mathematics

MDPI

Article

ZNN-Based Gait Optimization for Humanoid Robots with ALIP and Inequality Constraints

Yuanji Liu [1,2], Hao Jiang [2], Haiming Mou [3], Qingdu Li [1,2,*] and Jianwei Zhang [4]

[1] School of Optoelectronic Information and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 211240188@st.usst.edu.cn
[2] Institute of Machine Intelligence, University of Shanghai for Science and Technology, Shanghai 200093, China; 222302294@st.usst.edu.cn
[3] Shanghai Droid Robot Co., Ltd., Shanghai 200093, China; mhming@droidup.com
[4] Department of Informatics, University of Hamburg, 20146 Hamburg, Germany
* Correspondence: liqd@usst.edu.cn

Abstract: This paper presents a zeroing neural networks (ZNN)-based gait optimization strategy for humanoid robots. First, the algorithm converts the angular momentum linear inverted pendulum (ALIP)-based gait planning problem into a time-varying quadratic programming (TVQP) problem by adding adaptive adjustment factors and physical limits as inequality constraints to avoid system oscillations or instability caused by large fluctuations in the robot's angular momentum. Secondly, This paper proposes a real-time and efficient solution for TVQP based on an integral strong predefined time activation function zeroing neural networks (ISPTAF-ZNN). Unlike existing ZNN approaches, the proposed ISPTAF-ZNN is enhanced to achieve convergence within a strong predefined-time while exhibiting noise tolerance. This ensures the desired rapid convergence and resilience for applications requiring strict time efficiency. The theoretical analysis is conducted using Lyapunov stability theory. Finally, the comparative experiments verify the convergence, robustness, and real-time performance of the ISPTAF-ZNN in comparison with existing ZNN approaches. Moreover, comparative gait planning experiments are conducted on the self-built humanoid robot X02. The results demonstrate that, compared to the absence of an optimization strategy, the proposed algorithm can effectively prevent overshoot and approximate energy-efficient responses caused by large variations in angular momentum.

Keywords: ZNN-based gait optimization strategy; ALIP-based gait planning; TVQP problem with inequality constraints; ISPTAF-ZNN solver; Lyapunov stability theory

MSC: 37F05

check for updates

Academic Editor: Cheng-Hung Huang

Received: 28 February 2025
Revised: 9 March 2025
Accepted: 12 March 2025
Published: 13 March 2025

Citation: Liu, Y.; Jiang, H.; Mou, H.; Li, Q.; Zhang, J. ZNN-Based Gait Optimization for Humanoid Robots with ALIP and Inequality Constraints. *Mathematics* **2025**, *13*, 954. https://doi.org/10.3390/math13060954

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Humanoid robots have high-dimensional, hybrid, and nonlinear characteristics. As a crucial intermediate module between the upstream perception module and the downstream whole-body control system in robotics, foot placement must exhibit strong stability and robustness [1]. For lightweight-legged robots, researchers simplify the model into a linear inverted pendulum (LIP) for gait planning, allowing quick adjustment of the center of mass (CoM) dynamics [2]. Compared to traditional methods that use the CoM velocity as a simplified state, using the angular momentum at the contact point as the system state can capture more system dynamics, such as center of mass angular momentum, linear momentum, and CoM trajectory [3]. Therefore, the angular momentum linear inverted pendulum (ALIP) model has been widely used to plan foot placement [4,5]. Existing foot

placement based on the ALIP model directly uses the difference between expected and current angular momentum for deadbeat control [6], where the expected system state reaches the target value in the shortest time. However, this control strategy has drawbacks, such as strong model dependence, large control inputs, sensitivity to noise, and poor disturbance resistance [7].

Time-varying quadratic programming (TVQP) is an effective strategy to address these issues [8–10]. By transforming the gait planning problem into an optimization problem and using actual physical constraints and adaptive adjustment techniques, the stability and robustness of the system can be improved. To ensure real-time planning, high accuracy, and robustness, an efficient TVQP solver needs to be designed. Recurrent neural networks (RNNs) have feedback structures, which use the design formulas of dynamical systems and are good at handling sequential data [11]. Due to their parallel processing characteristics and hardware implementability, RNNs have become powerful TVQP solvers [12]. Hopfield neural networks or gradient-based neural networks (GNNs) have been proven to be efficient QP solvers [13]. However, GNNs are inherently designed for solving time-invariant problems. When applied to TVQP problems, GNNs inevitably introduce lagging residuals. Zeroing neural networks (ZNNs), a special form of RNN, are capable of theoretically zeroing each element of the error monitoring function in matrix/vector form, making them suitable for time-varying systems like robots [14]. Traditional ZNNs solve TVQP problems with only equality constraints and cannot account for inequality physical constraints in robotic systems. To overcome this limitation, various techniques, including slack variables [15], nonlinear complementarity problem (NCP) functions [16], and penalty functions [17], have been used to handle inequality constraints. These methods generally rely on Lagrange multiplier sub-methods for modifying inequality constraints, but they show different results. NCP function ensures that inequality constraints are always satisfied by enforcing complementarity conditions. When constraints tend to become invalid, penalty or correction terms automatically adjust the optimization direction, keeping the solution within the feasible region. According to a comparison in the literature [18], the NCP function method demonstrates stronger stability and robustness in handling TVQP problems, followed by the penalty function method, while the slack variable method often leads to solving failures. However, achieving fast convergence and strong robustness in ZNN remains a challenging problem.

The sign-bi-power activation function (SBPAF) adjusts its nonlinearity by modifying the power exponent, enabling finite-time or fixed-time convergence in ZNN [19]. On the basis of the original SBPAF, the combination of sliding mode control and SBPAF has been used to enhance the robustness of solving chaotic systems [20]. Similarly, Jin J. [21] proposed a strongly robust SBPAF with adaptive parameters as a general method for applications in robotic control and image processing. However, the inherent drawback of SBPAF-based ZNN is the inability to accurately predict the upper bound of convergence time, leading to uncertainty and potential time-out issues in high real-time systems. Therefore, the Sign-exponential-power Activation Function (SEPAF), which inherently possesses weak predefined-time convergence properties, was proposed for matrix square root finding [22], chaotic systems [23], and Sylvester equation solving [24]. Nevertheless, SEPAF-ZNN methods still fail to simultaneously achieve both strong predefined-time convergence and robustness. Therefore, designing an activation function that simultaneously possesses strong predefined-time convergence and robustness is of great significance.

Based on the background above, the goal of this paper is to design a humanoid robot gait optimization algorithm based on an adaptive optimization strategy to improve the system's real-time performance, stability, and robustness. First, we transform the ALIP-based gait planning problem into a TVQP problem, design a minimization objective function

based on angular momentum error and adaptive parameter error, and use adaptive adjustment parameters and stride limits to constrain the foot placement. The foot placement is optimized online for the next time step to avoid system oscillation caused by step-speed commands or external disturbances. Second, for the TVQP problem with inequality constraints, the NCP function is adopted to transform inequality constraints into a standard form that can be solved by ZNN. We propose a novel ISPTAF, a piecewise nonlinear AF, to achieve strong predefined-time convergence properties. Additionally, ISPTAF leverages the anti-disturbance characteristics of integral functions, incorporating them into the designed activation function to enhance robustness. Compared to traditional AF, ISPTAF exhibits strong predefined-time convergence properties, enabling accurate estimation of the convergence time in ZNN solvers. Additionally, its integral term effectively eliminates the impact of low-frequency disturbances, enhancing the robustness of ZNN. Therefore, ISPTAF-ZNN is suitable for solving high-real-time, high-dynamic gait optimization problems based on TVQP.

Finally, through MATLAB numerical simulations, we verify the strong predefined-time convergence, initial value insensitivity, and robustness of ISPTAF-ZNN. Through the MuJoCo physics simulation engine, we validate the effectiveness of the proposed optimization algorithm on the X02 robot.

The main contributions of this chapter are as follows:

1. Proposing an adaptive gait optimization algorithm based on angular momentum: A cost function is constructed to balance tracking errors and system smoothness. Adaptive parameters and physical limits are used as constraints, and adaptive parameters adjust the error between the expected angular momentum and the current estimated angular momentum online, dynamically adjusting the next foot placement to avoid oscillations or tipping problems caused by deadbeat control.
2. Proposing a novel neural network ISPTAF-ZNN: A strong predefined-time convergence activation function is designed, with an integral term in the activation function to enhance the ZNN's disturbance resistance. The ISPTAF-ZNN neural dynamics formula is derived, and Lyapunov stability theory is used to prove its strong predefined-time convergence and robustness.

2. Preliminaries and Problem Formulation

In this section, the preliminaries for the gait controller based on ALIP and the modeling of the adaptive foot placement quadratic programming (AFP-QP) problem are discussed.

2.1. Preliminaries

Robot walking is a periodic process. In each cycle, the foot in contact with the ground is called the support leg, and the leg in the air is called the swing leg. Depending on whether the legs are in contact with the ground, the phases can be divided into double support, single support, and double flight. In this study, it is assumed that, except for the case where the robot is standing still with both feet, the robot always moves in the single support phase, where the support foot remains in stable contact with the ground without slipping. To derive the CoM dynamics equation for the single support phase, a foot contact coordinate system $({}^{c}X, {}^{c}Y, {}^{c}Z)$ is established, as shown in Figure 1. The x direction is forward along the body, the y direction is to the left along the body, and the z direction is upward along the body, following the right-hand coordinate system.

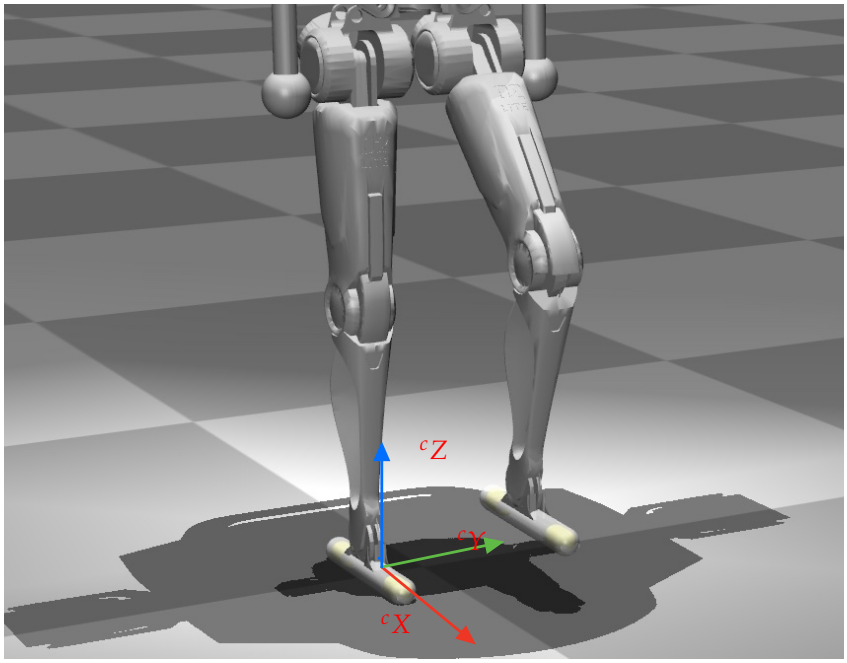


Figure 1. Support foot contact coordinate system.

Gait planning includes the x and y directions. By combining the characteristics of the 2D LIP model and the fact that there is no coupling between the longitudinal and lateral directions, we can divide the gait planning into x-direction foot placement planning in the ${}^cX^cZ$ plane and y-direction foot placement planning in the ${}^cY^cZ$ plane. This section first derives the foot placement planning algorithm for the x direction, and the algorithm for the y direction follows the same principle as that for the x direction.

According to the literature [25], the ALIP dynamics equation in the x direction is given by:

$$\dot{x}_c = \frac{L_y}{mz_c} + \frac{\dot{z}_c}{z_c}x_c - \frac{L_{cy}}{mz_c},\ \dot{L}_y \quad = mgx_c + u_a. \tag{1}$$

where $L_y = I\omega + p_c^x \wedge mv_c^x$ is the component of the angular momentum at the contact point in the x direction, $p_c^x = (x_c, z_c)$ is the position of the CoM in the ${}^cX^cZ$ plane, $v_c^x = (\dot{x}_c, \dot{z}_c)$ is the CoM velocity, $L_{cy} = I\omega$ is the component of the CoM angular momentum in the x direction, m is the total mass of the model, and u_a is the torque applied by the ankle motor.

Assuming the following conditions during steady-state walking: first, the height of the CoM remains nearly constant; second, the CoM pitch angle does not change drastically; and third, the torque from the ankle actuator has a negligible effect on the CoM. Based on these assumptions, the literature [25] simplifies Equation (1) by ignoring u_a and provides the following formula for calculating the desired foot placement for the next time step based on the current state:

$$p_{\text{sw}}^{x,\text{des}}(T_{k+1}^-) = \frac{L_y^{\text{des}} - \cosh(\alpha T)\hat{L}_y(T_k^-)}{mz_c\alpha\sinh(\alpha T)} \tag{2}$$

where T represents the full cycle of the swing leg motion. T_k^- represents the time after the k-th step ends but before the foot touches the ground. T_k^+ represents the time after the k-th step ends and the $k+1$-th step starts after the foot contacts the ground. $p_{\text{sw}}^{x,\text{des}}(T_{k+1}^-)$ represents the desired position vector of the swing leg relative to the CoM at time T_{k+1}^-. $\alpha = \sqrt{\frac{g}{z_c}}$ is the time constant. L_y^{des} represents the desired angular momentum at the end of the $k+1$-th step, which is the design input. $\hat{L}_y(T_k^-)$ is the real-time estimated angular momentum at the current gait cycle.

2.2. AFP-QP Scheme

By analyzing Equation (2), it can be seen that $p_{\text{sw}}^{x}(T_{k+1}^{-})$ mainly depends on the following factors: desired velocity v_x^{des}, gait cycle T, standing height z_c, and the real-time velocity v_c^x required to calculate $\hat{L}_y(T_k^{-})$. When the desired velocity is too large or when the real-time velocity changes drastically due to disturbances acting on the robot, it often leads to a foot placement that differs significantly from the current foot position. This can result in issues such as system oscillation, high energy loss, joint torque exceeding limits, and instability. Therefore, to improve the system's robustness and smooth transition capability, a smoothing coefficient γ is introduced to formulate the following optimization problem:

$$\min_{p_{\text{sw}}^{x,\text{des}}(T_{k+1}^{-}),\gamma} \quad ||L_y^{\text{des}} - \hat{L}_y(T_{k+1}^{-})||_{W_1} + ||\gamma - \gamma_{\text{pre}}||_{W_2} \tag{3}$$

$$\text{s.t.} \quad p_{\text{sw}}^{x,\text{des}}(T_{k+1}^{-}) = \frac{1-\gamma}{mz_c\alpha\sinh(\alpha T)} L_y^{\text{des}} + \frac{\gamma - \cosh(\alpha T)}{mz_c\alpha\sinh(\alpha T)} \hat{L}_y(T_k^{-}) \tag{4}$$

$$0 \le \gamma \le 1 \tag{5}$$

$$|p_{\text{sw}}^{x,\text{des}}(T_{k+1}^{-})| \le \Delta p \tag{6}$$

where the first term in the objective function (3) represents the error between the estimated angular momentum $\hat{L}_y(T_{k+1}^{-})$ generated by the planned swing foot $p_{\text{sw}}^{x,\text{des}}(T_{k+1}^{-})$ and the desired angular momentum L_y^{des}. A smaller error indicates faster convergence. The second term represents the smoothness of the system, preventing instability caused by drastic changes in the system. W_1 and W_2 are weight factors using the L_2 norm to adjust the weights of the two optimization objectives. The weights are adjusted using an empirical tuning method. If the gait adjustment is too drastic, it indicates that the adaptive factor weight is too small, so W_2 should be increased. Conversely, if the adjustment is too slow, W_1 should be increased. The constraint (4) represents the level of adjustment of γ to the swing foot planning. Equation (5) limits the range of γ. When γ approaches 0, a foot placement with a large variation is planned to quickly converge to the desired angular momentum L_y^{des}. When γ approaches 1, a foot placement with smaller variation is planned to achieve smoother motion. Equation (6) indicates that the desired foot placement should not exceed the safety threshold Δp defined by the design, to avoid system oscillations or even falls caused by foot placements that are too far.

3. ISPTAF-ZNN Solver

The traditional ZNN can only be used to solve optimization problems with equality constraints. For optimization problems with inequality constraints, we first need to transform the problem (3)–(6) into a standard form that can be handled by ZNN using the NCP function.

We first transform the problem into a standard TVQP form:

$$\begin{aligned} &\text{minimize } X(t)^{\text{T}}H(t)X(t)/2 + c^{\text{T}}X(t) \\ &\text{s.t. } A(t)X(t) = b(t) \\ &\quad C(t)X(t) \le d(t) \end{aligned} \tag{7}$$

where

$$X(t) = \begin{bmatrix} p_{\text{sw}}^{x,\text{des}}(T_{k+1}^{-}) \\ \gamma \end{bmatrix} \in \mathbb{R}^2, H(t) = \begin{bmatrix} D^2 & \\ & I \end{bmatrix} \in \mathbb{R}^{n\times n},$$

$$A(t) = \left[D, L_y^{\text{des}} - \hat{L}_y(T_k^{-})\right] \in \mathbb{R}^{m\times n}, b = L_y^{\text{des}} - E\hat{L}_y(T_k^{-}) \in \mathbb{R}^m,$$

$$c(t) = \begin{bmatrix} D(L_y^{\text{des}} - E\hat{L}_y(T_k^-)) \\ \alpha_{\text{pre}} \end{bmatrix} \in \mathbb{R}^n, C(t) = [I, -I]^{\mathrm{T}} \in \mathbb{R}^{p\times n}, d(t) = \begin{bmatrix} \Delta p \\ 1 \\ -\Delta p \\ 0 \end{bmatrix} \in \mathbb{R}^p,$$

$$D = mz_c\alpha \sinh(\alpha T), E = \cosh(\alpha T).$$

Define the following Lagrangian function [26], and the problem (7) can be restructured as:

$$\begin{aligned} \mathcal{L}(X(t), \lambda_1(t), \lambda_2(t), t) = X(t)^{\mathrm{T}} H(t) X(t)/2 + c^{\mathrm{T}} X(t) \\ + \lambda_1^{\mathrm{T}}(t)(A(t)X(t) - b(t)) \\ + \lambda_2^{\mathrm{T}}(t)(C(t)X(t) - d(t)) \end{aligned}$$

where $\lambda_1(t) \in \mathbb{R}^m$ and $\lambda_2(t) \in \mathbb{R}^p$ are the Lagrange multiplier vectors.

According to the Karush-Kuhn-Tucker (KKT) conditions, solving problem (7) is equivalent to solving the following nonlinear system:

$$\begin{cases} HX + c + A^{\mathrm{T}}\lambda_1 + C^{\mathrm{T}}\lambda_2 = 0, \\ AX - b = 0, \\ \phi_{FB}^{\delta}(\epsilon, \lambda_2) = \epsilon + \lambda_2 - z = 0, \end{cases} \tag{8}$$

where $\phi_{FB}^{\delta}(\epsilon, \lambda_2)$ represents one of the commonly used NCP functions, namely the perturbed Fischer–Burmeister function [27]:

$$\epsilon := d - CX, \quad z := \sqrt{\epsilon \circ \epsilon + \lambda_2 \circ \lambda_2 + \delta},$$

where $\circ$ is the Hadamard product, which represents element-wise multiplication of corresponding elements in matrices or vectors; $\delta \in \mathbb{R} \to 0^+$ is a perturbation term. Naturally, the following expression is introduced:

$$W(t)y(t) = r(t) \tag{9}$$

where

$$W = \begin{bmatrix} H & A^{\mathrm{T}} & C^{\mathrm{T}} \\ A & O & O \\ -C & O & I \end{bmatrix} \in \mathbb{R}^{(n+m+p)(n+m+p)}, y = \begin{bmatrix} X \\ \lambda_1 \\ \lambda_2 \end{bmatrix} \in \mathbb{R}^{n+m+p}, r = \begin{bmatrix} -c \\ b \\ z - d \end{bmatrix} \in \mathbb{R}^{n+m+p}$$

W is an invertible matrix. Since the elements of the coefficient matrix W and the vector r both vary smoothly with time, it follows from the above transformation that solving the TVQP problem (7) is equivalent to solving the time-varying matrix Equation (9).

For Equation (9), define the residual function as:

$$\mathcal{E} = Wy - r \tag{10}$$

where, $\mathcal{E} \in \mathbb{R}^{n+m+p}$. The residual error function is used to track the evolution process of the ZNN. According to the design method of ZNN [28], the standard evolution form of the ZNN is:

$$\dot{\mathcal{E}} = -k_0\mathcal{F}(\mathcal{E}) \tag{11}$$

where the left-hand side of Equation (11) represents the derivative form of $\mathcal{E}(t)$. $\mathcal{F}(\cdot): \mathbb{R}^{n+m+p} \to \mathbb{R}^{n+m+p}$ consists of scalar AF $f(\cdot)$ and has a monotonically increasing property. k_0 is the gain factor that regulates the convergence speed. By differentiating Equation (10) and combining it with Equation (11), the implicit ZNN dynamics equation is obtained as follows:

$$M\dot{y} = -Ny - k_0\mathcal{F}(Wy - r) - v \tag{12}$$

where,

$$M = \begin{bmatrix} H & A^T & C^T \\ A & 0 & 0 \\ (\Lambda(\epsilon \oslash z) - I)C & 0 & I - \Lambda(\lambda_2 \oslash z) \end{bmatrix} \in \mathbb{R}^{(n+m+p)\times(n+m+p)}, \tag{13}$$

$$N = \begin{bmatrix} \dot{H} & \dot{A}^T & \dot{C}^T \\ \dot{A} & 0 & 0 \\ (\Lambda(\epsilon \oslash z) - I)\dot{C} & 0 & 0 \end{bmatrix} \in \mathbb{R}^{m+n+p}, \tag{14}$$

$$\dot{y} = \begin{bmatrix} \dot{X} \\ \dot{\lambda_1} \\ \dot{\lambda_2} \end{bmatrix} \in \mathbb{R}^{n+m+p}, v = \begin{bmatrix} \dot{c} \\ -\dot{b} \\ (I - \Lambda(\epsilon \oslash z))\dot{d} \end{bmatrix} \in \mathbb{R}^{n+m+p} \tag{15}$$

The Λ operator transforms a vector into a diagonal matrix, and $\oslash$ is the Hadamard division operator, which represents element-wise division of corresponding elements in matrices or vectors.

In practical applications, the ZNN model is often disturbed by additional noise. The evolution form of the noisy ZNN is:

$$\dot{\mathcal{E}}(t) = -k_0\mathcal{F}(\mathcal{E}(t)) + \Delta(t)$$

Therefore, the noisy ZNN can be described as:

$$M\dot{y} = -Ny - k_0\mathcal{F}(Wy - r) - v + \Delta(t) \tag{16}$$

where $\Delta(t) \in \mathbb{R}^{n+m+p}$ represents the additive noise, such as measurement fluctuations or computational precision limitations [29].

A novel piecewise activation function is designed to enable the noisy ZNN (16) to converge quickly and suppress disturbances. First, in the absence of disturbances, we adopt a time-segmentation strategy to enhance the convergence properties of SEPAF, enabling it to achieve strong predefined-time convergence. The ZNN (12) not affected by noise can be designed as:

$$f(\mathcal{E}) = \begin{cases} \frac{\exp(|\mathcal{E}|^q)|\mathcal{E}|^{1-q}\text{sign}(\mathcal{E})}{T_c q} & t \in [0, T_c) \\ \mathcal{E} + |\mathcal{E}|^p\text{sign}(\mathcal{E}) + \xi\text{sign}(\mathcal{E}) & t \in [T_c, +\infty) \end{cases} \tag{17}$$

where $0 < p, q < 1, \xi > 0$; T_c is the manually set convergence time; $\text{sign}(\cdot)$ satisfies:

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

However, in engineering applications, the ZNN (16) is typically affected by additional noise. We utilize the integral term's insensitivity to low-frequency noise to enhance the robustness of ISPTAF [30], with the improved form given as

$$f(\mathcal{E}) = \begin{cases} \frac{h(t-T_{c_i})}{(T_{cn}-T_{c_i})} g_{q_n}(\mathcal{E}) + \frac{1}{T_{c_i}} g_{q_i}(\eta) + k_1 \text{sign}(\eta) & t \in [0, T_c) \\ \mathcal{E} + |\mathcal{E}|^p \text{sign}(\mathcal{E}) + \xi \text{sign}(\mathcal{E}) & t \in [T_c, +\infty) \end{cases} \tag{18}$$

where $g(x) = \frac{\exp(|x|^q)|x|^{1-q}\text{sign}(x)}{q}$, $h(\cdot)$ is the Heaviside step function, $\eta = \mathcal{E} + \int_0^t \dot{\mathcal{Z}}(\tau)d\tau$, and $\dot{\mathcal{Z}} = k_2 \frac{h(t-T_{c_i})}{(T_{cn}-T_{c_i})} g_{q_n}(\mathcal{E})$, where $T_{cn} > T_{ci} > 0$, and $0 < p, q_i, q_n < 1$.

Remark 1. *Based on the definition of integral sliding mode in the reference [31]: "If the system's motion equations under this sliding mode have the same order as the original system, then this sliding mode is called an integral sliding mode." The characteristic of an integral sliding mode is that the state space dimension is consistent with the system's dimension that uses the integral term. Its advantage lies in the fact that the system's state always starts from the sliding surface, eliminating the need for a reaching phase, thereby enhancing the system's robustness. In this study, an integral term $\mathcal{Z}$ is designed for $t \in [0, T_{c_i})$, allowing the disturbed ZNN (16) to rapidly decrease to the original ZNN (12) within the preset time T_{c_i}. For $t \in [T_{c_i}, T_{c_n}]$, the system's residual error strictly converges to zero within the preset time T_{c_n} and has strong disturbance rejection capability. For $t \in (T_{c_n}, +\infty)$, due to the robustness term $|\mathcal{E}|^p sign(\mathcal{E}) + \xi sign(\mathcal{E})$, the neural model can also resist additional noise. The design principles of the two preset times need to satisfy $T_{c_n} > T_{c_i} > 0$. Theorem 2 provides a detailed derivation of this process.*

4. Analysis of Convergence and Robustness

This section analyzes the convergence and robustness of the ISPTAF-ZNN (16) using AF (18).

Definition 1 ([32]). *Given a system:*

$$\dot{x} = f(t, x; \rho), \tag{19}$$

For the system parameterρ and a constant $T_c(\rho) > 0$, if a non-empty set $M \subseteq \mathbb{R}^n$ satisfies:

(i) For the system (19), if any solution $x(t, x_0)$ reaches M within some finite time $t = t_0 + T(x_0)$, and the residence time function $T : \mathbb{R}^n \to \mathbb{R}$ satisfies

$$T(x_0) \leq T_c \quad \forall x_0 \in \mathbb{R}^n,$$

then M is said to be globally weakly predefined-time attractive for the system (19). In this case, T_c is called the weak predefined-time.

(ii) For the system (19), if any solution $x(t, x_0)$ reaches M within some finite time $t = t_0 + T(x_0)$, and the residence time function $T : \mathbb{R}^n \to \mathbb{R}$ satisfies

$$\sup_{x_0 \in \mathbb{R}^n} T(x_0) = T_c,$$

then M is said to be globally strongly predefined-time attractive for the system (19). In this case, T_c is called the strong predefined time.

Theorem 1. *Under the assumption of no external noise, and by using the AF (17), it is assumed that TVQP (7) has a unique solution. The output $y(t)$ of the ZNN (12), starting from any initial state $y(0)$, will theoretically converge to the optimal solution $y^*(t)$ within the strong predefined time T_c. The residual error $\mathcal{E}$ will also decay to zero.*

Proof.

1. For $t \in [0, T_c)$, the ZNN (12) can be represented as the dynamic nonlinear equation $\dot{\mathcal{E}}(t) = -k_0 f(\mathcal{E}(t))$, where the autonomous and independent subsystems are given by

$$\dot{e}_i(t) = -k_0 \frac{\exp(|e_i(t)|^q)|e_i(t)|^{1-q}\text{sign}(e_i(t))}{T_c q} \tag{20}$$

where $i \in 1, 2, \ldots, n+m+p$. $ei(t)$ and $\dot{e}i(t)$ are the scalar components of the vector $\mathcal{E}(t)$ and $\dot{\mathcal{E}}(t)$. Choose a Lyapunov function as $V_i(t) = |e_i(t)|$. The corresponding time derivative is

$$\begin{aligned}\dot{V}_i(t) &= \dot{e}_i(t)\text{sign}(e_i(t)) \\ &= -k_0 \frac{\exp(|e_i(t)|^q)|e_i(t)|^{1-q}}{T_c q} \\ &= -k_0 \frac{1}{T_c q}\exp(V^q)V^{1-q}\end{aligned}$$

Solving this ordinary differential equation yields

$$V_i(t) = \left(-\ln\left(q\left(\frac{k_0 t}{q T_c} - C[1]\right)\right)\right)^{\frac{1}{q}}$$

Substituting the initial condition $V_i(t_0) = V_i(0)$ into the above equation gives $C[1] = \frac{e^{V_i(0)^q} k_0 t_0 - T_c}{e^{V_0^q} q T_c}$. Substituting $C[1]$ into the equation, we get

$$V_i(t) = \left(\ln\left(\frac{1}{\frac{k_0(t-t_0)}{T_c} + e^{-V_i(0)^q}}\right)\right)^{\frac{1}{q}} \tag{21}$$

Define the residence time as $T(e_i(0)) = t - t_0$. According to the solution (21), it is clear that when $T(e_i(0)) \to T_c \frac{(1-\exp(-|e_i(0)|^q))}{k_0}$, we have $\lim_{T(e_i(0))\to T_c \frac{(1-\exp(-|e_i(0)|^q))}{k_0}} V_i(t) = 0$. Moreover, since the given $|e_i(0)|$ is radially unbounded, we have $\sup_{e_i(0)\in\mathbb{R}}[1 - \exp(-|e_i(0)|^q)] = 1$. According to Definition (1), we have

$$\sup_{e_i(0)\in\mathbb{R}} T(e_i(0)) = T_c \tag{22}$$

This means that the ISPTAF-ZNN (12) is strongly predefined-time stable within the time T_c.

2. For $t \in [T_c, +\infty)$, the subsystem can be simplified as

$$\dot{e}_i(t) = -k_0(|e_i(t)| + |e_i(t)|^p + \xi)\text{sign}(e_i(t)) \tag{23}$$

The derivative of $V_i(t)$ is

$$\begin{aligned}\dot{V}_i(t) &= \dot{e}_i(t)\text{sign}(e_i(t)) \\ &= -k_0[|e_i(t)| + |e_i(t)|^p + \xi] \leq 0\end{aligned} \tag{24}$$

The Equation (24) shows that for $t \in [T_c, +\infty)$, $V_i(t)$ will not monotonically increase. Therefore, for all subsystems, the residual error vector $\mathcal{E}$ remains zero within $t \in [T_c, +\infty)$.

The two parts of the proof above show that under the strong predefined-time T_c, the residual error $\mathcal{E}$ of the ZNN (12) will strictly converge to zero. Correspondingly, the output of ZNN, $y(t)$, will also approach the optimal solution $y^*(t)$. □

Theorem 2. *For TVQP (7), the ZNN (16) activated by the AF (18) has its neural state perturbed by external noise $|\Delta_i(t)| \leq \delta$, where $0 < \delta < +\infty$. Starting from any initial state $y(0)$, $y(t)$ will strictly converge to the theoretical solution $y^*(t)$ within the strong predefined-time T_{c_n}, and the additional noise can be suppressed within the PT T_{c_i}, as long as the following conditions are met: $T_{c_n} > T_{c_i} > 0$, $k_0 k_1 > \delta$, $k_0 \xi \geq \delta$.*

Proof.

1. For $t \in [0, T_{c_n})$, the perturbed ISPTAF-ZNN (16) consists of $m + n + p$ subsystems in the time interval $t \in [0, T_c \frac{(1-\exp(-V(0)^q))}{k_0})$. Let $k_0 = k_2$, each subsystem can be represented as

$$\begin{aligned} \dot{e}_i(t) = &- k_0 \Bigg(\frac{h(t - T_{c_i})}{(T_{c_n} - T_{c_i})} g_{q_n}(e_i(t)) \\ &+ \frac{1}{T_{c_i}} g_{q_i}(\eta_i(t)) + k_1 \text{sign}(\eta_i(t)) \Bigg) + \Delta_i(t) \end{aligned} \tag{25}$$

$$\dot{\eta}_i(t) = -k_0 \Bigg(\frac{1}{T_{c_i}} g_{q_i}(\eta_i(t)) + k_1 \text{sign}(\eta_i(t)) \Bigg) + \Delta_i(t) \tag{26}$$

Choose $V_i(t) = |\eta_i(t)|$, and assume the parameters satisfy $k_0 k_1 > \delta$. Taking the derivative of $V_i(t)$ and substituting Equation (25), we get

$$\begin{aligned} \dot{V}_i(t) &= \dot{\eta}_i(t) \text{sign}(\eta_i(t)) \\ &= -k_0 \frac{1}{T_{c_i} q} \exp(V^q) V^{1-q} - (k_0 k_1 - \Delta_i(t)) \\ &\leq -k_0 \frac{1}{T_{c_i} q} \exp(V^q) V^{1-q} - (k_0 k_1 - \delta) \\ &\leq -k_0 \frac{1}{T_{c_i} q} \exp(V^q) V^{1-q} \end{aligned}$$

According to Theorem 1, the closed-loop system (26) weakly or strongly converges within the predefined-time $T_{c_i}^- = T_{c_i} \frac{(1-\exp(-|\eta_i(0)|^{q_i}))}{k_0}$. When $t > T_{c_i}^-$, the entire system simplifies to

$$\begin{cases} \dot{e}_i(t) = 0 & T_{c_i}^- \leq t < T_{c_i} \\ \dot{e}_i(t) = -k_0 \left(\frac{1}{T_{c_n} - T_{c_i}} g_{q_n}(e_i(t)) \right) & t \geq T_{c_i} \end{cases} \tag{27}$$

According to Theorem 1, the residual error $\dot{e}_i(t)$ of each subsystem (27) decays to zero in a strongly predefined-time manner. The convergence time is $\sup_{e_i(0) \in \mathbb{R}} T(e_i(0)) = T_{c_i} + T_{c_n} - T_{c_i} = T_{c_n}$. Therefore, for the ISPTAF-ZNN (16), the external disturbance $\Delta(t)$ can be suppressed within the predefined-time T_{c_i}, and the residual error $\mathcal{E}$ converges to zero within the strong predefined-time T_{c_n}.

2. For $t \in [T_{c_n}, +\infty)$, the subsystem can be represented as

$$\dot{e}_i(t) = -k_0 (|e_i(t)| + |e_i(t)|^p + \xi) \text{sign}(e_i(t)) + \Delta_i(t) \tag{28}$$

The derivative of $V_i(t)$ is

$$\dot{V}_i(t) = \dot{e}_i(t)\text{sign}(e_i(t)) \tag{29}$$
$$= -k_0(|e_i| + |e_i|^p) - (k_0\xi - \Delta_i) \tag{30}$$
$$\leq -k_0(|e_i| + |e_i|^p) - (k_0\xi - \delta_i) \leq 0 \tag{31}$$

When the parameters satisfy $k_0\xi \geq \delta$, Equation (24) shows that $V_i(t)$ will not monotonically increase within $t \in [T_{c_n}, +\infty)$ and will be able to suppress the external disturbance. □

The robustness proof over the entire time period shows that both the ZNN (12) driven by ISPTAF (17) and the ZNN (16) driven by ISPTAF (18) can not only strongly converge within the predefined-time T_{c_n} but also resist external disturbances within the shorter predefined-time T_{c_i}, ensuring that the ZNN's output $y(t)$ tightly approximates the optimal solution $y^*(t)$.

5. Simulation Results and Analysis

5.1. ZNN Solver Performance Verification

This section verifies the convergence and robustness of the proposed ISPTAF-ZNN solver, which will be used for effectively solving the AFP-QP problem. First, we validate the strong predefined-time convergence performance of the ZNN (16) with ISPTAF (18) under undisturbed conditions and check its adherence to inequality constraints, comparing it with SEPAF and SBPAF. Then, we experimentally verify the impact of the initial values on the ISPTAF-ZNN solver. Furthermore, under disturbed conditions, we validate the disturbance rejection capability of the proposed solver for different types of disturbances. Finally, by setting different predefined-time T_{cn}, we verify the real-time performance of the proposed solver. The hardware platform is a Lenovo Y9000R laptop (Lenovo, Beijing, China) with an Ubuntu 22.04 operating system, Intel i9-14900HX×32 CPU, and 32GB of memory. The experiments were conducted in Matlab 2021b, using the ode45 solver to handle the ZNN described by the ordinary differential equations. The relative error tolerance and absolute error tolerance were both set to 1×10^{-5}.

Given a TVQP problem:

$$\begin{aligned}\text{minimize } &\left(\frac{\sin(t)}{4} + 2\right)x_1^2(t) + \left(\frac{\sin(t)}{4} + 2\right)x_2^2(t) \\ &+ \cos(t)x_1(t)x_2(t) + \cos(t)x_1(t) + \sin(t)x_2(t), \end{aligned} \tag{32}$$
$$\text{s.t. } \sin(t)x_1(t) + \cos(t)x_2(t) = \cos(t) \tag{33}$$
$$x_1(t) + x_2(t) \leq 1.4 \tag{34}$$

The above TVQP problem (32) can be transformed into a compact form of the optimization problem with the following parameter matrix:

$$H(t) = \begin{bmatrix} \sin(t)/2 + 4 & \cos(t) \\ \cos(t) & \sin(t)/2 + 4 \end{bmatrix}, c(t) = \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$$

$$A(t) = [\sin(t), \cos(t)], \quad b(t) = \cos(t)$$

$$X(t) = [x_1(t), x_2(t)]^{\mathrm{T}}, \quad C(t) = [1, 1], \quad d(t) = 1.4$$

The AFs used to conduct comparative experiments are listed in Table 1

Table 1. Comparison of different AF.

AF	Expression	Predefined-Time Convergence	Robustness
SBPAF [22]	$f(\mathcal{E}) = \mathcal{E} + \lvert\mathcal{E}\rvert^{p}\text{sign}(\mathcal{E}) + \lvert\mathcal{E}\rvert^{\frac{1}{p}}\text{sign}(\mathcal{E}) + \xi\text{sign}(\mathcal{E}), \quad t \in [0, +\infty)$	No	Yes
SEPAF [22]	$f(\mathcal{E}) = \frac{1}{p}\exp(\lvert\mathcal{E}\rvert^{p})\lvert\mathcal{E}\rvert^{1-p}\text{sign}(\mathcal{E}) + \xi\text{sign}(\mathcal{E}), \quad t \in [0, +\infty)$	Weakly	Yes
PTAF [33]	$f(\mathcal{E}) = \begin{cases} \frac{\exp(\mathcal{E})-1}{(T_{c_n}-t)\exp(\mathcal{E})}, & t \in [0, T_{c_n}) \\ \mathcal{E}, & t \in [T_{c_n}, +\infty). \end{cases}$	Strong	No
ISPTAF (18)	$f(\mathcal{E}) = \begin{cases} \frac{h(t-T_{c_i})}{(T_{c_n}-T_{c_i})}g_{q_n}(\mathcal{E}) + \frac{1}{T_{c_i}}g_{q_i}(\eta) + k_1\text{sign}(\eta) & t \in [0, T_{c_n}) \\ \mathcal{E} + \lvert\mathcal{E}\rvert^{p}\text{sign}(\mathcal{E}) + \xi\text{sign}(\mathcal{E}) & t \in [T_{c_n}, +\infty) \end{cases}$	Strong	Yes

5.1.1. Without Noise Case

This section verifies two superior performance characteristics of the ZNN (12) driven by ISPTAF (18) (the undisturbed case of ZNN): (1) strong predefined-time convergence performance; (2) insensitivity to the initial value $y(0)$. The shared parameters for different types of activation functions are the same: $p = 0.5, \xi = 4$; the unique parameters for ISPTAF (18) are $T_{c_n} = 0.1, T_{c_i} = 0, q_i = 0, q_n = 0.6, k_1 = 2$. The parameters for ZNN (12) are $k_0 = 1.0$.

Figure 2 shows the convergence performance of ZNN (12) driven by three different AFs. The red curves in Figure 2a–c represent the KKT theoretical solution of the problem (32) when the inequality constraints are not active, and two elements satisfy:

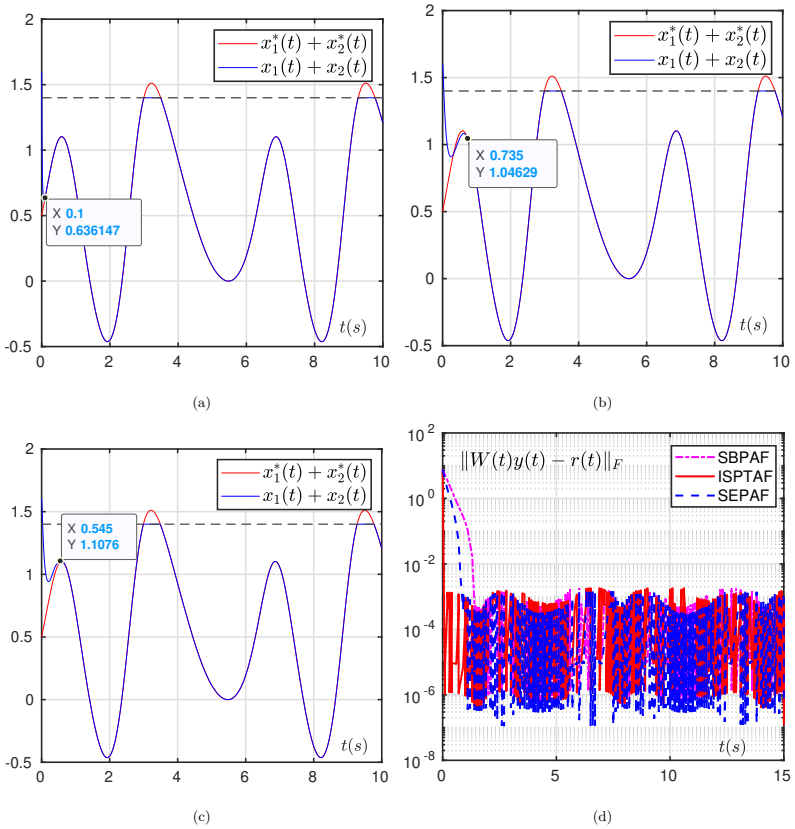


Figure 2. The transient performance of the variable to be solved, $X(t)$, approaching the optimal solution under different activation functions: (**a**) Output $X(t)$ of ZNN driven by ISPTAF (18); (**b**) Output $X(t)$ of ZNN driven by SBPAF; (**c**) Output $y(t)$ of ZNN driven by SEPAF; (**d**) Comparison of residual error $\mathcal{E}$.

$$\hat{x}_1^*(t) + \hat{x}_2^*(t) = -\frac{3\cos(t) - 2\sin(t) + 8\cos(t)\sin(t) + 8\cos^2(t) - 7\cos^3(t) + 3\cos^2(t)\sin(t)}{\sin(3t) - 8}$$

The black dashed line represents the constraint $x_1(t) + x_2(t) \leq 1.4$. The blue curves show the neural network's approximated solution. Figure 2a–c demonstrates that all three AFs ensure that the ZNN's numerical optimization solution effectively approaches the theoretical solution without violating the inequality constraints. However, the ISPTAF-ZNN exhibits strong predefined-time convergence characteristics, accurately converging to within the set error threshold at $T_{c_n} = 0.1$ s, while SBPAF and SEPAF converge at 0.735 s and 0.545 s, respectively. Combined with Figure 2d, it is observed that under the effect of inequality constraints, the solution of the original problem $y = [x_1, x_2, \lambda_1, \lambda_2]$ is compatible under the influence of the NCP function $\phi_{FB}^{\delta}(\epsilon, \lambda_2)$, driving the original problem's residual $||Wy - r||$ to a high accuracy of 1×10^{-5}.

To study the impact of the initial state $y(0)$, $y(0)$ was uniformly sampled within the interval $[-8, 8]$. The zoomed-in subfigure of Figure 3 shows that the ZNN (12) driven by ISPFAF (18) converges strictly to the optimal solution of the TVQP problem (32) within the predefined-time, regardless of the differences in the initial values. This result indicates that, unlike ZNN (12) driven by traditional SEPAF and SBPAF, the residual error convergence of the ZNN (12) driven by ISPTAF (18) is independent of the initial state. In practical applications, even if the system has uncertainties causing large deviations between each initial state and the optimal solution, and hot-starting is not feasible, the ZNN (12) driven by ISPTAF (18) can still guarantee fast convergence, improving the system's stability.

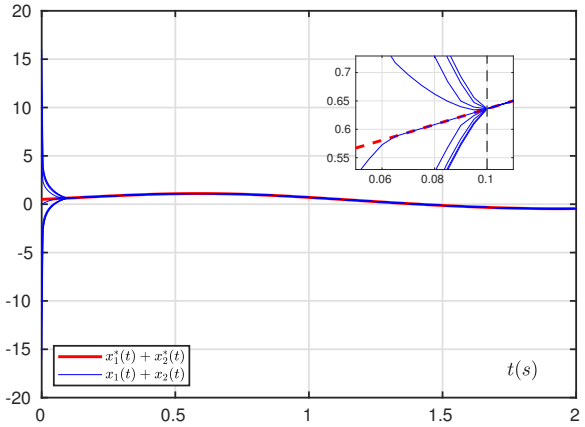


Figure 3. The transient performance of the ZNN output $y(t)$ driven by ISPFAF (18), with initial values $y(0)$ uniformly sampled from $[-8, 8]$.

5.1.2. Under Noise Case

First, this section verifies the robustness of the ZNN (16) driven by ISPTAF (18) under different disturbances and compares it with three other AFs listed in Table 1. Apart from $\xi \in 4, 6, 8$, the ZNN parameters remain consistent with those in the undisturbed case. The unique parameters of ISPTAF (18) are $T_{c_n} = 0.1, T_{c_i} = 0.03$. Time-varying disturbances are represented as $\Delta = \frac{3}{2}\sin(t)$, and constant disturbances are represented as $\Delta = 1.5$. Both disturbances are added after the system reaches a steady state ($t \geq 1.5$ s).

From the data in Table 2, it can be seen that ISPTAF outperforms under all test conditions (different ξ values) with higher convergence accuracy (10^{-6}), shorter convergence time (0.1 s), and strong robustness. This indicates that the ZNN driven by the ISPTAF has good fast response capabilities and solution accuracy. By comparing Figure 4 with

Table 2, both ISPTAF and PTAF exhibit strong predefined-time convergence characteristics, while SEP and SBP types can only achieve weak predefined-time convergence or finite-time convergence. The convergence time of SEPAF and SBPAF is negatively correlated with ξ. Furthermore, ISPTAF, SEPAF, and SBPAF activation functions exhibit strong anti-interference ability, whether dealing with constant or periodic disturbances, and the gain of the sign function demonstrates stronger robustness. However, PTAF fails to show any anti-interference capability under all test conditions, even diverging under constant disturbances.

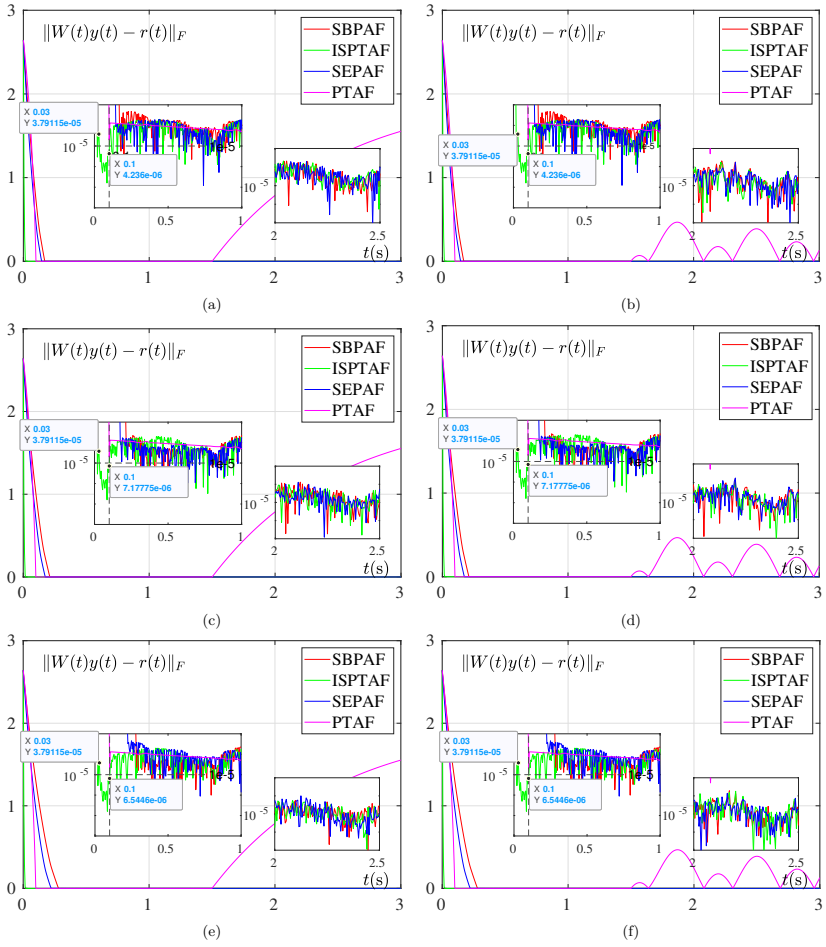


Figure 4. Transient performance of the residual error $\mathcal{E} = W(t)y(t) - r(t)$ under different activation functions, noise types, and ξ conditions. (**a**) Residual error $\mathcal{E}$ when the noise $\Delta = \frac{3}{2}\sin(t)$ and $t \geq 1.5$, with $\xi = 8$; (**b**) Residual error $\mathcal{E}$ when the noise $\Delta = 1.5$ and $t \geq 1.5$, with $\xi = 8$; (**c**) Residual error $\mathcal{E}$ when the noise $\Delta = \frac{3}{2}\sin(t)$ and $t \geq 1.5$, with $\xi = 6$; (**d**) Residual error $\mathcal{E}$ when the noise $\Delta = 1.5$ and $t \geq 1.5$, with $\xi = 6$; (**e**) Residual error $\mathcal{E}$ when the noise $\Delta = \frac{3}{2}\sin(t)$ and $t \geq 1.5$, with $\xi = 4$; (**f**) Residual error $\mathcal{E}$ when the noise $\Delta = 1.5$ and $t \geq 1.5$, with $\xi = 4$.

Next, Figure 5 shows that the ZNN (16) driven by the ISPFAF (18) exhibits strong predefined-time convergence even under disturbances with different preset T_{c_n}. Specifically, when the parameter T_{c_n} is set to 0.0001, 0.001, 0.01, and 0.1 s, the residual error of ISPTAF-ZNN reaches 10^{-7}, 10^{-6}, 10^{-6}, and 10^{-5} within preset times of 0.0001, 0.001, 0.01, and 0.1 s, respectively. This result indicates that ISPTAF-ZNN can achieve a solution frequency of 1×10^4 Hz in real-time TVQP problem solving, making it highly suitable for high-precision and high-frequency computing scenarios such as robot foot placement.

Table 2. Comparison of convergence accuracy and convergence time of different ξ values of AF under the disturbance $\Delta = \frac{3}{2}\sin(t)$.

		Convergence Accuracy	**Convergence Time**
AF	ξ	$t > 1.5$	$t > 1.5$
ISPTAF	8	4.2×10^{-6}	0.1
	6	7.2×10^{-6}	0.1
	4	6.5×10^{-6}	0.1
PTAF	8	1.9×10^{-5}	0.1
	6	2.1×10^{-5}	0.1
	4	2.4×10^{-5}	0.1
SEPAF	8	1.1×10^{-5}	0.19
	6	7.8×10^{-6}	0.32
	4	9.8×10^{-6}	0.408
SBPAF	8	1.5×10^{-5}	0.18
	6	5.7×10^{-6}	0.315
	4	3.2×10^{-5}	0.39

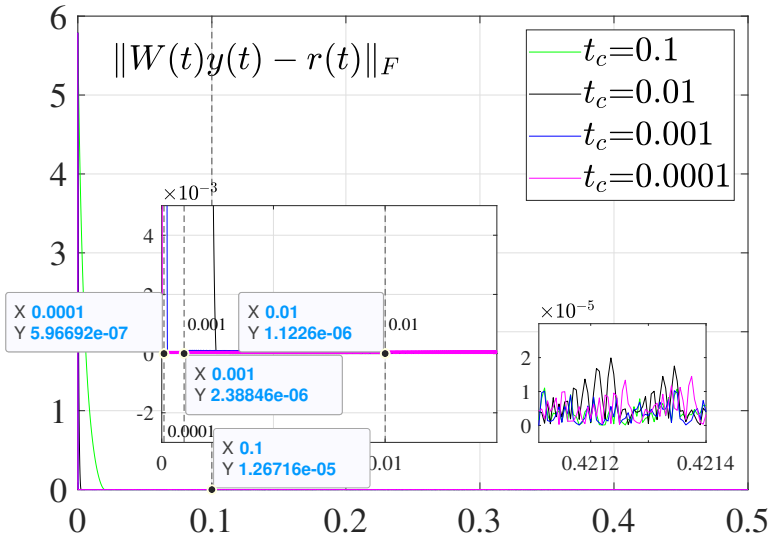


Figure 5. The convergence time and accuracy of the ZNN (16) driven by ISPTAF (18) when solving TVQP (32) under the noise $\Delta = \frac{3}{2}\sin(t), t > 0$ for different values of T_{c_n}.

5.1.3. Foot Placement Optimization VERIFICATION

This section performs gait optimization performance verification for the lightweight tendon-driven robot X02 using an AFP-QP scheme (3)–(6) and ISPTAF-ZNN (18) solver. The comparative experiment uses the gait planning formula (2) without an optimization algorithm.

First, we export the mechanical design files of the X02 into Unified Robot Description Format (URDF) and eXtensible Markup Language (XML). The URDF is used for calculating the kinematic and dynamic states required for robot simulation, and the XML is used to load the robot model in the simulation platform MuJoCo. The running frequency of the optimization algorithm is 1 kHz. The ISPTAF-ZNN solver parameters, except for $T_{cn} = 10^{-4}$ s, are set the same as in Section 5. The parameters in the optimization problem are as follows: the swing foot gait cycle parameter $T = 0.25$ s, robot mass $m = 24.19$ kg, standing height $z_c = 0.86$ m, initial adaptive parameter $\alpha = 0$, and gait limit $\Delta p_x = 0.15$ m (Figure 6).

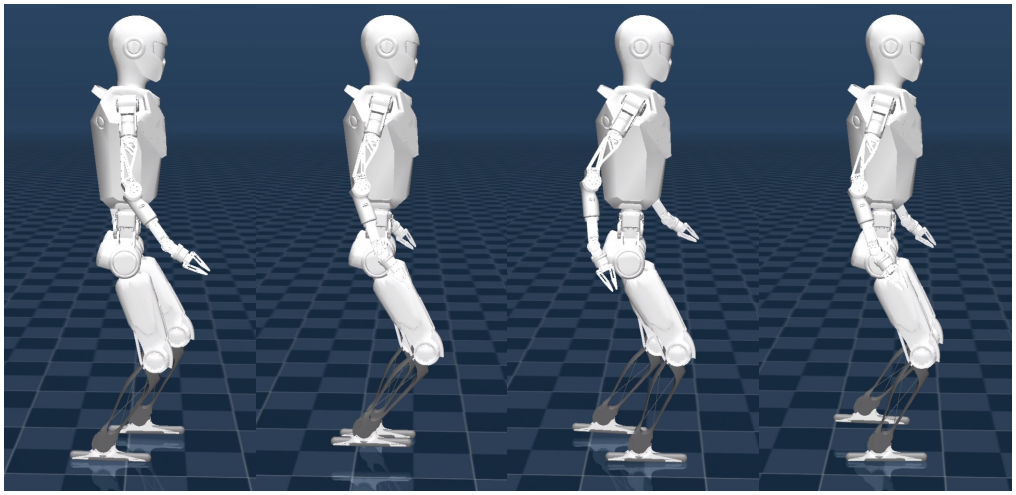


Figure 6. X02 motion trajectory.

Figure 7 shows the performance differences in the gait planning in the x direction with and without the optimization algorithm. After the experiment starts, the robot performs place stepping, and around 6 s, the desired velocity $v_x^{\text{des}} = 0.32$ m/s is triggered. By comparing Figure 7a,b, it is observed that after the velocity trigger, the footstep positions in the x direction using the optimization algorithm exhibit a smooth increase, and the footstep size is consistent in each gait cycle. However, the results without using the optimization method show violent oscillations, followed by fluctuations in each gait cycle. By comparing Figure 7c–f, it is found that the velocity tracking and angular momentum tracking with the optimization algorithm show asymptotic convergence without overshoot. In contrast, the velocity and angular momentum tracking without the optimization method show an overshoot of approximately 40% and 41%, respectively. Figure 7g shows that after the velocity trigger, the smoothing coefficient γ increases rapidly to 1. According to Equation (4), this is because a large angular momentum discrepancy drives the optimization algorithm to solve for a larger γ, which in turn reduces the step effect of the desired angular momentum L_y^{des}. As γ increases, the solver ensures smooth and gradual convergence to the desired velocity, preventing oscillations or falls caused by large strides.

Figure 8 shows the performance differences in the y-direction foot placement with and without the optimization algorithm proposed. After the experiment starts, the robot performs place-stepping, and the desired velocity $v_x^{\text{des}} = 0.2$ m/s is triggered around the 6th second. Unlike the x-direction results, the y-direction footstep alternates periodically between the left and right feet. From subplots Figure 8a,e, it can be seen that the footstep in the y-direction with the optimization algorithm keeps the robot stable within the step limit of 0.15 m, and the error between the desired angular momentum $\hat{L}_x^{\text{des}}$ and the estimated angular momentum $\hat{L}_x$ is small. This helps maintain the robot's position and posture stability, avoiding large fluctuations and excessive energy loss. However, from subplots Figure 8b,f, it can be seen that without the optimization algorithm, the footstep in the y-direction reaches up to 0.25 m on one side, causing an angle between the leg and the ground, generating lateral impact forces, and increasing the estimated angular momentum $\hat{L}_x$, which in turn increases the risk of system instability.

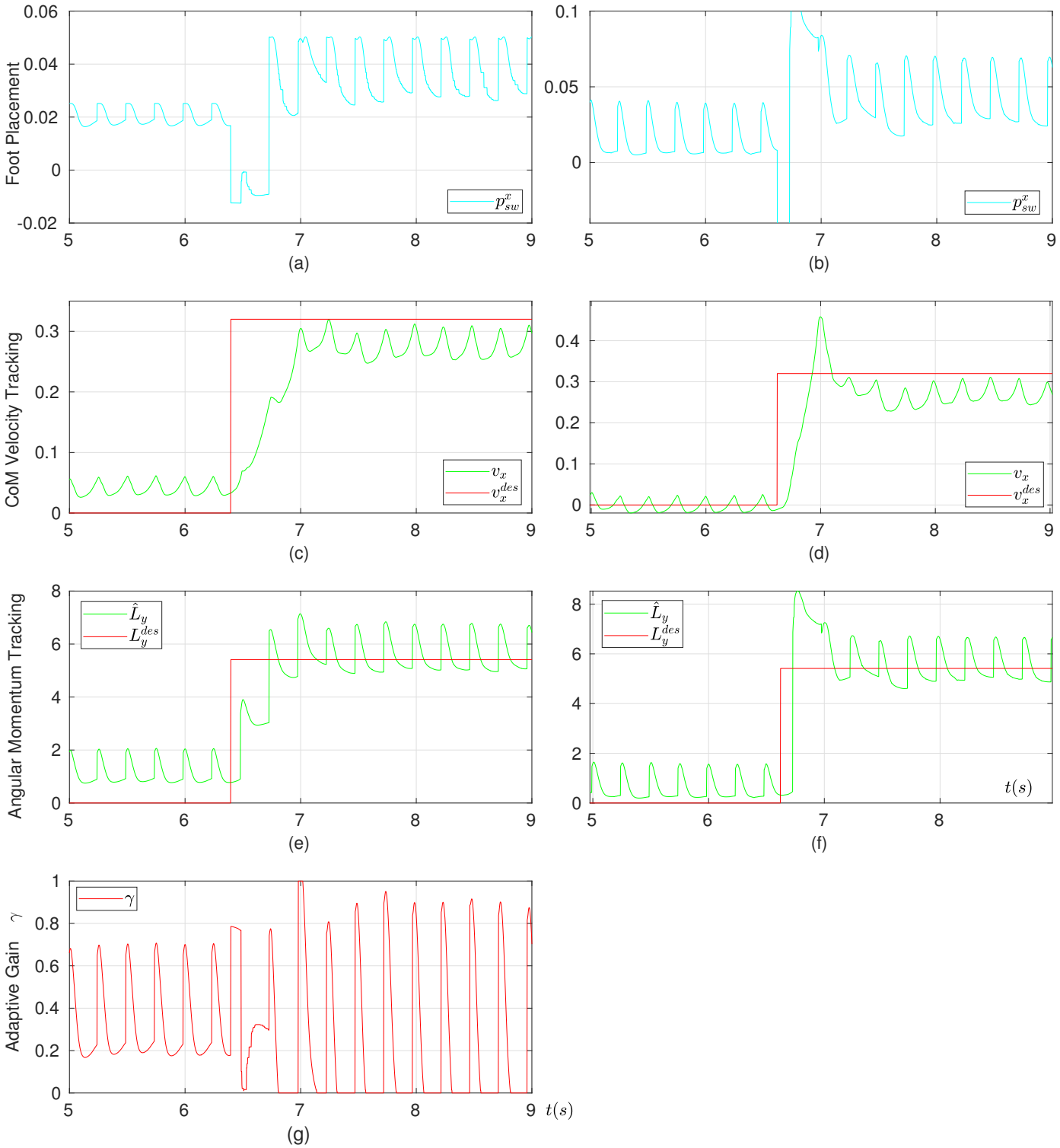


Figure 7. ZNN (16) x-direction foot placement results. (**a**,**c**,**e**,**g**) are performance plots using the optimization algorithm (3)–(6); (**b**,**d**,**f**) are performance plots without using the optimization algorithm.

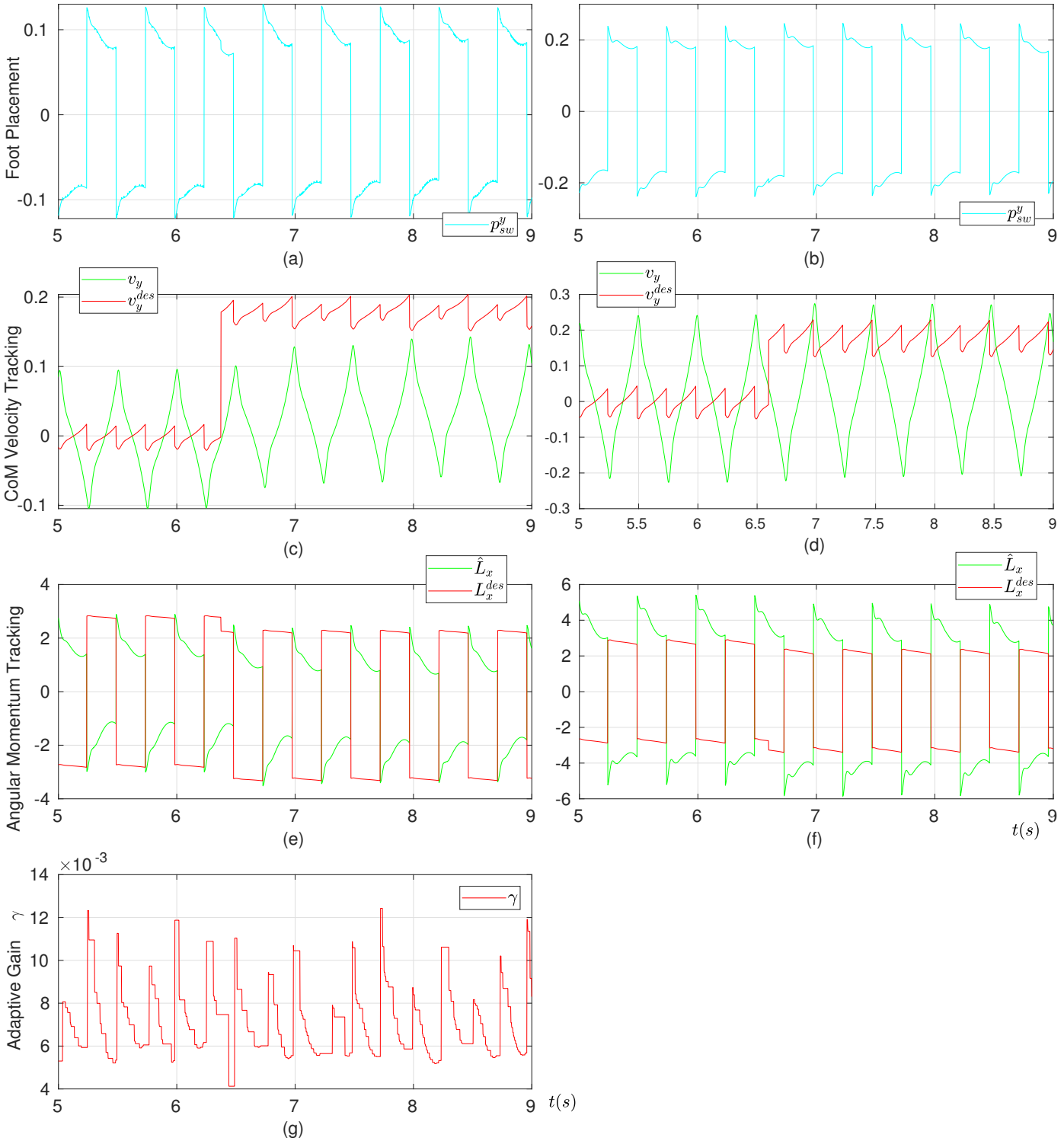


Figure 8. ZNN (16) y-direction foot placement results. (**a**,**c**,**e**,**g**) are performance plots using the optimization algorithm (3)–(6); (**b**,**d**,**f**) are performance plots without using the optimization algorithm.

6. Conclusions

This paper presents a novel ZNN-based gait optimization strategy for humanoid robots with ALIP and inequality constraints. The proposed method effectively converts the ALIP-based gait planning problem into a TVQP problem and incorporates physical constraints and adaptive adjustment factors to enhance system stability and robustness. We designed a piecewise nonlinear AF, ISPTAF, combining the SEPAF and SBPAF, which ensures predefined-time convergence and disturbance rejection. The algorithm was evaluated

using Lyapunov stability theory, demonstrating its theoretical soundness. Experimental results confirmed the convergence, robustness, and real-time performance of the solver. Moreover, comparative foot placement experiments on the X02 humanoid robot showed that the proposed algorithm significantly improves gait stability, reducing overshoot and energy loss compared to traditional approaches without optimization.

Although the proposed ISPTAF-ZNN solver demonstrates strong performance in gait optimization for humanoid robots, there are several directions for future research. First, further optimization of the ISPTAF-ZNN activation function can be explored to improve computational efficiency, especially for high-frequency real-time control scenarios. Additionally, integrating this approach with other control frameworks, such as reinforcement learning or hybrid MPC-based control strategies, could further enhance the adaptability and robustness of the robot's foot placement under more complex and dynamic environments. Future work can also involve testing the method in real-world scenarios to evaluate its performance under unexpected disturbances and various environmental conditions. Lastly, exploring the application of this optimization strategy in multi-robot systems or collaborative tasks would be an exciting avenue for extending its utility in broader robotic applications.

Author Contributions: Conceptualization, Y.L. and Q.L.; Methodology, Y.L. and Q.L.; Formal analysis and investigation, Y.L.; Writing—original draft preparation, Y.L.; Writing—review and editing, Q.L.; Software, Y.L., H.M. and H.J. Resources, H.M. Supervision, Q.L. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant 92048205 and the Pujiang Talents Plan of Shanghai under grant 2019PJD035.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: Thank you to Shanghai Droid Robot Co., Ltd. for their assistance in robot debugging.

Conflicts of Interest: Author Haiming Mou was employed by the company Shanghai Droid Robot Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Nomenclature

ALIP	Angular Momentum Linear Inverted Pendulum
AF	Activation Function
AFP-QP	Adaptive Foot Placement Quadratic Programming
CoM	Center of Mass
GNN	Gradient-based Neural Networks
NCP	Nonlinear Complementarity Problem
ISPTAF	Integral Strong Predefined-time Activation Function
RNN	Recurrent Neural Networks
LIP	Linear Inverted Pendulum
TVQP	Time-Varying Quadratic Programming
SAPAF	Sign-bi-power Activation Function
SEPAF	Sign-exponential-power Activation Function
ZNN	Zeroing Neural Networks

References

1. Hu, K.; Chen, Z.; Kang, H.; Tang, Y. 3D vision technologies for a self-developed structural external crack damage recognition robot. *Autom. Constr.* **2024**, *159*, 105262. [CrossRef]
2. Wang, T.; Chevallereau, C. Stability analysis and time-varying walking control for an under-actuated planar biped robot. *Robot. Auton. Syst.* **2011**, *59*, 444–456. [CrossRef]
3. Westervelt, E.R.; Grizzle, J.W.; Koditschek, D.E. Hybrid zero dynamics of planar biped walkers. *IEEE Trans. Autom. Control* **2003**, *48*, 42–56. [CrossRef]
4. Powell, M.J.; Ames, A.D. Mechanics-based control of underactuated 3D robotic walking: Dynamic gait generation under torque constraints. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 555–560.
5. Gong, Y.; Grizzle, J. One-step ahead prediction of angular momentum about the contact point for control of bipedal locomotion: Validation in a lip-inspired controller. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2832–2838.
6. Wang, W.; Liu, C.; Zhao, H.; Song, Z. Improved Deadbeat-Direct Torque and Flux Control for PMSM With Less Computation and Enhanced Robustness. *IEEE Trans. Ind. Electron.* **2023**, *70*, 2254–2263. [CrossRef]
7. Wang, Z.; Chai, J.; Sun, X.; Lu, H. Predictive deviation filter for deadbeat control. *IET Electr. Power Appl.* **2020**, *14*, 1041–1049. . [CrossRef]
8. Zheng, L.; Yu, W.; Xu, Z.; Zhang, Z.; Deng, F. Design, Analysis, and Application of a Discrete Error Redefinition Neural Network for Time-Varying Quadratic Programming. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 13646–13657. [CrossRef]
9. Zhang, Z.; Chen, T.; Zheng, L.; Luo, Y. A quadratic programming based neural dynamic controller and its application to UAVs for time-varying tasks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6415–6426. [CrossRef]
10. Wang, B.; Sun, S.; Ren, W. Distributed continuous-time algorithms for optimal resource allocation with time-varying quadratic cost functions. *IEEE Trans. Control Netw. Syst.* **2020**, *7*, 1974–1984. [CrossRef]
11. Zhang, Z.; Lu, Y.; Zheng, L.; Li, S.; Yu, Z.; Li, Y. A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. *IEEE Trans. Autom. Control* **2018**, *63*, 4110–4125. [CrossRef]
12. Yi, Z. *Convergence Analysis of Recurrent Neural Networks*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 13.
13. Wen, U.P.; Lan, K.M.; Shih, H.S. A review of Hopfield neural networks for solving mathematical programming problems. *Eur. J. Oper. Res.* **2009**, *198*, 675–687. [CrossRef]
14. Zhang, Y.; Jiang, D.; Wang, J. A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Trans. Neural Networks* **2002**, *13*, 1053–1063. [CrossRef]
15. Jin, L.; Li, S. Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints. *Neurocomputing* **2017**, *267*, 107–113. [CrossRef]
16. Li, W.; Wu, H.; Jin, L. A Lower Dimension Zeroing Neural Network for Time-Variant Quadratic Programming Applied to Robot Pose Control. *IEEE Trans. Ind. Inform.* **2024**, *20*, 11835–11843. [CrossRef]
17. Zuo, Q.; Li, K.; Xiao, L.; Li, K. Robust finite-time zeroing neural networks with fixed and varying parameters for solving dynamic generalized Lyapunov equation. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *33*, 7695–7705. [CrossRef]
18. Li, W.; Ma, X. Comparisons of Slack Variable, NCP Function, and Penalty Function Based ZNNs for Solving Equality-and Inequality-Constrained QP Problems With Robotic Applications. In Proceedings of the 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 3–5 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 226–231.
19. Li, S.; Chen, S.; Liu, B. Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function. *Neural Process. Lett.* **2013**, *37*, 189–205. [CrossRef]
20. Xiao, L.; Zhang, Y.; Jia, L.; He, Y. A ZNN-based sliding mode control strategy for solving synchronization problems of chaotic systems with application in memristor oscillator. *J. Frankl. Inst.* **2023**, *360*, 12101–12124. [CrossRef]
21. Jin, J.; Chen, W.; Qiu, L.; Zhu, J.; Liu, H. A noise tolerant parameter-variable zeroing neural network and its applications. *Math. Comput. Simul.* **2023**, *207*, 482–498. [CrossRef]
22. Li, W.; Liao, B.; Xiao, L.; Lu, R. A recurrent neural network with predefined-time convergence and improved noise tolerance for dynamic matrix square root finding. *Neurocomputing* **2019**, *337*, 262–273. [CrossRef]
23. Jin, J.; Chen, W.; Ouyang, A.; Yu, F.; Liu, H. A time-varying fuzzy parameter zeroing neural network for the synchronization of chaotic systems. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *8*, 364–376. [CrossRef]
24. Jin, J.; Zhu, J.; Gong, J.; Chen, W. Novel activation functions-based ZNN models for fixed-time solving dynamirc Sylvester equation. *Neural Comput. Appl.* **2022**, *34*, 14297–14315. [CrossRef]
25. Gong, Y.; Grizzle, J.W. Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion. *J. Dyn. Syst. Meas. Control* **2022**, *144*, 121006. [CrossRef]
26. Rockafellar, R.T. Lagrange multipliers and optimality. *SIAM Rev.* **1993**, *35*, 183–238. [CrossRef]

27. Chen, B.; Chen, X.; Kanzow, C. A penalized Fischer-Burmeister NCP-function. *Math. Program.* **2000**, *88*, 211–216. [CrossRef]
28. Wang, J.; Zhang, Y. Recurrent neural networks for real-time computation of inverse kinematics of redundant manipulators. In *Machine Intelligence: Quo Vadis?*; World Scientific: Singapore, 2004; pp. 299–319.
29. Cui, Z.; Huang, Y.; Li, W.; Chiu, P.W.Y.; Li, Z. Noise-Resistant Adaptive Gain Recurrent Neural Network for Visual Tracking of Redundant Flexible Endoscope Robot with Time-Varying State Variable Constraints. *IEEE Trans. Ind. Electron.* **2023**, *71*, 2694–2704. [CrossRef]
30. Song, Y.; Zheng, L.; Zhang, Z. An anti-disturbance integral recursive neural network for solving time-varying matrix inversion. In Proceedings of the 2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanchang, China, 28–30 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 271–275.
31. Utkin, V.; Shi, J. Integral sliding mode in systems operating under uncertainty conditions. In Proceedings of the Proceedings of 35th IEEE Conference on Decision and Control, Kobe, Japan, 13 December 1996; IEEE: Piscataway, NJ, USA, 1996; Volume 4, pp. 4591–4596.
32. Sánchez-Torres, J.D.; Gómez-Gutiérrez, D.; López, E.; Loukianov, A.G. A class of predefined-time stable dynamical systems. *IMA J. Math. Control Inf.* **2018**, *35*, i1–i29. [CrossRef]
33. Li, W.; Ma, X.; Luo, J.; Jin, L. A strictly predefined-time convergent neural solution to equality-and inequality-constrained time-variant quadratic programming. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 4028–4039. [CrossRef]

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.