EAI Endorsed Transactions
on AI and Robotics
Research Article EAI.EU

Robust Robotic Arm Calibration via Multi-Distance Optimization Approach and Lagrange Starfish Algorithm

Yongtao Qu[1,4], Zhiqiang Li[1,*], Long Liao[1], Xun Deng[5], Yuanchang Lin[4], Tinghui Chen[4,*], Linlin Chen[2], Jia Liu[2], Peiyang Wei[2], Jianhong Gan[2,6], ZhenZhen Hu[2,6], Can Hu[2], Yonghong Deng[2,6], Wei Li[5,6], Zhibin Li[2,3,6]

[1] Dongfang Electric Corporation Dongfang Turbine Co., Ltd., Deyang 618000, China
[2] School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China
[3] Xinjiang Technical Institute of Physics &Chemistry, Chinese Academy of Sciences, Urumqi 830011, China
[4] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Science, Chongqing 400714, China
[5] School of Urban Rail Transit, Sichuan Railway Vocational College, Chengdu 610097, China
[6] Dazhou Key Laboratory of Government Data Security, Sichuan University of Arts and Science, Dazhou, Sichuan 635000, China

Abstract

In response to the limitations of existing robotic parameter calibration methods in terms of computational complexity, convergence speed, data requirements, and accuracy, this study proposes an innovative calibration scheme that combines an improved Lagrangian Starfish Optimization Algorithm (LSFA) with a Support Vector Machine (SVM) algorithm. By incorporating Lagrange interpolation and a multi-dimensional distance metric model (including Mahalanobis distance, Manhattan distance, Chebyshev distance, cosine distance, standardized Euclidean distance, and Euclidean distance), the enhanced starfish optimization algorithm significantly improves global search capabilities and local search accuracy. This effectively addresses issues such as initial value sensitivity, noise, and outliers, with the algorithm specifically designed for kinematic parameter calibration of robotic arms. Furthermore, the improved local search mechanism optimizes the position update strategy of starfish through a weighted system, preventing the algorithm from becoming trapped in local optima. To further enhance the accuracy of dynamic parameter calibration, this study integrates the SVM algorithm into the LSFA framework, proposing the LSFA-SVM method specifically for dynamic parameter calibration of robotic arms. Experiments demonstrate a 43.93% reduction in error compared to traditional SVM. The results indicate that LSFA excels in kinematic calibration of robotic arms, achieving a root mean square error (RMSE) of 0.29 mm, a 29.27% improvement over the traditional Starfish Optimization Algorithm (SFOA). This study provides an efficient and precise solution for robotic parameter calibration in complex environments.

Keywords: Lagrangian Starfish Optimization Algorithm (LSFA); Support Vector Machine (SVM); multi-dimensional distance metrics; robotic arm calibration; dynamic parameter estimation; intelligent optimization algorithm

Received on 01 April 2025, accepted on 07 May 2025, published on 26 June 2025

doi: 10.4108/airo.9002

1. Introduction

With the advancement of industrial automation, robotic technology is increasingly utilized in manufacturing, logistics, and material handling. The performance of robotic arms, critical to automated production lines, directly impacts production efficiency and product quality.

*Corresponding author. Email: Etesop0712@outlook.com; Chenth199208@163.com

Ensuring high-precision positioning and motion control in complex tasks is essential for efficient automated production [1,2,3]. Robotic arm precision depends critically on accurate kinematic and dynamic parameters [4,5,6]. Kinematic parameters, such as joint lengths and offsets, define motion

trajectories, while dynamic parameters, including joint mass and inertia, influence movement responses. Accurate calibration of both parameters is therefore essential for optimal robotic arm performance [7,8,9].

To enhance robotic arm positioning precision, various calibration algorithms have been proposed, including least squares [10], Extended Kalman Filter (EKF)[11], Beetle Antennae Search [12], the Levenberg-Marquardt (LM) method [13], genetic algorithms [14], and particle swarm optimization [15]. Recent hybrid approaches integrate LM with differential evolution for kinematic deviation assessment [16], parallel arm calibration for error estimation [17], and hybrid genetic algorithms for over-constrained mechanisms [18]. Advanced techniques leverage neural networks for real-time parameter updates in large-scale tasks [19], genetic algorithms for optimal pose selection to reduce errors [20], and fuzzy-based multi-algorithm systems to achieve high-precision calibration [21], collectively improving flexibility and accuracy in complex robotic applications.

Existing calibration methods have notable limitations: genetic algorithms are computationally intensive with slow convergence, while neural networks require substantial training data, as evidenced in recent robotics studies [22].To address these challenges, this study proposes an innovative calibration scheme combining the Lagrange Starfish Optimization Algorithm (LSFA) based on a multi-dimensional distance metric model and improved local search with the Support Vector Machine (SVM) algorithm [23], building upon foundational work in robotic mathematical modeling [24].

As a novel intelligent optimization algorithm, SFOA includes two core phases: exploration and exploitation. The exploration phase mimics starfish foraging behavior, using a hybrid strategy that combines five-dimensional and one-dimensional search patterns to enhance computational efficiency and global search capabilities. The exploitation phase simulates starfish predatory and regenerative behaviors, employing bidirectional search and specialized motion mechanisms to ensure convergence [25]. Lagrange interpolation optimizes SFOA-processed data, further improving calibration accuracy. The SVM algorithm demonstrates exceptional capability in high-dimensional feature processing and generalization, precisely modeling complex dynamics for accurate parameter estimation. The main contributions include:

(i)This study introduces the Lagrange Starfish Optimization Algorithm (LSFA) by integrating Lagrange interpolation into the Starfish Optimization Algorithm (SFOA), effectively addressing initial value sensitivity, noise, and outliers in parameter calibration. LSFA leverages SFOA's optimization capabilities to refine state estimation and enhances output smoothness through Lagrange interpolation. A weighting system further reduces noise and outliers during calibration. Experiments confirm improved accuracy and reduced measurement noise errors [26].

(ii)To enhance the precision of local exploration and determine the optimal starfish position, this paper proposes an improved local search method for the Starfish Optimization Algorithm (SFOA). During the local search phase of SFOA, five starfish individuals are randomly selected from the initial population, and their fitness values are evaluated. The fitness values of these five starfish positions are summed, and their respective weight ratios are calculated and sorted in ascending order. The position of the starfish with the highest fitness value is multiplied by the largest weight, while the position of the starfish with the lowest fitness value is multiplied by the smallest weight, thereby updating the starfish positions. This enhanced mechanism focuses on the internal optimization of the local search process—specifically, the weighted position update strategy—which prevents the algorithm from converging to local optima and significantly improves the efficiency and accuracy of SFOA.

(iii)Traditional robotic arm kinematic calibration methods often rely solely on Euclidean distance, limiting the comprehensiveness of accuracy assessment. This study introduces a multidimensional distance metric model that integrates Mahalanobis, Manhattan, Chebyshev, cosine, standardized Euclidean, and Euclidean distances to evaluate kinematic parameters. The proposed method incorporates an external multidimensional evaluation framework, validating the optimization precision and algorithmic robustness of individual effects and synergistic effects in complex environments. Experimental results demonstrate that the Lagrange Starfish Optimization Algorithm (LSFA) outperforms existing methods, providing an effective solution for robotic kinematic calibration [27, 28].

(iv)Building upon the LSFA framework, we introduce the Support Vector Machine (SVM) algorithm, termed the LSFA-SVM method. This approach significantly enhances the calibration accuracy of robotic arm dynamic parameters compared to the standalone SVM algorithm, achieving a 43.93% reduction in error and enabling high-precision calibration of dynamic parameters.

2. Preliminary Work

The kinematic calibration of industrial robots involves four key steps: kinematic modeling, measurement, parameter identification, and compensation [29,30], as illustrated in Figure 1.

(i)Kinematic Modeling: The kinematic model defines the geometric relationships between a robot's joints and links. The Denavit-Hartenberg (D-H) model is widely used for this purpose [31,34].

(ii)Measurement: Actual operational data are collected using sensors and measuring devices, providing essential information for calibration analysis and optimization.

(iii)Parameter Identification: This step determines the robot's kinematic and dynamic parameters. Kinematic parameters are optimized using the Lagrange Starfish Optimization Algorithm (LSFA), based on a multidimensional distance metric and improved local search. Dynamic parameters are refined using the Support Vector Machine (SVM) algorithm. The integrated LSFA-SVM

model further minimizes dynamic parameter errors, enhancing calibration accuracy.

(iv)Compensation: The kinematic parameters are adjusted based on the identified errors to improve the robot's absolute positioning accuracy.

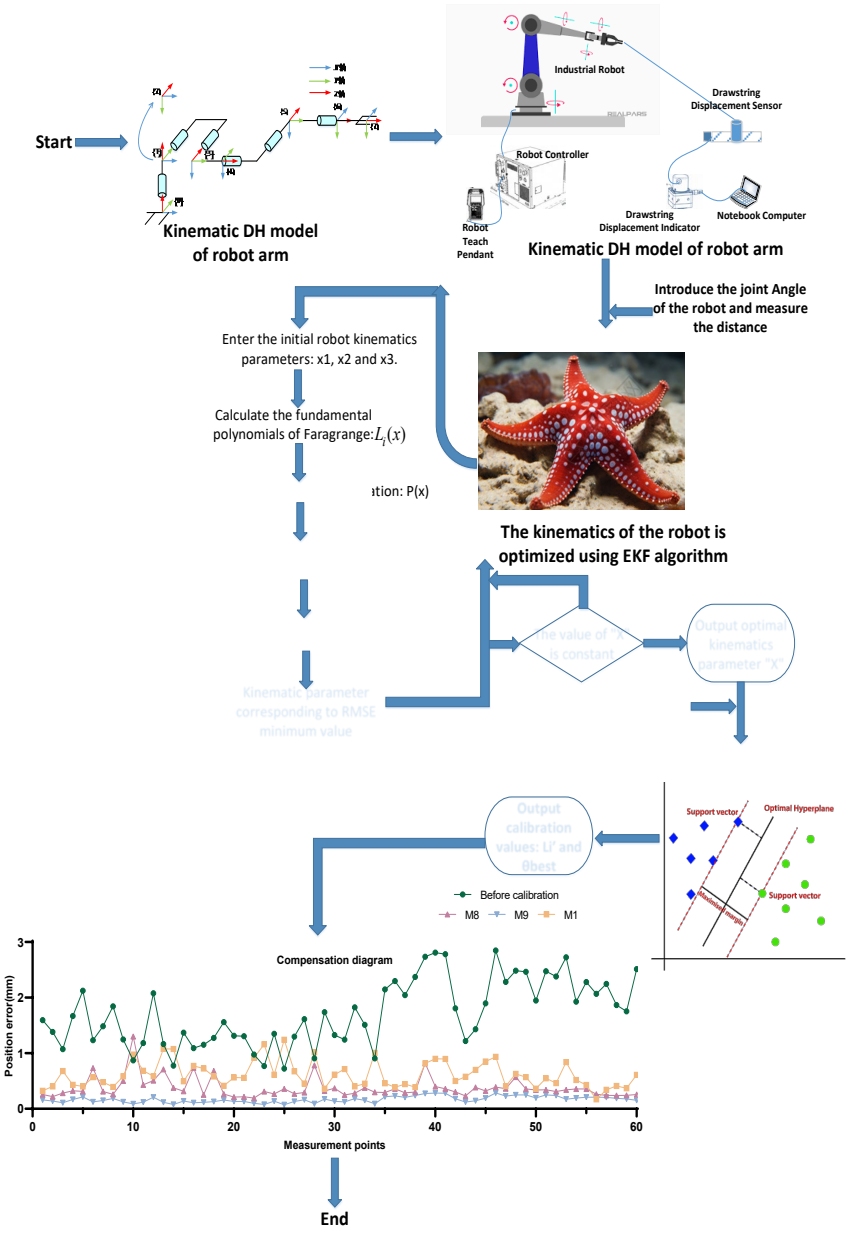


Figure 1. Flowchart illustrating the algorithm calibration procedure.

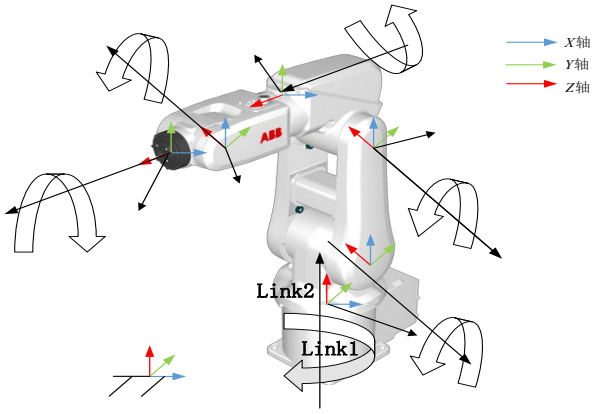


Figure 2. ABB irb 120 industrial robot.

The D-H model is a standardized method for robot modeling, defining geometric relationships between joints and links through a set of parameters. As shown in Figure 2, this study uses the ABB IRB 120 multi-joint robot as a case study, applying the D-H model to establish the robot's kinematic model through the following steps.

2.1. Parameter Definition

Table 1 lists ABB's standard D-H values for this robot, where "α" is the link twist angle, "a" the link length, "θ" the joint angle, and "d" the joint offset. These parameters define joint transformations and relationships under the Denavit-Hartenberg (D-H) convention.

Table 1. Nominal D-H values for the abb irb 120 manipulator.

	Joint_i	α_i (mm)	a_i (mm)	θ_i (mm)	d_i (mm)
	1	-90	0	0	290
	2	0	270	-90	0
i	**3**	-90	70	0	0
	4	90	0	0	302
	5	-90	0	0	0
	6	0	0	0	72

The Denavit-Hartenberg (D-H) model defines the transformation matrix for the i-th joint as follows:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1) $$

The transformation matrix T^{i}_{i-1} represents the shift from the (i-1)-th coordinate frame to the i-th coordinate frame. It encapsulates both translation and rotation, facilitating the conversion of points from the (i-1)-th frame to the i-th frame. The transformation matrices of neighboring joint are successively multiplied to create the total transformation matrix ${}^{0}_{n}T$ from the robot's base coordinate frame to the end-effector frame.

$$ {}^{0}_{n}T = {}^{0}_{1}T \times {}^{1}_{2}T \times {}^{2}_{3}T \times \cdots \times {}^{n-1}_{n}T , \quad (2) $$

The matrix ${}^{0}_{n}T$ represents the complete coordinate transformation from the robot's base to the end-effector, specifying both its orientation and position. It plays as a critical role in the calculation of calibration errors. During the experimental phase of this work, a gripper-like instrument was attached to the robot's end-effector to facilitate data measurement. Consequently, the comprehensive equation from the robot's base coordinate system to the gripper coordinate system is articulated as follows [20]:

$$ {}^{0}_{7}T = {}^{0}_{1}T \times {}^{1}_{2}T \times {}^{2}_{3}T \times \cdots \times {}^{6}_{7}T . \quad (3) $$

When performing robot kinematic modeling, it is essential to account for errors in the kinematic parameters of each link. By introducing an error model, we can express the deviations in the robot's transformation matrix. These inaccuracies predominantly stem from deviations between the actual parameters and their theoretical parameters, including link lengths, joint angles, link offsets, and link twist angles. The following section provides a detailed explanation and mathematical formulations [35,36].

2.2. Introduction of Errors

We may describe the robotic model's actual kinematic errors as follows, assuming that they differ from the theoretical kinematic parameters:

$$ \text{Actual link length: } a_i^{'} = a_i + \Delta a_i , \quad (4) $$

$$ \text{Actual link offset: } d_i^{'} = d_i + \Delta d_i , \quad (5) $$

$$ \text{Actual joint angle: } \theta_i^{'} = \theta_i + \Delta \theta_i , \quad (6) $$

$$ \text{Actual link twist angle: } \alpha_i^{'} = \alpha_i + \Delta \alpha_i . \quad (7) $$

In this context, Δa_i , Δd_i , $\Delta \theta_i$,and $\Delta \alpha_i$ represent the errors in link length, link offset, joint angle, and link twist angle, respectively.

2.3. The robotic kinematic error model has a transformation matrix as follows

The actual link transformation matrix can be articulated in a manner that incorporates the errors introduced:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c(\theta_i+\Delta\theta_i) & -s(\theta_i+\Delta\theta_i)c(\alpha_i+\Delta\alpha_i) & \cdots \\ s(\theta_i+\Delta\theta_i) & c(\theta_i+\Delta\theta_i)c(\alpha_i+\Delta\alpha_i) & \cdots \\ 0 & s(\alpha_i+\Delta\alpha_i) & \cdots \\ 0 & 0 & \cdots \\ s(\theta_i+\Delta\theta_i)s(\alpha_i+\Delta\alpha_i) & (a_i+\Delta a_i)c(\theta_i+\Delta\theta_i) & \\ -c(\theta_i+\Delta\theta_i)s(\alpha_i+\Delta\alpha_i) & (a_i+\Delta a_i)s(\theta_i+\Delta\theta_i) & \\ c(\alpha_i+\Delta\alpha_i) & d_i+\Delta d_i & \\ 0 & 1 & \end{bmatrix} . \quad (8) $$

Consequently, the error matrix $\Delta^{i-1}_{i}T$,which denotes the discrepancy between the actual transformation matrix ${}^{i-1}_{i}T'$ and the theoretical transformation matrix ${}^{i-1}_{i}T$,is defined as follows:

$$ \Delta^{i-1}_{i}T = {}^{i-1}_{i}T' - {}^{i-1}_{i}T , \quad (9) $$

This error matrix illustrates the variations in the robot's end-effector attitude resulting from small adjustments in various parameters. For multi-link robotic systems, the total error can be represented as the product of the error matrices of each link, thus leading to the total end-effector pose error $\Delta^{0}_{7}T$.

Based on linearization approximation theory and the actual kinematic model, we typically employ Taylor expansion to derive the error formula for the end-effector pose [37]. According to the Taylor expansion, the error in the transformation matrix of each link can be approximated as a linear superposition of the errors in the kinematic parameters. This linear approximation approach aligns well with contemporary constrained optimization methods used in kinematic control of redundant manipulators [38]. Specifically, we can derive a linear expression for the transformation matrix of each link. Assume that the transformation matrix of each link is a function of the parameters α, a, θ, and d.

$$ {}^{i-1}_{i}T = {}^{i-1}_{i}T(\alpha_i, a_i, \theta_i, d_i), \quad (10) $$

In the case of small variations in the parameters, let the infinitesimal change in the current parameters be denoted as Δα, Δa, Δθ and Δd. The linearized expression for the transformation matrix can be expressed as follows [39]:

$$ \Delta^{i-1}_{i}T = \frac{\partial^{i-1}_{i}T}{\partial\alpha_i}\Delta\alpha_i + \frac{\partial^{i-1}_{i}T}{\partial a_i}\Delta a_i + \frac{\partial^{i-1}_{i}T}{\partial\theta_i}\Delta\theta_i + \frac{\partial^{i-1}_{i}T}{\partial d_i}\Delta d_i . \quad (11) $$

Ultimately,the linearized representation of the transformation matrix can be expressed as [39]:

$$ \Delta^{i-1}_{i}T = J_i \times \Delta p . \quad (12) $$

Here, “J” denotes the Jacobian matrix [40], which can be expressed as:

$$ J_i = \left[\frac{\partial^{i-1}_{i}T}{\partial\alpha_i} \quad \frac{\partial^{i-1}_{i}T}{\partial a_i} \quad \frac{\partial^{i-1}_{i}T}{\partial\theta_i} \quad \frac{\partial^{i-1}_{i}T}{\partial d_i} \right] . \quad (13) $$

Moreover, Δp represents the vector of parameter variations:

$$ \Delta p = \left[\Delta\alpha_i \quad \Delta a_i \quad \Delta\theta_i \quad \Delta d_i \right]^T . \quad (14) $$

In this context, each of Δα, Δa, Δθ and Δd comprises six parameters. $\Delta\omega$ denotes the deviation of the robotic kinematic parameters,it represented as a 24×1 vector, thus [15]:

$$ \Delta\omega = \left[\Delta\alpha^T \quad \Delta a^T \quad \Delta\theta^T \quad \Delta d^T \right] . \quad (15) $$

In summary, by analyzing the effects of small variations in each Denavit-Hartenberg (DH) parameter on the robotic transformation matrix, we can develop a kinematic error model. This model elucidates the specific impacts of inaccuracies in different parameters on the robot's localization accuracy, thereby providing a more systematic approach and strategy for robot calibration and error compensation [19].

The error transformation matrix $\Delta^{i-1}_{i}T$ represents the posture deviation of the robot’s end effector. The first three rows of the fourth column in this matrix clearly represent the positional error of the end effector[41].

2.4. Kinematic Parameter Error Identification Model for Robots

In the calibration of industrial robot dynamic and kinematic parameters, commonly used instruments include laser trackers, 3D measuring arms, optical sensors, vision systems, torque sensors, accelerometers, and gyroscopes. While these instruments provide high measurement accuracy, they suffer from drawbacks such as high costs, operational complexity, and sensitivity to environmental conditions [42,43].

Additionally, conversion errors occur between the robot's base coordinate system, end effector's coordinate system, and the measuring apparatus's coordinate system. To reduce calibration costs and complexity, this study proposes a measurement scheme based on spatial fixed-point constraints for evaluating the calibration procedure.

First, a wire encoder measures the distance between the end effector of the ABB IRB 120 industrial robot and a fixed ground point. The wire length and joint angles are recorded from the encoder displacement display and teaching pendant. Using forward kinematics theory, the wire length is calculated. The difference between the calculated and measured wire lengths represents the end effector's position error [39]. This study establishes a kinematic parameter error identification model and calculates the end effector's position inaccuracy using Euclidean distance [44,46].

$$ f(\omega) = \frac{1}{n}\sum_{i=1}^{n}\left\| Y_i - Y_i' \right\| . \quad (16) $$

With this framework, f represents the goal function ω denotes the kinematic parameters, n denotes the number of samples, Y_i represents the quantified wire length, and $Y_i^{'}$ indicates the calculated wire length. The value of P^r can be obtained from the Cartesian coordinate system displayed on the teach pendant, while P_i^u is the computed position of the i-th end effector, specifically the first three rows of the last column of the transformation matrix ${}_n^0T$ .The fixed P_0 point of the wire encoder on the ground is then computed as expressed follows:

$$Y_i^{'} = \sqrt{\left(P_i^u - P_0\right)^2} . \tag{17}$$

ABB robots typically use a stationary reference coordinate system at the base. Cartesian coordinates and joint angles from the teach pendant are measured relative to this base system. Defining a tool coordinate system enhances precision and adaptability in specific applications. During calibration, a gripper attached to the end effector measures the distance between the end effector and a fixed point. Hand-eye calibration is essential for kinematic modeling, establishing a coordinate system at a specific position. Additionally, converting the gripper's coordinate system to one relative to the fixed point is critical [47,48].

3. Calibration of Robotic Kinematic and Dynamic Parameters Based on LSFA and SVM Algorithm

3.1. Establishment of a Calibration Device for Robot calibration Based on the LSFA

The Starfish Optimization Algorithm (SFOA) is a novel metaheuristic algorithm characterized by rapid convergence, strong high-dimensional optimization capabilities, and the ability to avoid local optima. Leveraging these advantages, this study applies SFOA to the calibration of kinematic parameters for ABB robots. This paper proposes an innovative approach that integrates SFOA with Lagrange interpolation, termed the Lagrange-Starfish Fusion Algorithm (LSFA). The following section details the steps for calibrating kinematic parameters using LSFA [49].

3.1.1 Initialization of the SFOA

The state vector X is defined to represent all the kinematic parameters of the joints.

$$X = \left[\alpha_1, a_1, \theta_1, d_1, \cdots, \alpha_n, a_n, \theta_n, d_n\right]^T . \tag{18}$$

Initialization of the Population: In the SFOA, the population is initialized by generating the positions of starfish individuals within the predefined bounds of the search space. The population position matrix is constructed as follows:

$$X_{\text{pos}} = rand(N, D) \bullet (ub - lb) + lb . \tag{19}$$

$$X_{\text{pos}} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1D} \\ X_{21} & X_{22} & \cdots & X_{2D} \\ X_{31} & X_{32} & \ddots & X_{3D} \\ X_{N1} & X_{N2} & \cdots & X_{ND} \end{bmatrix} . \tag{20}$$

Equation (18) represents the measurement equation of the state space model. X_{pos} denotes the population position matrix, N represents the number of individuals in the population, and D signifies the number of decision variables, the lower and upper bounds of the search space are denoted as lb ,and ub ,showing in Table 5 as follows.

After generating the initial position matrix, the fitness function values of the starfish positions are evaluated using the objective function. For the i-th individual in the population, its fitness value is calculated as shown in Equation (21):

The fitness values of all starfish are stored in vector F ,This method of storing fitness values simplifies the comparison and optimization of individual performances, thereby enhancing the efficiency of algorithm iterations, as shown in Equation (22).

$$Fit(X_i) = fh(X + X_{pos}(i,:)^{'}, \theta_i, Y_i) . \tag{21}$$

$$F = \left[Fit(X_1) \quad Fit(X_2) \quad \cdots \quad Fit(X_N)\right]_N^T . \tag{22}$$

3.1.2 Exploration of the SFOA

The exploration phase in the SFOA is designed to perform global search, aiming to extensively explore the solution space and identify potential promising regions while avoiding premature convergence to local optima.

During the exploration phase, the five arms of the starfish represent five potential search directions. The position of the starfish is updated by incorporating both the current best solution and a random solution. Specifically, the formula for adjusting the position of the starfish is presented as follows:

$$\begin{cases} \alpha = (2r-1)\pi \\ \theta = (\pi * T)/(2 * T_m) \\ X_i^{new} = X_i + \alpha(X_{best} - X_i)\cos\theta, & r \le 0.5 \\ X_i^{new} = X_i - \alpha(X_{best} - X_i)\sin\theta, & r > 0.5 \end{cases} . \tag{23}$$

Where X_i represents the current position of the starfish, X_{best} denotes the position of the starfish with the minimum fitness value in the population; α and θ are control parameters used to balance global search; X_i^{new} is the updated position; T is the current iteration count of the algorithm; T_m is the maximum number of iterations; and r is a random number between 0 and 1.

Characteristics of the exploration phase include an emphasis on diversity, thereby avoiding premature

convergence. This phase is particularly suited for identifying new promising regions within the solution space.

3.1.3 Exploitation of the SFOA

The primary objective of the exploitation phase is to perform local search, aiming to further refine the quality of solutions within promising regions identified during the exploration phase. This phase simulates the predatory and regenerative behaviors of starfish to update and optimize solutions. The specific update strategies are described as follows:

Calculate the vector difference between the global best position X_{best} of the starfish population and the positions of five randomly selected starfish X_j:

$$V_{qj} = (X_{best} - X_j). \tag{24}$$

where V_{qj} represents the vector between the global best position and the position of the j-th starfish.

Randomly select two vectors V_{qj1} and V_{qj2} from the five computed vectors and use them to update the position of the current starfish:

$$X_i^{new} = X_i + r1 * V_{qj1} + r2 * V_{qj2}. \tag{25}$$

where $r1$ and $r2$ are random numbers within the range $[0,1]$, V_{qj1} and V_{qj2} are the randomly selected vectors.

To enhance the precision of local exploration and determine the optimal starfish position, this study proposes an improved local search method for the Starfish Optimization Algorithm (SFOA). The specific steps are as follows:

(i)Fitness Value Calculation and Sorting: Five starfish positions are randomly selected from the population, and their corresponding fitness function values $f_i = (i = 1,2,...,5)$ are calculated.These values are sorted in ascending order and stored in vector $V_f = [f_1, f_2, \cdots, f_5]$, where $f_1 \le f_2 \le \cdots \le f_5$.

(ii)Weight Calculation and Sorting: The weight ratio r_i for each fitness function value is calculated, sorted in descending order, and stored in vector $V_r = [r_1, r_2, \cdots, r_5]$, where $r_1 \ge r_2 \ge r_3 \ge r_4 \ge r_5$.

(iii)Position Difference Calculation: The vector difference $p_i = X_{best} - X_i$ between the starfish positions corresponding to the fitness function values in V_f and the optimal starfish position X_{best} is calculated and stored in vector $X_p = [p_1, p_2, ...p_5]$.

(iv)Position Update: The sorted weight ratios V_r are multiplied by the position differences X_p and summed to obtain the updated local starfish position X_{new},as shown in the following equation:

$$r_i = f_i \bullet (\sum_{j=1}^{5} f_i)^{-1}. \tag{26}$$

$$X_i^{new} = X_i + \sum_{i=1}^{5} r_i \bullet p_i. \tag{27}$$

This method significantly improves the precision of local search while effectively avoiding the risk of the algorithm converging to local optima.

Furthermore, when a starfish is attacked or captured by a predator, it activates a self-protection mechanism by detaching an arm to escape. In the algorithm, this mechanism is employed to repair or update the positions of starfish with poor fitness function values. The specific steps are as follows:

$$X_{newi} = \exp\left(-T * N * T_m^{-1}\right) * X_i. \tag{28}$$

The exploration phase emulates the five-arm search capability of starfish to comprehensively explore the solution space, thereby mitigating the risk of converging to local optima. The exploitation phase replicates the predatory and regenerative behaviors of starfish to perform precise searches within identified promising regions, with the objective of locating the global optimum. These two phases operate in a synergistic manner, enabling the Starfish Optimization Algorithm (SFOA) to maintain an effective balance between global exploration and local exploitation, thus providing an efficient approach to solving complex optimization problems.

3.2. Calibration results based on Lagrange interpolation with the SFOA.

ABB industrial robots are widely used in high-precision tasks like painting, laser cutting, and electronic assembly, making precise and smooth motion control essential. This study proposes a calibration method (LSFA) combining the Lagrange interpolation algorithm with the Starfish Optimization Algorithm (SFOA) to refine results. Key advantages include:

Enhanced Accuracy: Lagrange interpolation constructs a polynomial through fixed points, improving kinematic parameter accuracy for reliable robot control.

Motion Smoothness: The continuous and differentiable interpolation function ensures smooth trajectories, reducing instabilities from abrupt parameter changes.

Robustness to Sparse Data: Lagrange interpolation effectively fills data gaps in sparse SFOA outputs, enhancing practicality.

Broad Applicability: As a versatile numerical technique, it handles discrete data points across diverse scenarios.

After SFOA calibration, Lagrange interpolation defines interpolation functions for the results.

$$L_i(x) = \prod_{\substack{j=0 \\ j \ne i}}^{n} \frac{x - x_j}{x_i - x_j}. \tag{29}$$

$$P(x) = \sum_{i=0}^{n} y_i L_i(x). \tag{30}$$

In this context, $L_i(x)$ denotes the i-th Lagrange basis polynomial, while y_i represents the function value corresponding to the variable x.

Let x_1 represent the initial kinematic parameters of the robotic arm, x_2 denote the outcomes derived from calibration with the Starfish optimization algorithm(SFOA), and x_3 represent the predicted values of the robot's kinematic parameters.

$$L_1(x)=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}. \quad (31)$$

$$L_2(x)=\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}. \quad (32)$$

$$L_3(x)=\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}. \quad (33)$$

$$P(x)=y_1L_1(x)+y_2L_2(x)+y_3L_3(x). \quad (34)$$

In order to obtain optimal interpolation results, we can take the derivative of the Lagrange interpolation polynomial.

$$P^{'}(x)=y_1L_1^{'}(x)+y_2L_2^{'}(x)+y_3L_3^{'}(x). \quad (35)$$

$$P_k=(I-K_kJ_k)P_k^{-}. \quad (36)$$

The ideal solution value x is presented as follows.

$$x=\frac{(x_2^2-x_3^2)L_1+(x_1^2-x_3^2)L_2+(x_1^2-x_2^3)L_3}{2L_1(x_2-x_3)+2L_2(x_1-x_3)+2L_3(x_1-x_2)}. \quad (37)$$

The Lagrange Starfish Optimization Algorithm (LSFA) efficiently calibrates robotic kinematic parameters (e.g., DH parameters) by integrating the global search capability of SFOA (avoiding local optima), Lagrange interpolation (smoothing parameter updates), and a multidimensional distance metric (robust error evaluation). Based on geometric and nonlinear error model characteristics, LSFA significantly enhances multi-parameter optimization, noise suppression, and adaptability to complex environments.

3.3. Calibration of dynamic parameters based on the Support Vector Machine (SVM) algorithm.

Subsequently, the optimal kinematic parameters from the Lagrange Starfish Optimization Algorithm (LSFA) were used to calibrate dynamic parameters. Industrial robot dynamic models are typically built using joint position, velocity, and acceleration parameters to describe operational forces and motion. In practice, joint angle errors often approximate dynamic errors, as they reflect dynamic model deviations through linearization. Studies indicate that correcting joint angle errors under certain conditions significantly enhances robotic arm motion control accuracy.Support Vector Machine (SVM), a supervised learning technique, evaluates joint angle deviations in robot calibration, with performance relying on the loss function design. The following section details the steps for dynamic parameter calibration using SVM based on measured joint angles [50-51]. The loss function of a linear SVM includes two components: hinge loss and regularization, expressed as:

$$\delta=\min(\frac{1}{2}\lambda_1\|\omega_1\|^2+\frac{1}{n}\sum_{i=1}^{n}(L_i-L_i^{'})). \quad (38)$$

where λ_1 is the regularization coefficient and ω_1 represents the weights of the hidden layer. In SVM, the input can be mapped to a high-dimensional space to facilitate classification by a linear classifier. The equation delineating the transition from the input layer to the hidden layer is expressed as:

$$z=w_1x+b_1. \quad (39)$$

$$x=\theta_i*\theta_j. \quad (40)$$

where z is the input function of the concealed layer. x is the kernel function used to compute the similarity between samples, and θ represents the joint angles. The equation delineating the transition from the hidden layer to the output layer is expressed as:

$$h=f(z)=(1+e^{-z})^{-1}. \quad (41)$$

The equation describing the output layer is given by:

$$L_i^{'}=\omega_2h+b_2. \quad (42)$$

where h is the output function of the hidden layer, $L_i^{'}$ is the output layer function, ω_2 represents the regularization coefficient. $f(\bullet)$denotes the activation function of the buried layer, and b_1 and b_2 are the bias vectors. Following the principles of back propagation, the loss function is minimized using gradient descent. The weight and bias update scheme is formulated as:

$$\frac{\partial\delta}{\partial\omega_1}=-\sum_{i=1}^{n}\sum_{j=1}^{6}(L_i-L_i^{'})\omega_2h(1-h)x. \quad (43)$$

$$\frac{\partial\sigma}{\partial\omega_2}=-\sum_{i=1}^{n}(L_i-L_i^{'})h_2+\lambda_1\omega_2. \quad (44)$$

The parameter values of LSFA are shown in Table 5 as follows.

The Support Vector Machine (SVM), with its advantages in nonlinear modeling, noise resistance, small-sample adaptability, and high-dimensional optimization, effectively calibrates robotic arm dynamic parameters. By training on real-world joint angle data, SVM captures the relationship between joint angle deviations and dynamic parameters, enabling accurate dynamic parameter predictions. Calibrated joint angle data compensate for errors, thereby enhancing the robotic arm's repeat positioning accuracy through improved motion control precision.

3.4. Kinematic Parameter Evaluation Framework Based on Multi-Dimensional Distance Metrics

This study proposes a multidimensional distance metric method based on the Lagrange Starfish Optimization Algorithm (LSFA) for optimizing the kinematic parameters of robotic arms. First, the distance between the end-effector of the robotic arm and a fixed reference point is precisely measured using a cable encoder. This distance is then mathematically characterized using Mahalanobis Distance, Manhattan Distance, Euclidean Distance, Chebyshev Distance, Cosine Distance, and Standardized Euclidean Distance. For each distance metric, the LSFA algorithm calculates the fitness function values for both the initial and updated positions of the starfish.Table 5.Multi-Distance Metric Computation.

By comparing these values, the optimal fitness function values corresponding to the six distance metrics are determined. Through iterative computation, the best starfish positions under the six distance metrics are ultimately obtained. Furthermore, the fitness function values corresponding to the six distance metrics are sorted in ascending order and summed. The weight ratio of each fitness function value to the total distance is calculated and sorted in descending order. Finally, the six starfish positions corresponding to the sorted fitness function values are multiplied by their respective weight ratios and summed to derive a comprehensive starfish position based on multidimensional distance metrics. By integrating the advantages of multiple distance metrics, this method effectively addresses the limitations of the starfish optimization algorithm in exploration and foraging within complex environments, significantly improving the accuracy and robustness of kinematic parameter optimization.

Here, Y^{m} , Y^{mh} , Y^{e} , Y^{cb} , Y^{co} ,and Y^{se} represent the distances measured by the cable encoder between the end-effector of the robotic arm and the fixed reference point, characterized by Mahalanobis Distance, Manhattan Distance, Euclidean Distance, Chebyshev Distance, Cosine Distance, and Standardized Euclidean Distance, respectively. Similarly, Y^{mu} , Y^{mhu} , Y^{eu} , Y^{cbu} , Y^{cou} ,and Y^{seu} denote the distances after calibrating the kinematic parameters, characterized by the aforementioned six distance metrics.

In robotic kinematic calibration, Mahalanobis Distance evaluates the discrepancy between the end-effector and target positions while accounting for correlations between dimensions (e.g., positional and orientation errors), providing comprehensive error analysis. Manhattan Distance rapidly assesses the total discrepancy, offering high computational efficiency in high-dimensional spaces. Chebyshev Distance measures the maximum deviation, making it suitable for scenarios sensitive to extreme errors. Cosine Distance evaluates orientation discrepancies by measuring the angle between vectors. Standardized Euclidean Distance eliminates unit and scale influences, assessing standardized errors and enhancing calibration robustness.Mathematical Expressions of Distance Metrics:

$$\begin{cases} Y_i^m = \sqrt{\left(p_i^r - p_0\right)^T S^{-1}\left(p_i^r - p_0\right)} \\ Y_i^{mh} = \sum_{i=1}^{n} \left|x_i - x_0\right| + \left|y_i - y_0\right| + \left|z_i - z_0\right| \\ Y_i^e = \sqrt{\sum_{i=1}^{n} (p_i^r - p_0)^2} \\ Y_i^{cb} = \max_i \left|x_i - y_0\right| + \left|x_i - y_0\right| + \left|x_i - y_0\right| \\ Y_i^{co} = 1 - \dfrac{p_i^r * p_0}{\left\|p_i^r\right\| \left\|p_0\right\|} \\ Y_i^{se} = \sqrt{\sum_{i=1}^{n} \dfrac{(p_i^r - p_0)^2}{\sigma_i^2}} \end{cases} \quad (45)$$

After updating the kinematic parameters using the Lagrangian Starfish Optimization Algorithm (LSFA), this study recalculates the distances between the robotic arm's end-effector and the fixed point using the aforementioned six distance metrics, denoted as Y^{mu} , Y^{mhu} , Y^{eu} , Y^{cbu} , Y^{cou} ,and Y^{seu} .Subsequently, the error between the initial measured distances and the computed distances is calculated using Equation (17) to evaluate the optimization performance of the kinematic parameters.Here, Y^{m} denotes the fixed point, Y^{mh} represents the measured data point; Y^{e} is the covariance matrix of the vector, Y^{cb} is the total number of data points measured by the cable encoder and Y^{co} is the standard deviation of Y^{se} .

Table 2. Design of the Calibration Algorithm.

	Algorithm 1: LSFA-SVM	
	Input: $X,N,D,D_0,ub,lb,\alpha,\beta,\theta,r1,r2,T,max\_T,\{q_{i1},q_{i2},...,q_{i6}\},\{Y_1,Y_2,...,Y_m\},L_i,EP$.	
	Operation	***Cost***
	/* Initialization */	
1.	**Initialize:** $X=X_0,\omega_1,\omega_2 Ep=0.5$	
	/* LSFA-Step */	
2.	**while** not converge and $T \leq max\text{-}T$ **do**	
3.	**If** $rand<Ep$ ……%Exploration phase	
4.	···calculate θ by Eq.(23) and the fitness of starfish populations	
5.	**For** i=1 to N **do**	
6.	set five randomly p-dimensions among D(D>5)	
7.	set the starfish populations location between ub and lb	
8.	**For** j=1to 5 **do**	
9.	calculate α by Eq.(23)	
10.	compute the location of starfish populations using Eq.(23)	T_{12}
11.	**End for**	
12.	**End for**	
13.	**Else** ……%Exploitation phase	
14.	compute V_{gl} with Eq.(26)	
15.	**for** i=1 to N **do**	
16.	compute the location of starfish populations using Eq.(27)	
17.	**if** i=N **do**	
18.	update the positions using Eq.(28)	
19.	**end if**	
20.	**end for** ……%updating finished	
21.	**End if**	
22.	**for** i=1 to N **do**	
23.	update the fitness values for all starfish individuals with Eq.(21)	
24.	update the positions of starfish by Lagrange interpolation algorithm using Eq.(37)	
25.	**end for**	
26.	obtain the global best solution in iteration T step	
27	T=T+1	
28.	**End while**	
29.	obtain global solution	
	Output: L'_i, X_{best}	
	/* SVM-Step */	
30.	**while** not converge and $t \leq N$ **do**	
31.	set X_0=x_{left}	
32.	**for** i=1 to $\lvert n \rvert$	
33.	calculation δ, with (38)	
34.	update z with(39)	
35.	update x with(40)	
36.	compute h based on(41)	T_{13}
37.	compute L_i' based on(42)	
38.	update ω_1 based on(43)	
39.	update ω_2 with(44)	
40.	t=t+1	
41.	**end for**	
42.	**out put** L_i', θ_{best}	
	/* Operation Ending */	
	Output: X	

4. Experiment and Analysis

4.1. calibration experiment

Building on the theoretical analysis, experiments were conducted using an ABB IRB120 robotic arm, cable-driven encoders, a displacement display, LabVIEW software, and a laptop (Fig. 3). Figure 4 illustrates the data collection interface, which gathered training data from 110 spatial positions. To evaluate the effectiveness of the calibration algorithm, 10 test samples were validated near the robotic arm's workspace boundaries and singularity points. Table 6. Parameter tables of LSFA and SVM.

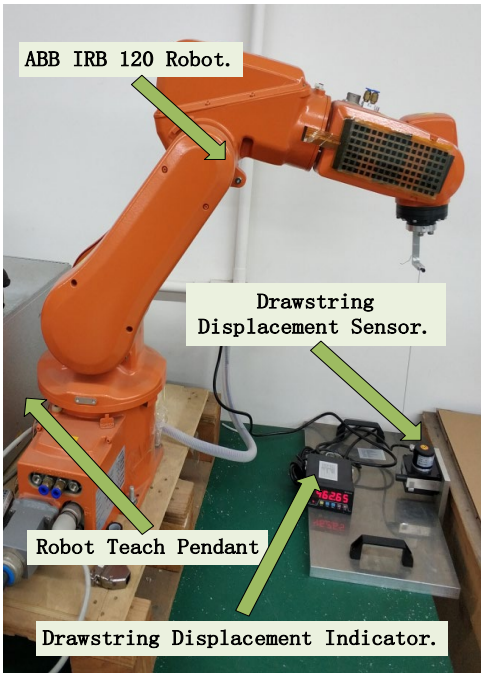


Figure 3. Experimental platform for robot calibration.

To accurately evaluate the performance of the calibration algorithm, this study selects uses the Root Mean Square Error (RMSE), Standard Deviation (STD), and Maximum Error (MAX) as important indicators for evaluating the accuracy of the robot calibration [52]. The following provides a detailed description of these three metrics.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(L_i - L_i^{'}\right)^2}\ . \tag{46}$$

$$STD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(L_i - \bar{L}_i^{'}\right)}\ . \tag{47}$$

$$MAX = \max\left\{\sqrt{\left(L_i - L_i^{'}\right)^2}\right\}. \tag{48}$$

4.2. Analysis of the Calibration Result of the Algorithm

In robot kinematics and dynamics calibration, smaller RMSE, STD, and MAX values indicate higher accuracy.Experiment Performance Analysis: This experiment evaluates the calibration accuracy of kinematic and dynamic parameters using the Lagrange Starfish Algorithm (LSFA) with a multidimensional distance metric and improved local search, combined with the Support Vector Machine (SVM) algorithm. Table 3 compares algorithm performance, Table 4 summarizes the algorithms, and Figure 5 shows post-calibration localization errors for each approach.

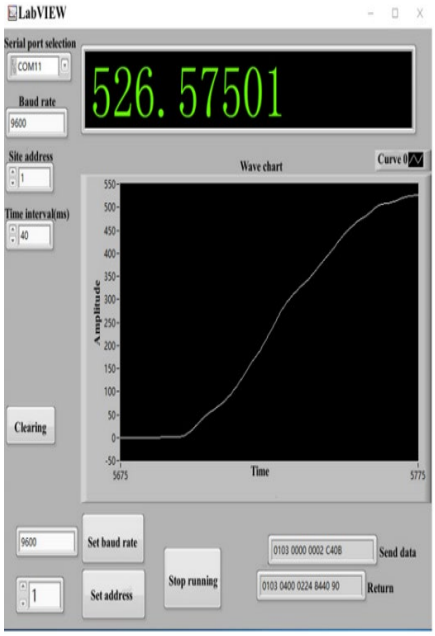


Figure 4. illustrates the interface of the data collection program.

Table 3. Performance Comparison of Various Calibration Algorithms.

Item		RMSE (mm)	STD (mm)	MAX (mm)	Iterations
Calibrate robotic arm kinematics	Before	2.09	2.00	3.36	53
	M1	0.66	0.56	1.71	13
	M2	0.64	0.53	1.38	16
	M3	0.50	0.41	1.16	16
	M4	0.54	0.44	1.26	10
	M5	0.70	0.56	1.76	17
	M6	0.61	0.48	1.51	30
	M7	0.73	0.61	1.60	29
	M8	0.41	0.32	1.03	30
	M9	0.29	0.21	0.91	30
Calibrate robotic arm dynamics	M10	0.66	0.54	1.57	40
	M11	0.37	0.34	0.96	25

4.3. Based on the experimental result,we summarize the following

As shown in Figures 5(a) and (b), both SFOA and LSFA outperform other algorithms in convergence speed and accuracy during kinematic parameter calibration. The LSFA-SVM integration significantly enhances dynamic parameter calibration precision.

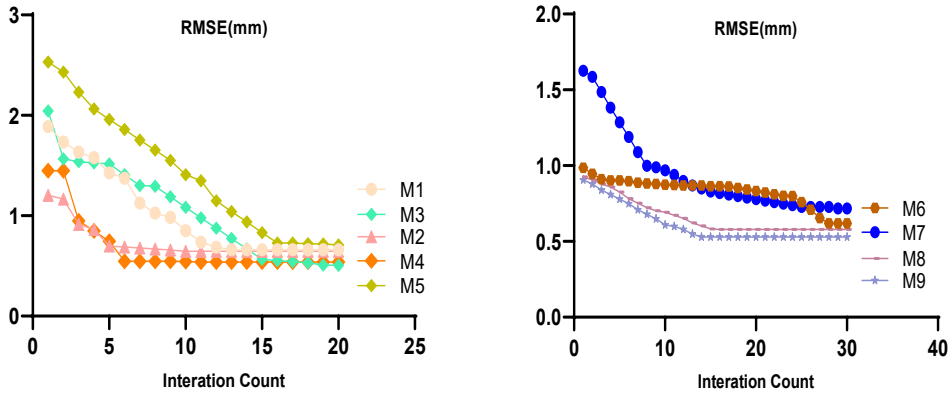
In Figure 6, M1 to M8 denote algorithms for calibrating the robotic arm's kinematic parameters. M8 shows results from the Starfish Optimization Algorithm (SFOA), with an RMSE of 0.41 mm, STD of 0.32 mm, and MAX of 1.03 mm. M9, based on the Lagrange Starfish Optimization Algorithm (LSFA) with a multidimensional distance metric and improved local search, achieves RMSE, STD, and MAX values of 0.29 mm, 0.21 mm, and 0.91 mm, respectively, improving accuracy by 29.27%, 34.4%, and 11.65% over M8.

M10 represents the Support Vector Machine (SVM) algorithm for dynamic parameter calibration, while M11 denotes the integrated LSFA-SVM algorithm. M11 improves calibration accuracy over M10 by 43.93%, 37.03%, and 38.85%, respectively.

Table 4. Comparison of Calibration Algorithms.

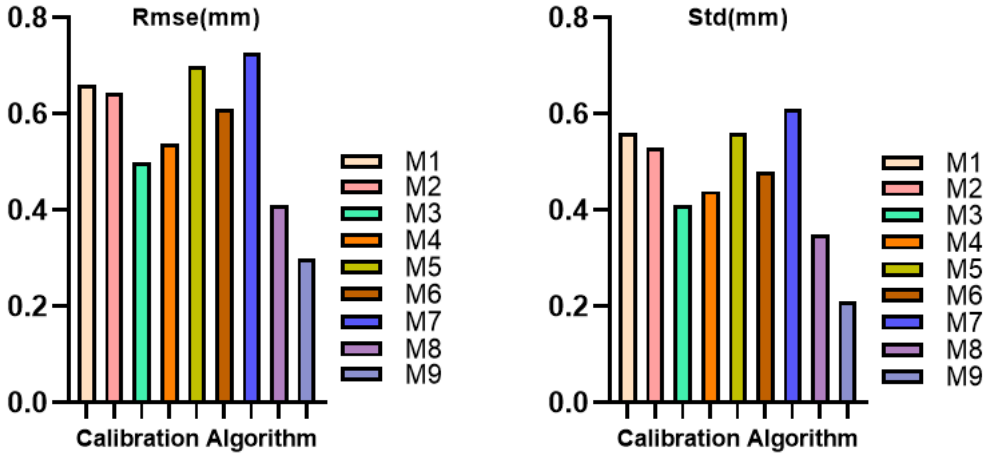
Method	Description
M1	The Extended Kalman Filter (EKF) effectively mitigates non-Gaussian noise during robot calibration.
M2	The Particle Filter (PF) algorithm is a Bayesian filtering technique suitable for nonlinear and non-Gaussian state estimation [53].
M3	The Levenberg-Marquardt algorithm is an optimization technique used for nonlinear least squares problems[13].
M4	The LMGA algorithm integrates the Levenberg-Marquardt (LM) algorithm with Genetic Algorithms (GA) for robotic parameter calibration [45].

M5	The Beetle Antennae Search (BAS) method is a bio-inspired optimization approach based on intelligent design [54,55].
M6	The Genetic Algorithm (GA) pursues optimal solutions using the principles of "survival competition" and "survival of the fittest" [56,57].
M7	The EPF algorithm is a technique for calibrating robotic parameters that integrates the Extended Kalman Filter (EKF) with the Particle Filter (PF) [58].
M8	The Starfish optimization algorithm (SFOA) is a newly proposed intelligent optimization algorithm.
M9	The LSFA integrates the Lagrange interpolation algorithm and the Starfish Optimization Algorithm.
M10	SVM is a supervised learning algorithm for classification and regression tasks.
M11	LSFA-SVM integrates the LSFA and SVM algorithms.

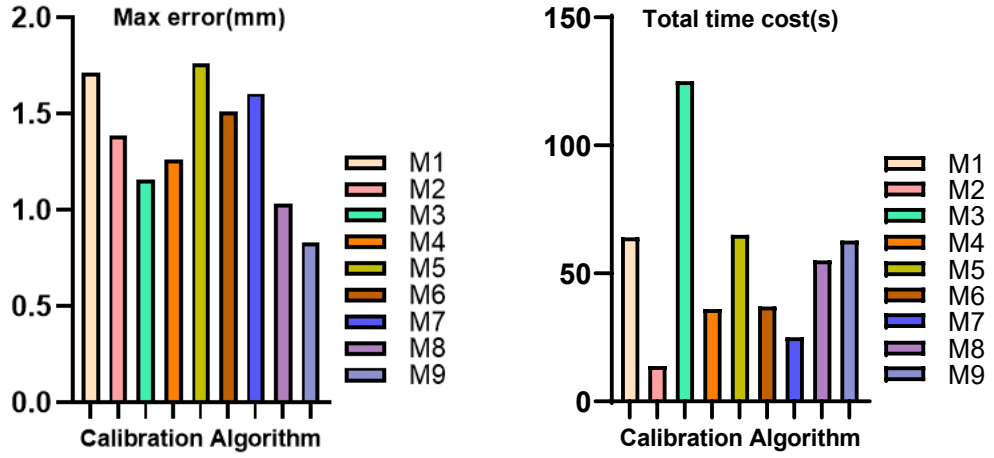


(a) (b)

Figure 5. Calibration method training curve.



(a) (b)



(**c**) (**d**)

Figure 5. The result of the calibration method and total time cost.

Table 5. Multi-Distance Metric Computation and LSFA-SVM Interaction Flow

Multi-Distance Metric Computation Steps	**LSFA-SVM Interaction Logic**
Input Data: Robot end-effector position P_i and fixed reference point P_0	LSFA Input: Initial kinematic parameters
Parallel Computation of Six Distances(Eq.45)	LSFA Optimization:(Eq.29-37)
Weight Assignment: Generate weights r_i based on error ranking (Eq. 26).	Output to SVM: Optimized parametersand dynamic data
Integrated Distance Output: Weighted summation	SVM Modeling:(Eq.38-44)
Output compensation:Δkinemic parameters	Output compensation:Δdynamic parameters

Table 6. Parameter tables of LSFA and SVM

	LSFA	**Value**	**SVM**	**Value**
Parameter	Population Size (N)	30	Kernel Function	x
	Max Iterations	1000	Regularization(λ_1)	10.0
	Variables(D)	24	Kernel Coefficient(λ)	0.1
	ub	0.28	Max Iterations	120
	lb	-1.58	h	6

As shown in Figure 7, the Lagrange Starfish Optimization Algorithm (LSFA), integrating a multidimensional distance metric model and improved local search, employs six distance measures Mahalanobis, Manhattan, Chebyshev, cosine, standardized Euclidean, and Euclidean distances—to optimize kinematic parameter calibration. By sequentially aggregating D1-D6 (120 sample points), LSFA significantly enhances calibration accuracy, demonstrating a clear downward trend in test errors. Specifically:(a) Model aggregation reduces RMSE;(b) Model aggregation reduces STD;(c) Model aggregation reduces MAX;(d) Model aggregation reduces POSITION.

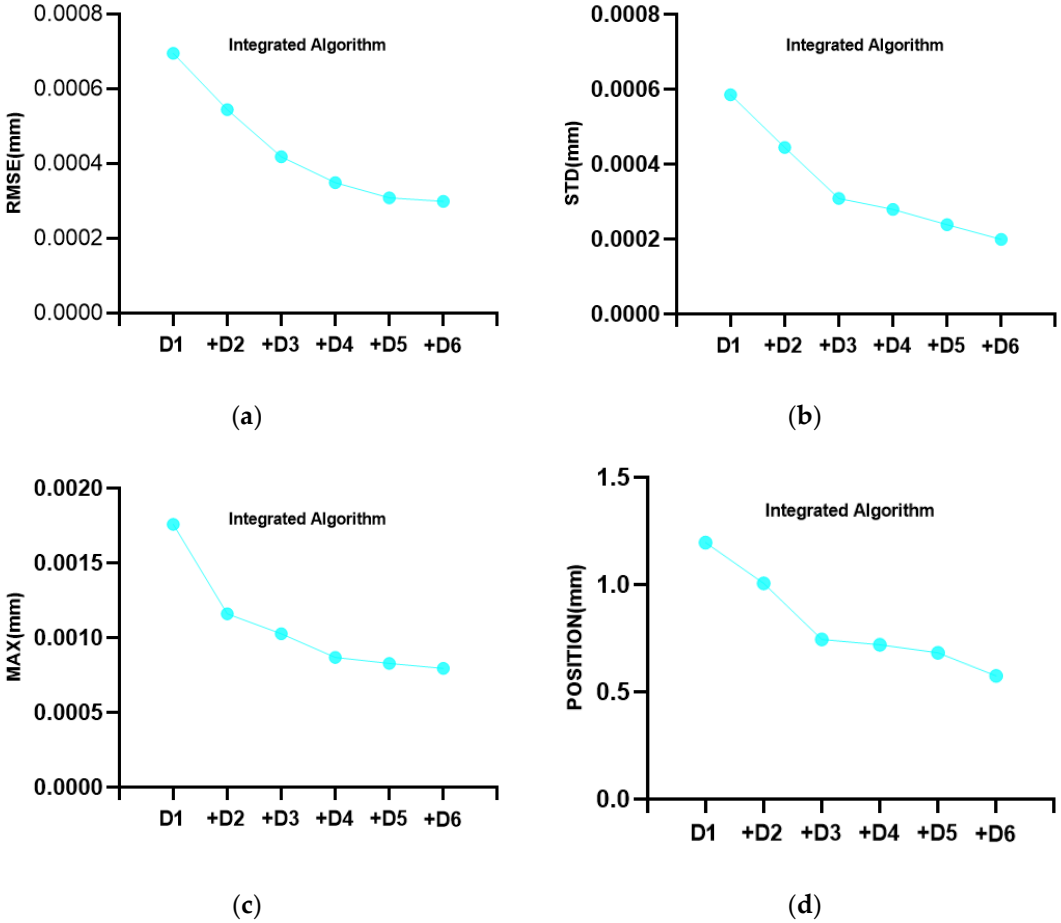


Figure 7. The calibration accuracy of model aggregation with six distance metrics.

5. Conclusions and prospective research

This study proposes the Lagrange Interpolation Local Search Improved Starfish Optimization Algorithm (LSFA), a comprehensive method for calibrating industrial robotic arms. LSFA integrates a multidimensional distance metric model—including Mahalanobis, Manhattan, Chebyshev, cosine, standardized Euclidean, and Euclidean distances—to optimize kinematic parameters. Lagrange interpolation reduces noise and outliers, while the Starfish Optimization Algorithm (SFOA) provides robust global search capabilities. Enhanced local search and multiple distance metrics further improve the algorithm's robustness and adaptability. Additionally, integrating the Support Vector Machine (SVM) algorithm into LSFA significantly enhances the accuracy of dynamic parameter calibration.

Experimental results demonstrate the improved algorithm's high precision and robustness in complex environments. Future research will focus on combining intelligent optimization with neural networks, multi-robot collaborative calibration, deep learning for automation, and validating the method in diverse practical scenarios. These efforts aim to advance efficient and precise robotic technologies for industrial automation.

Funding:

This research was funded by the National Funded Postdoctoral Research Program GZC20241900, Natural Science Foundation Program of Xinjiang Uygur Autonomous Region 2024D01A141, National Natural Science Foundation of China 62101076, Tianchi Talents Program of Xinjiang Uygur Autonomous Region and Postdoctoral Fund of Xinjiang Uygur Autonomous Region, Cooperation projects between Chongqing universities in Chongqing and institutions affiliated with the Chinese Academy of Science(HZ2021011), Key Projects of Open Fund ZSAQ202401, ZSAQ202414, ZSAQ202422, ZSAQ202423, ZSAQ202425.

References

[1] Guo Y, Gao X, Wei L, et al. Robot joint space grid error compensation based on three-dimensional discrete point space circular fitting. CIRP J Manuf Sci Technol. 2024;50:140-150.

[2] 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. 2023;72:1-12.

[3] Wu X, Zuo W, Lin L, Jia W, Zhang D. F-SVM: combination of feature transformation and SVM learning via convex relaxation. IEEE Trans Neural Netw Learn Syst. 2018;29(11):5185-5199.
[4] Jiang Z, Zhou W, Li H, Mo Y, Ni W, Huang Q. A new kind of accurate calibration method for robotic kinematic parameters based on the extended Kalman and particle filter algorithm. IEEE Trans Ind Electron. 2018;65(4):3337-3345.
[5] Cao HQ, Nguyen HX, Nguyen TT, Nguyen VQ, Jeon JW. Robot calibration method based on extended Kalman filter–dual quantum behaved particle swarm optimization and adaptive neuro-fuzzy inference system. IEEE Access. 2021;9:132558-132568.
[6] Chen T, Li S. Highly accurate robot calibration using adaptive and momental bound with decoupled weight decay. arXiv preprint arXiv:2408.12087. 2024.
[7] Dzedzickis A, Subačiūtė-Žemaitienė J, Šutinys E, et al. Advanced applications of industrial robotics: new trends and possibilities. Appl Sci. 2021;12(1):135.
[8] Javaid M, Haleem A, Singh RP, et al. Substantial capabilities of robotics in enhancing industry 4.0 implementation. Cogn Robot. 2021;1:58-75.
[9] Zhu D, Feng X, Xu X, et al. Robotic grinding of complex components: a step towards efficient and intelligent machining challenges, solutions, and applications. Robot Comput Integr Manuf. 2020;65:101908.
[10] Branco MP, et al. Encoding of kinetic and kinematic movement parameters in the sensorimotor cortex: a brain-computer interface perspective. Eur J Neurosci. 2019;50(5):2755-2772.
[11] 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 Manuf. 2024;86:102660.
[12] Chen C, Cao L, Chen Y, Chen B, Yue Y. A comprehensive survey of convergence analysis of beetle antennae search algorithm and its applications. Artif Intell Rev. 2024;57(6):141.
[13] Miao L, Zhang Y, Song Z, et al. A two-step method for kinematic parameters calibration based on complete pose measurement—verification on a heavy-duty robot. Robot Comput Integr Manuf. 2023;83:102550.
[14] Chen F, Zhao G, Zhao J, et al. Calibration of a parallel mechanism in a serial-parallel polishing machine tool based on genetic algorithm. Int J Adv Manuf Technol. 2015;81:27-37.
[15] Li H, Hu X, Zhang X, Wei S. Kinematic parameters calibration of industrial robot based on RWS-PSO algorithm. Proc Inst Mech Eng C J Mech Eng Sci. 2023;237(14):3210-3220.
[16] Luo G, Zou L, Wang Z, et al. A novel kinematic parameters calibration method for industrial robot based on Levenberg-Marquardt and differential evolution hybrid algorithm. Robot Comput Integr Manuf. 2021;71:102165.
[17] Liu Y, Wu J, Wang L, Wang J. Parameter identification algorithm of kinematic calibration in parallel manipulators. Adv Mech Eng. 2016;8(9):1687814016667908.
[18] Sun T, Lian B, Zhang J, Song Y. Kinematic calibration of a 2-DoF over-constrained parallel mechanism using real inverse kinematics. IEEE Access. 2018;6:67752-67761.
[19] Kong Y, Yang L, Chen C, Zhu X, Li D. Online kinematic calibration of robot manipulator based on neural network. Measurement. 2024;238:115281.
[20] Le PN, Kang HJ. A new manipulator calibration method for the identification of kinematic and compliance errors using optimal pose selection. Appl Sci. 2022;12(11):5422.
[21] Luo X, Li Z, Yue W, Li S. A calibrator fuzzy ensemble for highly-accurate robot arm calibration. IEEE Trans Neural Netw Learn Syst. 2025;36(2):2169-2181.
[22] Moshayedi AJ, Roy AS, Kolahdooz A, et al. Deep learning application pros and cons over algorithm. EAI Endorsed Trans AI Robot. 2022;1(1):e7.
[23] Zanfei A, Menapace A, Brentan BM, et al. Shall we always use hydraulic models? A graph neural network metamodel for water system calibration and uncertainty assessment. Water Res. 2023;242:120264.
[24] Moshayedi AJ, Roy AS, Sambo SK, et al. Review on: the service robot mathematical model. EAI Endorsed Trans AI Robot. 2022;1(1):1-19.
[25] Wang Z, Cao B, Xie Z, et al. Kinematic calibration of a space manipulator based on visual measurement system with extended Kalman filter. Machines. 2023;11(3):409.
[26] Bao W, Bi M, Xiao S, Fang J, Huang T, Zhang Y. Lagrange interpolation based extended Kalman filter for phase noise suppression in CO-OFDM system. Opt Commun. 2019;435:221-226.
[27] Liu J, Huang Q, Ulishney C, Dumitrescu C. A support-vector machine model to predict the dynamic performance of a heavy-duty natural gas spark ignition engine. SAE Tech Pap. 2021.
[28] Sarin K, Bardamova M, Svetlakov M, et al. A three-stage fuzzy classifier method for Parkinson's disease diagnosis using dynamic handwriting analysis. Decis Anal J. 2023;8:100274.
[29] Qi J, Chen B, Zhang D. A calibration method for enhancing robot accuracy through integration of kinematic model and spatial interpolation algorithm. J Mech Robot. 2021;13(6):061013.
[30] Lin X, Zhu H, Ennocent AF. Locomotion trajectory optimization for quadruped robots with kinematic parameter calibration and compensation. Measurement. 2024:115622.
[31] Yu D. Kinematic parameter identification for a parallel robot with an improved particle swarm optimization algorithm. Appl Sci. 2024;14(15):6557.
[32] Gao G, Guo X, Li G, et al. Kinematic parameter identification and error compensation of industrial robots based on unscented Kalman filter with adaptive process noise covariance. Machines. 2024;12(6):406.
[33] Zhu Y, Yang C, Wei Q, et al. Human–robot shared control for humanoid manipulator trajectory planning. Ind Robot. 2020;47(3):395-407.
[34] Fan Y, Lv X, Lin J, Ma J, Zhang G, Zhang L. Autonomous operation method of multi-DOF robotic arm based on binocular vision. Appl Sci. 2019;9(24):5294.
[35] Woodside MR, Fischer J, Bazzoli P, et al. A kinematic error controller for real-time kinematic error correction of industrial robots. Procedia Manuf. 2021;53:705-715.
[36] Lao D, Quan Y, Wang F, Liu Y. Error modeling and parameter calibration method for industrial robots based on 6-DOF position and orientation. Appl Sci. 2023;13(19):10901.
[37] Liu J, Yang Y, Xu B, et al. RSTC: residual Swin transformer cascade to approximate Taylor expansion for image denoising. Comput Vis Image Underst. 2024;248:104132.
[38] Liao B, Li J, Li S, Li Z. Briefly revisit kinematic control of redundant manipulators via constrained optimization. 2022.
[39] Gan Y, Duan J, Dai X. A calibration method of robot kinematic parameters by drawstring displacement sensor. Int J Adv Robot Syst. 2019;16(5):1729881419883072.
[40] Xu J, Liu X, Li P, et al. Calibration method of industrial robot kinematic parameters based on improved SSA

algorithm. In: Proc 6th Int Conf Softw Eng Comput Sci. IEEE; 2023:1-7.
[41] Soleymani M, Kiani M. Planar soft space robotic manipulators: dynamic modeling and control. Adv Space Res. 2024;74(1):384-402.
[42] Kana S, Gurnani J, Ramanathan V, et al. Fast kinematic re-calibration for industrial robot arms. Sensors. 2022;22(6):2295.
[43] Xiang T, Jiang X, Qiao G, Gao C, Zuo H. Kinematics parameter calibration of serial industrial robots based on partial pose measurement. Mathematics. 2023;11(23):4802.
[44] Cheng B, Wang B, Chen S, Huang D, et al. Investigation of axis-fitting-based measurement and identification techniques for kinematic parameters in multi-joint industrial robots. Precis Eng. 2024;91:1-13.
[45] Kang Z, Wang L, Sun A, et al. Two-step calibration of 6-DOF industrial robots by grouping kinematic parameters based on distance constraints. Measurement. 2024;235:114906.
[46] Zhao C, Qian L, Wang S, Jia Y, Luo X. A closed-loop kinematic approach to self-calibration of joint offset for multi-branched robots. Measurement. 2024;238:115237.
[47] Cao D, Liu W, Liu S, Chen J, Liu W, Ge J, Deng Z. Simultaneous calibration of hand-eye and kinematics for industrial robot using line-structured light sensor. Measurement. 2023;221:113508.
[48] Feng X, Tian D, Wu H, Qian C, Zhu D. A matrix-solving hand-eye calibration method considering robot kinematic errors. J Manuf Process. 2023;99:618-635.
[49] Song Y, Tian W, Tian Y, et al. An efficient calibration method for serial industrial robots based on kinematics decomposition and equivalent systems. Robot Comput Integr Manuf. 2023;84:102607.
[50] Ma S, Deng K, Lu Y, et al. Error compensation method of industrial robots considering non-kinematic and weak rigid base errors. Precis Eng. 2023;82:304-315.
[51] Ahmad NS, Goh P. Material classification via embedded RF antenna array and machine learning for intelligent mobile robots. Alex Eng J. 2024;106:60-70.
[52] Liu J, Deng Y, Liu Y, Chen L, Hu Z, Wei P, Li Z. A logistic-tent chaotic mapping Levenberg Marquardt algorithm for improving positioning accuracy of grinding robot. Sci Rep. 2024;14(1):9649.
[53] Deng X, Ge L, Li R, Liu Z. Research on the kinematic parameter calibration method of industrial robot based on LM and PF algorithm. In: Proc Chin Control Decis Conf. IEEE; 2020:2198-2203.
[54] Wang Y, Chen Z, Zu H, Zhang X, Mao C, Wang Z. Improvement of heavy load robot positioning accuracy by combining a model-based identification for geometric parameters and an optimized neural network for the compensation of nongeometric errors. Complexity. 2020;2020(1):5896813.
[55] Wang J, Chen H. BSAS: Beetle swarm antennae search algorithm for optimization problems. arXiv preprint arXiv:1807.10470. 2018.
[56] Sun T, Liu C, Lian B, Wang P, Song Y. Calibration for precision kinematic control of an articulated serial robot. IEEE Trans Ind Electron. 2020;68(7):6000-6009.
[57] Fan C, Zhao G, Zhao J, Zhang L, Sun L. Calibration of a parallel mechanism in a serial-parallel polishing machine tool based on genetic algorithm. Int J Adv Manuf Technol. 2015;81:27-37.
[58] Deng X, Ge L, Li R, Liu Z. Research on the kinematic parameter calibration method of industrial robot based on LM and PF algorithm. In: Proc Chin Control Decis Conf. IEEE; 2020:2198-2203.