacent layer is zero, then the weights are not assigned. Thus, the length of code string in the structural layer is equal to the number of "1" in the control layer. The code length of structural layer will be dynamically changed with the variation for the number of "1" in the control layer. Thus, the code length of weight coefficient need not be fixed, so the computational complexity can be reduced. Still, taking Fig. 5 as an example, the encoding of the corresponding weight coefficient is derived as

$$\boldsymbol{s}^{l_1 l_2} \xrightarrow{is\ corresponding\ to} 0.2, 0.4, -0.2, 0.1, 0.3$$

$$\boldsymbol{s}^{l_2 l_3} \xrightarrow{is\ corresponding\ to} -0.3, 0.3, -0.1, 0.4$$

$$\boldsymbol{s}^{l_3 l_4} \xrightarrow{is\ corresponding\ to} 0.1, -0.1, 0.2.$$

The formula to calculate the individual fitness is given as follows:

$$f_{\text{itness}} = \frac{1}{\frac{1}{p}\sum_{p=1}^{p}\left(\sqrt{\sum_{i=1}^{n}(t_{pi} - y_{pi})^2}\right) + \varepsilon} \tag{11}$$

where t_{pi} and y_{pi} are the pth sample and the ith outputs' nominal value and actual value, p is the total number of samples, n is the number of output of neural network, and ε is an appropriate small positive number.

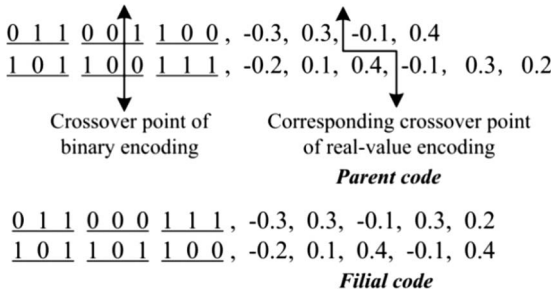


Fig. 6. Crossover operator for hybrid coding.

D. Crossover Operator

Since the proposed coding structure is different from traditional coding methods, appropriate genetic operators are required for the design. Single-point, multipoint, and consistent crossover operators can be conducted for the manipulation of binary code of individuals. The determination for the intersection point of real-valued string depends on the location of the intersection point for binary string. As the length of real-valued code string is equal to the number of "1" in the binary code string, the code length of real-valued code string before the crossover point should be equal to the number of "1" before the crossover point of binary code string to ensure that the total length of real-valued code string is still equivalent to the number of "1" in the binary code string after crossover, which is called equivalent crossover.

Fig. 6 shows the case of single-point crossover between one individual in the second subpopulation in Fig. 5 and another individual string in the same subpopulation. One crossover point is randomly selected for the structural code string of neural network, and equivalent crossover is implemented for the weight code.

The mutation operator of standard genetic algorithm is performed for the manipulation of binary code string which represents the connection relationship of nodes in the adjacent layer. As the weight encoding is influenced by the structural code string, if a particular bit is changed from one to zero, it means that the connection between two nodes is canceled. Then, the corresponding weight coefficients in real-valued code string should also be canceled and vise versa. The coding value is similar with the initialization of population and is confirmed in a given region randomly.

Obviously, the aforementioned mutation operator only generates strong mutation effectiveness for the binary code string. Hence, we propose the hybrid searching method to create a new offspring. Namely, the mutation operation of binary code string is executed on a part of the individuals while using nonuniform mutation operator for other individuals.

Nonuniform mutation is described as follows. Assuming that this is a vector $[w_1, \ldots, w_i, \ldots]$ in the coding region of real-valued weights in one subnetwork, w_i is chosen for nonuniform mutation. If the changing range is (LD, UD), the mutation result is $w' = (w'_1, \ldots, w'_i, \ldots)$, and

$$w'_i = \begin{cases} w_i + \Delta(t, UD - w_i) & \text{rand}(0,1) = 0 \\ w_i - \Delta(t, w_i - LD) & \text{rand}(0,1) = 1 \end{cases} \tag{12}$$

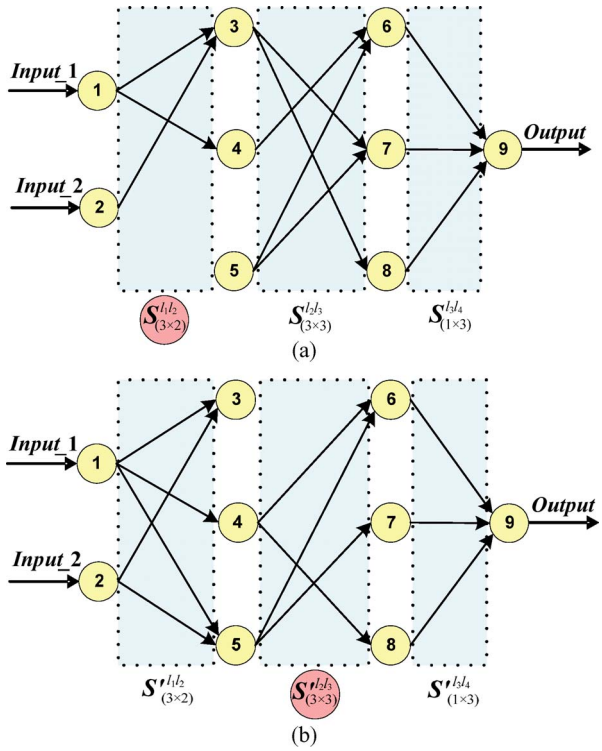


Fig. 7. (a) Phenotype of representative individual $\boldsymbol{s}^{l_1 l_2}_{(3\times 2)}$. (b) Phenotype of representative individual $\boldsymbol{s}'^{l_2 l_3}_{(3\times 3)}$.

where $\Delta(t, y)$ is a function with the range of $(0, y)$ and t is the evolutionary generation. When t is increasing, the changing range of w'_i is reduced, which improves the precision. Here, $\Delta(t, y)$ is expressed as

$$\Delta(t, y) = y \cdot \left(1 - r^{(1-t/T)^b}\right) \tag{13}$$

where r is random between [0, 1], T is the maximal evolutionary generation, and b is a system parameter which is usually equal to two.

E. Structural Optimization

When encoding the network topology, the matrix expression is equivalent with the code string. Some matrix is so special that some rows or columns are zeros, such as

$$\boldsymbol{s}^{l_1 l_2}_{(3\times 2)} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}_{3\times 2} \quad \boldsymbol{s}'^{l_2 l_3}_{(3\times 3)} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}_{3\times 3}. \tag{14}$$

The phenotypes of these two kinds of encoding matrix are shown in Fig. 7.

The coding structure should be monitored when calculating the individual fitness. If there is all-zero row or all-zero column, the encoding structure of other corresponding individual in the cooperative groups should be dynamically adjusted. Regarding to the node which has output and no input (structure matrix exists in all-zero row), or the node which has input and no output (structure matrix exists in all-zero column), it has no

contribution to the network, thereby deleting such node to reduce the length of the binary code.

However, due to the changes of the length for binary code string, the phenomenon of structure mismatch will occur for the individuals in different subpopulations which create difficulties for the evaluation of fitness. Moreover, the changes of the binary length will induce incontinence for the disposal of mutation operator. Therefore, to maintain the invariance of structure matrix, the redundant nodes are not temporarily dealt with in the evolutionary process. The information of structural matrix of binary code can be utilized when calculating the fitness value to reduce computational complexity. That is, if the kth row of structural matrix $\mathbf{S}^i$ is all zero, the kth column of structural matrix $\mathbf{S}^{i+1}$ should also be all zero. Similarly, if the jth column of structural matrix $\mathbf{S}^i$ is all zero, the jth row of structural matrix $\mathbf{S}^{i-1}$ should also be all zero. The length of corresponding real-valued code string is changed accordingly.

F. Algorithm Steps

The overall steps of the algorithm are given as follows:

Step 1) Initialize population $p(0) = (p_1(0), \ldots, p_i(0), \ldots, p_k(0))$, where k is the number of subpopulation.

Step 2) Calculate the fitness of all the individuals in $p(0)$.

Step 3) Implement the operators of selection, crossover, and mutation for the subpopulation to create new subpopulation of next generation.

Step 4) Adjust the network structure and calculate the fitness of the individuals in the subpopulation.

Step 5) If the network error related with the best individual in the solution space is convergent to a minimal value or if the evolutional generation is reached, the algorithm ends. Otherwise, switch to Step 3).

IV. Case Study

Two case studies were conducted in this section to verify the advantages of the proposed CCNN algorithm for the optimal kinematic calibration of parallel manipulators.

A. Case Study One: A Parallelogram-Based 3-DOF Parallel Manipulator

As a representative parallel manipulator, the 3-DOF tripod is composed of a base structure, a moving platform, and three legs connecting the base and platform. Among those three legs, two of them are in the same plane and consist of identical planar four-bar parallelograms as chains connected to the moving platform by revolute joints, while the third leg is one rectangular bar connected to the moving platform by a spatial joint. There is one revolute joint on the top end of each leg, and the revolute joint is linked to the base by an active prismatic joint. The computer-aided design (CAD) model of the 3-DOF parallel mechanism is shown in Fig. 8(a).

A kinematic model of the manipulator is shown in Fig. 8(b). The vertices of the moving platform are p_i ($i = 1, 2, 3$), and the vertices of the base are b_i ($i = 1, 2, 3$). A global reference system $O : O - xyz$ is located at the point of intersection b_1b_2 and Ob_3. Another reference system, called the moving frame $O' : O' - x'y'z'$, is located at the center of p_1p_2 on the moving platform. Robot workspace is the ability of a robot's end-effector to reach a collection of points from the origin of a coordinate system attached to the moving plate which depends on the configuration and size of their links and wrist joint [55]. The reachable workspace of the parallelogram-based parallel manipulator is shown in Fig. 8(c).

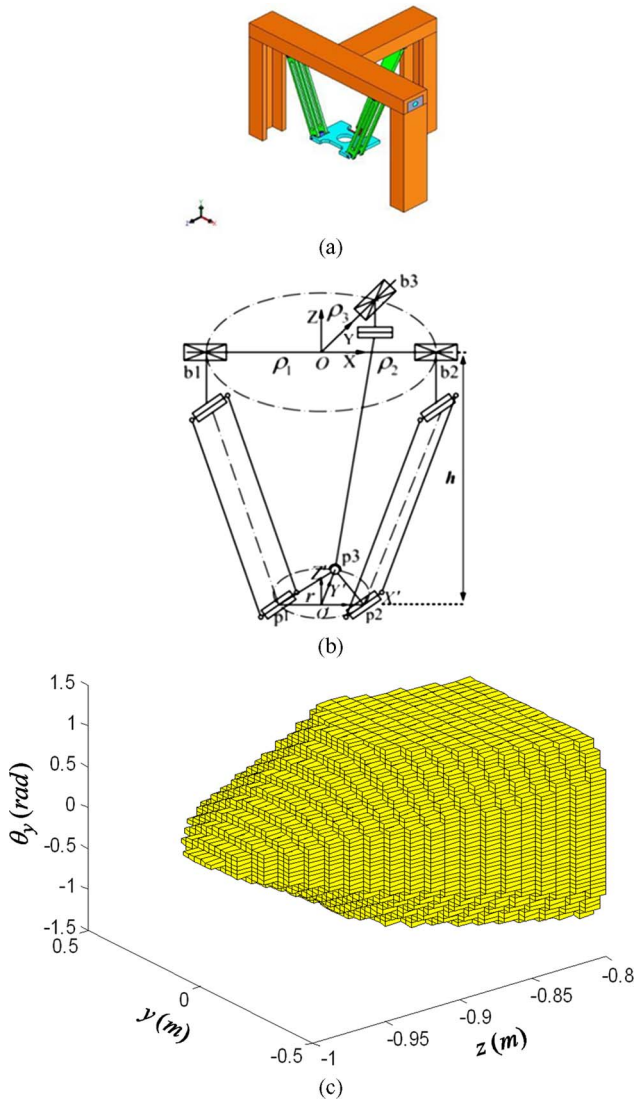


Fig. 8. (a) CAD modeling of parallelogram-based parallel manipulator. (b) Schematic representation. (c) Reachable workspace.

The inverse kinematics is given as

$$\rho_1 = \sqrt{L^2 - y^2 - (z + r\sin\phi)^2} + r\cos\phi \quad (15a)$$

$$\rho_2 = \sqrt{L^2 - y^2 - (z - r\sin\phi)^2} + r\cos\phi \quad (15b)$$

$$\rho_3 = \sqrt{L^2 - z^2} + y + r \quad (15c)$$

where ρ_i is the required actuator input, L is the fixed length of the supporting legs between the moving platform and base, and r is the radius of the moving platform.

According to the characteristics of errors for the parallel manipulators, it can be divided into three categories, i.e., the deterministic error, the random error, and the time-varying

TABLE I
PRECISION BEFORE CALIBRATION FOR THE PARALLELOGRAM-BASED PARALLEL MANIPULATOR

Sample	Training sets (100)		Verification Set (25)	
Joint error	Mean	Maximum	Mean	Maximum
Error_ρ_1 (m)	1.4863 $\times 10^{-4}$	1.8394 $\times 10^{-4}$	1.5050 $\times 10^{-4}$	1.6847 $\times 10^{-4}$
Error_ρ_2 (m)	9.2481 $\times 10^{-5}$	1.1746 $\times 10^{-4}$	9.0889 $\times 10^{-5}$	1.0883 $\times 10^{-4}$
Error_ρ_3 (m)	-9.3605 $\times 10^{-5}$	-6.6808 $\times 10^{-5}$	-9.2562 $\times 10^{-5}$	-6.7563 $\times 10^{-5}$

error. Without loss of generality, suppose that all three types of errors exist, namely

$$Error = e_{\text{constant}} + e_{\text{random}} + e_{\text{time\_variant}}. \quad (16)$$

The errors of given joint variables were

$$\begin{aligned} Error\_\rho_1 =& -10^{-4} + N(u=0, \sigma = 10^{-5}) \\ &+ 10^{-5}\sin(0.5/180 \times \pi \times t) \\ Error\_\rho_2 =& 10^{-4} + N(u=0, \sigma = 10^{-5}) \\ &- 10^{-5}\cos(0.5/180 \times \pi \times t) \\ Error\_\rho_3 =& 1.2 \times 10^{-4} + N(u=0, \sigma = 10^{-5}) \\ &+ 5 \times 10^{-5}\sin(0.5/180 \times \pi \times t) \\ &+ 5 \times 10^{-6}\cos(0.5/180 \times \pi \times t). \end{aligned} \quad (17)$$

In the workspace, 125 pose points of end-effector were chosen. The related ideal joint values were derived through the calculation of inverse kinematics. One hundred groups of the ideal joint values were randomly chosen as network input vectors, with the corresponding 100 groups of joint errors as the network output vectors. The rest 25 groups of ideal joint values and joint errors were the verification samples. The different transfer functions may strongly affect the actual performance and computing complexity of the proposed neural network. Since sigmoidal function exhibited excellent property when tackling with highly nonlinear large data sets, it was selected as the transfer function for hidden layer. Although, in many cases, one hidden layer was enough, one network with a higher order approximation capability was desirable in this scenario. For the back-propagation (BP) neural network, using more than one hidden layers may exacerbate the problem of local minima. While CCNN algorithm was based on the parallel processing of multiple subpopulations, one more hidden layer would increase the calibration precision accompanying with only an additional subpopulation. Thus, two hidden layers were used in this case. The size of each subpopulation depended on the nature of the problem, and it should be larger than the total number of the variables including the real-valued code and binary code; thus, the individuals in each subpopulation can span the searching space. Table I shows the statistical precision of driven joints of parallel robot before calibration.

When considering the integration methodology for the proposed pseudoerror approach and CCNN, the parameters were set as follows.

TABLE II
RESULT COMPARISON OF JOINT COMPENSATION AFTER CALIBRATION OF THE PARALLELOGRAM-BASED PARALLEL MANIPULATOR WITH LM, GRNN, ENN, AND CCNN METHODS IN ASPECTS OF MEAN SQUARED ERROR (MSE), MEAN ABSOLUTE ERROR (MAE), AND SUM SQUARED ERROR (SSE)

	Joint ρ_1	Joint ρ_2	Joint ρ_3	All joints
mse (BP_LM)	7.9493 $\times 10^{-10}$	9.0117 $\times 10^{-10}$	2.5542 $\times 10^{-9}$	1.4168 $\times 10^{-9}$
mae (BP_LM)	2.2556 $\times 10^{-5}$	2.7766 $\times 10^{-5}$	4.0760 $\times 10^{-5}$	3.0361 $\times 10^{-5}$
sse (BP_LM)	1.9873 $\times 10^{-8}$	2.2529 $\times 10^{-8}$	6.3856 $\times 10^{-8}$	1.0626 $\times 10^{-7}$
mse (GRNN)	2.3939 $\times 10^{-10}$	3.0567 $\times 10^{-10}$	1.3712 $\times 10^{-9}$	6.3875 $\times 10^{-10}$
mae (GRNN)	1.2989 $\times 10^{-5}$	1.4726 $\times 10^{-5}$	3.1408 $\times 10^{-5}$	1.9708 $\times 10^{-5}$
sse (GRNN)	5.9848 $\times 10^{-9}$	7.6418 $\times 10^{-9}$	3.4280 $\times 10^{-8}$	4.7906 $\times 10^{-8}$
mse (ENN)	1.6788 $\times 10^{-10}$	1.6346 $\times 10^{-10}$	2.0214 $\times 10^{-10}$	1.8089 $\times 10^{-10}$
mae (ENN)	1.0227 $\times 10^{-5}$	9.9658 $\times 10^{-6}$	1.1833 $\times 10^{-5}$	1.0527 $\times 10^{-5}$
sse (ENN)	4.1972 $\times 10^{-9}$	4.0865 $\times 10^{-9}$	5.0535 $\times 10^{-9}$	1.3567 $\times 10^{-8}$
mse (CCNN)	**9.4504 $\times 10^{-11}$**	**1.0614 $\times 10^{-10}$**	**1.2658 $\times 10^{-10}$**	**1.0907 $\times 10^{-10}$**
mae (CCNN)	**8.2787 $\times 10^{-6}$**	**7.5606 $\times 10^{-6}$**	**9.3438 $\times 10^{-6}$**	**8.3944 $\times 10^{-6}$**
sse (CCNN)	**2.3626 $\times 10^{-9}$**	**2.6535 $\times 10^{-9}$**	**3.1644 $\times 10^{-9}$**	**8.1805 $\times 10^{-9}$**

1) Input–output sample number is 100.
2) Dimension number is three.
3) Size of each subpopulation is 80.
4) Evolutional generation number is 120.
5) Crossover probability is 0.72.
6) Structural mutation probability is 0.0015.
7) Weight mutation probability is 0.0025.

Table II verifies the capacity of network generalization after the calibration of joint compensation for the parallelogram-mechanism-based tripod. The 25 groups of input–output sample pairs in the verification set were used to test the effectiveness of network training. To compare the proposed CCNN algorithm with the other methods given in the literature, the LM algorithm as one of the enhanced BP approaches, together with GRNN and ENN, was selected. The simulation results showed that the CCNN algorithm had a significant accuracy improvement, which illuminated the feasibility and advantage of the proposed method.

B. Case Study Two: A Parallel Robotic Machine Tool

Unlike most existing 3-DOF parallel manipulators, the parallel robotic machine tool shown in Fig. 9 contains a hybrid and uncoupled motion. The new design of this manipulator aims to

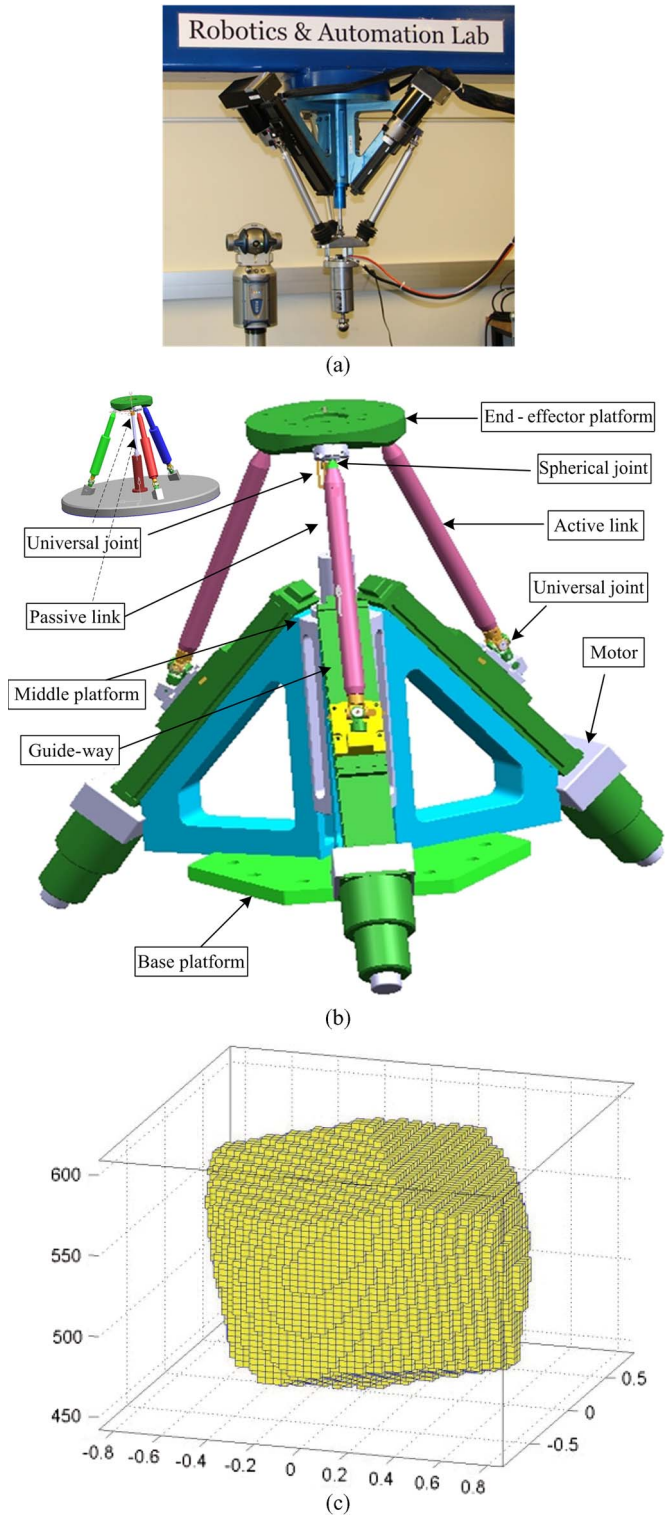


Fig. 9. (a) Prototype of the parallel robotic machine tool. (b) CAD modeling. (c) Reachable workspace.

achieve higher stiffness and implement pure 3-DOF motions, i.e., all side-effect motions can be eliminated.

The new manipulator has three platforms. The base platform is fixed on the ground, and the middle platform is used to support the path of the actuated links. The moving platform is used to mount a tool and helps to support the passive link, which is joined to the middle platform at the other end. The manipulator consists of three identical legs with active ball screw actuators, and a passive leg is installed between the base and the moving platform. Each of the actuated legs is connected with the moving platform by a spherical joint, and the passive link is fixed on the ground and connected to the moving platform by a universal joint; thus, redundant motions can be eliminated by the universal joint.

The proposed parallel manipulator includes two innovative features. First, the universal joint of the passive link is located on the moving platform rather than on the base platform, thus eliminating the motions along the x and y translations and the z rotation. Second, the reference point on the moving platform has hybrid and independent motions with x and y rotations and a z translation. The reachable workspace of the parallelogram-based parallel manipulator is shown in Fig. 9(c).

The inverse kinematic model is represented as

$$\begin{aligned}&[l_e\cos\beta_i\cos\theta_y + z_0\sin\theta_y - (l_b - u_i\cos\gamma)\cos\alpha_i]^2\\&\quad+[l_e\cos\beta_i\sin\theta_x\sin\theta_y + l_e\sin\beta_i\cos\theta_x\\&\qquad-z_0\sin\theta_x\cos\theta_y - (l_b - u_i\cos\gamma)\sin\alpha_i]^2\\&\quad+[-l_e\cos\beta_i\cos\theta_x\sin\theta_y + l_e\sin\beta_i\sin\theta_x + z_e\\&\qquad+z_0\cos\theta_x\cos\theta_y - u_i\sin\gamma]^2 = l^2\end{aligned}\tag{18}$$

where u_i $(i = 1, 2, 3)$ is the actuated joint input, l_e is the radius of the moving platform, l_b is the radius of the base platform, z_o is the offset of the spherical joints on the moving platform, γ is the direction of a guideway, a_i $(i = 1, 2, 3)$ is the angle of the attachment point on the base, and β_i $(i = 1, 2, 3)$ is the angle of the attachment point on the moving platform.

Without loss of generality, the hypothetical errors applied for the deviation of the actuated joint parameters were set as

$$\begin{aligned}Error\_\rho_1 =& 6\times10^{-4} + N(u=0,\sigma=2\times10^{-5})\\&+10^{-5}\sin(t)\cdot\sin(t^2/\pi)\\Error\_\rho_2 =& 3\times10^{-4} + N(u=0,\sigma=5\times10^{-4})\\&+10^{-5}\cos(t)\cdot e^{-t}\\Error\_\rho_3 =& 4.8\times10^{-4} + N(u=0,\sigma=2.5\times10^{-5})\\&+10^{-5}/1+e^{-t}.\end{aligned}\tag{19}$$

Similar with the first case, 152 pose configurations in the reachable workspace of the end-effector were utilized for calibration. Through inverse kinematic model, the ideal joint values were accordingly deduced, in which 125 groups were elected as training samples, and the rest 27 groups were elected as verification sets. Table III describes the mean and maximum of the pseudoerror which affected the precision of the actuated joints before calibration.

Table IV shows the result comparison between LM-based BP algorithm, GRNN, ENN, and CCNN methods. It can be observed that the manipulator structure and the error component were more complex than the aforementioned case. This illustrated that the CCNN algorithm possessed the characteristics of stronger robustness to ensure the accuracy after calibration than other compared methods.

TABLE III
PRECISION BEFORE CALIBRATION FOR THE DRIVEN JOINTS OF THE PARALLEL ROBOTIC MACHINE TOOL

Sample	Training sets (125)		Verification Sets (27)	
Joint error	Mean	Maximum	Mean	Maximum
Error_u_1 (m)	6.0141×10^{-4}	6.1819×10^{-4}	6.0087×10^{-4}	6.1127×10^{-4}
Error_ u_2 (m)	2.9986×10^{-4}	3.5472×10^{-4}	3.0216×10^{-4}	3.5766×10^{-4}
Error_ u_3 (m)	4.8962×10^{-4}	4.9896×10^{-4}	4.9001×10^{-4}	4.9667×10^{-4}

TABLE IV
RESULT COMPARISON OF JOINT COMPENSATION AFTER THE CALIBRATION OF THE PARALLEL ROBOTIC MACHINE TOOL WITH LM, GRNN, ENN, AND CCNN METHODS

	Joint u_1	Joint u_2	Joint u_3	All joints
mse (BP_LM)	2.2596×10^{-9}	9.4589×10^{-7}	6.0174×10^{-9}	3.1806×10^{-7}
mae (BP_LM)	3.8286×10^{-5}	8.0882×10^{-4}	6.7983×10^{-5}	3.0503×10^{-4}
sse (BP_LM)	6.1010×10^{-8}	2.5539×10^{-5}	1.6247×10^{-7}	2.5762×10^{-5}
mse (GRNN)	4.6922×10^{-10}	3.0567×10^{-8}	2.1424×10^{-9}	1.1059×10^{-8}
mae (GRNN)	1.8185×10^{-5}	1.4726×10^{-4}	3.9260×10^{-5}	6.8234×10^{-5}
sse (GRNN)	1.2669×10^{-8}	8.2531×10^{-7}	5.7845×10^{-8}	8.9578×10^{-7}
mse (ENN)	1.5321×10^{-10}	1.3586×10^{-8}	9.5220×10^{-10}	4.8970×10^{-9}
mae (ENN)	1.0392×10^{-5}	9.8172×10^{-5}	2.6173×10^{-5}	4.4912×10^{-5}
sse (ENN)	4.1367×10^{-9}	3.6682×10^{-7}	2.5709×10^{-8}	3.9666×10^{-7}
mse (CCNN)	**8.6038×10^{-11}**	**3.6269×10^{-10}**	**4.5644×10^{-11}**	**1.6479×10^{-10}**
mae (CCNN)	**8.3452×10^{-6}**	**1.5635×10^{-5}**	**5.7632×10^{-6}**	**9.9144×10^{-6}**
sse (CCNN)	**2.3230×10^{-9}**	**9.7925×10^{-9}**	**1.2324×10^{-9}**	**1.3348×10^{-8}**

V. CONCLUSION

Kinematic calibration is an urgent issue for the more extensive applications of parallel manipulators in industry, as it has the direct influence on operational accuracy. In this paper, a comprehensive algorithm for optimal kinematic calibration and joint compensation of parallel manipulators has been proposed through the integration of pseudoerror theory and CCNN. Generally, manufacturing and assembly error, thermal error, and nonlinear stiffness error constitute the multierror sources. Pseudoerror theory works in such a way that the various and complex error sources can be considered together as a single function generator which is applied on the actuated joint input and causes its deviation. Thus, pose compensation of end-effector can be solved in a natural and compact way. Inspired by the cooperation of biological swarm intelligence, the CCNN algorithm aims to model the highly nonlinear relationship between ideal joint variables and controller compensation input. The proposed methodology supplies a widely applicable solution dedicated to parallel manipulators, which is suitable for external calibration, self-calibration, and constrained calibration of general serial and parallel mechanisms with high-accuracy requirement.

REFERENCES

[1] W. L. Xu, J. S. Pap, and J. Bronlund, "Design of a biologically inspired parallel robot for foods chewing," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 832–841, Feb. 2008.

[2] D. Zhang, Z. Gao, B. Song, and Y. J. Ge, "Configuration design and performance analysis of a multidimensional acceleration sensor based on 3RRPRR decoupling parallel mechanism," in *Proc. IEEE Int. Conf. Decision Control*, 2009, pp. 8304–8309.

[3] G. R. Dunlop and T. P. Jones, "Position analysis of a two DOF parallel mechanism—Canterbury tracker," *Mech. Mach. Theory*, vol. 34, no. 4, pp. 599–614, May 1999.

[4] D. Zhang and C. M. Gosselin, "Parallel kinematic machine design with kinetostatic model," *Robotica*, vol. 4, no. 7, pp. 429–438, 2002.

[5] J. A. Carretero, R. P. Podhorodeski, M. N. Nahon, and C. M. Gosselin, "Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator," *J. Mech. Des.*, vol. 122, no. 1, pp. 17–24, Mar. 2000.

[6] D. Zhang and L. Wang, "Conceptual development of an enhanced tripod mechanism for machine tool," *Robot. Comput.-Integr. Manuf.*, vol. 21, no. 4/5, pp. 318–327, Aug.–Oct. 2005.

[7] Z. M. Bi, S. Y. T. Lang, D. Zhang, P. E. Orban, and M. Verner, "An integrated design toolbox for tripod-based parallel kinematic machines," *J. Mech. Des.*, vol. 129, no. 8, pp. 799–807, Aug. 2007.

[8] S. Lessard, P. Bigras, and I. A. Bonev, "A new medical parallel robot and its static balancing optimization," *J. Med. Devices*, vol. 1, no. 4, pp. 272–278, Dec. 2007.

[9] J. P. Merlet, "Optimal design for the micro parallel robot MIPS," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 1149–1154.

[10] D. Zhang, L. Wang, and E. Esmailzadeh, "PKM capabilities and applications exploration in a collaborative virtual environment," *Robot. Comput.-Integr. Manuf.*, vol. 22, no. 4, pp. 384–395, Aug. 2006.

[11] J. A. Soons, "On the geometric and thermal errors of a Hexapod machine tools," in *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements*. New York: Springer-Verlag, 1999, ser. Advanced Manufacturing Series, pp. 151–170.

[12] A. Yu, I. A. Bonev, and P. Z. Murray, "Geometric approach to the accuracy analysis of a class of 3DOF planar parallel robots," *Mech. Mach. Theory*, vol. 43, no. 3, pp. 364–375, Mar. 2008.

[13] X. J. Liu, J. S. Wang, and G. Pritschow, "A new family of spatial 3-doF fully-parallel manipulators with high rotational capability," *Mech. Mach. Theory*, vol. 40, no. 4, pp. 475–494, Apr. 2005.

[14] P. Chiacchio, F. Pierrot, L. Sciavicco, and B. Siciliano, "Robust design of independent joint controllers with experimentation on a high-speed parallel robot," *IEEE Trans. Ind. Electron.*, vol. 40, no. 4, pp. 393–403, Aug. 1993.

[15] S. Staicu, D. Zhang, and R. D. Rugescu, "Dynamic modelling of a 3-DOF parallel manipulator using iterative matrix relations," *Robotica*, vol. 24, pp. 125–130, 2006.

[16] R. Saltaren, R. Aracil, O. Reinoso, and M. A. Scarano, "Climbing parallel robot: A computational and experimental study of its performance around structural nodes," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1056–1066, Dec. 2005.

[17] H. B. Choi, A. Konno, and M. Uchiyama, "Design, implementation, and performance evaluation of a 4-DOF parallel robot," *Robotica*, vol. 28, no. 1, pp. 107–118, Jan. 2010.

[18] J. P. Merlet, *Parallel Robots*. New York: Springer-Verlag, 2006.

[19] P. Maurine and E. Dombre, "A calibration procedure for the parallel robot Delta 4," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 975–980.

[20] H. S. Kim, "Kinematic calibration of a Cartesian parallel manipulator," *Int. J. Control Autom. Syst.*, vol. 3, no. 3, pp. 453–460, Sep. 2005.

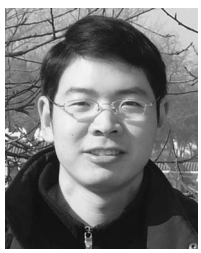
[21] P. Renaud, N. Andreff, F. Marquet, and P. Martinet, "Vision-based kinematic calibration of a H4 parallel mechanism," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1191–1196.

[22] P. Renaud, N. Andreff, J. M. Lavest, and M. Dhome, "Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 12–22, Feb. 2006.
[23] D. Daney, N. Andreff, G. Chabert, and Y. Papegay, "Interval method for calibration of parallel robots: Vision-based experiments," *Mech. Mach. Theory*, vol. 41, no. 8, pp. 929–944, Aug. 2006.
[24] J. I. Jeong, D. S. Kang, Y. M. Cho, and J. W. Kim, "Kinematic calibration for redundantly actuated parallel mechanisms," *J. Mech. Des.*, vol. 126, no. 2, pp. 307–318, Mar. 2004.
[25] A. Rauf and J. Ryu, "Fully autonomous calibration of parallel manipulators by imposing position constraint," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, pp. 2389–2394.
[26] D. Daney and I. Z. Emiris, "Robust parallel robot calibration with partial information," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, pp. 3262–3267.
[27] A. Nahvi and J. M. Hollerbach, "Calibration of a parallel robot using multiple kinematic closed loops," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 407–412.
[28] A. Mansour, P. Hodjat, A. Aria, and V. Gholamreza, "Kinematic calibration of the Hexaglide parallel robot using a simple measurement system," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, 2009, pp. 853–859.
[29] P. Last, A. Raatz, J. Hesselbach, N. Pavlovic, and R. Keimer, "Parallel robot calibration utilizing adaptronic joints," in *Proc. ASME Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2009, pp. 1277–1284.
[30] W. Y. Hsu and J. S. Chen, "Error analysis and auto-calibration for a Cartesian-guided tripod machine tool," *Int. J. Adv. Manuf. Technol.*, vol. 24, no. 11/12, pp. 899–909, Dec. 2004.
[31] C. W. Wampler, J. M. Hollerbach, and T. Arai, "An implicit loop method for kinematic calibration and its application to closed-chain mechanisms," *IEEE Trans. Robot. Autom.*, vol. 11, no. 5, pp. 710–724, Oct. 1995.
[32] P. Vischer and R. Clavel, "Kinematic calibration of the parallel Delta robot," *Robotica*, vol. 16, no. 2, pp. 207–218, Mar. 1998.
[33] M. J. Hwang, O. Bebek, F. Liang, B. W. Fei, and M. C. Cavusoglu, "Kinematic calibration of a parallel robot for small animal biopsies," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4104–4109.
[34] L. Savoure, P. Maurine, D. Corbel, and S. Krut, "An improved method for the geometrical calibration of parallelogram-based parallel robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 769–776.
[35] D. Daney, Y. Papegay, and A. Neumaier, "Interval methods for certification of the kinematic calibration of parallel robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 1913–1918.
[36] M. S. Varziri and L. Notash, "Kinematic calibration of a wire-actuated parallel robot," *Mech. Mach. Theory*, vol. 42, no. 8, pp. 960–976, Aug. 2007.
[37] N. A. Zaghloul and M. A. Abu Kiefa, "Neural network solution of inverse parameter used in the sensitivity-calibration analysis of the SWMM model simulations," *Adv. Eng. Softw.*, vol. 32, no. 7, pp. 587–595, Jul. 2001.
[38] R. Abielmona, V. Groza, and E. Petriu, "Evolutionary neural network-based sensor self-calibration scheme using IEEE 1451 and wireless sensor networks," in *Proc. IEEE Int. Symp. Comput. Intell. Meas. Syst. Appl.*, 2003, pp. 38–43.
[39] X. M. Bian, X. R. Li, H. M. Chen, D. Q. Gan, and J. Qiu, "Joint estimation of state and parameter with synchrophasors—Part I: State tracking," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1196–1208, Aug. 2011.
[40] S. Zhang, L. Ding, J. Xu, F. Zhang, S. Wang, and Y. Chang, "Digital background calibration of MDAC stage gain error and DAC error in pipelined ADC," in *Proc. IEEE Int. Conf. Solid-State Integr. Circuit Technol.*, 2010, pp. 251–253.
[41] R. L. Czaplewski and G. P. Catts, "Calibration of remotely sensed proportion or area estimates for misclassification error," *Remote Sens. Environ.*, vol. 39, no. 1, pp. 29–43, Jan. 1992.
[42] C. de Vries, T. Danaher, R. Denham, P. Scarth, and S. Phinn, "An operational radiometric calibration procedure for the Landsat sensors based on pseudo-invariant target sites," *Remote Sens. Environ.*, vol. 107, no. 3, pp. 414–429, Apr. 2007.
[43] J. Tian, M. Li, and F. Chen, "Dual-population based coevolutionary algorithm for designing RBFNN with feature selection," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 6904–6918, Oct. 2010.
[44] N. Garcia-Pedrajas and D. Ortiz-Boyer, "A cooperative constructive method for neural networks for pattern recognition," *Pattern Recognit.*, vol. 40, no. 1, pp. 80–98, Jan. 2007.
[45] M. H. Nguyen, H. A. Abbass, and R. I. Mckay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 70, pp. 155–163, 2006.
[46] N. Franken and A. P. Engelbrecht, "Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma," *IEEE Trans. Evol. Comput.*, vol. 9, no. 6, pp. 562–579, Dec. 2005.
[47] M. Potter and K. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, 2000.
[48] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
[49] V. Machado, A. Doria Neto, and J. Dantas de Melo, "A neural network multiAgent architecture applied to industrial networks for dynamic allocation of control strategies using standard function blocks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1823–1834, May 2010.
[50] C. Xia, C. Guo, and T. Shi, "A neural network identifier and fuzzy controller based algorithm for dynamic decoupling control of permanent magnet spherical motor," *IEEE Trans. Ind. Electron.*, vol. 57, no. 8, pp. 2868–2878, Aug. 2010.
[51] E. Echenique, J. Dixon, R. Cardenas, and R. Pena, "Sensorless control for a switched reluctance wind generator, based on current slopes and neural networks," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 817–825, Mar. 2009.
[52] G. Chang, C. Chen, and Y. Teng, "Radial basis function-based neural network for harmonics detection," *IEEE Trans. Ind. Electron.*, vol. 57, no. 6, pp. 2171–2179, Jun. 2010.
[53] F. El-Sousy, "Hybrid $H\infty$-based wavelet-neural-network tracking control for permanent-magnet synchronous motor servo drives," *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3157–3166, Sep. 2010.
[54] V. N. Ghate and S. V. Dudul, "Cascade neural-network-based fault classifier for three-phase induction motor," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1555–1563, May 2011.
[55] O. Masory and J. Wang, "Workspace evaluation of Stewart platforms," *Adv. Robot.*, vol. 9, no. 4, pp. 443–461, 1995.

Dan Zhang (M'05–SM'08) received the Ph.D. degree in robotics and mechatronics from Laval University, Quebec City, Canada, in June 2000.

He was an R&D Engineer and a Materials and Manufacturing Ontario (MMO) Industrial Fellow at a company for a while. From 2000 to 2004, he was a Research Scientist with the Integrated Manufacturing Technologies Institute, National Research Council, Canada. He is a Full Professor and a Canada Research Chair in robotics and automation with the Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Canada. His research interests include robotics and mechatronics; high-performance parallel robotic machine development; sustainable/green manufacturing systems; micro-/nanomanipulation and microelectromechanical system (MEMS) devices (sensors), micromobile robots and control of multirobot cooperation, and intelligent servo control system for the MEMS-based high-performance microrobot; web-based remote manipulation; wearable power assist hip exoskeleton; and rescue robot. He has published 150 journal and conference papers, two books, five book chapters, and numerous other technical publications. He is the Editor-in-Chief for the *International Journal of Mechanisms and Robotic Systems*.

Dr. Zhang was the Director of the Board of Directors of Durham Region Manufacturing Association, Canada, and is the Director of the Board of Directors of Professional Engineers Ontario, Lake Ontario Chapter, Canada. He is a Registered Professional Engineer of Canada, a Fellow of Canadian Society for Mechanical Engineering (CSME), a Senior Member of Society of Manufacturing Engineers (SME), and a member of American Society of Mechanical Engineers (ASME). He was the recipient of the MMO Industrial Fellowship.



Zhen Gao received the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in July 2009.

He was a Research Assistant with Robot Sensor and Human–Machine Interaction Laboratory, Institute of Intelligent Machines, Chinese Academy of Sciences, Beijing, China. He is a Postdoctoral Research Fellow with the Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, where he has been the Manager of Robotics and Automation Laboratory since May 2010. His current research interests include parallel robots, medical robots, microelectromechanical systems, sensors, and artificial intelligence.