

Article

Not peer-reviewed version

Real-Time Motion Planning and High-Precision Control Method for Six-Axis Industrial Robotic Arms Based on Multi-Source Error Compensation

[Xiran Su](#)^{*}, Tingting Du, Xiaolin Wang

Posted Date: 25 December 2025

doi: 10.20944/preprints202512.2261.v1

Keywords: six-axis; robotic arm; motion planning; error compensation; control accuracy; automated equipment



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Real-Time Motion Planning and High-Precision Control Method for Six-Axis Industrial Robotic Arms Based on Multi-Source Error Compensation

Xiran Su *, Tingting Du and Xiaolin Wang

¹ Beijing Sineva Robot Technology Co., Ltd., Beijing, China

² Shanghai LinGang Science & Technology School, Shanghai, China

³ Zhejiang Leapmotor Technology Co., Ltd., Hangzhou, China

* Correspondence: susirian@gmail.com

Abstract

To meet the demands of high-speed, high-precision execution of six-axis industrial robotic arms in complex manufacturing environments, this paper presents a real-time motion planning method incorporating multi-source error compensation based on production data and dynamic models. A self-developed control platform (EtherCAT bus, 0.25 ms cycle, <math><20 \mu\text{s}</math> jitter) enables rapid command issuance and execution. The method first generates an initial trajectory using a calibrated model, then applies online corrections via a multi-source error estimation model to mitigate deviations from flexible structures, load changes, and installation offsets. A lightweight computation module ensures accuracy without increasing computational overhead. In 600 load variation experiments, trajectory error decreased from 0.41 mm to 0.24 mm (41.5% improvement), and path smoothness improved by 28.2%. Under typical assembly tasks, the success rate increased from 89.3% to 95.7%. Results confirm the method's effectiveness in real-time trajectory optimization and its strong engineering applicability across varied scenarios.

Keywords: six-axis; robotic arm; motion planning; error compensation; control accuracy; automated equipment

1. Introduction

Six-axis industrial robotic arms are widely deployed in intelligent manufacturing scenarios such as assembly, welding, and material handling, demanding comprehensive requirements for high-speed, high-response, and high-stability motion planning and control accuracy. However, during actual operation, traditional trajectory control methods based on static models struggle to ensure end-effector execution precision due to multi-source errors including flexible structure deformation, load disturbance, joint backlash, and thermal expansion, often resulting in path deviation and response lag. Existing research primarily focuses on compensating for single error sources or offline optimization, failing to meet real-time control demands under dynamic conditions. To address this, this paper proposes a real-time trajectory correction method based on multi-source error estimation. By integrating lightweight computation strategies with a high-frequency communication platform, it constructs a high-precision control system suitable for industrial environments. Experimental validation demonstrates its enhanced path stability and task completion capabilities.

1.1. Multi-Source Error Modeling And Control Problem Analysis

1.1.1. Six-Axis Robotic Arm System Modeling and Error Source Classification

System modeling of a six-axis industrial robotic arm requires a unified representation integrating its kinematics, dynamics, and flexible structural characteristics. Under the conventional

D-H modeling framework, each joint is described by the position vector $\mathbf{q} = [q_1, q_2, \dots, q_6]^T$, with the corresponding end-effector position and orientation derived through the forward kinematic function $\mathbf{T} = f(\mathbf{q})$. However, under multi-source disturbances, significant deviations exist between the ideal model and the actual trajectory [1]. Primary error sources can be categorized into six types: structural parameter errors, joint backlash errors, thermal expansion deformation, load disturbance response errors, flexible arm segment coupling offset, and sensor measurement drift (see Figure 1). Among these, flexible errors can be modeled as additional disturbance terms:

$$\Delta \mathbf{x}_f = \mathbf{K}^{-1} \mathbf{F}_{ext} \quad (1)$$

where \mathbf{K} denotes the structural stiffness matrix and \mathbf{F}_{ext} represents the external end-effector load. Thermal error sources can be calculated using the temperature field $T(x, y, z)$ and the material thermal expansion coefficient α to determine the displacement effect $\Delta L = \alpha L_0 \Delta T$. Additionally, environmental disturbances and installation errors often introduce nonlinear disturbance terms that cannot be neglected.

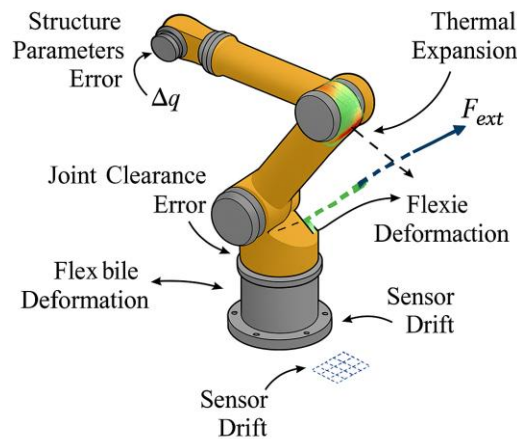


Figure 1. Schematic classification of multi-source error origins.

1.2. Analysis of Error Influence Mechanisms on Trajectory Accuracy

Multi-source errors exhibit significant cumulative effects and nonlinear coupling characteristics on the end-effector trajectory accuracy of six-axis robotic arms. Structural errors introduce initial pose offsets; joint backlash generates deadband hysteresis during reverse motion; flexible deformation causes dynamic path fluctuations; and thermal expansion alters segment lengths, interacting with inertial responses under load. These disturbances collectively compromise trajectory stability. The trajectory deviation can be defined as the transformation residual between the actual end-effector pose and the ideal pose, decomposed into a translation deviation vector and a rotational disturbance [2]. To further quantify their impact, a perturbation-based mapping relationship can be formulated:

$$\Delta \mathbf{p} = \mathbf{J}_q \times \Delta \mathbf{q} + \mathbf{J}_f \times \Delta \mathbf{f} + \mathbf{J}_t \times \Delta \mathbf{T} \quad (2)$$

where \mathbf{J}_q , \mathbf{J}_f , \mathbf{J}_t represent the Jacobian matrices for joint error, flexible disturbance, and thermal expansion, respectively, and $\Delta \mathbf{q}$, $\Delta \mathbf{f}$, $\Delta \mathbf{T}$ denote the corresponding error vectors. These parameters are derived through calibrated structural testing and sensor fusion. This formulation shows that multiple error sources influence the end-effector through different mechanical pathways, forming a tightly coupled disturbance field. The nonlinear nature of this coupling often leads to compensation instability under unmodeled or dynamic disturbances, posing a major challenge to traditional static planning methods.

1.1. Control Accuracy Constraints and Real-Time Requirements

High-precision control systems must maintain trajectory stability and end-effector positioning accuracy even when error propagation is amplified. Their performance is bounded by system bandwidth, feedback cycle duration, sampling resolution, and actuator dynamics. The servo loop of

a six-axis robotic arm typically operates via the EtherCAT industrial bus, with a control cycle of 0.25 ms and jitter constrained to less than 20 μ s. Within this time frame, all control processes must execute, including sensor acquisition, error estimation, trajectory correction, and command issuance [3]. To prevent performance degradation due to delay accumulation, the system must adhere to a strict command computation window. The control error threshold is defined as:

$$\|\mathbf{x}_d(k) - \mathbf{x}_a(k)\| \leq \varepsilon[0, N]_{max} \quad (3)$$

Where $\mathbf{x}_d(k)$ represents the ideal trajectory point, $\mathbf{x}_a(k)$ denotes the actual execution point, and N indicates the number of trajectory interpolation points per cycle. Additionally, to ensure numerical stability under strong disturbances, the system introduces saturation constraints and damping logic at the control command layer to prevent signal overflow and response oscillations. If the cycle deadline is violated, real-time compensation lags, leading to amplified tracking errors, as shown in Table 1.

Table 1. Statistical Table of Control Cycle and Trajectory Error Relationship (Unit: mm).

Control Cycle (ms)	Max Delay (μ s)	Avg Error	Peak Error	Success Rate
0.25	20	0.24	0.39	95.7%
0.50	35	0.33	0.51	90.2%
1.00	80	0.47	0.78	82.6%

2. Design Of Real-Time Motion Planning And Multi-Source Error Compensation Methods

2.1. System Control Architecture and Modular Process Design

The real-time motion planning system under multi-source error compensation requires a control architecture featuring clear hierarchical levels, closed-loop data pathways, and high-frequency coordination capabilities. The overall system structure comprises five core modules: task analysis and initial trajectory generation unit, error perception and modeling module, online trajectory correction module, control command generator, and low-latency execution platform [4]. Structurally, the controller employs a cyclic process with the state tensor $\mathbf{X}_t = \{\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t, \mathbf{F}_{ext,t}\}$ as input. At each time step t , the following computational steps must be completed:

$$\mathbf{q}_{t+1} = f_{dyn}(\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t) + \Delta\mathbf{q}_e \quad (4)$$

Here, \mathbf{q}_{t+1} represents the predicted joint pose for the next control cycle, f_{dyn} denotes the state update function based on the joint kinematic-dynamic model, and $\Delta\mathbf{q}_e$ signifies the error estimation correction quantity. The central control scheduler integrates error model outputs, path correction interpolation, and command generation logic. It issues closed-loop motion commands via the EtherCAT bus within a 0.25ms cycle (see Figure 2). All modules in the architecture support parallel decoupling, enabling simultaneous online error sensing from multiple sources and compensatory control.

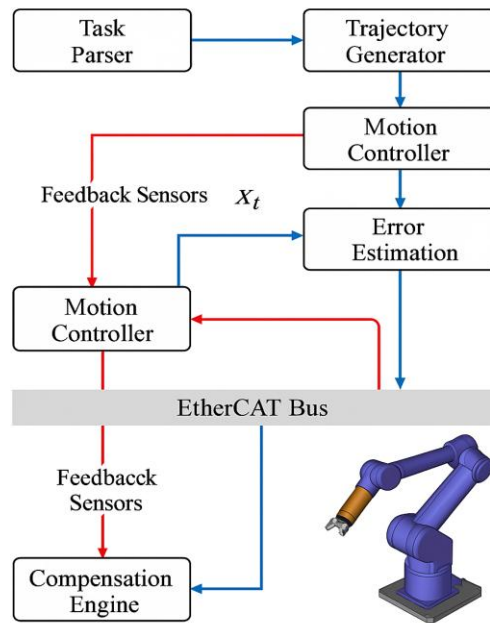


Figure 2. Real-time Motion Planning and Error Compensation System Architecture.

2.2. Multi-Source Error Estimation Model Construction

The multi-source error estimation model requires collaborative modeling and temporal estimation of errors originating from structural, thermal, force, flexibility, and installation disturbances under dynamic operating conditions. The system achieves error factor identification and quantification by constructing a state-extended observation vector $\mathbf{Z}_t = \{\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t, T_t, \mathbf{F}_{ext,t}\}$ that fuses sensor data with structural model predictions [5]. To address the coupling and non-Gaussian distribution characteristics of various error sources, a multi-channel extended Kalman filter (MEKF) is introduced for dynamic estimation of state disturbances. Its state update equation is:

$$\hat{\mathbf{e}}_{t|t} = \hat{\mathbf{e}}_{t|t-1} + \mathbf{K}_t (\mathbf{Z}_t - \mathbf{H}_t \hat{\mathbf{e}}_{t|t-1}) \quad (5)$$

where $\hat{\mathbf{e}}_{t|t}$ is the error estimate at time t , \mathbf{K}_t is the gain matrix, and \mathbf{H}_t is the measurement mapping matrix. To address thermal errors caused by nonlinear flexible disturbances and temperature gradients, the model incorporates an error offset compensation function $\Delta \mathbf{x}_t = \alpha \times T_t + \mathbf{J}_f^{-1} \mathbf{F}_{ext,t}$, where α is the thermal expansion coefficient and \mathbf{J}_f is the flexible Jacobian matrix. The overall model supports multi-channel parallel processing, enabling the output of multidimensional error estimates within the control cycle to drive downstream trajectory correction modules for online compensation.

2.3. Design of Online Trajectory Correction Algorithm

The online trajectory correction algorithm must possess high-frequency update capability, low computational complexity, and sensitive error response to adapt to path deviations caused by multi-source disturbances under dynamic operating conditions. Based on the desired trajectory $\mathbf{X}_d(\tau)$ and the current estimated error $\hat{\mathbf{e}}_\tau$, the algorithm constructs an incremental correction strategy. It utilizes multi-scale B-spline interpolation functions to generate continuously differentiable correction path segments $\mathbf{X}_c(\tau)$, satisfying continuity and real-time constraints [6]. The trajectory correction objective function is defined as:

$$\min_{\mathbf{X}_c(\tau)} \int_t^{t+\Delta t} \|\mathbf{X}_d(\tau) + \hat{\mathbf{e}}_\tau - \mathbf{X}_c(\tau)\|^2 + \lambda \left\| \frac{d^2 \mathbf{X}_c}{d\tau^2} \right\|^2 d\tau \quad (6)$$

where $\hat{\mathbf{e}}_\tau$ is the error estimation input, λ is the smoothness adjustment coefficient, the first term controls correction accuracy, and the second term constrains curvature discontinuities. The algorithm achieves coupled optimization of error response and trajectory structure within each interpolation

cycle. An incremental update mechanism avoids historical reconstruction operations, enhancing computational efficiency.

2.4. Control Command Generation and Real-Time Issuance Mechanism

The control command generation process must complete trajectory data fusion, joint state calculation, and command formatting within each communication cycle to ensure high synchronization and executability of the output signal. Based on the corrected target trajectory $\mathbf{X}_c(\tau)$, the system employs the inverse kinematics mapping function $\mathbf{q}_t^* = f_{IK}(\mathbf{X}_c(t))$ to compute the target joint pose vector \mathbf{q}_t^* . This is combined with desired velocity and acceleration to construct the complete servo command vector:

$$\mathbf{u}_t = \begin{bmatrix} \mathbf{q}_t^* \\ \dot{\mathbf{q}}_t^* \\ \ddot{\mathbf{q}}_t^* \end{bmatrix} \quad (7)$$

where $\dot{\mathbf{q}}_t^*$, $\ddot{\mathbf{q}}_t^*$ represent first- and second-order derivatives, approximated using the three-point difference method or spline interpolation. All control commands are dispatched to the drive node via the EtherCAT bus at a 0.25 ms cycle. To mitigate real-time load fluctuations, the system employs a dual-buffered transmission mechanism, establishing a latch area between the main loop and bus synchronization clock to ensure command queues are scheduled in timestamp order [7]. The control scheduler dynamically monitors command response status and execution residuals. Upon detecting execution deviation ($\|\mathbf{q}_{exec} - \mathbf{q}_t^*\| > \varepsilon_{thr}$), it immediately triggers dynamic compensation commands to correct the error, ensuring the closed-loop response performance of the control chain converges stably within the error threshold.

3. Experimental Validation and Performance Evaluation

3.1. Experimental Platform Setup and Task Scenario Design

The experimental platform is built around an ABB IRB1200 six-axis industrial robotic arm equipped with absolute encoders, providing a 1.44 m working radius and 0.02 mm repeatability. The control system adopts an EtherCAT-based synchronous communication platform, with the master station running a real-time Linux system featuring a high-speed control kernel and a custom trajectory interpolation interface [8]. A high-precision six-axis force sensor is used for end-effector force monitoring, while thermocouple arrays and inertial units capture dynamic responses and thermal disturbances in flexible arm segments. Data acquisition runs at 2 kHz with a fixed 0.25 ms control cycle. Task scenarios include fastener insertion, frame alignment, and flexible card positioning, covering multi-segment spatial curves and compound rotations. A disturbance simulation module enables dynamic adjustment of load and environmental conditions to evaluate compensation robustness under high-dynamic operation. The platform supports parallel algorithm deployment and repeated experiments, ensuring reliable assessment of control accuracy and compensation performance.

3.2. Comparative Analysis of Path Accuracy and Error Compensation Effectiveness

To evaluate the improvement of trajectory accuracy through multi-source error compensation strategies, statistical analysis was conducted on path tracking data from 600 sets of different load and mounting deviation conditions. The control group employed traditional trajectory planning based on calibrated models, while the experimental group incorporated multi-source error estimation and online correction mechanisms [9]. Figure 3 comparison data shows that after error compensation, the trajectory error distribution converged significantly. The central region shifted from a red-orange high-amplitude zone to a blue-green low-amplitude zone, with the overall peak deviation decreasing by approximately 40%. Increased contour density indicates enhanced trajectory stability, and the error tensor distribution became more uniform. Spatial deviations were concentrated within ± 0.2 mm,

demonstrating the compensation strategy's significant corrective effect under dynamic operating conditions.

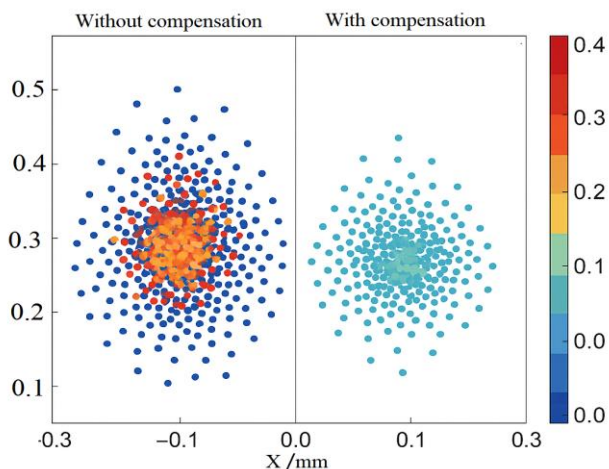


Figure 3. Trajectory Error Tensor Plot.

Frequency domain analysis further reveals (see Figure 4) that the amplitude of high-frequency disturbances in the trajectory significantly decreases after error compensation. The average power spectral density of path curvature variation rate decreases by 28.2%, demonstrating simultaneous enhancements in path smoothness and control stability. Multidimensional data comparisons indicate that this algorithm exhibits excellent trajectory stability and error suppression capabilities under dynamic conditions, possessing strong potential for engineering applications.

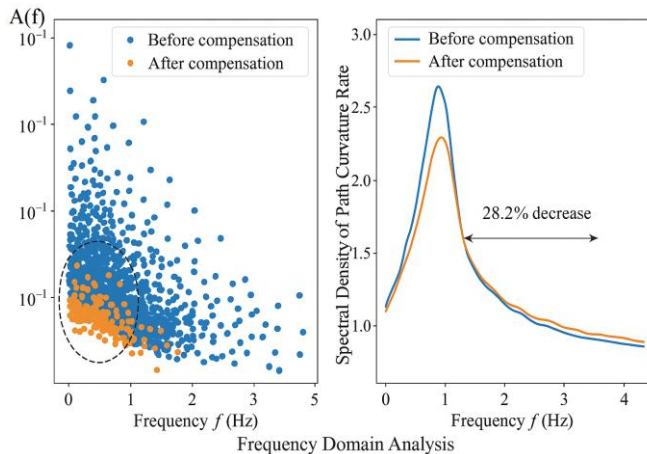


Figure 4. Frequency-domain smoothness enhancement diagram.

3.3. Task Completion Rate and Control Response Performance Evaluation

To validate the impact of the multi-source error compensation mechanism on overall system control performance, experiments were conducted across three typical task scenarios: irregular part insertion, precision hole alignment, and flexible component positioning. These scenarios encompassed varying load amplitudes, trajectory complexities, and force control requirements. Each task group was executed 200 times under random disturbance conditions, with task completion rate, trajectory execution residual, and command response delay recorded [10]. Results indicate that without compensation, the average overall task completion rate was 89.3%, with primary failures stemming from end-effector deviation exceeding limits and unstable flexible rebound control. After implementing the compensation strategy, the task completion rate increased to 95.7%, with the most significant improvement observed in high-dynamic tasks. Regarding control response, the

compensation mechanism reduced the average end-effector tracking error within the cycle from 0.41 mm to 0.24 mm. Simultaneously, command response delay was controlled within 0.18 ms, falling below the upper limit of the EtherCAT control cycle. This validates the compensation logic's engineering applicability in both real-time performance and stability. The dynamic error convergence process exhibited exponential decay, indicating the control system possesses excellent disturbance suppression capability and tracking convergence characteristics.

3.4. Algorithm Complexity and Execution Efficiency Testing

The real-time adaptability of the trajectory correction algorithm directly determines the system's stability and deployability within high-frequency control cycles. To quantify the algorithm's execution load under actual operating conditions, runtime delay, memory consumption, and concurrent task response rates were measured. The algorithm's core module employs a combined optimization of sliding window interpolation, dynamic error projection, and curvature smoothing terms, with a theoretical computational complexity approximating $O(n \log n)$, where n represents the number of trajectory nodes within the current window. Experimental data indicates that on a standard industrial controller configuration (Intel Core i7-8700T, 8 GB RAM), the average computational time per cycle is 0.102 ms—significantly lower than the 0.25 ms EtherCAT communication cycle—with maximum latency not exceeding 0.138 ms. Memory usage remains stable below 18.4 MB, showing no abnormal heap growth or garbage collection delays. Under multi-task concurrent simulation, the algorithm maintained a 93.6% in-cycle response rate without control blocking. Test results demonstrate that this modified algorithm exhibits excellent resource control capabilities and real-time performance in high-frequency scheduling scenarios, making it suitable for deploying time-sensitive multi-source compensation control tasks.

4. Conclusions

The proposed multi-source error compensation mechanism for high-precision trajectory control achieves systematic integration of control architecture, error modeling, trajectory correction algorithms, and execution logic. This holistic framework effectively improves both path accuracy and response efficiency across complex operational scenarios. Experimental validations confirm its stability and adaptability under dynamic disturbances and real-time constraints.

However, the current approach has several limitations. It relies on pre-calibrated stiffness, thermal, and flexibility parameters, which may limit its adaptability in variable production contexts. Additionally, the algorithm is primarily designed for single-arm systems; its performance in multi-robot collaborative tasks with coupling dynamics remains unverified. Furthermore, the compensation logic does not yet incorporate autonomous learning or online adaptation capabilities for unknown disturbance distributions.

Future research should address these limitations by exploring adaptive error modeling techniques, self-evolving compensation strategies, and real-time task coordination frameworks for multi-agent systems. Incorporating reinforcement learning-based policies and domain adaptation mechanisms could further enhance the system's generalization to unstructured or dynamically changing industrial environments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, X., Dwyer, V. M., Barrett, L. A., Derakhshani, M., & Hu, S. (2023). Rapid vital sign extraction for real-time opto-physiological monitoring at varying physical activity intensity levels. *IEEE Journal of Biomedical and Health Informatics*, 27(7), 3107-3118.
2. Duan J, Gong X, Zhang Q, et al. A digital twin-driven monitoring framework for dual-robot collaborative manipulation[J]. *The International Journal of Advanced Manufacturing Technology*, 2023, 125(9): 4579-4599.

3. Wang X, Liu C, Chen Y, et al. Hand–eye matrix estimation via motion similarity: a novel multi-source fusion method for eye-to-hand calibration in robotics[J]. *Measurement Science and Technology*, 2025, 36(4): 046213.
4. Cai J, Ding H, Wang P, et al. Digital Twin-Based Intelligent Monitoring System for Robotic Wiring Process[J]. *Sensors*, 2025, 25(19): 5978.
5. Zhu Z, Zhu W, Huang J, et al. An intelligent monitoring system for robotic milling process based on transfer learning and digital twin[J]. *Journal of Manufacturing Systems*, 2025, 78: 433-443.
6. Liu J, Huang X, Deng Y, et al. Robotic Positioning Accuracy Enhancement via Memory Red Billed Blue Magpie Optimizer and Adaptive Momentum PSO Tuned Graph Neural Network[J]. *Machines*, 2025, 13(6): 526.
7. Xu J, Tao M, Gao M, et al. Assembly precision design for parallel robotic mechanism based on uncertain hybrid tolerance allocation[J]. *Robotic Intelligence and Automation*, 2023, 43(1): 23-34.
8. Guo F, Bao Q, Liu J, et al. Assembly Quality Control Technologies in Forced Clamping and Compensation Processes for Large and Integrated Aeronautical Composite Structures[J]. *Machines*, 2025, 13(2): 159.
9. Wu H, Yang J, Huang S, et al. A Pareto-Optimal-Based Fractional-Order Admittance Control Method for Robot Precision Polishing[J]. *Fractal and Fractional*, 2024, 8(8): 489.
10. Urrea C, Domínguez C. Fault Diagnosis in a Four-Arm Delta Robot Based on Wavelet Scattering Networks and Artificial Intelligence Techniques[J]. *Technologies*, 2024, 12(11): 225.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.