



OPEN Accurate robot calibration via cascaded adaptive momentum LM and B-spline interpolated PSO

Dong Li¹✉, Yonghong Deng^{2,4}✉ & Zhibin Li^{3,4,5}

To improve robotic positioning accuracy and enhance overall manufacturing precision, this paper proposes an adaptive momentum Levenberg–Marquardt cascaded B-spline interpolation particle swarm optimization (AMLM-BIPSO) algorithm for calibrating robotic geometric errors. Initially, a momentum term is incorporated into the traditional Levenberg–Marquardt algorithm to suppress overshooting and oscillation, thereby improving the preliminary estimation of geometric parameters. Subsequently, inspired by the concept of Knowledge-based artificial Neural Networks, B-spline interpolation is embedded into the standard particle swarm optimization framework to refine the final calibration. By cascading these two enhanced techniques, the proposed method achieves higher accuracy in parameter identification. Experimental validation on an industrial robot confirms that the AMLM-BIPSO algorithm yields substantial improvements in positioning accuracy and calibration reliability.

Keywords Kinematic parameters calibration, Positioning accuracy, Levenberg–Marquardt, B-spline interpolation, Particle swarm optimization algorithm

With the widespread integration of robots into modern manufacturing systems, their performance has become a decisive factor in determining both production efficiency and product quality. As a result, there is a growing emphasis on improving the positioning accuracy and overall manufacturing precision of robotic systems. A robot's positioning accuracy is typically characterized by two key metrics: absolute positioning accuracy and repeatability. The former denotes the robot's ability to reach a designated spatial target accurately, while the latter reflects its consistency in returning to the same location across repeated operations¹. Repeatability is primarily governed by the mechanical design and fabrication quality of the robot. In contrast, absolute accuracy is significantly affected by discrepancies between the robot's actual and nominal geometric parameters, which arise from factors such as machining tolerances, assembly misalignments, and long-term wear. These discrepancies are collectively referred to as geometric parameter errors, which have been reported to contribute up to 90% of the total absolute positioning error in industrial robots². Therefore, precise calibration of geometric parameters is critical for enhancing positioning accuracy, and has become a focal point of research in robotic optimization. To address this challenge, extensive efforts have been directed toward developing advanced modeling techniques, intelligent optimization algorithms, and effective error compensation strategies to support high-precision robotic applications.

To calibrate robotic geometric parameters effectively, one must address the challenge of solving a high-dimensional and nonlinear optimization problem. Extensive research has been conducted in this domain, leading to various algorithmic approaches aimed at improving positioning accuracy. These approaches can be broadly categorized into four main types: gradient-based optimization, filtering-based estimation, step-size adaptive methods, and swarm intelligence-based strategies.

Gradient-based optimization algorithms minimize the error function by leveraging gradient information, with the nonlinear Least Squares (NLS) and Levenberg–Marquardt (LM) methods being the most widely used. Mao et al.³ introduced a separable NLS algorithm to mitigate joint deformation and convergence issues, achieving a 92.47% improvement in positioning accuracy under variable loading conditions. Wang et al.⁴ incorporated spherical constraints into an NLS framework to simultaneously account for geometric and end-

¹Department of Mechanical and Electrical Engineering, Liaocheng University Dongchang College, Liaocheng 252000, Shandong, China. ²School of Science, Xihua University, Chengdu 610039, China. ³School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China. ⁴Dazhou Key Laboratory of Government Data Security, Sichuan University of Arts and Science, Dazhou 635000, China. ⁵Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China. ✉email: ldjylidong@163.com; dengyhcd@163.com

effector deformation errors. Deng et al.⁵ proposed a Chebyshev-interpolated LM (CILM) algorithm for robotic smoothing systems, significantly enhancing absolute positioning precision in optical component processing. Li et al.⁶ further improved calibration accuracy by integrating an unscented Kalman filter (UKF) with a variable step-size LM algorithm, reporting a 19.51% enhancement over the best-performing LM-based method.

Filtering-based optimization methods focus on real-time compensation and dynamic parameter estimation. Techniques such as the Extended Kalman Filter (EKF), Particle Filter (PF), and UKF have been extensively applied. Liu et al.⁷ employed an EKF approach supported by an optical tracking system for improved target tracking accuracy. Deng et al.⁸ developed an adaptive residual EKF that alleviated gradient vanishing issues and improved identification performance. Du et al.⁹ integrated a hybrid filtering strategy combining the UKF with iterative PF to facilitate online calibration with high autonomy. Gao et al.¹⁰ proposed a UKF enhanced by adaptive process noise covariance to overcome the limitations of linearized models in capturing high-order system dynamics.

Step-size adaptive methods employ flexible adjustment strategies to improve convergence speed. A representative example is the Beetle Antennae Search (BAS) algorithm. Li et al.¹¹ proposed a robot arm calibration method by fusing EKF with a quadratic-interpolated BAS, effectively enhancing convergence and accuracy. Similarly, Chen et al.¹² developed a calibration technique based on an improved Beetle Swarm Optimization (BSO) algorithm, specifically tailored for industrial drilling and riveting tasks.

Swarm intelligence-based methods emulate collective biological behaviors, exhibiting robust global search capability. Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Squirrel Search Algorithm (SSA) are prominent representatives. Cao et al.¹³ proposed a hybrid method combining EKF, dual quantum-behaved PSO, and an adaptive neuro-fuzzy inference system (ANFIS). Liu et al.¹⁴ integrated an adaptive GA with simulated annealing to minimize motion uncertainty. Ma et al.¹⁵ introduced a Lévy flight-assisted SSA framework to optimize measurement poses and reduce end-effector error.

Despite their respective advantages, each approach faces inherent limitations due to the complex, nonlinear, and highly coupled nature of robotic kinematic calibration problems. A summary of these limitations is presented in Table 1.

Existing approaches for robot kinematic parameter calibration can be broadly categorized into four main classes: gradient-based methods, filtering-based estimators, adaptive step-size algorithms, and swarm intelligence techniques.

Gradient-based methods, such as the LM algorithm, are widely adopted for their fast convergence. However, they are often sensitive to initial conditions and prone to entrapment in local minima. Filtering-based estimators, including the EKF and UKF, offer improved adaptability for online calibration, yet typically rely on Gaussian noise assumptions and accurate prior models. Adaptive step-size algorithms, such as BAS and BSO, enhance robustness in convergence, though they can suffer from instability in parameter tuning. Recently, swarm intelligence techniques, such as GA, PSO, and SSA, have attracted considerable interest due to their global optimization capabilities. Despite their potential, these algorithms are still challenged by premature convergence, population stagnation, and insufficient local exploitation in high-dimensional spaces.

To address these limitations, several studies have explored hybrid frameworks that combine traditional optimization with intelligent strategies. For instance, the integration of Kalman filtering with heuristic search algorithms has shown promise in balancing global exploration and local estimation. Nevertheless, limited research has focused on reconstructing population structures via interpolation-based mechanisms or mitigating overshooting and instability issues inherent in LM-type methods through dynamic momentum correction.

In this work, a novel Adaptive Momentum Levenberg–Marquardt Cascaded B-spline Interpolation Particle Swarm Optimization (AMLMB-IPSO) algorithm is proposed for calibrating robot geometric parameters. The PSO algorithm serves as the foundational framework owing to its simplicity, strong global search ability, and proven effectiveness across various engineering optimization problems. By incorporating B-spline interpolation into the PSO update mechanism, we introduce a smooth, continuous adjustment strategy that is algorithm-agnostic and potentially extensible to other metaheuristics such as GA, DE, or SSA. This design offers a generalized pathway for enhancing population diversity and search stability.

The main contributions of this study are as follows:

- (a) Momentum-corrected Levenberg–Marquardt: A dynamic momentum term is incorporated into the LM update rule to account for historical trends and local curvature, thereby suppressing overshooting and improving parameter identification stability.
- (b) B-spline-based population interpolation: Drawing inspiration from Knowledge-based Artificial Neural Networks, B-spline interpolation is used to reconstruct the particle distribution, improving continuity in the search space, and enhancing convergence smoothness.

| Algorithm category | Main drawbacks |
|--|--|
| Gradient-based (LS, LM) | There is truncation error, making it prone to converging to a local optimum rather than the global optimum |
| Filter-based (EKF, PF, UKF) | Highly dependent on covariance matrix selection; EKF and UKF assume Gaussian noise, limiting effectiveness in complex environments |
| Step-length update (BAS, etc.) | Convergence speed varies with step-size tuning; may still fall into local optima without an effective update strategy |
| Swarm intelligence algorithms (PSO, GA, SSA, etc.) | It may lose diversity due to population clustering, leading to convergence to a local optimum |

Table 1. Main drawbacks of robot current calibration methods.

- (c) Cascaded optimization framework: The integration of AMLM and BIPSO leverages the strengths of both local precision and global exploration, leading to more reliable and accurate parameter calibration.

The remainder of the paper is organized as follows: Section “Kinematic and error models of robot” introduces the kinematic modeling and error formulation; Section “Identification of robot kinematic parameters” describes the proposed calibration approach; Section “Experimental verifications” presents experimental results; and Section “Conclusion” concludes the study.

Kinematic and error models of robot

The robot system used in this study is an ABB IRB 120 industrial robot, as illustrated in Fig. 1. A typical robotic operation begins with task planning, where a motion path is designed based on specified task requirements. This path is subsequently post-processed and converted into executable code, which is then deployed to the robot controller. Over time, factors such as mechanical wear, structural deformation, and thermal expansion can lead to deviations in the robot’s nominal Denavit-Hartenberg (DH) parameters, which define its kinematic structure. These deviations introduce discrepancies between the planned and actual motion trajectories, compromising both accuracy and manufacturing quality. As a result, the robot may deviate from the intended path, potentially leading to defects such as misalignment, uneven seam widths, or poor fusion quality in welding or assembly operations. To address these issues, robot calibration plays a critical role. By systematically identifying the actual DH parameters through a calibration procedure, geometric deviations can be effectively quantified and compensated. This reduces trajectory errors and enables the robot to follow the planned path more precisely, thereby improving operational accuracy, consistency, and product quality.

As shown in Fig. 1, when using a wire encoder for calibration, the position error of the robot’s end effector can be converted into the measurement error of the wire length. By accurately measuring the changes in wire length, the positional deviation of the robot’s end effector can be further derived and analyzed, enabling quantitative error evaluation and compensation. Thus, the following relationship holds:

$$\Delta L = \|P_{BF} - P_E\|_2 - L_{EF} \quad (1)$$

where ΔL represents the deviation between the actual and theoretical cable lengths. P_{BF} denotes the position of the robot flange end center in the robot base coordinate system, which is obtained through the robot kinematic model and includes the geometric parameter error to be identified. P_E refers to the position of the cable encoder port in the robot base coordinate system, directly provided by the measurement software. L_{EF} is the actual measured cable length, obtained directly from the encoder.

Robot calibration consists of four fundamental stages: modeling, measurement, parameter identification, and error compensation. Among these, modeling plays a crucial role in determining robot positioning accuracy. The DH model has long been the standard kinematic representation for serial robots. Consequently, this paper employs the DH convention to formulate the kinematic model. The initial kinematic parameters of the ABB IRB 120 industrial robot are provided in Table 2.

The matrix T_i^{i-1} representing the transformation between consecutive joint coordinate systems is given:

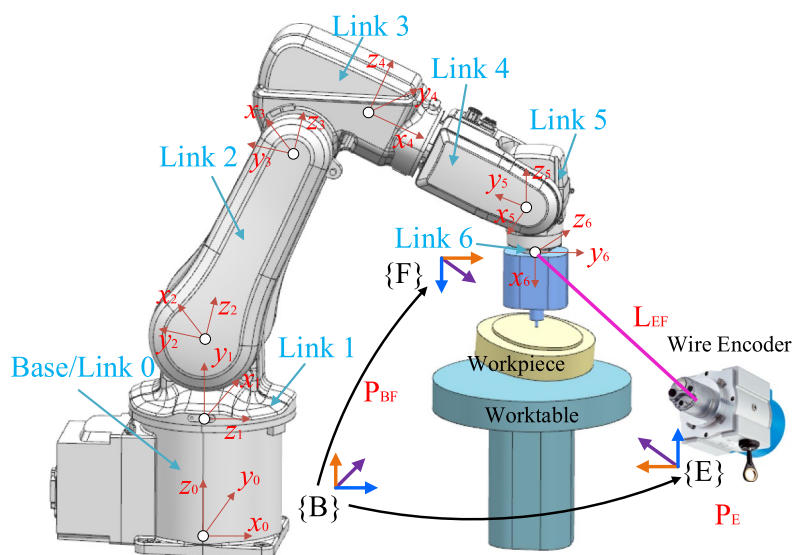


Fig. 1. Schematic of the coordinate systems in robot system. *Note that {B} represents the robot’s base coordinate system, {F} denotes the coordinate system at the center of the end flange, {E} refers to the coordinate system at the end of the cable encoder.

| Joint i | Link length a_i/mm | Link offset distance d_i/mm | Link twist angle α_i/deg | Joint angle θ_i/deg |
|-----------|-----------------------------|--------------------------------------|--|-----------------------------------|
| 1 | 0 | 290 | -90 | $q_1 + 0$ |
| 2 | 270 | 0 | 0 | $q_2 - 90$ |
| 3 | 70 | 0 | -90 | $q_3 + 0$ |
| 4 | 0 | 302 | 90 | $q_4 + 0$ |
| 5 | 0 | 0 | -90 | $q_5 + 0$ |
| 6 | 0 | 72 | 0 | $q_6 + 0$ |

Table 2. Nominal kinematic parameters of ABB IRB 120.

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

For a six-joint robot, the overall transformation matrix defining the end-effector’s position with respect to the base frame is derived:

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \tag{3}$$

The relationship between the end-effector’s position and the joint variables is given by:

$$T_6^0 = \prod_{i=1}^6 T_i^{i-1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $\mathbf{n} = (n_x, n_y, n_z)^T$, $\mathbf{o} = (o_x, o_y, o_z)^T$, and $\mathbf{a} = (a_x, a_y, a_z)^T$ are rotational column vectors, $P = (p_x, p_y, p_z)^T$ is a translational column vector, that is, the end position.

Thus, the expression for the robot’s end position P_{BF} is obtained as:

$$P_{BF} = P = T_6^0((1 : 3), 4) \tag{5}$$

where T_6^0 is the homogeneous transformation matrix from frame 6 to frame 0, and $T0\ 6((1:3),4)$ denotes the sub-vector consisting of the first three elements of the fourth column, which corresponds to the translational position $(p_x, p_y, p_z)^T$ of the end-effector.

In deriving the kinematic parameter error vector X , it is assumed that the deviation of the robot’s positioning error is small enough such that a first-order approximation is valid. Thus, higher-order terms are neglected. Additionally, it is assumed that the transformation matrix T_{i-1}^i varies smoothly with respect to the D-H parameters. Under these assumptions, the differential change in the robot end-effector’s position can be expressed linearly by the Jacobian J and the parameter error vector X , as shown in Eq. (6):

$$\delta P_{BF} = \sum_{i=1}^6 \frac{\partial P_{BF}}{\partial a_i} \delta a_i + \sum_{i=1}^6 \frac{\partial P_{BF}}{\partial d_i} \delta d_i + \sum_{i=1}^6 \frac{\partial P_{BF}}{\partial \alpha_i} \delta \alpha_i + \sum_{i=1}^6 \frac{\partial P_{BF}}{\partial \theta_i} \delta \theta_i = JX \tag{6}$$

where δP_{BF} is the robot positioning error, J stands for Jacobian matrix. X denotes the kinematic parameter error, which can be represented as:

$$X = (\delta a_1, \dots, \delta a_6, \delta d_1, \dots, \delta d_6, \delta \alpha_1, \dots, \delta \alpha_6, \delta \theta_1, \dots, \delta \theta_6)^T \tag{7}$$

The differential model expressed in Eq. (6) is based on a first-order Taylor expansion of the forward kinematic function with respect to the D-H parameters. This formulation assumes that the deviations in the geometric parameters are sufficiently small, such that the nonlinear terms beyond the first order can be neglected. This local linearization is a common practice in robot calibration and is typically valid when the kinematic parameter perturbations remain within a limited range. However, it is acknowledged that in scenarios involving large structural deviations or joint flexibility, the linear model may not fully capture the system’s nonlinear characteristics. In such cases, the linearized model should be interpreted as an approximation that facilitates efficient computation and parameter sensitivity analysis, rather than an exact representation of the full nonlinear behavior.

Based on the principles of robot calibration, the following loss function can be formulated to optimize the D-H parameters.

$$\begin{aligned}
 f(X) &= \min \left[\frac{1}{n} \sum_{i=1}^n (\Delta L_i(D + X))^2 \right] \\
 &= \min \left[\frac{1}{n} \sum_{i=1}^n (\|P_{BF}(D + X) - P_E\|_2 - L_{EF})^2 \right]
 \end{aligned} \tag{8}$$

where D represents the nominal geometric parameters.

Identification of robot kinematic parameters

The least squares objective function is formulated based on the robot kinematic error model to derive D-H parameters that better approximate the actual values. The AMLM algorithm is employed to obtain near-optimal estimates of kinematic parameter deviations, leveraging its rapid convergence properties to swiftly approach the true D-H parameter deviations. These near-optimal estimates serve as central values for generating a candidate solution set in the BIPSO algorithm. By utilizing its global search capability, the BIPSO algorithm enhances accuracy in estimating kinematic parameter deviations. Consequently, the AMLM-BIPSO approach effectively achieves higher-precision identification of these deviations.

AMLM-based initial calibration

The traditional LM algorithm seeks a balance between gradient descent and the Newton method to solve nonlinear least squares problems, making it a common approach for identifying robot kinematic parameters. Its core mechanism lies in adjusting the damping factor (μ), ensuring that update steps exhibit both local quadratic convergence and global stability. However, in practical applications of robot geometric parameter calibration, two main issues arise. First, when the curvature of the objective function changes sharply or when local flat regions exist, single-step updates may easily lead to convergence to suboptimal local solutions or oscillatory behavior, affecting the accuracy of kinematic parameter identification. Second, the LM algorithm typically relies on local quadratic approximations, which may lose effectiveness in highly nonlinear kinematic calibration problems, particularly when the Jacobian matrix has a high condition number, leading to poor numerical stability. To address these challenges, this paper introduces momentum mechanisms and uncertainty estimation within the LM framework for robot kinematic parameter identification, enabling adaptive adjustments to the update direction, improving convergence stability, and ensuring higher accuracy in D-H parameter optimization.

In the traditional LM algorithm, the update step is:

$$\Delta x_k = -(J(x_k)^T J(x_k) + \mu I)^{-1} J(x_k)^T f(x_k) \tag{9}$$

where x_k represents the D-H parameters error vector. I denote the unit matrix. $f(x_k)$ is the position error vector. μ stands for the damping factor.

After introducing momentum, the corrected update step is defined as:

$$\Delta x'_k = \gamma_k \Delta x_{k-1} + (1 - \gamma_k) \Delta x_k \tag{10}$$

where $\gamma \in [0,1)$, is the momentum coefficient at step k , and its value is adaptively adjusted based on the current local curvature and the descent performance of the objective function.

Adjustment strategy for the momentum coefficient. When the current step exhibits good descent performance (e.g., the actual decrease is significantly higher than the expected decrease), γ can be appropriately increased to leverage the advantage of the previous direction, thereby accelerating convergence. Conversely, when the descent performance is poor, γ should be reduced, making the update rely more on the gradient information computed in the current iteration. This helps prevent the algorithm from drifting in the wrong direction.

Update γ_k using the following strategy:

$$\gamma_k = \gamma_{\max} \exp(-\alpha |\rho_k - 1|) \tag{11}$$

where γ_{\max} is the predefined maximum value, such as 0.95. α is a proportional scaling factor that controls the decay sensitivity of the momentum coefficient γ_k . Its value is constrained to the open interval (0, 1) to ensure meaningful exponential scaling. The optimal value of α is problem-dependent. In this study, based on repeated calibration experiments using real-world robot kinematic data, α was empirically set to 0.6. A sensitivity analysis showed that the algorithm is robust within the range $0.4 \leq \alpha \leq 0.8$, with $\alpha = 0.6$ providing a consistent trade-off between convergence speed and stability across various robot types and calibration scenarios. ρ_k is the error reduction ratio, determined by the following equation:

$$\rho_k = \frac{f(x_{k-1}) - f(x_k)}{f(x_{k-1})} \tag{12}$$

If $\rho_k \approx 1$, it indicates a significant error reduction, and a relatively large momentum is maintained. If $\rho_k \approx 0$, it suggests a slow error reduction, and the momentum is decreased to prevent oscillations.

The adaptive momentum strategy described in Eq. (11) is designed to improve the stability and convergence rate of the optimization process. The function is based on an exponential decay of the absolute deviation of the error reduction ratio ρ_k from 1, ensuring that γ_k increases when the update direction is effective ($\rho_k \approx 1$) and

decreases when the descent performance deteriorates ($\rho_k \rightarrow 0$). This formulation draws conceptual inspiration from adaptive momentum methods used in deep learning, such as Adam¹⁶, RMSProp¹⁷, and Nesterov-accelerated gradient (NAG)¹⁸, which scale the learning rate or momentum based on gradient statistics. However, the proposed method differs by linking the adjustment directly to the optimization outcome (ρ_k) rather than gradient magnitude or variance, making it more interpretable and applicable to classical numerical optimization. Several function types, including linear, polynomial, and exponential forms, were tested in preliminary calibration experiments. The exponential function exhibited superior robustness and smoother convergence behavior, especially under ill-conditioned Jacobian matrices.

Thus, the position update formula of the AMLM algorithm is as follows:

$$x_{k+1} = x_x + \Delta x'_k \quad (13)$$

Stability analysis of the AMLM update strategy

The AMLM algorithm modifies the standard LM update by introducing a momentum term γ_k , which is adaptively adjusted based on the error reduction ratio ρ_k . For the update step to remain stable, the following stability conditions are considered:

1. $\gamma_k \in [0, \gamma_{max})$ ensures the weight of historical updates does not dominate, avoiding divergence.
2. When $\rho_k \rightarrow 1$, the update step is confidently accelerated in a consistent descent direction. Conversely, when $\rho_k \rightarrow 0$, the update step becomes more conservative, effectively damping oscillations.
3. The damping factor μ dynamically adjusts the step scale, ensuring numerical stability, particularly when the Jacobian is ill-conditioned.

Under these conditions, the iterative updates behave similarly to a discrete-time dynamic system with bounded energy, exhibiting asymptotic stability in the presence of decreasing residuals.

BIPSO-based algorithm for Parameter Identification

The suboptimal solution x_{AMLM} derived from the AMLM algorithm serves as the central reference for generating the initial population in the BIPSO algorithm. This set comprises N individuals, each characterized by dim dimensions. The initial position of each individual is determined as follows:

$$x(i, j) = x_{AMLM}(i, j) + (ub(j) - lb(j)) \cdot (2 \cdot rand - 1), \quad i = 1, 2, \dots, N, j = 1, 2, \dots, dim \quad (14)$$

Here, $x(i, j)$ denotes the position of the i -th individual in the j -th dimension. $ub(j)$ and $lb(j)$ represent the upper and lower deviation limits for the i -th individual along the j -th dimension, respectively. $rand(\cdot)$ generates random values within the interval $[0, 1]$. Specifically, the translation-related errors (e.g., Δa , Δd) are assigned a range of ± 1 mm, and the rotation-related errors (e.g., $\Delta \theta$, $\Delta \alpha$) are bounded by $\pm 1^\circ$. These constraints ensure that the initial population remains within a physically plausible domain, while also providing sufficient diversity to allow global exploration.

Compute the fitness function for each particle:

$$fit_i = f(x(i, :)) \quad (15)$$

where $f(\cdot)$ is the objective function in section “Kinematic and error models of robot”.

Update each particle's personal best:

$$P_k(i, :) = \begin{cases} x_k(i, :), & f(x_k(i, :)) < f(P_k(i, :)) \\ P_k(i, :), & otherwise \end{cases} \quad (16)$$

Update the global best:

$$G_k \leftarrow \arg \min f(P_k) \quad (17)$$

Update velocity and position using the standard PSO equations:

$$\begin{aligned} V_{k+1}(i, :) &= \omega V_k(i, :) + c_1 r_1 (P_k(i, :) - x_k(i, :)) + c_2 r_2 (G_k - x_k(i, :)) \\ x_{k+1}(i, :) &= x_k(i, :) + V_{k+1}(i, :) \end{aligned} \quad (18)$$

where ω represents inertia weight. c_1 -individual learning factor, c_2 -social learning factor. $r_1, r_2 \sim U(0, 1)$ are random values.

Define the interpolation points: Use the updated particle positions \mathbf{X}_{k+1} as the knots (i.e., the curve must pass through these points).

$$Pt_j = \{x_{k+1}(1, j), x_{k+1}(2, j), \dots, x_{k+1}(N, j)\} \quad (19)$$

Compute the Euclidean distance between consecutive points:

$$d_j = \|x(i+1, j) - x(i, j)\| \quad (20)$$

Compute the cumulative chord length:

$$t_i = \sum_{m=1}^i d_{m-1}, t_0 = 0 \tag{21}$$

Normalize the parameters to the range [0,1]:

$$t_i = \frac{t_i - t_0}{t_{N-1} - t_0}, i = 0, 1, \dots, N - 1 \tag{22}$$

where t_{N-1} is the total accumulated chord length.

As the B-spline curve is required to pass through all given points, it is necessary to determine the control points C_m such that:

$$x(i, j) = S_j(t_i), i = 0, 1, \dots, N - 1 \tag{23}$$

where the B-spline curve is represented as:

$$S_j(t_i) = \sum_{b=0}^{N-1} N_{b,o}(t)C_b \tag{24}$$

where C_m are the unknown control points. $N_{d,o}(t)$ are the B-spline basis functions of degree o , defined recursively:

$$N_{m,0}(t) = \begin{cases} 1, & t_m \leq t \leq t_{m+1} \\ 0, & otherwise \end{cases} \tag{25}$$

$$N_{m,o}(t) = \frac{t - t_m}{t_{m+o} - t_m} N_{m,o-1}(t) + \frac{t_{m+o+1} - t}{t_{m+o+1} - t_{m+1}} N_{m+1,o-1}(t)$$

The key equation to solve is:

$$AC = Pt \tag{26}$$

where A is the B-spline basis function matrix evaluated at t_i . C is the vector of control points. P_i is the vector of given data points.

After solving for the control points C_m , M new interpolated points can be computed:

$$X_{interp}(m, j) = \sum_{b=0}^{N-1} N_{b,o}(t_m)C_b, m = 1, 2, \dots, M \tag{27}$$

where t_m is evenly spaced interpolation parameters in the range of the knot vector. M is the number of interpolated points.

This results in M interpolated candidate solutions, enhancing the exploration capability of the PSO algorithm. Compute the fitness function for each interpolated point:

$$fit_interp = f(x_{interp}) \tag{28}$$

Sort the fitness values in ascending order and assign the top N values to x_{k+1} :

$$[\sim, indices] = sort(fit_interp, 'ascend') \tag{29}$$

$$x_{k+1} = x_{interp}(indices(1 : N), :) \tag{30}$$

Assign the minimum fitness value to G_{best} :

$$G_{best} = x_{interp}(indices(1), :) \tag{31}$$

where G_{best} represents the final optimization result of BIPSO, which is the ultimate identified geometric parameter error of the robot.

In general, the AMLM-BIPSO algorithm enhances robot geometric parameter error identification by integrating the advantages of AMLM and BIPSO. AMLM improves local search efficiency by dynamically adjusting update corrections, mitigating overshooting and oscillations, thereby enhancing convergence stability and accuracy. Meanwhile, BIPSO reconstructs the population distribution using B-spline interpolation, ensuring continuity in the search space and enhancing global exploration capabilities. The cascading of AMLM and BIPSO effectively balances local refinement and global optimization, achieving high-precision identification of geometric parameter errors while accelerating convergence and improving overall robustness.

Table 3 outlines the detailed design of the proposed algorithm. Here, K_1 and K_2 represent the maximum iteration rounds for the AMLM and BIPSO algorithms, respectively. N denotes the number of sampling points, while x corresponds to the deviations in kinematic parameters. The complete initialization process and parameter

| AMLM-BIPSO cascaded Algorithm | |
|--|---|
| Input: function f , $\{q_1, q_2, q_3, q_4, q_5, q_6\}$, $\{L_{EF1}, L_{EF2}, \dots, L_{EFn}\}$ | |
| /--Note: Initialization--/ | |
| 1. | Initialize: $x = x_0$ |
| 2. | Initialize: $K_1=30, K_2=30$ |
| /--Note: Training Starts--/ | |
| /--Note: AMLM-Step--/ | |
| 3. | Set $\Delta x_0=0, \mu=5, \alpha=0.6, \gamma_{\max}=0.95$ |
| 4. | for $k=1$ to K_1 |
| 5. | Make J_k with (6) |
| 6. | $\Delta x_k = -(J(x_k)^T J(x_k) + \mu I)^{-1} (J(x_k)^T f(x_k))$ |
| 7. | $\rho_k = (f(x_{k-1}) - f(x_k)) / f(x_{k-1})$ |
| 8. | $\gamma_k = \gamma_{\max} \exp(-\alpha \rho_k - 1)$ |
| 9. | $\Delta x_k = \gamma_k \Delta x_{k-1} + (1 - \gamma_k) \Delta x_k$ |
| 10. | Updating $x_{k+1} = x_k + \Delta x_k$ |
| 11. | end for |
| 12. | $x_{\text{AMLM}} = x_k$ |
| /--Note: BIPSO-Step--/ | |
| 13. | Set $\omega=0.8, c_1=1.2, c_2=1.5, N=100, M=100$ |
| 14. | Initialize Generate initial population based on (14) |
| 15. | Calculate $fit_i = f(x(i, :))$ |
| 16. | Calculate personal best P and global best G |
| 17. | for $k=1$ to K_2 |
| 18. | Update V_{k+1} and x_{k+1} with (18) |
| 19. | Set interpolation points |
| 20. | $P_{t_j} = \{x_{k+1}(1, j), x_{k+1}(2, j), \dots, x_{k+1}(N, j)\}$ |
| 21. | for $m=1$ to M |
| 22. | Calculate new interpolated points |
| 23. | $x_{\text{interp}}(m, j) = \sum_{b=0}^{N-1} N_{b,0}(t_m) C_b$ |
| 24. | end for |
| 25. | Calculate $fit_{\text{interp}} = f(x_{\text{interp}})$ |
| 26. | $[\sim, \text{indices}] = \text{sort}(fit_{\text{interp}}, 'ascend')$ |
| 27. | $x_{k+1} = x_{\text{interp}}(\text{indices}(1:N), :)$ |
| 28. | $G_{\text{best}} = x_{\text{interp}}(\text{indices}(1), :)$ |
| 29. | end for |
| 30. | $x_{\text{best}} = G_{\text{best}}$ |
| /--Note: Training Ends--/ | |
| Output: x_{best} | |

Table 3. The pseudocode of the AMLM-BIPSO algorithm.

settings, including particle count, node count, and learning coefficients, are provided in the pseudocode in Table 3.

Experimental verifications

Experiment conditions

The experimental setup comprises an ABB IRB120 industrial robotic arm, which offers a repeat positioning accuracy of 0.01 mm, along with a drawstring displacement sensor featuring a resolution of 0.004 mm. Additionally, a PC and a drawstring displacement indicator are utilized, as illustrated in Fig. 2. Furthermore, the specific performance parameters of the primary equipment used in the experiment are publicly accessible in references²¹.

To ensure comprehensive coverage of the tested robot arm's workspace, sampling points are strategically selected. A total of 1042 points are manually taught throughout the robot's entire workspace, after which the sampling points are measured using a drawstring displacement sensor. A drawstring displacement indicator is employed to display the cable length while simultaneously recording the end-effector's position coordinates, joint angle, and corresponding cable length, with the data stored on a PC. Upon completing the data acquisition process, a calibration system is applied to the collected dataset to determine the optimal kinematic parameters.

The primary focus of this study is the accuracy of robot calibration, which serves as a direct indicator of the calibrator's effectiveness in precisely adjusting the robot^{19,20}. To assess calibration performance, root mean square error (RMSE), standard deviation (STD), and maximum error (MAX) are utilized as key evaluation criteria:

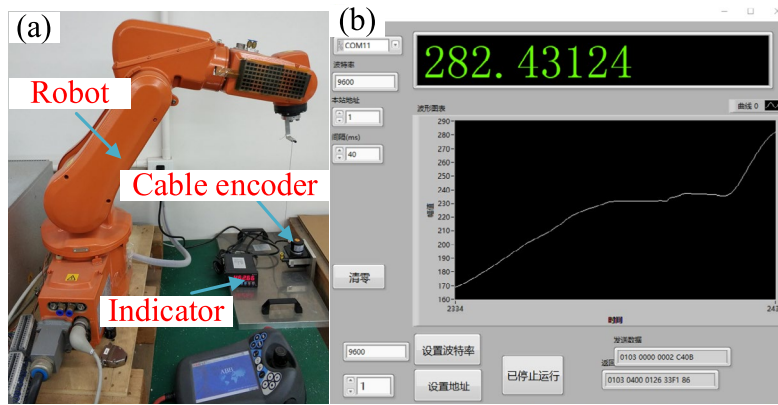


Fig. 2. Experimental system. (a) Experimental hardware; (b) Data collection software interface.

| Models | Description |
|--------|---|
| M1 | The EKF algorithm, as described in ⁷ , is capable of handling non-Gaussian noise in the calibration system |
| M2 | The LM algorithm introduced in ^{5,6} is utilized for identifying deviations in kinematic parameters |
| M3 | The PSO algorithm ¹³ , inspired by the foraging patterns of birds, is employed to obtain an optimal solution |
| M4 | An efficient robot calibration method based on unscented Kalman filter and variable step-size Levenberg–Marquardt algorithm (UKF-VSLM) ⁶ |
| M5 | A Robot Calibration Method Based on Extended Kalman Filter and Improved Covariance Matrix Adaptive Evolution Strategy (EKF-ICMA-ES) ²¹ |
| M6 | A calibration method for industrial robot based on Levenberg–Marquardt and beetle antennae search algorithm (LM-BASA) ²² |
| M7 | The proposed AMLM method enhances the LM algorithm by utilizing momentum-based corrections, integrating historical updates |
| M8 | The BIPSO method uses B-spline interpolation to enhance search efficiency by reducing reliance on discrete positions |
| M9 | The proposed AMLM-BIPSO algorithm enhances robot calibration, significantly boosting its accuracy |

Table 4. Comparative study of algorithms.

$$\begin{aligned}
 MAX &= \max_{1 \leq i \leq n} \left\{ \sqrt{(L_i - L_i^*)^2} \right\} \\
 STD &= \frac{1}{n} \sum_{i=1}^n \sqrt{(L_i - L_i^*)^2} \\
 RMSE &= \sqrt{\frac{1}{n} \sum_{i=1}^n (L_i - L_i^*)^2}
 \end{aligned}
 \tag{32}$$

To demonstrate the advantages of the proposed algorithm, a comparative analysis is conducted against several cutting-edge algorithms, as presented in Table 4. For models M1 to M9, each dataset consists of six joint angles and the measured distance from the robot end-effector to a fixed target point. From each dataset, 120 samples are randomly selected under a uniform distribution to create a test case, and this process is repeated ten times to generate ten distinct cases, minimizing potential data bias. Each test case follows an 80%-20% split for training and validation to assess model performance. The results, including mean values and standard deviations for each model, are systematically recorded to ensure objective reporting. For each model in every test case, training terminates when the objective function’s reduction between consecutive iterations falls below 0.001 or when the iteration count surpasses 60. For the cascade method, the former iteration runs 30 times, followed by the latter iteration running 30 times.

All experimental results reported in Section “Experimental verifications” are obtained using the exact algorithmic parameters detailed in Sections “AMLM-based initial calibration” and “Stability analysis of the AMLM update strategy”. As shown in the pseudocode of Table 3, the AMLM module employs an adaptive decay parameter $\alpha = 0.6$ and maximum momentum coefficient $\gamma_{\max} = 0.95$. The stopping condition is defined by a convergence threshold of 30 iterations per stage. For BIPSO, the number of particles is set to 100, and the B-spline interpolation involves 100 control nodes. The initial population is centered around the AMLM output (as per Eq. 14), with bounds of ± 1 mm for translational parameters and $\pm 1^\circ$ for angular parameters. These settings, explicitly stated in Table 3, are consistently applied throughout all test cases, ensuring transparency, reproducibility, and a fair basis for performance comparison.

In practical robot calibration, actuator limitations—such as joint motion range, torque saturation, and mechanical tolerances—play an important role in determining the feasibility of optimization results. In implementation, the optimization process explicitly incorporated parameter bound constraints based on

| Joint i | a_i/mm | d_i/mm | α_i/deg | θ_i/deg |
|-----------|-----------------|-----------------|-----------------------|-----------------------|
| 1 | -2.951 | 295.699 | -90.757 | 2.016 |
| 2 | 267.839 | 1.178 | 0.217 | -87.023 |
| 3 | 71.236 | 0.118 | -89.581 | 0.117 |
| 4 | -2.754 | 306.181 | 90.179 | -0.173 |
| 5 | -0.837 | -0.312 | -90.148 | 0.044 |
| 6 | 0.152 | 73.615 | 0.154 | -2.138 |

Table 5. Kinematic parameters of the robot after AMLM-BIPSO-based calibration.

| Item | RMSE (mm) | Std (mm) | Max (mm) | Time (s) |
|-----------|--------------|--------------|--------------|----------------|
| Before | 2.177 | 2.006 | 3.99 | - |
| M1 | 0.604 | 0.483 | 1.676 | 89.635 |
| M2 | 0.452 | 0.372 | 1.032 | 87.512 |
| M3 | 0.581 | 0.459 | 1.647 | 140.216 |
| M4 | 0.412 | 0.337 | 0.988 | 92.147 |
| M5 | 0.43 | 0.352 | 1.017 | 120.643 |
| M6 | 0.401 | 0.328 | 0.965 | 106.215 |
| M7 | 0.372 | 0.306 | 0.911 | 92.156 |
| M8 | 0.489 | 0.395 | 1.236 | 160.258 |
| M9 | 0.332 | 0.272 | 0.840 | 110.218 |

Table 6. Calibration performance and total execution time of M1–M9. Values in bold indicate the calibration results of the proposed cascaded AMLM-BIPSO method.

physical actuator capabilities and manufacturer specifications. Specifically, the geometric parameter deviations were constrained within ± 5 mm for translational components and $\pm 8^\circ$ for rotational parameters, ensuring the identified D-H deviations remain within realistic and executable bounds. These constraints not only preserve mechanical safety and avoid joint singularities, but also reflect real-world tolerances due to assembly and machining errors. By embedding these bounds into the search space of those algorithms, so the calibration remains applicable to real industrial settings. Future work will explore the integration of full actuator models, such as joint velocity and acceleration constraints, to further improve robustness in dynamic or online calibration tasks.

The proposed AMLM-BIPSO algorithm (M9) is constructed by cascading two enhanced components: an AMLM algorithm and a BIPSO. To evaluate the effectiveness of this two-stage framework, the following comparative setup is designed: Single-algorithm baselines: M1 (EKF), M2 (LM), and M3 (PSO) represent classical filter-, gradient-, and swarm-based methods, respectively. Single enhancements: M7 and M8 are the respective improvements of M2 and M3. M7 is the AMLM method, while M8 is the BIPSO method. Cascade-based benchmarks: M4, M5, and M6 are recent hybrid methods combining filtering and optimization techniques, representing typical multi-stage calibration frameworks. Proposed method: M9 integrates AMLM and BIPSO into a unified cascade, enabling joint refinement of local precision and global search. The comparisons between M9 and M1–M3 highlight the advantage of the proposed enhancements; comparisons with M4–M6 assess the performance relative to other cascade frameworks; and comparisons with M7 and M8 evaluate the added value of cascading over individual enhancements.

Performance assessment alongside advanced contemporary methods

The experiment evaluates the performance of the proposed method in comparison with state-of-the-art approaches. Table 5 presents the kinematic parameters calibrated using AMLM-BIPSO, while Table 6 summarizes the calibration results and computation times for methods M1–M9. Table 7 presents the Wilcoxon signed-ranks test on iteration. Figures 3 and 5 illustrate the accuracy of the compared methods post-calibration. Specifically, Fig. 3 examines calibration accuracy and total computation time across different approaches, Fig. 4 visualizes the training progress of various calibration methods, and Fig. 5 depicts the residual position errors after applying different calibration techniques. The key findings derived from the experimental results are summarized as follows:

- The AMLM-BIPSO method, denoted as M9, demonstrates a significant improvement in calibration accuracy. After calibration, the RMSE, Std, and Max of M9 are reduced to 0.332, 0.272, and 0.840, respectively, compared to their pre-calibration values of 2.177, 2.006, and 3.99. This corresponds to accuracy improvements of 84.75%, 86.44%, and 78.95%, respectively. These substantial enhancements indicate the high effectiveness of the AMLM-BIPSO method in error suppression and accuracy optimization, providing a robust technical foundation for the geometric parameter calibration of robotic systems.

| Comparison | R+ | R- | <i>p</i> value |
|--------------|----|----|----------------|
| M9 versus M1 | 55 | 0 | 0.002 |
| M9 versus M2 | 55 | 0 | 0.002 |
| M9 versus M3 | 55 | 0 | 0.002 |
| M9 versus M4 | 55 | 0 | 0.002 |
| M9 versus M5 | 55 | 0 | 0.002 |
| M9 versus M6 | 55 | 0 | 0.002 |
| M9 versus M7 | 51 | 4 | 0.0273 |
| M9 versus M8 | 55 | 0 | 0.002 |

Table 7. The Wilcoxon signed-rank test results.

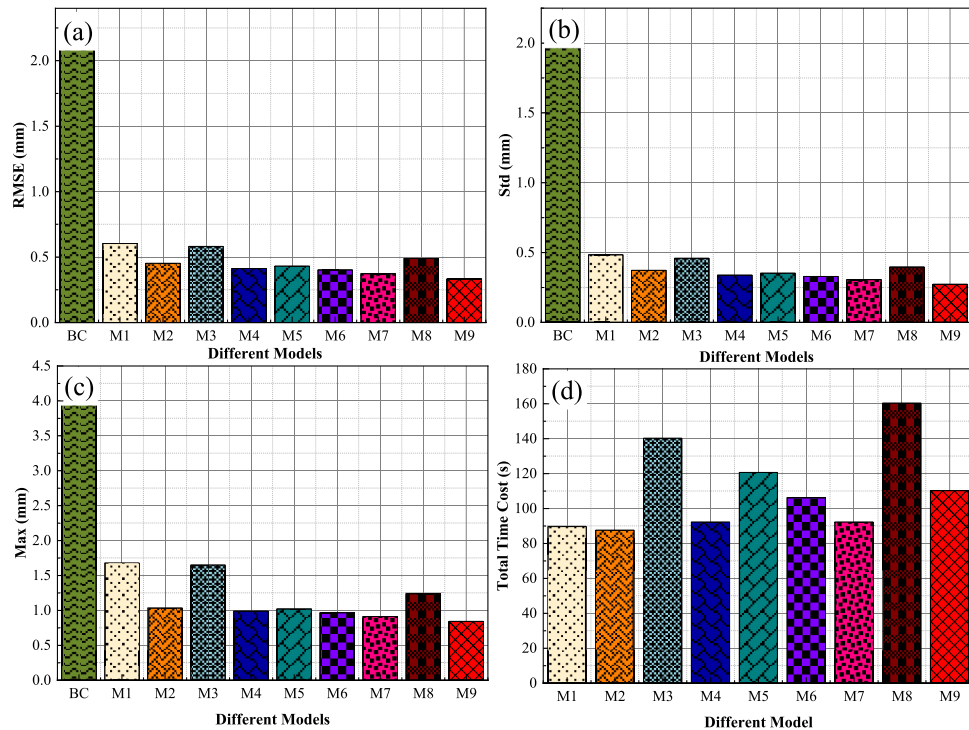


Fig. 3. Evaluation of calibration accuracy and total time cost across different approaches. (a) RMSE; (b) Std; (c) Max; (d) Time.

- (b) Among the evaluated methods (M1–M8), M9 demonstrates the highest accuracy in robot calibration. As presented in Table 6 and Fig. 3, M9 achieves an RMSE of 0.332, a Std of 0.272, and a Max of 0.840. Compared to the most accurate alternative, M7, which has an RMSE of 0.372, a Std of 0.306, and a Max of 0.911, M9 improves accuracy by 10.75%, 11.11%, and 7.79%, respectively. These findings underscore the effectiveness of the proposed method in improving robot calibration accuracy. Statistical significance analysis. To further validate the observed performance improvements, The Wilcoxon signed-rank test was conducted between the proposed AMLM-BIPSO method (M9) and each of the baseline methods (M1–M8), based on the RMSE results across ten test cases. As shown in Table 7, all comparisons yielded *p*-values less than 0.05, indicating that the improvements achieved by M9 are statistically significant. These results confirm the robustness of the proposed method in enhancing calibration accuracy.
- (c) As shown in Table 6 and Fig. 3, M9, the proposed AMLM-BIPSO cascade method, demonstrates better computational efficiency than M3 (140.216), M5 (120.643), and M8 (160.258), while being slightly less efficient than M1 (89.635), M2 (87.512), M4 (92.147), M6 (106.215), and M7 (92.156). The lower time cost of M1 and M2 suggests that single methods can be computationally lightweight, whereas the significantly higher costs of M3 and M8 indicate that some single methods impose a substantial computational burden. Among cascade methods, M9 achieves comparable efficiency to M4 and M6 but performs better than M5. These results suggest that although the proposed M9 method incurs a moderate increase in computation time due to its cascaded structure, this increase is significantly lower than that of other single methods such as M3 and M8. More importantly, the time cost of M9 remains comparable to that of other cascaded methods (M4–M6), while delivering superior accuracy and convergence speed. This indicates that the trade-off

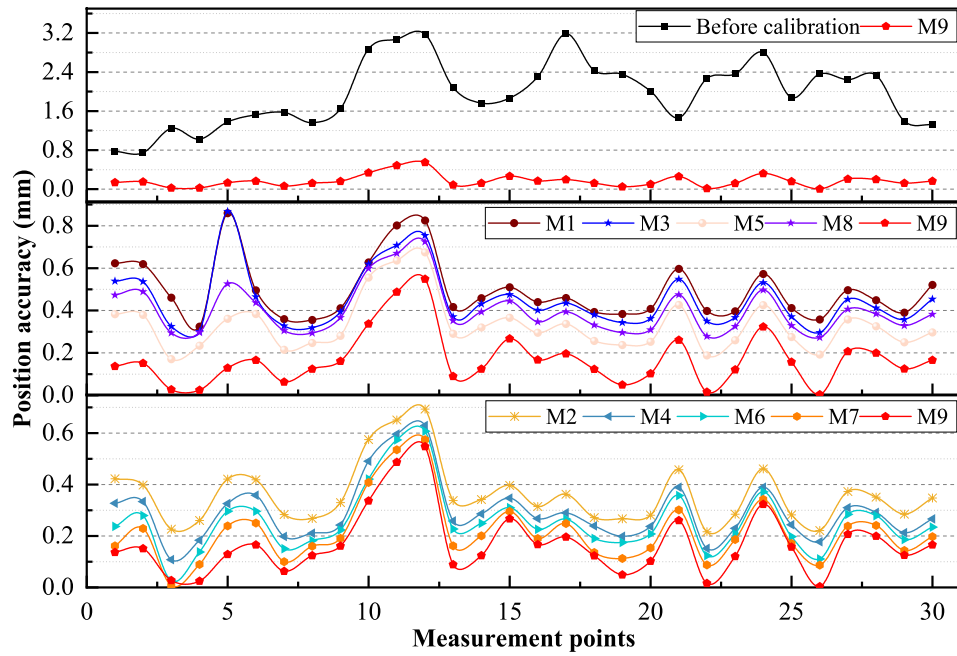


Fig. 5. Residual position errors following different calibration approaches.

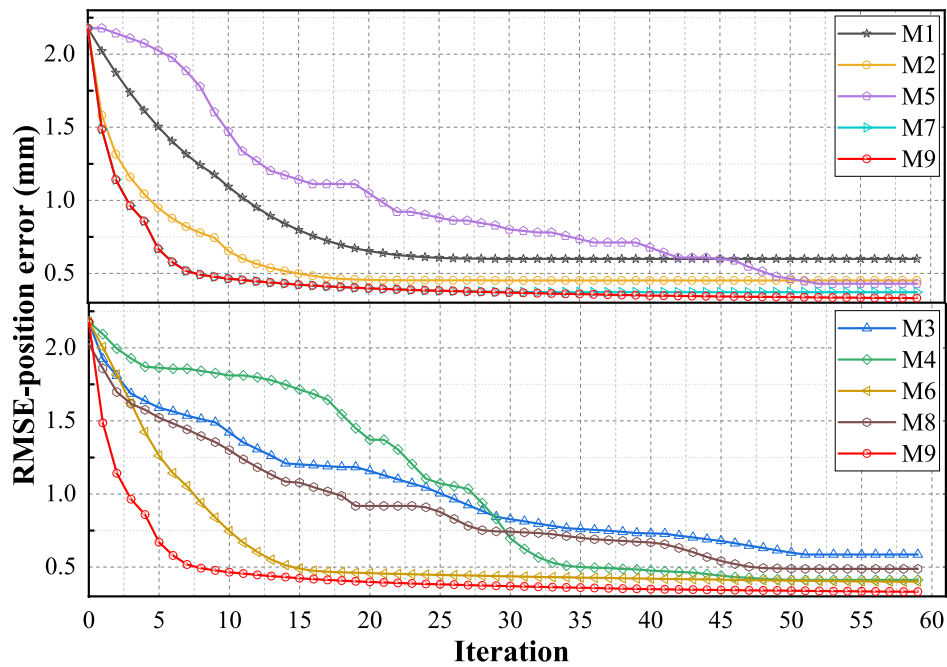


Fig. 4. Training progress of various calibration approaches.

between computational efficiency and calibration precision is well-balanced in the proposed AMLM-BIP-SO algorithm.

- (d) M9 exhibits a significantly higher convergence efficiency compared to other calibration approaches. As shown in Fig. 4, the RMSE-position error of M9 decreases rapidly within the first few iterations and stabilizes at a lower error level than the other methods. In contrast, single methods such as M1, M2, M3, M7, and M8 exhibit slower convergence rates and higher final errors, indicating their limited calibration effectiveness. Among the cascade methods (M4, M5, and M6), M9 still achieves faster convergence and lower final RMSE, demonstrating its superior optimization capability. These results highlight the effectiveness of M9 in improving calibration accuracy while maintaining high convergence efficiency. Figure 4 illustrates the training progress of various calibration methods in terms of RMSE over iterations. From a dynamic

system perspective, this curve reflects the transient behavior of the optimization process. For the proposed M9, the RMSE decreases rapidly in the initial iterations and stabilizes after approximately 15 iterations, indicating a settling time of around 15 steps. Additionally, the curve exhibits a low overshoot, as the RMSE consistently decreases without significant fluctuations above the final steady-state error. In contrast, some baseline methods (e.g., M2 and M8) display longer settling times and noticeable oscillations, suggesting less stable convergence behavior. This analysis confirms that the proposed method achieves fast and stable convergence, which is critical for efficient and reliable robot calibration.

- (e) To ensure accurate robot calibration, M9 is employed for kinematic parameter identification, with the results presented in Table 5. Additionally, 120 measurement points are collected from an ABB IRB120 industrial robot. After completing the calibration process, a comparative analysis of the calibration outcomes for different methods is conducted, as illustrated in Fig. 5. The results indicate a significant reduction in robot positioning error post-calibration. Among all methods (M1–M8), M9 achieves the highest calibration accuracy, demonstrating the effectiveness of the proposed approach.
- (f) As shown in Table 6 and Figs. 3, 4, and 5, M7 (AMLM), an improved version of M2 (LM), demonstrates notable enhancements in both calibration accuracy and convergence efficiency through the introduction of adaptive momentum. By incorporating this mechanism, M7 effectively accelerates convergence while reducing calibration errors compared to M2. The results indicate that M7 achieves a lower RMSE and faster error reduction, particularly in the early iterations, highlighting its improved optimization capability. These findings confirm that the integration of adaptive momentum into the LM algorithm enhances its performance, making M7 a more effective calibration approach.
- (g) As shown in Table 6 and Figs. 3, 4, and 5, M8 (BIPSO) is an improved version of M3 (PSO), incorporating B-spline interpolation, inspired by the KAN Neural Network, to enhance population distribution reconstruction. This modification improves the search capability by ensuring continuity in the fitness landscape, thereby reducing dependence on discrete particle positions. As a result, M8 achieves higher calibration accuracy and a faster convergence rate compared to M3. However, the introduction of interpolation necessitates additional particle computations, leading to an increased time cost. Despite this trade-off, the enhanced optimization performance of M8 demonstrates the effectiveness of integrating B-spline interpolation into the PSO framework for improved robot calibration.
- (h) M9, a cascade of M7 (AMLM) and M8 (BIPSO), achieves superior calibration accuracy and faster convergence than either method alone. As shown in Table 6 and Figs. 3, 4, and 5, M9 leverages the adaptive momentum of M7 and the B-spline interpolation of M8, further enhancing both accuracy and convergence efficiency. However, the increased computational complexity results in a higher time cost. Despite this, the improved calibration precision and efficiency demonstrate the effectiveness of M9 in high-precision robot calibration.

Discussion on potential error sources

Although the proposed AMLM-BIPSO algorithm demonstrates a high correlation between experimental calibration results and theoretical predictions, several potential sources of error may still affect the final accuracy. First, the mechanical wear and thermal deformation of the robot joints may cause time-dependent variations in the kinematic structure, which are not fully captured during static calibration. Second, sensor noise and resolution limits—particularly from the drawstring displacement sensor—introduce measurement uncertainty that propagates through the identification process. Third, manual teaching errors in the data collection phase can introduce initial pose deviations, potentially affecting the reliability of the training dataset. Lastly, modeling assumptions such as small-angle approximations and neglecting dynamic effects (e.g., inertial, or frictional forces) may also contribute to residual discrepancies. Despite these sources of error, the proposed method exhibits strong robustness and calibration precision, as evidenced by significant improvements in RMSE, STD, and MAX error metrics. Future work may consider incorporating dynamic modeling or real-time feedback to further mitigate these uncertainties.

Conclusion

This paper proposes an AMLM-BIPSO algorithm for robot calibration. Standard calibration experiments validate its effectiveness. The key conclusions are summarized as follows:

1. AMLM improves calibration accuracy and convergence speed. By introducing adaptive momentum into the traditional LM algorithm, AMLM considers historical update steps to dynamically adjust the optimization direction. This enhancement effectively suppresses overshooting, accelerates convergence, and improves the accuracy of robotic parameter identification.
2. BIPSO enhances search capability and calibration performance. By incorporating B-spline interpolation into the standard PSO algorithm, BIPSO reconstructs the population distribution, ensuring continuity in the fitness landscape and reducing reliance on discrete particle positions. This modification significantly increases calibration accuracy and convergence efficiency.
3. The AMLM-BIPSO cascade further improves calibration precision and convergence. By combining AMLM's fast convergence with BIPSO's enhanced global search capability, the proposed cascade method achieves superior calibration performance. Compared to state-of-the-art methods, AMLM-BIPSO improves calibration accuracy by 10.75%, 11.11%, and 7.79% in RMSE, Std, and Max error reduction, respectively.

In conclusion, AMLM-BIPSO effectively enhances robot calibration accuracy and efficiency, making it a promising solution for high-precision robotic applications. Although the proposed AMLM-BIPSO algorithm is experimentally validated on an ABB IRB120 6-DOF serial industrial robot, its structure is not specific to

this platform. The method relies solely on the forward kinematic model and D-H parameterization, which are commonly used in most serial manipulators. Therefore, the proposed approach can be generalized to other industrial robots with similar serial kinematic chains, provided that: (1) the robot follows the D-H convention; (2) geometric errors are the dominant contributors to positioning inaccuracy; and (3) enough end-effector measurements can be obtained for parameter identification. In systems with different configurations (e.g., SCARA, parallel robots), the modeling framework may need to be adapted, but the two-stage optimization structure remains applicable.

Data availability

The data that support the findings of this study are openly available in RobotCali: <https://github.com/Lizhibing1490183152/RobotCali>.

Received: 19 March 2025; Accepted: 2 June 2025

Published online: 01 July 2025

References

- Dehghani, M., McKenzie, R. A., Irani, R. A. & Ahmadi, M. Robot-mounted sensing and local calibration for high-accuracy manufacturing. *Robot. Comput.-Integrated Manufact.* **79**, 102429 (2023).
- Liu, J. et al. A logistic-tent chaotic mapping Levenberg Marquardt algorithm for improving positioning accuracy of grinding robot. *Sci. Rep.* **14**(1), 9649 (2024).
- Mao, C., Zheng, Y., Xing, Z. & Li, S. Joint deformation compensation algorithm for robust kinematic calibration of serial robots. *IEEE Trans. Instrum. Meas.* **73**, 1–11 (2024).
- Wang, X. et al. A self-calibration method for robot end-effector using spherical constraints. *Appl. Sci.* **14**(21), 9824 (2024).
- Deng, Y., Hou, X., Li, B., Wang, J. & Zhang, Y. A novel method for improving optical component smoothing quality in robotic smoothing systems by compensating path errors. *Opt. Express* **31**(19), 30359–30378 (2023).
- Li, Z., Li, S. & Luo, X. Efficient industrial robot calibration via a novel unscented Kalman filter-incorporated variable step-size Levenberg–Marquardt algorithm. *IEEE Trans. Instrum. Meas.* **72**, 1–12 (2023).
- Liu, Y. & Li, Y. Positioning accuracy improvement for target point tracking of robots based on Extended Kalman Filter with an optical tracking system. *Robot. Auton. Syst.* **179**, 104751 (2024).
- Deng, Y., Hou, X., Li, B., Wang, J. & Zhang, Y. A highly powerful calibration method for robotic smoothing system calibration via using adaptive residual extended Kalman filter. *Robot. Comput.-Integr. Manufact.* **86**, 102660 (2024).
- Du, G., Liang, Y., Li, C., Liu, P. X. & Li, D. Online robot kinematic calibration using hybrid filter with multiple sensors. *IEEE Trans. Instrum. Meas.* **69**(9), 7092–7107 (2020).
- Gao, G., Guo, X., Li, G., Li, Y. & Zhou, H. Kinematic parameter identification and error compensation of industrial robots based on unscented Kalman filter with adaptive process noise covariance. *Machines* **12**(6), 406 (2024).
- Li, Z., Li, S. & Luo, X. Using quadratic interpolated beetle antennae search to enhance robot arm calibration accuracy. *IEEE Robot. Autom. Lett.* **7**(4), 12046–12053 (2022).
- Chen, X. & Zhan, Q. The kinematic calibration of an industrial robot with an improved beetle swarm optimization algorithm. *IEEE Robot. Autom. Lett.* **7**(2), 4694–4701 (2022).
- Cao, H. Q., Nguyen, H. X., Nguyen, T. T., Nguyen, V. Q. & Jeon, J. W. Robot calibration method based on extended Kalman filter–dual quantum behaved particle swarm optimization and adaptive neuro-fuzzy inference system. *IEEE Access* **9**, 132558–132568 (2021).
- Liu, K., Xia, J., Zhong, F. & Zhang, L. Structural parameters identification for industrial robot using a hybrid algorithm. *Int. J. Adv. Rob. Syst.* **19**(2), 1–15 (2022).
- Ma, S., Lu, Y., Deng, K., Guan, Q. & Xu, X. Optimal measurement poses using LSSA for robot kinematics-flexibility calibration. *IEEE Robot. Autom. Lett.* **9**(6), 4974–4981 (2024).
- Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- Xu, D., Zhang, S., Zhang, H. & Mandic, D. P. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. *Neural Netw.* **139**, 17–23 (2021).
- Gu, P., Tian, S. & Chen, Y. Iterative learning control based on Nesterov accelerated gradient method. *IEEE Access* **7**, 115836–115842 (2019).
- Deng, Y., Hou, X., Li, B., Wang, J. & Zhang, Y. A novel positioning accuracy improvement method for polishing robot based on Levenberg–Marquardt and opposition-based learning squirrel search algorithm. *J. Intell. Rob. Syst.* **110**(1), 8 (2024).
- Li, S., Zhang, Y. & Jin, L. Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(10), 2243–2254 (2016).
- Li, Z., Li, S., Francis, A. & Luo, X. A novel calibration system for robot arm via an open dataset and a learning perspective. *IEEE Trans. Circuits Syst. II Express Briefs* **69**(12), 5169–5173 (2022).
- Fan, M., Zhao, H., Wen, J., Yu, L. & Xia, H. A novel calibration method for kinematic parameter errors of industrial robot based on Levenberg–Marquardt and Beetle Antennae Search algorithm. *Meas. Sci. Technol.* **34**(10), 105024 (2023).

Acknowledgements

This study was supported by the Key R&D Program of Liaocheng City (No.2023YD85), Dazhou Key Laboratory of Government Data Security (No. ZSAQ202422 and No. ZSAQ202401), the National Funded Postdoctoral Research Program (No. GZC20241900), Natural Science Foundation Program of Xinjiang Uygur Autonomous Region (No. 2024D01A141), Tianchi Talents Program of Xinjiang Uygur Autonomous Region (Li Zhibin) and Postdoctoral Fund of Xinjiang Uygur Autonomous Region (Li Zhibin).

Author contributions

Conceptualization, D.L. and Y.D.; methodology, D.L.; software, Y.D.; validation, Z.L.; formal analysis, D.L.; investigation, Z.L.; resources, Z.L.; data curation, Z.L.; writing—original draft preparation, D.L.; writing—review and editing, Y.D.; visualization, Y.D.; supervision, Y.D.; project administration, Z.L.; funding acquisition, D.L., Y.D. and Z.L. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.L. or Y.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025