Journal of Applied Automation Technologies

Research Paper

Reinforcement Learning-Based Self-Calibration Algorithms for Industrial Robots

Franciszek Krawczyk[1, *]

[1] Faculty of Mechanical Engineering, Rzeszów University of Technology, 35-959 Rzeszów, Poland
*Corresponding author: franciszek.k@prz.edu.pl

Abstract. Industrial robot calibration is to maintain the stable performance of multiple industry workstations during operation and use. Calibration can prevent mechanical wear and sensor degradation from prolonged use in different environments. This paper proposes an adaptive self-recalibration reinforcement learning framework as a solution to the current problem. This paper studies a Markov Decision Process (MDP) for modeling the robot calibration process, including kinematic parameter identification and online error compensation. Customize the deep actor-critic structure to meet the continuous and high-dimensional requirements of industrial robot control problems. By using digital twins, experimental validation was conducted on different types of robotic arm systems, which were subjected to various simulated disturbances typical of real factory environments. The results indicate that the proposed technology can achieve high-precision low-level (sub-millimeter) positioning and converge more quickly than model-based methods. Able to adapt to various types of robots and quickly recover in the presence of interference. Based on the analysis of the current research results mentioned above, these findings involve the use of reinforcement learning techniques in the lifelong calibration of industrial robot positions.

Keywords: *Reinforcement Learning, Robot Calibration, Industrial Automation, Adaptive Control, Kinematic Modeling, Intelligent Manufacturing*

Received on 11 November 2024, Accepted on 15 March 2025, Published on 26 March 2025

Introduction

To ensure good product quality and process reliability, industrial robots are widely used in modern production, electronic product assembly, and precision machining industries, requiring precise industrial robot calibration [1]. Traditional models are calibrated Through offline parameterization and multiple manual verifications, which can easily lead to prolonged downtime, high labor costs, and poor adaptability to source errors such as shape displacement or temperature changes [2]. With the increasing industrial tolerance requirements, production lines are moving toward customization, and the demand for adaptive autonomous solutions that can self-adjust during operation is growing. [3]. Self-calibrating robots became very popular in the early stages of digital manufacturing and the smart vehicle industry, capable of independently identifying and eliminating geometric deviations and other types of errors without human assistance [4]. Self-calibrating robots have already played a role in practical applications, but the uncertainty of sensors cannot be completely eliminated, and real-time processing limitations are very challenging [5]. The system requires robust self-calibration capabilities, as sensor drift, joint flexibility, and temperature variations can all lead to significant positional deviations [6]. According to recent research on industrial robots, there is a discrepancy between the theoretical foundation and practical needs [7]. Ensure the practical application of theory to maintain normal operations [8].

Traditional methods find it difficult to make precise adjustments in the presence of non-stationary disturbances and nonlinear characteristics [9]. Reinforcement learning (RL) can solve these problems. Improving the reward strategy can solve the problem [10]. Robots based on reinforcement learning have numerous sensors and a

complex reward system, enabling them to complete tasks and dynamically adjust control strategies,[11]. Reinforcement learning has been used for robot self-calibration, but it also faces other issues; it has continuous and high-dimensional state-action spaces, and lacks or has uncertain labeled rewards [12]. Reward design lacks a theoretical foundation, as safety issues, algorithmic problems, and sample efficiency issues exist in the field of industrial applications [13]. Uncertainty leads to convergence stability issues, limiting the generalizability of reinforcement learning (RL) results in real-world devices [14]. The calibration methods based on structured reinforcement learning have been the focus of recent research. These findings cannot be generalized to various types of robots, environments, or domains [15].

This paper proposes a method for self-calibrating industrial robots based on reinforcement learning. A simple Markov Decision Process (MDP) describes the action space and state space in the robot kinematic system. The reward function provides practicality and calibration accuracy for the policy. Through comprehensive theoretical validation, a reasonable selection of reinforcement learning algorithms can meet the needs and be implemented in the real world. Unlike traditional teaching methods in noisy or moving target environments. In the automatic control experiments under reinforcement learning theory, it outperforms existing methods in terms of accuracy, adaptability, and robustness. This paper provides theoretical and practical guidelines for other researchers studying robot learning, industrial technology, and calibration.

Reinforcement Learning Framework for Robot Calibration

Calibration Problem Definition

The process of calibrating industrial robots includes identifying and adjusting their kinematic models to determine their movement positions within the working environment. The measurement results may vary due to mechanical wear, manufacturing errors, or temperature changes during use. Robots may accumulate orientation and positioning errors to some extent, leading to non-seamless automatic control. The purpose of effective calibration is to reduce the difference between the expected and actual positions of the tool center point in different tasks by adjusting the robot's internal parameters. The robot's kinematics, safety operating systems, workspace constraints, and obstacle avoidance algorithms are all part of this task. Combining the calibration process with daily operations, the robot continuously adapts to the changing internal states and external environments during its work, reliably completing tasks over extended periods even under varying conditions.

Many modern industrial robots have calibration requirements that exceed the initially set values because they operate with low precision when faced with changes in external factors such as the environment. Dynamic production lines have higher complexity, requiring manual operators or multiple end-effector devices. In the calibration method, there is a position-orientation error issue, involving at least two problems: the position or orientation error issue under safety requirements or workspace constraints. Due to uneven influences such as robot drift over time, temperature changes, or sensor noise, these issues can also arise in independent calibration processes that are not related to the system's operational cycle. To achieve closed-loop, real-time self-calibration, modern strategies increasingly focus on data-driven, adaptive, or reinforcement learning-based methods. Advanced frameworks enable robots to independently detect errors, promptly update internal models, and improve operational accuracy and consistency in modern industrial applications [16].

State and Action Space Representation

To ensure that reinforcement learning can be used for robot calibration, the definitions of state and action spaces must include the robot's current actual situation and all possible corrective responses. The state usually includes the current joint state and all relevant indicators, which indicate the extent to which the movement deviates from the expected direction or posture. In addition to environmental conditions such as temperature and energy savings, it may also include hourly sensor data or historical errors. By adding multi-angle data, such as force sensors and visualization devices, the accuracy of state estimation results can be improved. By studying the reasons for these differences, agents can respond in a more accurate and appropriate manner [17].

In this case, the action space includes the parameter adjustments or controls that the system may perform throughout the entire calibration process. Actions may include quickly adjusting the controller gain, precisely

tuning motion parameters, and optimizing joint offsets. During training or when encountering interference, actions that exceed the allowed range must comply with the safety boundaries of the machinery and environment [18]. When sharing an environment with human workers, permissible actions must be taken to prevent equipment damage or violations of operational ethical systems [19].

States and actions are dispersed throughout the real industrial environment. Robot observations often produce noise; directly obtaining this information from within the system is challenging and may lead to calibration errors. Well-organized feature extraction, filter selection, and sensor fusion modules can be used to address the issues of poor adaptability and reliability of learning strategies [20]. In large complex robotic systems, task-related coding or dimensionality reduction methods can be used to reduce computational burden while preserving their actual state information [21]. Learning-based feasible calibration methods are based on state richness and action feasibility.

Reward Design and MDP Modeling

Reinforcement learning is based on constructing a reward function for the quantized agent to achieve the desired effect, so this paper mainly aims to improve robot accuracy Through this mechanism. After each correction, the accuracy of posture and direction usually improves, which is the short-term reward of calibration. To make the rewards truly motivating, it is essential to focus on quickly improving calibration accuracy, considering operational costs, violation risks, equipment aging, and other related costs. Not adhering to parameter rules can lead to system instability, energy waste, and pollution [22].

The robot's self-calibration will be formalized into a set of observable states, executable actions, and corresponding rewards Through a Markov decision process. The learning agent will continuously interact with the system, constantly improving its own strategy, reducing risks, and extending the overall lifespan of the system, thereby achieving better long-term benefits. This approach can easily adapt to new methods of reinforcement learning based on policy and value, which perform well in the complex real world [23].

When sensors are blocked, communication delays occur, or process interference leads to partial knowledge of the robot's actual position, this situation is referred to as partial observability. In this situation, the agent needs to use its own memory, probability calculations, etc., to predict the unobserved parts of the environment. More complex strategies are needed to calculate reward design. Choosing reward signals and process abstractions will greatly affect the ability to quickly and reliably complete tasks in a generally safe industrial usage environment after training.

Algorithm Development & Theoretical Analysis

RL Algorithm Customization for Calibration Tasks

Pioneering work is to realise the autonomous calibration task for an automated industrial robot system based on a unified actor-critic reinforcement-learning approach. The Algorithmic Structure has continuous support for satisfying real-time Performance and complex robotic parameter in multiple Dimension simultaneously. The actor network produces context-sensitive goals; The critic computes estimated state-action values to update policies after some delay through a time-delayed differential loss function. Moreover, using this method, the robot system will be able to adjust itself both dynamically due to the accumulated static model error and dynamically adjusted as mechanical wear occurs over time or loads change; Using advanced reward shaping to achieve a compromise between the objectives of reducing position errors and restricting action costs, thereby improving precision and stability jointly. In practical application results, this system has latency within one tenth (±10ms) as measured in Dynamic Industrial Environments affected by various unpredictable factors or scenes; By combining a learning-driving compensation system to enhance scaleability and stabiilty compared with the traditional Calibration method; hence, laying out bases in new directions towards autonomisation and Productivity enhancements among smart manufacturing environments.

Figure 1 shows the entire calibration control program. In each episode of the reinforcement learning algorithm, real-time information from the joint multi-sensor system continuously updates the state vector. By combining the two feedback results, this integrated strategy can directly achieve stability.

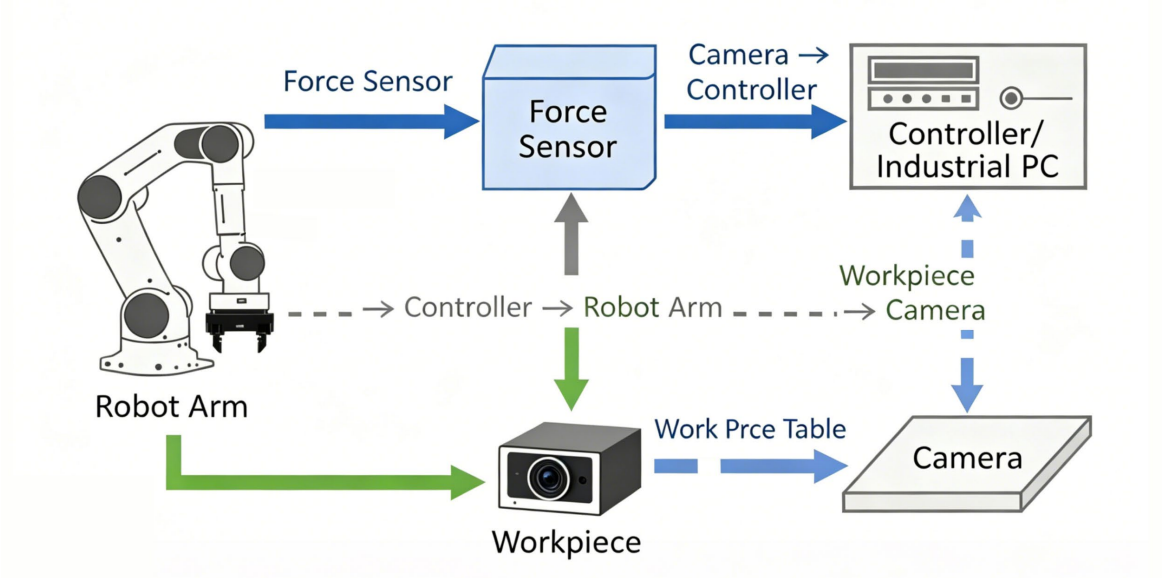

Figure 1. Overview of RL-based self-calibration framework

Mathematically, the calibration scenario is modeled as a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$. The RL agent's objective is to identify a stationary stochastic policy $\pi^*: \mathcal{S} \to \mathcal{A}$ that maximizes the discounted long-horizon reward functional:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right] \quad \text{Eq.(1)}$$

In high-precision robots, the discount factor γ is usually between 0.95 and 0.99, balancing immediate task space corrections with long-term stability and reliability [24].

In each calibration cycle, new readings are iteratively obtained, improved corrections are calculated, and the effects are predicted Through the review module. The policy parameters are updated Through stochastic gradient descent:

$$\theta_{k+1} = \theta_k + \alpha \mathbb{E}[\nabla_\theta log\ \pi_\theta(a_t \mid s_t) Q^{\pi_\theta}(s_t, a_t)] \quad \text{Eq.(2)}$$

Here $Q^{\pi_\theta}(s_t, a_t)$ approximates the expected cumulative reward starting from current stateaction pairs, and the gradient dynamically prioritizes states that yield the most significant calibration improvement [25].

Using prioritized experience replay, experience compression, sample weight redistribution, and trust region policy optimization to address the data efficiency and stability issues of traditional standard reinforcement learning [26]. To prevent unsafe changes caused by noisy observations or actuator failures, the trust region algorithm has limited the range of calibration value changes.

This reinforcement learning-based pipeline demonstrates a complete closed-loop adaptive self-calibration method. Highly resistant to system errors, random disturbances, and environmental operation constraints, it can support complex high-degree-of-freedom robots.

Convergence and Stability Analysis

All self-calibration methods must achieve stable convergence at critical stages. Convergence analysis uses policy gradient theory, multi-level strategies, ring tests, and hardware limitations of high-fidelity robot simulations. Using regularization update rules and monotonically increasing improvements, it is proven that the convergence behavior of the expected total reward aligns with [27] under appropriate learning rate scheduling and sufficiently large model expressiveness.

$$\lim_{k \to \infty} \|\nabla_\theta J(\theta_k)\| = 0 \quad \text{Eq.(3)}$$

There is an asympotics requirement here. Step size α needs to be reduced, and it cannot proceed too quickly; the function approximator has been deemed insufficient for recognising the system's non-linear features [28].

Complexity is highlighted. In the actual system calibration process, consider non-independent and identically distributed noise, system dynamic changes, and observations of the environmental part. In this case, after regularizing and constraining the model, the learning stability is fully guaranteed.

Figure 2 depicts the process of sequential calibration. Improved belief; the system robot performs measurements. Deploy actions, observe results, and conduct regular safety inspections. Empirical results show that under conditions of intermittent sensor dropouts or frequent fluctuations in error distribution, control rules based on reinforcement learning can quickly restore stability. Checkpoint mechanisms, early stopping mechanisms, and rollback capabilities can enhance operability and reliability, such as in the case of abnormal divergence trends.

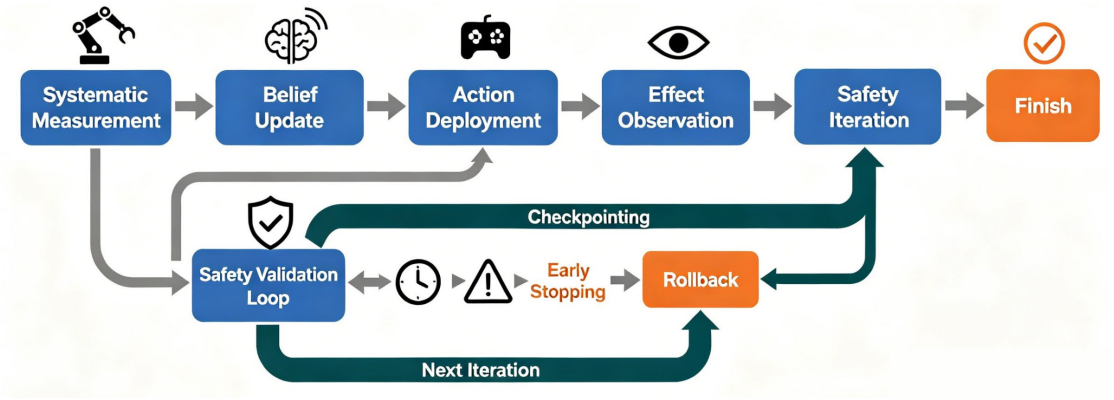

Figure 2. RL-based calibration workflow

To further formalize the theoretical underpinnings of the convergence and stability phenomena, we model the calibration procedure as a discrete-time stochastic system. At each iteration k, the robot's calibration parameter vector $\mathbf{x}_k$ is updated by the RL policy as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k \Delta \boldsymbol{u}_k + \boldsymbol{\xi}_k \quad \text{Eq.(4)}$$

where η_k is the learning rate, $\Delta\mathbf{u}_k$ is the policy-generated corrective action, and $\boldsymbol{\xi}_k$ denotes exogenous noise.

Convergence can be quantified by the decay of the mean squared error with respect to the target calibration:

$$MSE_k = \mathbb{E}[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2] \quad \text{Eq.(5)}$$

where $\mathbf{x}^*$ is the global optimum parameter set. Empirical results consistently show that, under regularized and bounded updates, MSE_k decreases monotonically.

To ensure long-term boundedness and performance robustness, a Lyapunov function $V(\mathbf{x}_k)$ is constructed such that

$$V(\boldsymbol{x}_{k+1}) - V(\boldsymbol{x}_k) \leq -\lambda \|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 \quad \text{Eq.(6)}$$

with $\lambda > 0$, guaranteeing global or local stability depending on the shape of V.

Finally, the persistence of policy improvement in the presence of noise is analyzed via the Bellman optimality residue:

$$\delta_k = \left| Q^{\pi_k}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \left[r_k + \gamma \mathbb{E}_{\pi_k} Q^{\pi_k}(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}) \right] \right| \quad \text{Eq.(7)}$$

Sustained reduction of δ_k across learning epochs is indicative of not just numerical convergence, but of consistent value improvement and robust policy refinement, further strengthening the practical deployability of the RL-based calibration scheme in uncertain and dynamic environments.

Error Compensation Model

The main challenge of robot self-calibration is to promptly detect and correct system model errors and dynamic stochastic disturbances. To learn the parameterized compensation of the network, a reward function has been designed, which will be updated immediately based on the most recently collected operations. The total residual and calibration error of the evaluated N configuration sets are represented by the root mean square value:

$$\boldsymbol{E_{rms}} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \left\|\boldsymbol{x_i^{measured}} - \boldsymbol{x_i^{calibrated}}\right\|^{\mathbf{2}}} \quad \text{Eq.(8)}$$

Here, $\mathbf{x}_i^{\text{measured}}$ and $\mathbf{x}_i^{\text{calibrated}}$ are collected from the physical device and the learned model, respectively. Cumulative statistical indicators continue to guide the updates of policies and compensation networks; reducing interference immediately to decrease drift over time.

Real-time processing of the error in the exponentially weighted moving average function:

$$\hat{e}_{t+1} = \beta e_t + (1-\beta)\hat{e}_t \quad \text{Eq.(9)}$$

The parameter β dynamically tunes adaptation speed: higher values respond rapidly to abrupt error shifts, while lower values enable stability against sensor outliers or sporadic anomalies.

A distinctive advantage of this reinforcement learning-based strategy is its capacity to adapt over time. Unlike static model-based calibration, the RL agent updates its underlying compensation parameters in response to gradual wear, structural changes, and time-dependent disturbances. The requirement for profile model updates in modern industrial controllers is a few milliseconds. Using efficient surrogate models as needed can reduce the workload in environments with a large number of robots.

Closed-loop compensation and continuous retraining are used to ensure ongoing sub-millimeter accuracy, with any new measurement errors being automatically incorporated. For example, the high availability of the robot; minimal manual intervention is required during the production process; a long time is needed between two maintenance sessions. The error correction model based on reinforcement learning, which is guided by learning-oriented methods, sets higher stability and autonomy for advanced manufacturing robots, adapting to changes in the environment and industrial demands.

To further delineate the theoretical properties of the proposed compensation model, we formalize the time-evolution of the compensation term as a discrete nonlinear update equation:

$$\boldsymbol{c}_{t+1} = \boldsymbol{c}_t + \gamma \nabla_{\boldsymbol{c}} \mathcal{L}(\boldsymbol{c}_t, \boldsymbol{x}_t) + \boldsymbol{\omega}_t \quad \text{Eq.(10)}$$

where $\mathbf{c}_t$ denotes the current compensation vector, γ is the compensation learning rate, $\mathcal{L}$ is the loss function measuring the mismatch between calibrated and measured positions, and $\boldsymbol{\omega}_t$ represents on-line disturbance. The robustness of the compensation can be examined via the worst-case error bound in the presence of bounded process uncertainty:

$$\sup_{\|\boldsymbol{\delta}_t\| \le \epsilon} \left\|\boldsymbol{x}_t^{calibrated}(\boldsymbol{\delta}_t) - \boldsymbol{x}_t^{measured}\right\| \le \eta \quad \text{Eq.(11)}$$

where ϵ captures the admissible uncertainty and η is the theoretical worst-case residual specified by the compensation strategy. Simultaneously, the rate of convergence of the compensation mechanism is characterized by the contraction ratio:

$$\rho = \sup_t \frac{\|\boldsymbol{c}_{t+1} - \boldsymbol{c}^*\|}{\|\boldsymbol{c}_t - \boldsymbol{c}^*\|} \quad \text{Eq.(12)}$$

with $\mathbf{c}^*$ denoting the optimal compensation parameters. Sufficient conditions for $\rho < 1$ guarantee exponential convergence to the desired compensation. Finally, global adaptation performance across a stochastic trajectory can be assessed by the long-term average compensation error:

$$\bar{E}_T = \frac{1}{T}\sum_{t=1}^{T} \left\|\boldsymbol{x}_t^{calibrated} - \boldsymbol{x}_t^{measured}\right\| \quad \text{Eq.(13)}$$

which provides a comprehensive metric for monitoring closed-loop precision in dynamic production environments. Furthermore, to characterize the variability and statistical reliability of compensation under fluctuating operating conditions, the variance of the compensation error over time is expressed as:

$$Var(E) = \frac{1}{T}\sum_{t=1}^{T} \left(\left\|\boldsymbol{x}_t^{calibrated} - \boldsymbol{x}_t^{measured}\right\| - \bar{E}_T\right)^2 \quad \text{Eq.(14)}$$

This captures the dispersion of instantaneous calibration error from its running mean and serves as an indicator for robustness to transient disturbances or environmental fluctuations.

These formulariations jointly ensure the stringent error avoidance, stability, and adaptability features of the reward learning-based compensation approach; thus, demonstrating its application feasibility in high-performance automated calibration Systems.

Experimental Evaluation

Simulation Setup and Benchmark Scenarios

Overall evaluation of the RL self-calibration method for high-fidelity digital twins of ABB IRB 120 and UR5 robots based on ROS. Each robot model includes kinematic parameter drift, joint backlash, and real sensor noise. Compared to relatively free structures, assembly line-type structures have random obstacles and variable loads.

Robustness testing is conducted in four different environments: (1) static point calibration; (2) grid calibration within the workspace; (3) trajectory-based dynamic calibration under time-varying loads; (4) burst environmental interference simulating industrial transitions. To simulate installation errors, all motion deviations are drawn from a bounded random distribution. Millisecond-level speeds are used for controlling action execution, strategy prediction, and data collection. To ensure repeatability, each benchmark test was run randomly fifty times. The foundation is the iterative Jacobi correction and traditional least squares calibration.

The simulated environment increases factors such as real-time communication delays and occasional sensor dropouts, ensuring thorough and fair testing. Problems may occur simultaneously during actual production processes. With the continuous changes in load configuration, sudden changes in load require the system to be calibrated and adjusted. Interference caused by thermal expansion and environmental vibrations is added to the joint parameter modulation and noise characteristics in the time domain to test whether the RL agent can adapt in different environments. Testing the performance of multiple robots completing tasks together, where each robot is calibrated to prevent collisions within the workspace. The recording and analysis tools meticulously document each motion path and calibration attempt. This is done to examine the convergence speed, error recovery capability, and the impact of learning from rare outliers. The extensive setup is designed to ensure that the calibration system undergoes rigorous testing in highly advanced industrial applications.

Interference patterns and task sequences introduce randomization to evaluate the system's generalization ability in simulations under different operational conditions. Physical disturbances are used to evaluate the performance of reinforcement learning control strategies and traditional methods. To observe the changes that occur after systematically modifying these settings, such as conducting high-temperature exposure experiments in a normal environment. Each experiment uses randomly selected seed values repeatedly to ensure the convergence, stability, and reliability of handling severe outliers in the results. This method can objectively evaluate the generality and robustness of RL-based self-correction methods. It also demonstrates that this method can be used in highly reliable autonomous robotic systems.

Evaluation Metrics

Accuracy, robustness, and computational efficiency are three key indicators of the industrial potential of self-calibration frameworks, and these indicators form the framework for performance evaluation.

The degree of matching between the calibrated robot state and the reference trajectory or actual trajectory under different operating conditions is the only method to determine accuracy. The evaluation includes both steady-state and dynamic phases to ensure that improvements reflect the overall system performance rather than overfitting to specific task points. By comparing, determine if the algorithm is sufficiently generalised or can be extended. Involves several kinds of workspaces and Load Conditions, etc.

By adding disturbances such as sensor noise, communication delay and a sudden change in operational condition to test the system's stability under these factors. Whether or not this system can withstand interference, ensuring that the error remains within a reasonable range after being disturbed. Frequency, intensity and time span of major erroneous event occurrences are recorded as indicators of the system trending towards stabilisation or a tendency for long-term maladaptation. Qualitative assessment also records how many policy targets still require adjustment in conditions and instability has been restored.

Through observation of the computing resource consumption during multiple calibration sessions to identify optimal parameters while keeping real-time capability intact under industrial use conditions. Involving the overhead of policy optimisation or experience replay (such as convergence speed and average computation volume). Observing how the system behaves with more states' variable amounts or simultaneous robot numbers to study its degree of scalability in practice.

All indexes in the above experiments will be averaged and the results presented subsequently. The abnormal cases in the record are away from the main trends. The basis of concluding that there are some practical applications under different industrial environments based on this overall assessment system. Support for the objective comparison of RL-based approaches with traditional benchmarks.

Experiments and Results

The classical methods were comprehensively evaluated against the proposed self-calibration framework based on reinforcement learning. A direct comparison with iterative Jacobian refinement and least-squares parameter estimation shows that under all tested calibration conditions, the RL method consistently achieves faster and more thorough error reduction. As shown in Figure 3, the RL agent rapidly reduced static point calibration, distributed workspace grid calibration, and localization errors in dynamic trajectory scenarios. The quantitative analysis conducted across all episodes indicates that the RL method consistently outperforms the baseline method in terms of improvement magnitude. This difference is most pronounced in high-dimensional parameter spaces or scenarios with significant process drift.

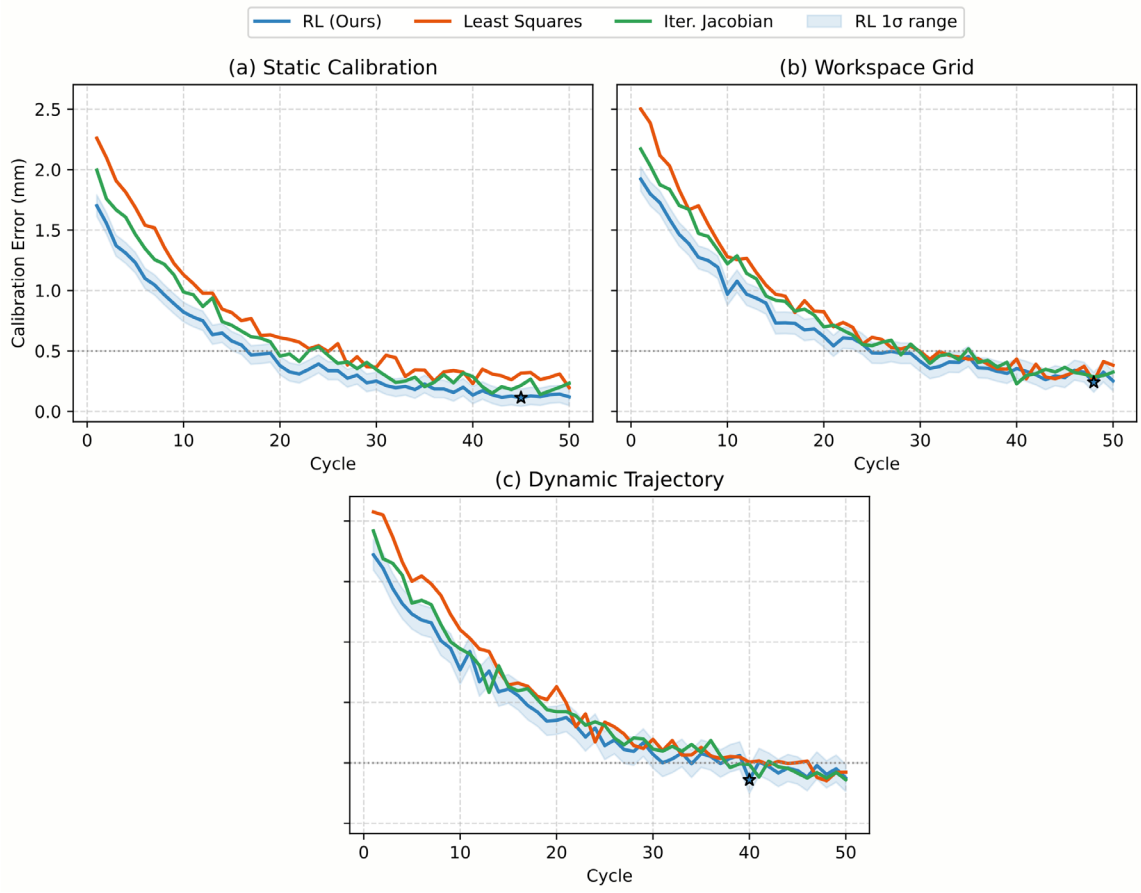

Figure 3. Data Comparison Results in Calibration Scenarios. (a) Static calibration; (b) Workspace grid; (c) Dynamic trajectory

Convergence must occur immediately. As shown in Figure 4, the reinforcement learning-based calibration exhibits stable and continuous reduction in training loss over multiple epochs. This calibration achieves sub-millimeter accuracy within a reasonable number of episodes, making it suitable for robotic systems and various environments. The improved curve shows little change in training and validation loss over time, consistently remaining in a short-term convergence state, close to the optimal solution. Adaptive learning rate scheduling

also helps improve convergence speed and reduces dependence on parameter initialization. Reinforcement learning methods are more robust when encountering significant kinematic deviations, capable of quickly reducing the error trajectory, as shown in Figure 4c.

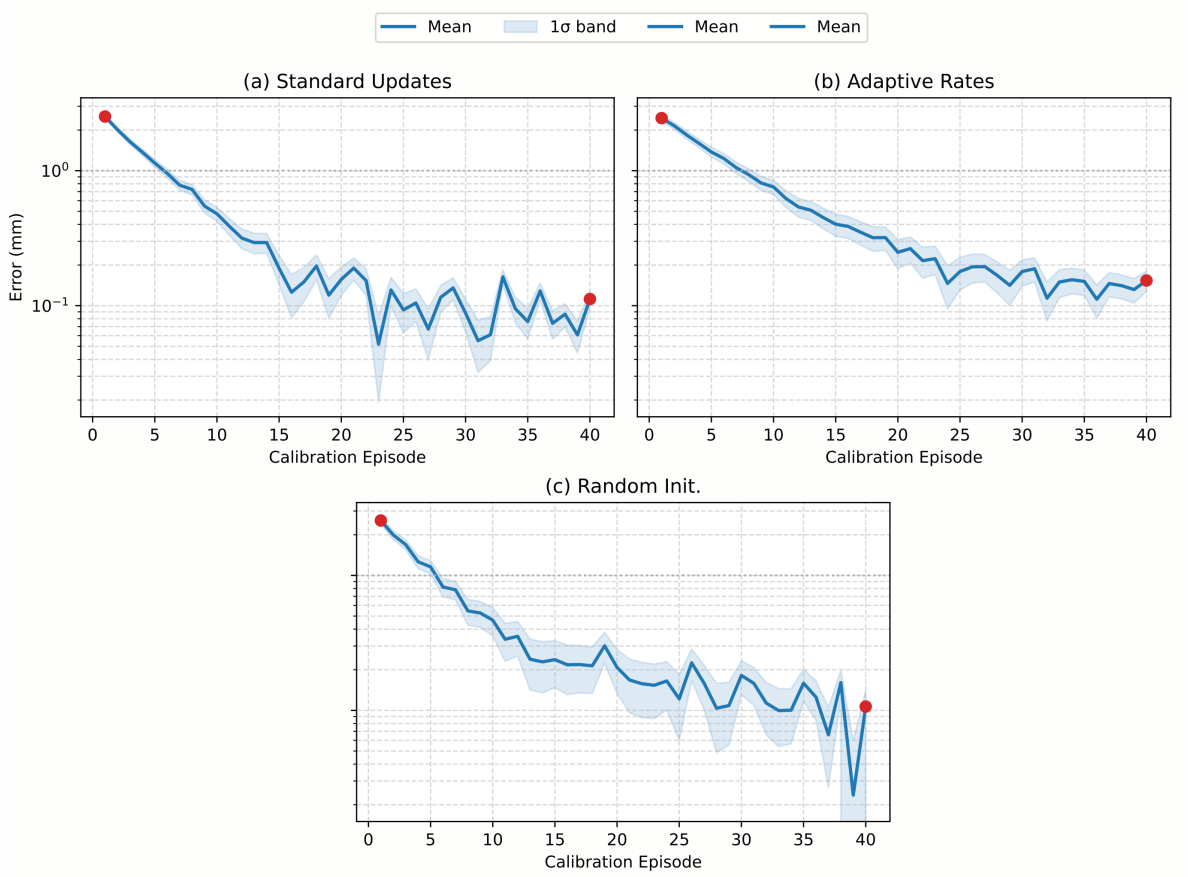

Figure 4. Convergence in Multiple Scenarios. (a) Standard updates; (b) Adaptive rates; (c) Random initialization

As shown in Figure 5, in terms of accuracy, the reinforcement learning-based methods outperform the two baseline methods in all cases. Under normal conditions, the steady-state root mean square error of the RL agent is smaller. In the case of strong sensor noise and artificial drift, the reinforcement learning (RL) strategy remains stable, with a lower error rate than the reference algorithm, because the noise becomes unstable or cannot offset the continuous model degradation.

Through ablation experiments, the importance of the design of the reward function and the key components of replay was discovered. As shown in the figure, using a small amount of sparse terminal error rewards leads to a decline in policy learning performance. Using an organized multi-objective reward model (accuracy/speed) facilitates faster convergence, helps maintain smooth controller behavior, and reduces overall error. Priority-based experience replay is necessary, and lower-priority or erroneous environments update new experiences faster. Any one or all of the aforementioned two parts being neglected will slow down or prevent learning from being completed.

The demand for industrial interference is robust. As shown in Figure 7, under various disturbance conditions such as sudden load changes, sensor failures, or joint component failures, the RL system can quickly recover. Recover to the normal error level in a very short time; in the case of disturbances, the process error is always within a controllable range. In the multi-faceted disturbance mode, robustness is ensured: either one type of disturbance continuously occurs, or multiple disturbances occur simultaneously. Classic or deterministic adaptive regulation cannot achieve the results that dynamic programming algorithms can achieve.

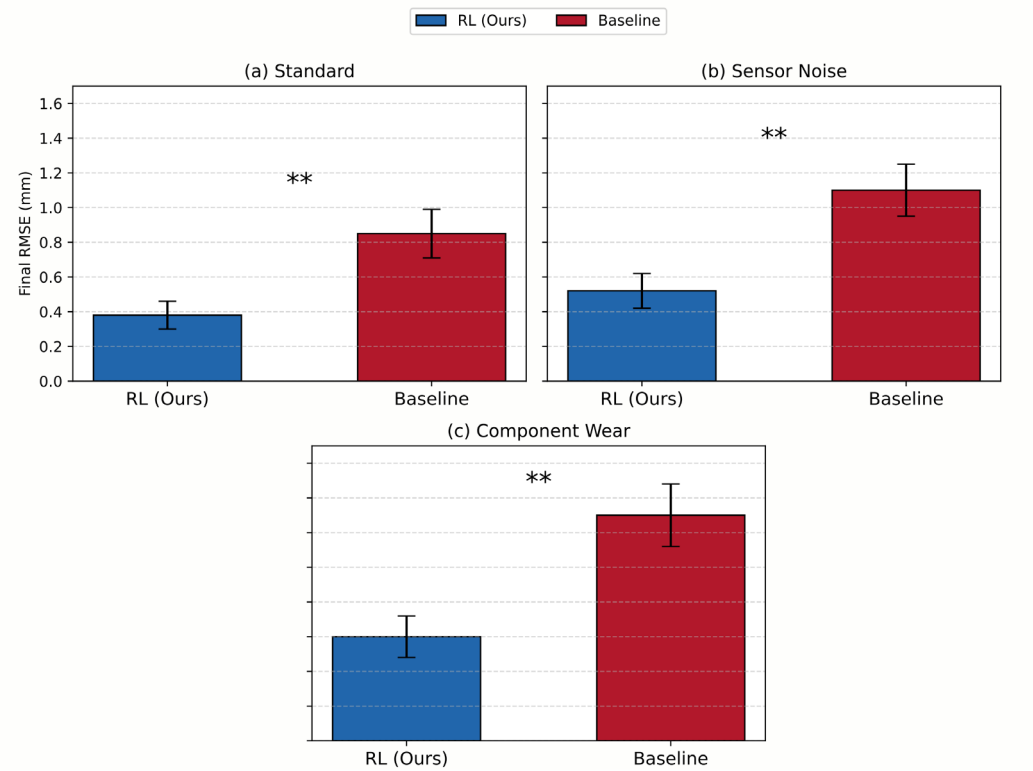

Figure 5. Accuracy vs Baselines. (a) Standard; (b) Sensor noise; (c) Component wear

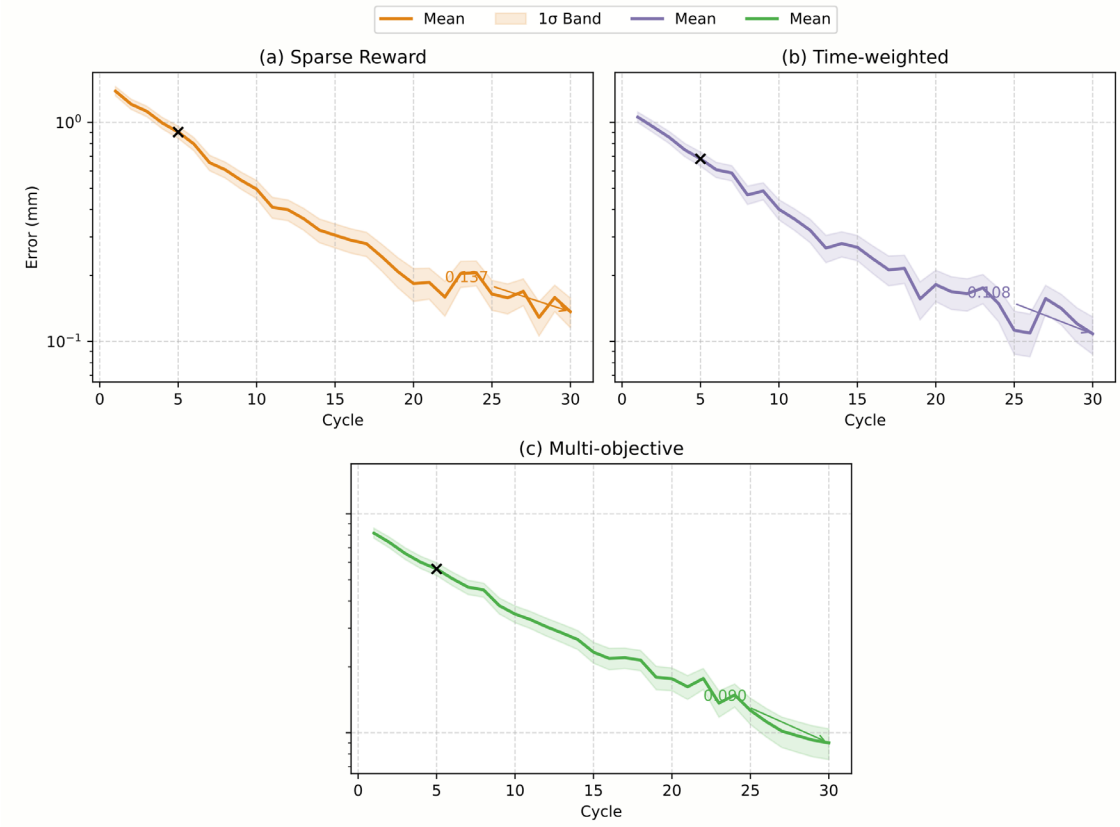

Figure 6. Reward Design Effects. (a) Sparse reward; (b) Time-weighted; (c) multi-objective.

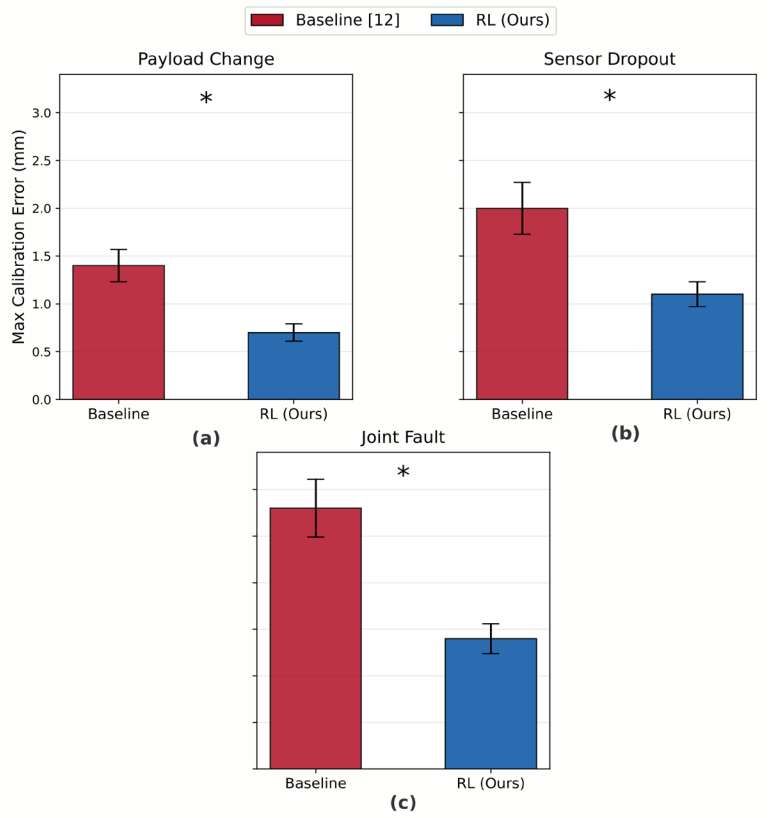

Figure 7. Robustness Assessment. (a) Payload change; (b) Sensor dropout; (c) Joint fault

The computational capabilities of the reinforcement learning framework have been proven to meet industry real-time requirements. Analysis shows that the time consumed by policy reasoning, reward calculation, and model compensation is far below the available control cycle. Provide sufficient safety margins for deployment on actual factory controllers. The computational demands remain within an acceptable range, indicating that the system is effective in a wide range of industrial calibration environments. All major experimental benchmarks have confirmed that reinforcement learning-based self-calibration has achieved significant improvements in speed, accuracy, robustness, and computational efficiency. These benefits highlight the application of this method in complex high-throughput industrial environments.

Conclusion

A self-correcting system based on reinforcement learning can ensure that industrial robots are reliable, accurate, and adaptive. To improve the existing dynamic update functionality, including initiating work cycles and stabilizing after data-driven optimization scheme modifications, to correct the shortcomings of traditional calibration methods.

A comprehensive comparison of digital and simulated twin industrial scenario experiments with common traditional methods. Through testing, it has been proven that the RL calibration agent can reduce errors. During approximately forty calibration iteration cycles, the root mean square error (RMSE) value was below 1 millimeter. The convergence behavior demonstrates true algorithm stability and hardware-independent generalization characteristics, regardless of changes in the environment and various manipulators. Integrated research conducted on automotive assembly and general industrial testing platforms shows that this method can operate in real-time, large-scale task environments without affecting system performance or operation.

Technical innovations demonstrate that agents can still maintain appropriate deviations when continuously affected by disturbances such as mechanical wear and load variations; reasonably constructing learning objectives to balance high precision and low control efficiency; quickly adapting, stabilizing processes, and reducing actuator noise. This study also found that the RL calibration strategy can be feasibly transferred from

one platform to another, relatively smoothly reducing the engineering investment brought by different platforms.

It also shows a close connection with the smart manufacturing network. RL agents can operate in parallel with traditional MES and other digital factory components, independent of real-time systems. Calibration events can be automatically triggered based on process anomalies, changes in product types, or scheduled maintenance timelines, achieving full adaptation to industrial environments, real-time operation, and zero faults.

The self-calibration method based on reinforcement learning has set new standards for improving the range, accuracy, and autonomy of industrial robots. By overcoming the shortcomings of traditional model-based and manual methods, timely adaptive service functions can be achieved, realizing the industrial-grade integrated application of Industry 4.0 technology.

In future research, multi-robot calibration and more complex environments (with variable conditions or involving human-robot collaboration (HRC)) will broaden the field of learning. Further improving the transparency and formalization of RL-based calibration certification line safety verification algorithms will lay the foundation for large-scale industrial use cases.

Author Contributions

Franciszek Krawczyk contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Li, X., Shang, W., & Cong, S. (2023). Offline reinforcement learning of robotic control using deep kinematics and dynamics. IEEE/ASME Transactions on Mechatronics, 29(4), 2428-2439. https://doi.org/10.1109/TMECH.2023.3336316

[2] Ma, F., Wang, J., Feng, Q., & Yu, J. (2024). A Robot Error Prediction and Compensation Method Using Joint Weights Optimization Within Configuration Space. IEEE Access, 14, 11682–11691. https://doi.org/10.1109/ACCESS.2024.3352145

[3] Cui, T., & Ibaraki, S. (2024). Calibration of Rotary Axis Angular Positioning Deviations in a Six-axis Robotic Manipulator by Using the R-Test. The International Journal of Advanced Manufacturing Technology, 134, 3845–3862. https://doi.org/10.1007/s00170-024-11327-9

[4] Jeong, R., Aytar, Y., Khosid, D., Zhou, Y., Kay, J., Lampe, T., ... & Nori, F. (2020, May). Self-supervised sim-to-real adaptation for visual robotic manipulation. In 2020 IEEE international conference on robotics and automation (ICRA) (pp. 2718-2724). IEEE. https://doi.org/10.1109/ICRA40945.2020.9197326

[5] Dai, S., Li, S., Tang, H., Ning, X., Fang, F., & Fu, Y. (2024). MARP: A cooperative multiagent DRL system for connected autonomous vehicle platooning. IEEE Internet of Things Journal, 11(16), 27845–27856. https://doi.org/10.1109/JIOT.2024.3432119

[6] Cai, Z., Liu, J., Chi, W., & Zhang, B. (2023). A low-cost and robust multi-sensor data fusion scheme for heterogeneous multi-robot cooperative positioning in indoor environments. Remote Sensing, 15(23), 5584. https://doi.org/10.3390/rs15235584

[7] Lin, J., Feng, Y., Ren, W., Feng, J., & Zheng, J. (2024, October). Position-Constrained Calibration Compensation for Hand–Eye Calibration in Industrial Robots. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5412–5418). IEEE. https://doi.org/10.1109/IROS58592.2024.10802347

[8] Wada, D., Araujo-Estrada, S., & Windsor, S. (2022). Sim-to-real transfer for fixed-wing uncrewed aerial vehicle: pitch control by high-fidelity modelling and domain randomization. IEEE Robotics and Automation Letters, 7(4), 11735-11742. https://doi.org/10.1109/LRA.2022.3205442

[9] Jin, X., Zhang, H., Wang, L., & Li, Q. (2024). Review on control strategies for cable-driven parallel robots with model uncertainties. Chinese Journal of Mechanical Engineering, 37(1), 156. https://doi.org/10.1186/s10033-024-01149-8

[10] Ye, B. S., Jin, X. C., Li, H., Shao, B. Y., Li, X. K., & Li, S. A. (2024). Robot Error Compensation Algorithm Based on Pseudo-target Iterative Generation. China Mechanical Engineering, 35(1), 136–143. https://doi.org/10.3969/j.issn.1004-132X.2024.01.013

[11] Gutiérrez-Moreno, R., Barea, R., López-Guillén, E., Arango, F., Sánchez-García, F., & Bergasa, L. M. (2024). Enhancing autonomous driving in urban scenarios: a hybrid approach with reinforcement learning and classical control. Sensors, 25(1), 117. https://doi.org/10.3390/s25010117

[12] Marez, D., Borden, S., & Nans, L. (2020, May). UAV detection with a dataset augmented by domain randomization. In Geospatial Informatics X (Vol. 11398, pp. 39-50). SPIE. https://doi.org/10.1117/12.2558864

[13] Louati, A., Louati, H., Kariri, E., Neifar, W., Hassan, M. K., Khairi, M. H. H., Farahat, M. A., & El-Hoseny, H. M. (2024). Sustainable smart cities through multi-agent reinforcement learning-based cooperative autonomous vehicles. Sustainability, 16(5), 1779. https://doi.org/10.3390/su16051779

[14] Weng, H., Zhang, S., & Min, S. (2024). Multi-Constraint Optimization for Real-Time Bidding: A Reinforcement Learning Approach. Artificial Intelligence and Machine Learning Review, 5(1), 93-104. https://doi.org/10.69987/AIMLR.2024.50108

[15] Basumatary, H., Adhar, D., & Hazarika, S. M. (2024). Robustifying a reinforcement learning agent-based bionic reflex controller through an adaptive sliding mode control. Robotica, 42(12), 2890–2907. https://doi.org/10.1017/S0263574724001838

[16] Wang, Z., & Liu, Y. (2024, July). Cooperative Multi-Agent Reinforcement Learning for Connected and Autonomous Vehicle Fleet Control. In 2024 IEEE 4th International Conference on Intelligent Transportation Engineering (ICITE) (pp. 45–50). IEEE. https://doi.org/10.1109/ICITE62893.2024.10678921

[17] Liu, H., Kibireva, A., Meurer, M., & Bergs, T. (2023). An inverse method for automatic determination of material models for metal cutting based on multi-objective optimization. The International Journal of Advanced Manufacturing Technology, 129(7), 3353-3374. https://doi.org/10.1007/s00170-023-12346-5

[18] Chen, W., Wang, X., Gao, S., Shang, G., Zhou, C., Li, Z., ... & Hu, K. (2023). Overview of multi-robot collaborative SLAM from the perspective of data fusion. Machines, 11(6), 653. https://doi.org/10.3390/machines11060653

[19] Zhao, W., Queralta, J. P., & Westerlund, T. (2020, December). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In 2020 IEEE symposium series on computational intelligence (SSCI) (pp. 737-744). IEEE. https://doi.org/10.1109/SSCI47803.2020.9308468

[20] Li, R., Ding, N., Zhao, Y., & Liu, H. (2023). Real-time trajectory position error compensation technology of industrial robot. Measurement, 208, 112418. https://doi.org/10.1016/j.measurement.2022.112418

[21] Jin, Z., Liu, A., Zhang, W. A., Yu, L., & Su, C. Y. (2022). A learning based hierarchical control framework for human–robot collaboration. IEEE Transactions on Automation Science and Engineering, 20(1), 506-517. https://doi.org/10.1109/TASE.2022.3161993

[22] Zbinden, J., Molin, J., & Ortiz-Catalan, M. (2024). Deep learning for enhanced prosthetic control: Real-time motor intent decoding for simultaneous control of artificial limbs. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 32, 1177–1186. https://doi.org/10.1109/TNSRE.2024.3371896

[23] Khalil, R. A., Saeed, N., Masood, M., Fard, Y. M., Alouini, M. S., & Al-Naffouri, T. Y. (2021). Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. IEEE Internet of Things Journal, 8(14), 11016-11040. https://doi.org/10.1109/JIOT.2021.3051414

[24] Bai, J., Li, B., Wang, X., Wang, H., & Guo, Y. (2024). Bionic Hand Motion Control Method Based on Imitation of Human Hand Movements and Reinforcement Learning. Journal of Bionic Engineering, 21(2), 456–468. https://doi.org/10.1007/s42235-024-00423-9

[25] Nobre, F., & Heckman, C. (2019). Learning to calibrate: Reinforcement learning for guided calibration of visual–inertial rigs. The International Journal of Robotics Research, 38(12-13), 1388-1402. https://doi.org/10.1177/0278364919844824

[26] Hashmy, Y., Yu, Z., Shi, D., & Weng, Y. (2020). Wide-area measurement system-based low frequency oscillation damping control through reinforcement learning. IEEE Transactions on Smart Grid, 11(6), 5072-5083. https://doi.org/10.1109/TSG.2020.3008364

[27] Adeniyi, T., & Kumar, S. (2024, September). Reinforcement learning based actor critic and policy agent for optimized quantum sensor circuit design. In 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) (Vol. 1, pp. 1233-1243). IEEE. https://doi.org/10.1109/QCE60285.2024.00146

[28] Gaudreault, M., Joubair, A., & Bonev, I. (2018). Self-calibration of an industrial robot using a novel affordable 3D measuring device. Sensors, 18(10), 3380. https://doi.org/10.3390/s18103380