World Journal of Engineering Research
Print ISSN: 2959-9865
Online ISSN: 2959-9873
DOI: https://doi.org/10.61784/wjer3064

LLM-GUIDED ADAPTIVE STEP-SIZE ZEROING NEURAL NETWORK FOR ROBOTIC MANIPULATOR TRAJECTORY TRACKING

GuangYu Long, ZhuoSong Fu*
College of Computer Science and Engineering, Jishou University, Jishou 416000, Hunan, China.
**Corresponding Author: ZhuoSong Fu*

Abstract: Zeroing Neural Network (ZNN) is widely used for robotic trajectory tracking due to their rapid convergence and strong error attenuation properties. In discrete implementations, however, the choice of step-size critically influences tracking accuracy, numerical stability, and computational cost. Conventional variable-step strategies typically rely on fixed heuristics or manually tuned rules, limiting their adaptability in dynamic task conditions. To address this limitation, this paper introduces a Large Language Model (LLM)-based adaptive step-size mechanism for discrete ZNN in manipulator trajectory tracking. The LLM receives natural-language instructions together with essential system-state descriptions and outputs step-size adjustments that guide the ZNN update. This enables intuitive human-robot interaction and allows the controller to flexibly shift between high-precision tracking and low-computation execution without modifying the underlying ZNN formulation. Results show that the proposed method improves tracking accuracy when finer steps are selected, reduces computational load when coarser steps suffice, and maintains high semantic consistency in interpreting step-size-related instructions. These findings demonstrate the potential of integrating LLM reasoning into step-size regulation to enhance the flexibility and interpretability of discrete ZNN-based robotic tracking.
Keywords: Zeroing neural network; Variable step-size; Large Language Model; Robotic manipulator

1 INTRODUCTION

Zeroing Neural Network (ZNN) has been extensively studied as an effective computational framework for solving time-varying problems such as inverse kinematics and trajectory tracking of robotic manipulators [1-3]. By embedding the solution of a dynamic equation into an exponentially convergent neural error system, ZNN provides fast response, strong disturbance rejection, and stable tracking properties [4-5]. When implemented in discrete time, however, the performance of ZNN strongly depends on the choice of the numerical step size τ. A small step size yields higher tracking accuracy but significantly increases computational load, while a large step size improves computational efficiency at the cost of reduced precision or even numerical instability [6-7]. Designing an adaptive step-size strategy that balances accuracy and efficiency has therefore become an important issue in discrete ZNN research.

Existing approaches to variable step-size selection typically rely on fixed analytical rules, derivative-based indicators, error thresholds, or predefined schedules [8-10]. Although these methods can adjust τ according to certain system metrics, they lack flexibility, semantic interpretability, and the ability to incorporate human intentions. In practica applications, high-level objectives such as "make the tracking more accurate" or "run faster even if precision decreases" using natural language are more intuitive than manually tuning numerical parameters. Conventional adaptive algorithms cannot directly interpret such natural-language instructions, nor can they generalize across linguistically varied expressions that imply similar operational adjustments.

Recent advances in Large Language Models (LLMs) have shown remarkable capability in semantic understanding, instruction following, and reasoning under loosely specified objectives [11-13]. These properties make LLMs promising candidates for assisting numerical algorithms that require flexible or context-dependent parameter regulation. While LLMs have been applied to robot task planning, code generation, and decision-making [14], their potential as interpreters for numerical step-size modulation in ZNN solvers has not yet been investigated. To the best of our knowledge, no prior work has explored integrating LLM-driven semantic guidance with discrete ZNN to achieve human-interpretable and adaptive step-size control for trajectory tracking tasks.

To bridge this gap, this paper proposes an LLM-guided step-size adjustment mechanism for discrete ZNN manipulator trajectory tracking. The LLM receives natural-language instructions together with key system-state information and outputs step-size τ. This design allows the controller to dynamically shift between high-accuracy tracking and low-cost computation while maintaining an intuitive interaction interface. More importantly, the method introduces semantic consistency into numerical regulation: instructions that imply similar intentions yield similar step-size decisions, even across diverse linguistic expressions.

The main contributions of this work are summarized as follows.

1) Introduce an LLM-guided step-size adjustment mechanism that interprets natural-language instructions and system-state descriptions to dynamically regulate the step-size τ of discrete ZNN without modifying its computational formulation.

2) Develop a variable step-size discrete ZNN scheme for robotic manipulator trajectory tracking that flexibly transitions between high-precision updates and low-computation operation based on LLM-generated decisions.
3) Validate the effectiveness and robustness of the LLM-driven step-size adjustments through experiments on robotic manipulator trajectory tracking.

2 PROBLEM AND METHOD

This section presents the discrete ZNN formulation used for trajectory tracking, the inverse kinematics mapping of the robotic manipulator, and the proposed LLM-guided step-size adjustment mechanism. To avoid ambiguity, the parameter τ is consistently referred to as the step-size, which also determines the real-time update interval of the controller.

2.1 Zeroing Neural Network for Inverse Kinematics

For a robotic manipulator with joint configuration $\mathbf{q}(t) \in R^n$ and end-effector pose $\mathbf{x}(t) \in R^m$, the forward kinematics is expressed as

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{q}(t)). \tag{1}$$

Given a desired trajectory $\mathbf{x}_{\mathrm{d}}(t)$, the tracking error is defined as

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{q}(t)) - \mathbf{x}_{\mathrm{d}}(t). \tag{2}$$

The continuous-time ZNN imposes an exponentially stable error dynamics:

$$\dot{\mathbf{e}}(t) = -\alpha \mathbf{e}(t), \tag{3}$$

where $\alpha > 0$ is a constant decay coefficient. Differentiating $\mathbf{e}(t)$ with respect to time gives

$$\dot{\mathbf{e}}(t) = J(\mathbf{q}(t))\dot{\mathbf{q}}(t) - \dot{\mathbf{x}}_{\mathrm{d}}(t), \tag{4}$$

where $J(\mathbf{q}(t))$ is the manipulator geometric Jacobian. Combining the above equations leads to the ZNN update law:

$$J(\mathbf{q}(t))\dot{\mathbf{q}}(t) = \dot{\mathbf{x}}_{\mathrm{d}}(t) - \alpha \mathbf{e}(t), \tag{5}$$

and the joint velocity can be obtained via

$$\dot{\mathbf{q}}(t) = J^{\dagger}(\mathbf{q}(t))(\dot{\mathbf{x}}_{\mathrm{d}}(t) - \alpha \mathbf{e}(t)) \tag{6}$$

where $J^{\dagger}(\mathbf{q}(t))$ denotes the Moore–Penrose pseudoinverse.

2.2 Discrete ZNN Update with Step Size τ

For real-time implementation, the continuous dynamics are discretized using a forward-Euler scheme. At discrete time index k, the update rule becomes

$$\mathbf{q}_{k+1} \doteq \mathbf{q}_k + \tau_k \dot{\mathbf{q}}_k \tag{7}$$

where τ_k is the step-size applied at iteration k.
Substituting the ZNN expression of $\dot{\mathbf{q}}_k$, the discrete update becomes

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \tau_k J^{\dagger}(\mathbf{q}_k)(\dot{\mathbf{x}}_{\mathrm{d},k} - \alpha \mathbf{e}_k). \tag{8}$$

A smaller τ_k increases update frequency and improves tracking accuracy, while a larger τ_k gives faster execution but reduce precision. Therefore, selecting an appropriate τ_k plays a critical role in discrete ZNN performance.

2.3 Step-Size Decision Based on System State Description

At iteration k, the system constructs a compact description of the current motion state, including: 1) the discrete time index k, or elapsed time t_k; 2) the current tracking error norm $\|\mathbf{e}_k\|$; and 3) the currently applied step-size τ_k. This information is formatted as a short text string and used as part of the input to the LLM and enables the step-size decision to depend on the real-time tracking condition.

2.4 Natural-Language-Guided Step-Size Modulation

In addition to the state description, the user provides a natural-language instruction such as “make the tracking more accurate” or “update less frequently”. The LLM processes 1) the human instruction and 2) the system-state description, and outputs a recommended step size:

$$\tau_k^{\mathrm{LLM}} = g_{\mathrm{LLM}}(\mathrm{instruction}, \mathrm{statedescriptioin}). \tag{9}$$

The output is constrained to a safe range:

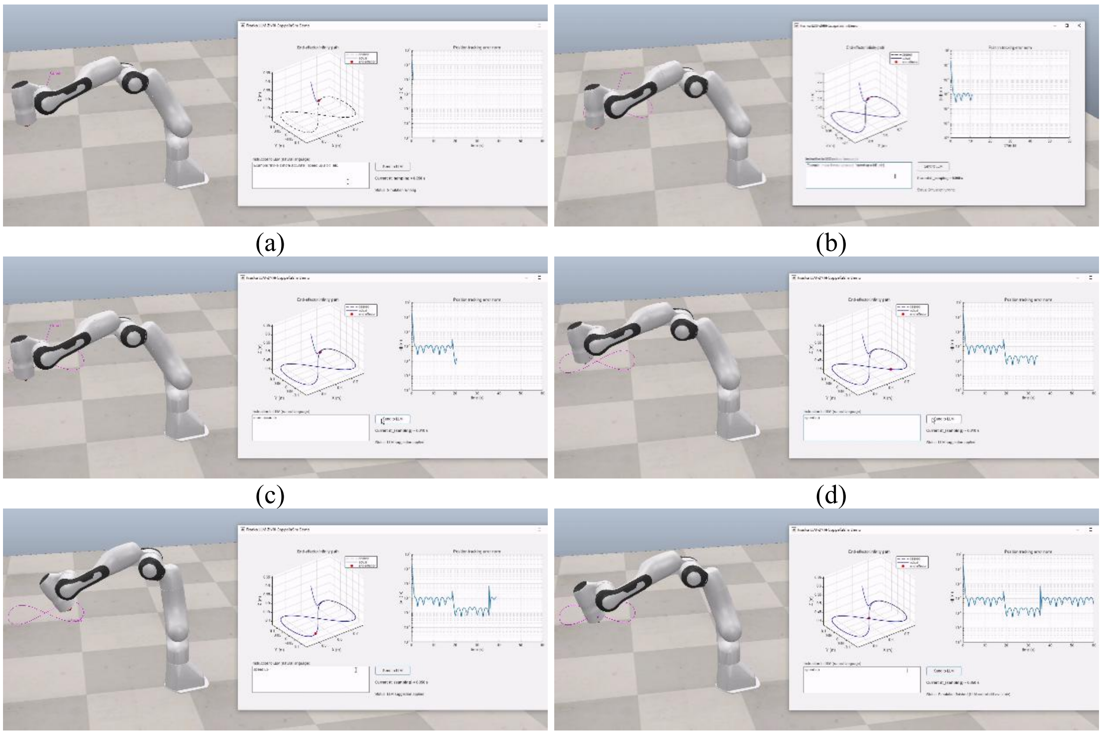
(a) (b) (c) (d) (e) (f)

Figure 1 Real-Time Path Tracking under the LLM-Guided Step-Size Adjustment Mechanism

$$\tau_{\min} \leq \tau_k^{\mathrm{LLM}} \leq \tau_{\max}, \tag{10}$$

ensuring numerical stability and preventing overly aggressive updates. On the basis of Eq. (10), the final step-size used in ZNN iteration is

$$\tau_k^* = \min(\max(\tau_k^{\mathrm{LLM}}, \tau_{\min}), \tau_{\max}). \tag{11}$$

This formulation allows: 1) intuitive human control through natural language; 2) semantically consistent adjustment of τ; adaptation to varying error levels and trajectory conditions. Notably, the ZNN computational structure remains unchanged. Only the numerical step size is modulated.

2.5 Overall Procedure

The algorithm for iteration k to change the step-size can be summarized as

1) Compute tracking error $\mathbf{e}_k$ and derivative $\dot{\mathbf{x}}_{\mathrm{d},k}$;
2) Assemble the state description including t_k, τ_k, and $\|\mathbf{e}_k\|$;
3) Query the LLM using the natural-language instruction and the state description.
4) Obtain the recommended step-size τ_k^* ;
5) Perform the ZNN update: $\mathbf{q}_{k+1} = \mathbf{q}_k + \tau_k^* J^{\dagger}(\mathbf{q}_k)(\dot{\mathbf{x}}_{\mathrm{d},k} - \alpha \mathbf{e}_k)$.

3 EXPERIMENTS

To evaluate the proposed LLM-guided step-size adjustment mechanism, a series of trajectory-tracking experiments are conducted using a Franka Emika Panda manipulator simulated in CoppeliaSim [15]. The robot follows a planar infinity-like path, while the discrete ZNN solver computes joint updates in real time. MATLAB serves as the controller environment, generates ZNN-based velocity updates, issues step-size decisions, and communicates with CoppeliaSim through the Remote API.
Three experiments are designed.

1) **Experiment 1** illustrates how the LLM modulates the step-size τ in response to human natural-language instructions and shows the resulting trajectory-tracking behavior.
2) **Experiment 2** compares fixed-step ZNN solvers with the LLM-guided variable-step scheme under identical conditions.
3) **Experiment 3** evaluates the robustness and semantic consistency of LLM-generated step-sizes using multiple natural-language expressions.

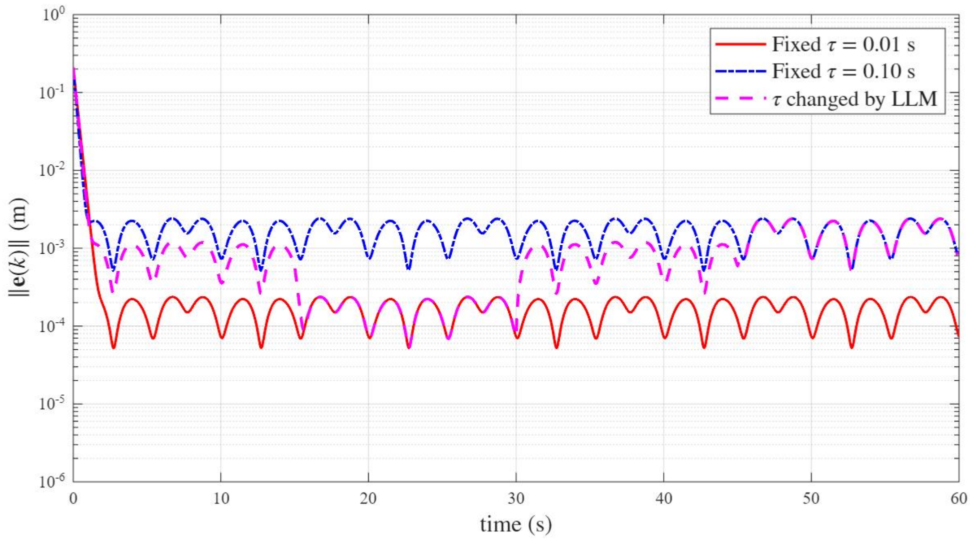

Figure 2 Tracking-Error Comparison between Fixed Step-Sizes and the LLM-Guided Step-Size

All experiments use the same ZNN formulation, robot model, and infinity-like path. The results confirm that the proposed mechanism adjusts τ appropriately, improves tracking precision when instructed to be "more accurate", and increases update efficiency when instructed to "speed up", while maintaining stable tracking behavior.

3.1 Experiment 1: Real-Time Step-Size Modulation via Natural-Language Instructions

Experiment 1 investigates how the LLM adjusts the step size in real time during robotic trajectory tracking. The user provides natural-language commands such as "speed up" or "make it more accurate", and the LLM outputs a suitable step-size based on both the instruction and the current system state. The manipulator begins tracking the infinity-like path with an initial step-size $\tau = 0.05$ s. Six representative snapshots from the experiment appear in Figure 1(a)-(f). These frames show the robot motion in CoppeliaSim, the desired and actual end-effector trajectories, and the corresponding error curves.
1) **Experiment begins**. At the start of the experiment, the manipulator moves from its initial configuration toward the infinity-like path. The tracking error is relatively high, as the end-effector has not yet reached the desired path.
2) **Error stabilizes under the current step size**. After several iterations, the ZNN solver drives the robot onto the desired path, and the error norm stabilizes at a characteristic level determined by the initial step-size. The tracking motion becomes smooth and periodic.
3) **The first natural-language instruction is issued**. When the user enters a command (e.g., "more accurate"), the LLM generates a new step-size. The change in τ modifies the update frequency of the ZNN solver, and the error curve responds immediately. A smaller step size produces a sharp decrease.
4) **Error stabilizes under the new step size**. After the initial transient, the tracking error settles into a new steady pattern that corresponds to the updated step-size. A reduced τ yields smaller error errors.
5) **The second natural-language instruction is issued**. The user inputs another natural-language instruction. The LLM interprets the new intent and updates τ. The tracking error immediately reflects this change, showing a rise when the LLM decreases or increases the step-size.
6) **Experiment concludes**. Near the end of the trajectory, the robot continues tracking the desired path using the most recently selected step-size. The tracking behavior remains stable, and the error remains bounded, demonstrating that LLM-guided step-size adjustment mechanism does not compromise the stability of the discrete ZNN solver.

3.2 Experiment 2: Comparison Between Fixed and LLM-Guided Step-Sizes

Experiment 2 evaluates the tracking performance of the proposed LLM-guided step-size adjustment mechanism by comparing it with two fixed-step ZNN baselines. All three methods operate on the same infinity-like path and use identical ZNN parameters except for the step-size τ. The goal is to assess whether the proposed mechanism produces behavior consistent with the numerical characteristics of the underlying solver and whether its performance falls

between the fine-step and coarse-step baselines, as theoretically expected. Three tracking-error curves are plotted in Figure 2.
1) The fixed-step baselines reveal the expected behavior of the discrete ZNN solver. The solid line ($\tau = 0.01$ s) shows the smallest steady-state error and the smoothest oscillation pattern. The error decreases rapidly at the beginning of the experiment and then remains in a narrow band around 10^{-4} m. This reflects the high numerical resolution achieved with a small step-size.

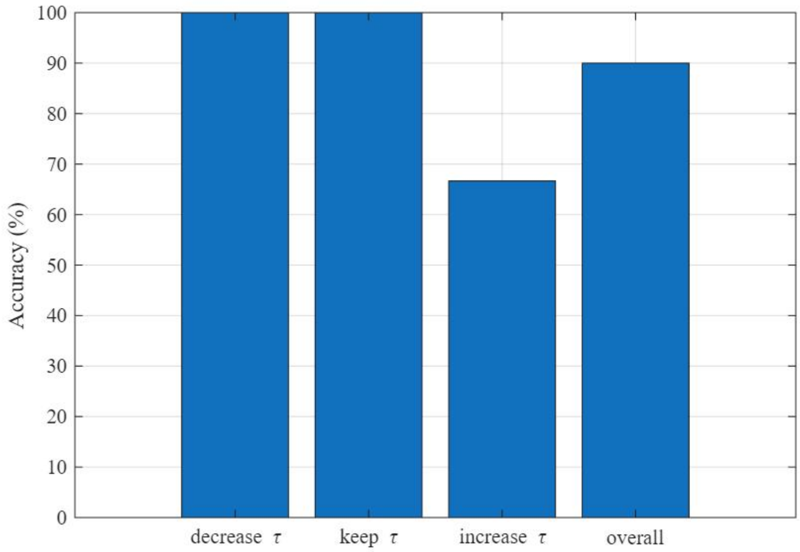

Figure 3 Accuracy of Instruction Interpretation

2) The dash-dotted line (($\tau = 0.10$ s) shows a noticeably larger steady-state error, remaining around 10^{-3} m. The oscillations are more pronounced and occur in a wider band, which is consistent with the reduced update frequency of the solver. Although the error remains bounded and stable, this configuration provides visibly lower precision.
3) The dashed line, corresponding to the LLM-guided step-size adjustment mechanism, oscillates between the behaviors of the two baselines. When the mechanism selects a smaller step-size, the error decreases and approaches the pattern of the solid line. When it selects a larger step-size, the error increases temporarily and moves toward the pattern of the dash-dotted line. After each change in step-size, the error settles into a predictable and stable range.
Overall, the dashed line remains entirely between the solid and dash-dotted lines, demonstrating that the LLM-guided step-size adjustment mechanism produces performance commensurate with the numerical properties of the selected step -sizes. In addition, the transitions between different steady-state levels occur smoothly. There is no sign of divergence, instability, or abrupt irregularities. This confirms that the mechanism respects the stability characteristics of the discrete ZNN solver while enabling adaptive behavior.

3.3 Experiment 3: Semantic Consistency

Experiment 3 evaluates the semantic consistency of the LLM-guided step-size adjustment mechanism when processing diverse natural-language expressions that convey similar intentions. The goal is to determine whether the mechanism reliably maps different verbal formulations to the correct type of step-size action: 1) decrease τ; 2) keep τ; and increase τ. To conduct the test, 100 generated instruction phrases are prepared for each of the three action categories. For example:
1) "Make it more accurate", "reduce the step size" and "increase update frequency" denote instructions that decrease τ.
2) "Maintain the current setting", "leave it unchanged" and "keep going as is" denote instructions that keep τ.
3) "Speed up", "use a larger step size" and "reduce update frequency" denote instructions that increase τ.
Each instruction is independently fed to the mechanism, together with a neutral system-state description, and the resulting step-size decision is labeled as correct or incorrect. Figure 3 shows the accuracy across the three action types and the overall performance.
The results indicate that the mechanism demonstrates high semantic reliability. For instructions that decrease τ, the accuracy reaches 100%, meaning every tested phrase is interpreted correctly. For instructions that keep τ, the accuracy also reaches 100%, showing that the mechanism recognizes neutrality consistently. For instructions that increase τ, the accuracy is slightly lower at 66.7%, indicating that upward step-size adjustments are somewhat harder for the mechanism to infer, possibly due to greater linguistic diversity in how users express speeding-up intentions. The overall accuracy reaches 90%, confirming strong robustness across all categories.
Across all tests, the mechanism does not produce unstable or extreme step-size recommendations, showing that the semantic interpretation remains well-behaved even when minor misclassifications occur.

4 CONCLUSION

This paper has presented an LLM-guided step-size adjustment mechanism for discrete ZNN-based trajectory tracking of robotic manipulators. By allowing the step-size τ to be determined through natural-language instructions and system-state descriptions, the proposed framework has enabled intuitive human–robot interaction without modifying the underlying ZNN formulation. The method has maintained the numerical properties of the solver while introducing semantic interpretability and adaptability into the control process. A series of experiments has been conducted to validate the effectiveness and robustness of the mechanism. The results have shown that integrating LLM reasoning into numerical step-size selection offers a promising direction for adaptive, human-interpretable robotic control. The framework has established a foundation for future extensions, such as richer multimodal instructions, more advanced ZNN formulations, and real-robot experiments.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

This research was supported by the Research Foundation of Jishou University under Grant JD24004.

REFERENCES

[1] Fu Z, Zhang Y, Tan N. Gradient-feedback Zhang neural network for unconstrained time-variant convex optimization and robot manipulator application. IEEE Transactions on Industrial Informatics, 2023, 19(10): 10489 -10500.

[2] Fu Z, Zhang Y, Li W. Solving future nonlinear equation system via ZNN and novel general ILR3S formula with multitype manipulator applications. IEEE Transactions on Industrial Electronics, 2024, 71(10): 12623-12633.

[3] Lai G, Lin C, Ouyang Y, et al. Finite-time uncalibrated visual servoing for robotic manipulators based on model-free zeroing neural networks. IEEE Transactions on Automation Science and Engineering, 2025, 22: 24347-24361.

[4] Li J, Qu L, Zhu Y, et al. A novel zeroing neural network for time-varying matrix pseudoinversion in the presence of linear noises. Tsinghua Science and Technology, 2025, 30(5): 1911-1926.

[5] Liao B, Han L, Cao X, et al. Double integral-enhanced Zeroing neural network with linear noise rejection for time-varying matrix inverse. CAAI Transactions on Intelligence Technology. 2024, 9(1): 197–210.

[6] Guo J, Zhang Y. Stepsize interval confirmation of general four-step DTZN algorithm illustrated with future quadratic programming and tracking control of manipulators. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(3): 1662-1670.

[7] Qiu B, Guo J, Li X, et al. New discretized zeroing neural network models for solving future system of bounded inequalities and nonlinear equations aided with general explicit linear four-Step rule. IEEE Transactions on Industrial Informatics, 2021, 17(8): 5164-5174.

[8] Park J, Park C, Park P. A variable step-size MSD-based diffusion LMS algorithm with nnoisy input and quantization link noise. IEEE Wireless Communications Letters, 2025, 14(11): 3734-3738.

[9] Liu J. An improved arithmetic optimization algorithm with multi-strategy variable step size for optimizing engineering problems. IEEE Access, 2025, 13: 156566-156599.

[10] Shen L, Jiang D, Zhu H. Control study for suppressing rotor vibration of the six-pole hybrid magnetic bearing system based on no-parameter variable-step-size least-mean-square algorithm. IEEE Transactions on Power Electronics, 2025, 40(8): 10672-10681.

[11] Hongying Z, Javed A, Abdullah M, et al. Large language models with contrastive decoding algorithm for hallucination mitigation in low-resource languages. CAAI Transactions on Intelligence Technology. 2025, 10: 1104-1117.

[12] Duan J, Li W, Bai Q, Nguyen M, et al. LLM-botguard: A novel framework for detecting LLM-driven bots with mixture of experts and graph neural networks. IEEE Transactions on Computational Social Systems, 2025, 12(5): 3488-3500.

[13] Chen W, Li G, Li M, et al. LLM-enabled incremental learning framework for hand exoskeleton control. IEEE Transactions on Automation Science and Engineering, 2025, 22: 2617-2626.

[14] Fakhoury S, Naik A, Sakkas G, et al. LLM-based test-driven interactive code generation: User study and empirical evaluation. IEEE Transactions on Software Engineering, 2024, 50(9): 2254-2268.

[15] Ferro M, Mirante A, Ficuciello F, et al. A Coppeliasim dynamic simulator for the Da Vinci research kit. IEEE Robotics and Automation Letters, 2023, 8(1): 129-136.