machines

MDPI

Article

An RBF-L1-WBC Approach for Bipedal Wheeled Robots

Renyi Zhou [1,2], Yisheng Guan [1], Xiaoqun Chen [3], Haobin Zhu [3], Qianwen Cao [3], Guangcai Ma [1,2], Tie Zhang [4] and Shouyan Chen [3,*]

[1] School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China
[2] Advanced Technology Department, Guangzhou Automobile Group Co., Ltd., Guangzhou 510000, China
[3] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China
[4] School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China
* Correspondence: maxcsy@gzhu.edu.cn

Abstract

Bipedal wheeled robots combine the advantages of wheeled mobility and legged agility, enabling high-speed locomotion and obstacle negotiation in complex environments. However, their dynamic behavior is inherently unstable and highly coupled, making robust control particularly challenging in the presence of task conflicts, external disturbances, and modeling uncertainties. This paper proposes an RBF–L1–WBC framework that integrates L1 adaptive control to compensate for model inaccuracies and disturbances, radial basis function (RBF) neural networks to approximate nonlinear variations in linear quadratic regulator (LQR) gains, and whole-body control (WBC) to coordinate multiple tasks while mitigating control conflicts. Experimental findings confirm that the proposed methodology yields statistically significant improvements in both attitude regulation precision and velocity tracking accuracy, surpassing the performance of benchmark controllers including classical LQR, adaptive LQR, and classical Virtual Model Control (VMC).

Keywords: bipedal wheeled robots; dynamic modeling; whole-body control; L1 adaptive control; linear quadratic regulator

1. Introduction

Wheeled-legged robots integrate the structural advantages of both wheeled and legged configurations, providing strong mobility, efficient obstacle negotiation, and high traversal speed. Among them, two-wheeled-legged robots are inherently unstable, nonlinear systems that require active control to maintain equilibrium [1]. Their dynamics exhibit strong coupling, time-varying characteristics, and multi-objective control requirements. Contemporary research commonly models bipedal wheeled-legged robots using inverted pendulum approximations and employs methods such as learning-based control [2], model predictive control [3], or linear quadratic regulator (LQR) [4,5].

Liu et al. [6] constructed a simplified second-order inverted pendulum model combined with an LQR controller to achieve dynamic balance under varying-height conditions. Kim and Yeom [7] realized coordinated wheel–leg motion by explicitly modeling the dynamic coupling among the wheel, leg, and arm within a model predictive control framework, enabling the mitigation of control conflicts in multi-actuator outputs. Xin et al. [8] introduced a composite control system that integrates uncertainty and disturbance assessment with virtual model control (VMC) [9], addressing motion balance and disturbance rejection under whole-body dynamic coupling and time-varying parameters. Guo et al. [10]

Check for updates

Academic Editors: Yugang Liu and Sidney Givigi

Received: 13 January 2026
Revised: 11 February 2026
Accepted: 14 February 2026
Published: 15 February 2026

Copyright: © 2026 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.

developed a fuzzy proportional–derivative jumping controller combined with an LQR controller to enhance obstacle-crossing performance and terrain adaptability. However, variations in leg length significantly alter the center-of-mass height and position, making a first-order inverted pendulum model insufficient to capture the coupled wheel–leg dynamics.

Whole-body control (WBC) [11], originally developed for legged robots, has been adapted to wheeled-legged platforms to handle multi-task coordination and potential control conflicts. Typical WBC approaches formulate a quadratic programming problem with inequality constraints derived from control objectives and dynamic constraints to determine feasible reaction forces and joint torques [12]. Techniques such as null-space projection enable the simultaneous execution of lower-priority tasks within the high-priority task structure, effectively handling multi-objective control in redundant, nonlinear systems. Klemm et al. [13] developed a hierarchical WBC framework for the two-legged wheeled robot Ascento, enabling real-time jump planning and coordinated wheel–leg–CoM control. Wang et al. [14] proposed a composite design incorporating linear output regulation, nonlinear damping allocation, and passive control to address whole-body dynamic coupling and disturbance rejection during height-varying motions. Xin et al. [15] further introduced a whole-body dynamic control framework for underactuated wheeled-legged robots to improve coordination under strong coupling, dynamic balance demands, and steering-induced disturbances. Follow-up work extended this into a hierarchical torque-control architecture that enhances adaptive locomotion over unfamiliar uneven terrain without external sensing [16]. Zhang et al. [17] adopted adaptive optimal output regulation for the wheeled-legged robot Ollie using a data-driven approach to cope with model inaccuracies and disturbances. Nevertheless, WBC approaches generally require relatively accurate dynamic models and impose high computational costs when addressing uncertainties.

Although the WBC framework effectively resolves multi-objective control conflicts, its performance critically depends on accurate dynamic models. In the presence of uncertainties such as model inaccuracies and external disturbances, WBC often requires computationally intensive online optimization, leading to substantial computational overhead that hampers real-time control implementation. Existing adaptive control schemes based on radial basis function (RBF) [18] networks, while capable of handling uncertainties, typically focus on parameter estimation for single control objectives and have not been effectively integrated with WBC to address model uncertainties. Moreover, conventional RBF networks, though competent in approximating linear disturbances, exhibit limited prediction accuracy when confronted with highly nonlinear and time-varying complex disturbances, thus failing to achieve precise disturbance compensation.

To address these limitations, this paper proposes a novel hybrid control framework termed RBF-L1-WBC, with principal contributions and innovations in three aspects. First, departing from existing RBF-based adaptive controllers, we innovatively leverage the nonlinear approximation capability of RBF networks to perform piecewise smooth fitting of offline-designed LQR control gains, rather than employing them directly for online parameter estimation or disturbance compensation. This approach substantially reduces the online computational burden associated with real-time optimal gain calculation in conventional WBC. Owing to the parsimonious hyperparameter configuration and lightweight architecture of RBF networks, the proposed method enhances deployment feasibility and enables real-time control on embedded platforms. Second, in contrast to classical WBC, the proposed framework incorporates an L1 adaptive law [19] as a fast compensator at the upper layer of WBC, dedicated to real-time estimation and compensation of model errors and external disturbances. This decouples the slow WBC optimization from the fast L1 adaptive compensation, effectively mitigating the framework's overreliance on

model precision and significantly improving system robustness. Finally, by synergistically integrating RBF gain scheduling, L1 fast adaptation, and WBC multi-objective optimization, the proposed framework addresses three critical challenges—computational efficiency, disturbance robustness, and task conflict resolution—simultaneously, leading to a marked improvement in overall control performance.

2. Robot Modeling and Problem Specification

2.1. Robotic Model

Figure 1 illustrates the bipedal wheeled robot model and the coordinate system used in this study, while Table 1 lists the corresponding Denavit–Hartenberg (DH) [20] convention parameters defined for the robot.The numbers in the figure represent the identifiers for each coordinate system.

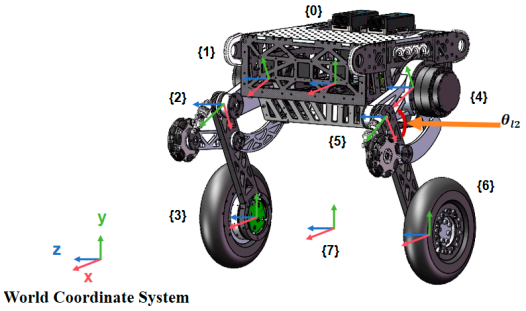


Figure 1. Bipedal wheeled robot model.

Table 1. DH parameters of the bipedal wheeled robot.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0°	0	$D/2$	θ_{r1}
2	0°	l_{r1}	0	θ_{r2}
3	0°	l_{r2}	0	θ_{r3}
4	0°	0	$-D/2$	θ_{l1}
5	0°	l_{l1}	0	θ_{l2}
6	0°	l_{l2}	0	θ_{l3}
7	0°	0	$-D/2(D/2)$	$\theta_{37}(\theta_{67})$

Based on the link structure of the bipedal wheeled robot, kinematic modeling is performed using the D-H method.

Here, i denotes the index of each link coordinate frame, $\boldsymbol{\alpha}$ is the link twist, $\boldsymbol{a}$ represents the link length, $\boldsymbol{d}$ is the link offset, and $\boldsymbol{\theta}$ is the joint angle. D denotes the distance between the two wheels. The subscripts for $\boldsymbol{\theta}$ are defined as follows: θ_{r1} corresponds to the right hip joint angle, θ_{r2} to the right knee joint angle, and θ_{r3} to the right wheel angle. Similarly, θ_{l1} denotes the left hip joint angle, θ_{l2} the left knee joint angle, and θ_{l3} the left wheel angle. The relationships $\theta_{37} = 0 - (\theta_{r1} + \theta_{r2} + \theta_{r3})$ and $\theta_{67} = 0 - (\theta_{l1} + \theta_{l2} + \theta_{l3})$ are established to maintain kinematic consistency.

Derive the kinematic relationship between the trunk and the two wheel hubs of the bipedal wheeled robot using the homogeneous transformation formulation.

$$
{}^{0}_{3}T = {}^{0}_{1}T{}^{1}_{2}T{}^{2}_{3}T = \begin{bmatrix} c_{r1r2r3} & -s_{r1r2r3} & 0 & l_{r2}c_{r1r2} + l_{r1}c_{r1} \\ s_{r1r2r3} & c_{r1r2r3} & 0 & l_{r2}s_{r1r2} + l_{r1}s_{r1} \\ 0 & 0 & 1 & D/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

Here, c_{li} and s_{li} denote $\sin(\theta_i)$ and $\cos(\theta_i)$ of the i-th joint of the left leg. The tilt angle of the robot's center of gravity is determined using the following formula.

$$^{0}X(q) = \frac{\sum m_i {}^{0}X_i(q)}{\sum m_i} \tag{2}$$

In the equation, $q = [\theta_{r1}, \theta_{r2}, \theta_{r3}, \theta_{l1}, \theta_{l2}, \theta_{l3}]$, represents the joint variable vector delineating the robot's current posture, where each joint angle θ_i obtained as the difference between its initial angle and the angle measured by the encoder; In Figure 1, the measurement convention for joint angles is explicitly illustrated using the left knee joint as an example. Specifically, the angle θ_{l2} is defined as the rotational displacement from the initial configuration to the current pose. Owing to the bilateral symmetry of the robot's kinematic architecture, the measurement conventions for the remaining joints—such as the right knee joint, left hip joint, and right hip joint—are identical to that of the left knee joint. To avoid visual clutter and redundancy in the illustration, these conventions are not individually annotated for each joint. m_i denotes the mass of the i-th component; $^{0}X_i(q)$ indicates the position of the center of mass of the i-th component relative to the coordinate system {0} when the robot assumes stance q, this angle can be obtained using Equation (1). The wheeled-legged robot is classified as a wheeled inverted pendulum, with its center of mass tilt angle illustrated in Figure 2.

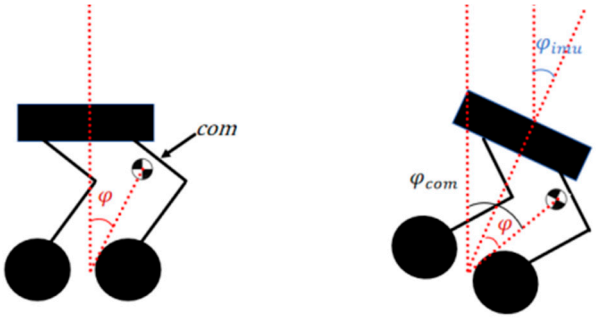


Figure 2. Schematic of the bipedal wheeled robot's centroid tilt angle.

The pitch angle φ_{com} of the robot's center of mass and the equivalent pendulum length L in the world coordinate system can be calculated using the following expressions:

$$L = ||^{7}X(q)_{xy}||_2 \tag{3}$$

Here, the superscript 7 denotes that the position vector is expressed in the body coordinate frame {7}, whose origin is located at the midpoint between the two wheels. The subscript $(\cdot)_{xy}$ denotes the component of a vector projected onto the horizontal x–y plane of coordinate frame {7}.

Figure 2 φ_{imu} denotes the tilt measurement from the Inertial Measurement Unit (IMU) [21], φ represents the tilt angle of the robot's center of mass in the base coordinate frame {0}, and L indicates the distance from the center of mass to the midpoint of the line connecting the two wheels. The dynamic equations of the variable-length inverted pendulum model of the bipedal wheeled robot are derived based on force analysis.

$$\begin{cases} \left(M_{com} + 2m_w + 2\frac{I_{zw}}{r^2}\right)\ddot{x} + M_{com}L(q)\ddot{\varphi}_{com} = \frac{\tau_{l3}+\tau_{r3}}{r} \\ M_{com}L(q)\ddot{x} + \left(I_{zcom} + M_{com}L(q)^2\right)\ddot{\varphi}_{com} = M_{com}gL(q)\varphi_{com} - (\tau_{l3} + \tau_{r3}) \\ \left(\frac{2I_{ycom}}{D} + m_w D + \frac{I_{zw}D}{r^2}\right)\ddot{\gamma} = \frac{\tau_{l3}-\tau_{r3}}{r} \end{cases} \tag{4}$$

Here, m_w and M_{com} denote the masses of the wheel and the equivalent pendulum, respectively, I_{zw} is the moment of inertia of the wheel about the z-axis, while I_{ycom} and I_{zcom} represent the moments of inertia of the equivalent pendulum about the y-axes and z-axes, respectively. In (5), τ_{l3} and τ_{r3} denote the driving torques applied to the left and right wheel actuators, respectively. The variable γ represents the yaw angle of the robot body, and γ denotes its angular acceleration about the vertical axis. Let the state vector $X = [\, x, \dot{x}, \varphi_{com}, \dot{\varphi}_{com}, \gamma, \dot{\gamma} \,]^T$, x denotes the horizontal displacement of the robot along the forward direction, expressed in the base coordinate frame {0}. The output vector $Y = X$, and the control vector $U = (\tau_{l3}, \tau_{l3})^T$. For bipedal wheeled robots, $L(q)$ changes as the hip and knee joint angles are varied.

2.2. LQR-Based Balance Control for Legged-Wheeled Robots

A linear feedback controller $U = -KX$ is designed to regulate balance, locomotion, and turning in bipedal wheeled robotic systems. Rewriting Equation (4) yields the following state-space representation:

$$\dot{X} = (A - BK)X \tag{5}$$

The following quadratic performance metric is defined:

$$W = \int_0^\infty X^T Q X + U^T R U \mathrm{d}t \tag{6}$$

Here, Q and R denote positive definite matrices and positive definite diagonal matrices, respectively, representing the weighting of state errors and control inputs. Solving the Algebraic Riccati Equation (ARE) [22] to derive the auxiliary matrix P, the control gain K that minimizes the performance metric W can be calculated:

$$K = R^{-1} B^T P \tag{7}$$

Here, P is a symmetric positive definite matrix.

2.3. Leg Control of a Wheeled-Legged Robot Based on Virtual Model Control

A spring-damper virtual model is applied at the end of the robot's leg to achieve control objectives, including regulation of robot height and leg compliance, as illustrated in Figure 3.



Figure 3. Virtual spring-damper model of the robot leg.

The virtual force for each leg is defined as:

$$F_{yi} = k(y_{id} - y_i) + d(\dot{y}_{id} - \dot{y}_i) \tag{8}$$

Here, $i = l, r$; k and d represent the stiffness and damping coefficients of the virtual model, respectively. y_{id} and y_i denote the actual leg and desired lengths, respectively. The desired length is derived from the robot's desired height h_d and desired orientation R_d, i.e., $y_{id} = g(h_d, R_d)$. To fulfill the roll angle control requirement, a PID controller is applied to the virtual force to adjust the roll angle.

$$\begin{aligned} u(t) &= K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \\ F_{Yl} &= F_{yl} + u(t) \\ F_{Yr} &= F_{yr} - u(t) \end{aligned} \tag{9}$$

In these equations, $e(t) = r_d(t) - r(t)$ is the control error, where $r_d(t)$ denotes the desired roll angle θ_{roll_d}, $y(t)$ is the actual roll angle θ_{roll}, and $u(t)$ is the control input applied to the robotic leg. The virtual forces F_{yl} and F_{yr} for the left and right legs respectively can be obtained via Equation (8).

2.4. Problem Description

Pitch and Velocity Control: In the application of LQR for the balance control of bipedal wheeled robots, the control gain matrix K is determined by the system matrix A and the input matrix B via the ARE. When variations in the robot's hip and knee joint angles affect the center-of-mass $L(q)$ length and the equivalent pendulum mass M_{com} the feedback gain K requires recalibration. However, solving the ARE incurs significant computational and temporal costs, and the modeling process introduces an error ε. Therefore, the feedback gain can be expressed as:

$$K = R^1 B\Big(q, \varepsilon)^T P \tag{10}$$

Roll Angle and Height Control: For roll angle regulation of bipedal wheeled robots, VMC is applied. Modeling errors ε' arise from the simplification of the Coriolis force term, while variations in leg length $L(q)$ modify the control model. Consequently, the PID controller coefficients K_p, K_I, and K_d must be optimized. The optimal control input $u^*(t)$ is determined according to the corresponding optimal gain parameters $K_p(L)^*$, $K_i(L)^*$, and $K_d(L)^*$ for the current leg configuration.

$$u^*(t) = K_p(L)^* e(t) + K_i(L)^* \int_0^t e(\tau) d\tau + K_d(L)^* \frac{de(t)}{dt} \tag{11}$$

Whole-Body Coordination Control: When traversing complex terrain, differences in leg elevation and forward velocity between the robot's left and right limbs may induce inconsistencies or conflicts in control outputs. To address this, RBF network is employed to approximate the control gains K for various system states, while adaptive techniques mitigate uncertainties. The WBC framework is then applied to resolve control conflicts and coupling among multiple tasks, ensuring coordinated locomotion and stable whole-body motion.

3. Control of a Two-Wheeled Legged Robot Utilizing RBF-L1 Adaptive-LQR and RBF-PID-VMC

This study presents the control architecture illustrated in Figure 4, a comprehensive summary diagram comparing all methods is provided in Figure 5.

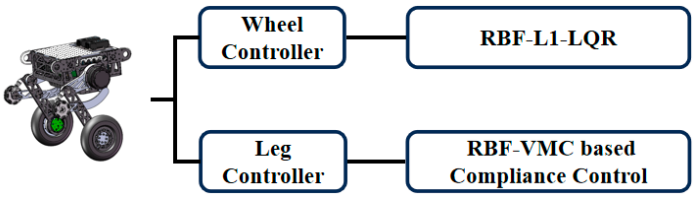


Figure 4. RBF-Enhanced Control Framework for a Bipedal-Wheeled Robot.

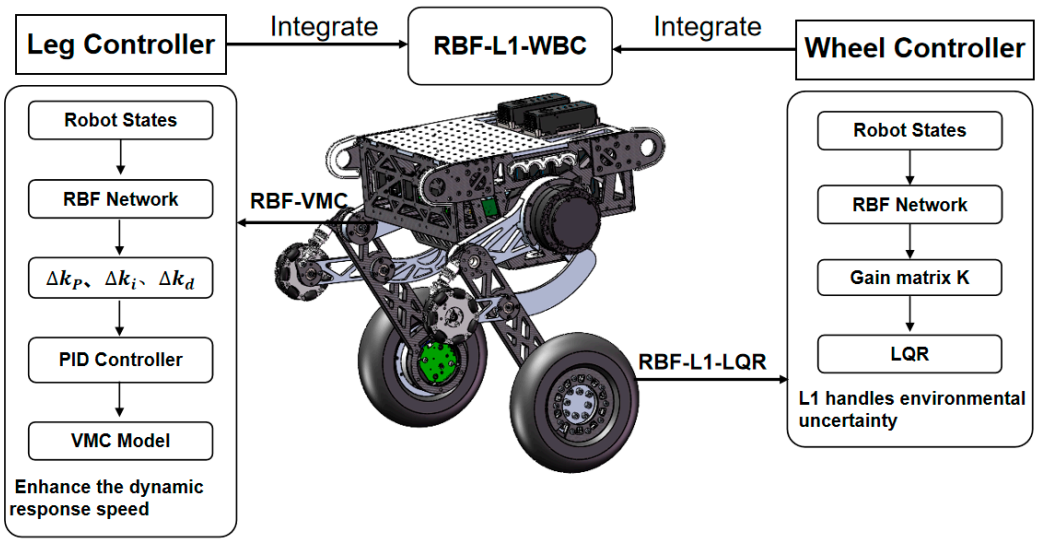


Figure 5. Comprehensive summary diagram comparing all methods.

3.1. Robot Balance Control Based on RBF-L1 Adaptive-LQR

This study addresses the challenge of dynamic computation of the feedback gains K by employing a segmented approximation approach utilizing RBF neural network for both K and the center-of-mass tilt angle φ_{com}, complemented with real-time compensation using an L1 adaptive control strategy.

3.1.1. RBF-Based LQR Control Parameter Tuning Method

In conventional LQR control, the feedback gain matrix Kis typically obtained by solving the ARE. However, as robotic motion control systems grow in complexity—particularly when encountering varying terrain heights or complex topographical changes—the robot's center-of-mass (CoM) parameters may undergo significant variations. Under such conditions, the LQR approach often requires frequent recalculation and readjustment of K, leading to computational bottlenecks. Moreover, as established in the preceding equivalent CoM analysis, determining the equivalent CoM parameters entails extensive matrix operations, which inevitably impose additional real-time computational burdens. To address the need for dynamic adjustment of both the feedback gain Kand the equivalent CoM parameters in response to changes in the robot's state, this paper employs a piecewise fitting strategy based on RBF neural networks. This approach enables adaptive regulation of control parameters, reduces computational overhead, and enhances the system's response efficiency.

The RBF neural network employed in this study comprises three layers: an input layer, a hidden layer, and an output layer, as illustrated in Figure 6. The weights of the network are then updated using the gradient descent algorithm. The RBF neural network in this experimental setup is configured with the following parameters: a learning rate $\eta = 10^{-9}$, a momentum factor $\alpha = 0.05$, a three-layer architecture, and six neurons. This specific parameter combination was determined through extensive preliminary experiments and

grid search optimization, achieving an optimal trade-off between control precision and real-time performance. The extremely low learning rate (10^{-9}) ensures gradual and stable parameter updates, effectively preventing oscillations during dynamic robot control. Meanwhile, the momentum factor accelerates the convergence process while mitigating the risk of becoming trapped in local optima.

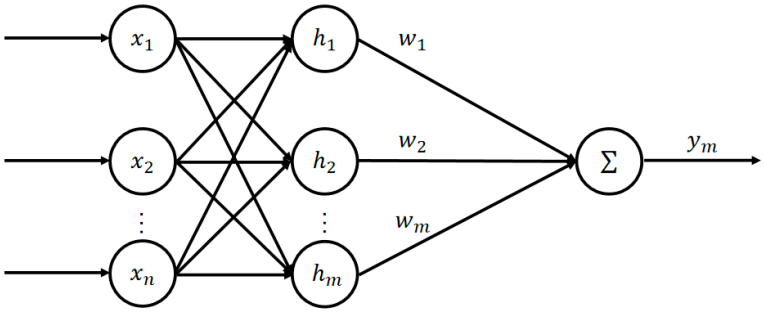


Figure 6. Structure of the RBF Neural Network.

To meet the high demands for computational efficiency and accuracy in real-time robot control, this study adopts a single-hidden-layer RBF neural network structure, primarily based on a comprehensive consideration of computational complexity adaptability, function approximation capability matching, and real-time constraints: robot control requires parameter fitting to be completed within milliseconds, and the RBF's single-hidden-layer structure only involves kernel function activation and linear weighting, significantly reducing computational load compared to multi-layer perceptrons; meanwhile, the fitting targets are the state-parameter mapping relationships of the robot, for which the RBF's local activation mechanism can efficiently capture their nonlinear characteristics.

The specific implementation is as follows: the optimal feedback gain matrix K and corresponding centroid parameters are fitted for different robot states, enabling real-time adaptive adjustment of control gains and centroid positions, thereby enhancing the system's responsiveness and control performance in complex environments. First, a systematic traversal of all possible joint angle combinations of the robot is conducted, covering typical motion states within its workspace. During the traversal, the key parameters (i.e., equivalent centroid inclination angle, control feedback gain, etc.) corresponding to each joint setup are precisely calculated, and the results are systematically stored to construct a dataset. This dataset comprehensively reflects the important feature parameters related to dynamic control under different robot postures. Subsequently, an RBF neural network is introduced to perform nonlinear function approximation on the constructed dataset. Through training, the network accurately fits the functional relationships between the robot's various states and its equivalent centroid inclination angle and control feedback gains. By using this fitting function, frequent real-time matrix operations and ARE solutions are avoided during the real-time control process, reducing computational burden, improving the real-time performance of the control system, and ensuring control accuracy. As shown in Figure 7, the implementation workflow is visualized for clarity:

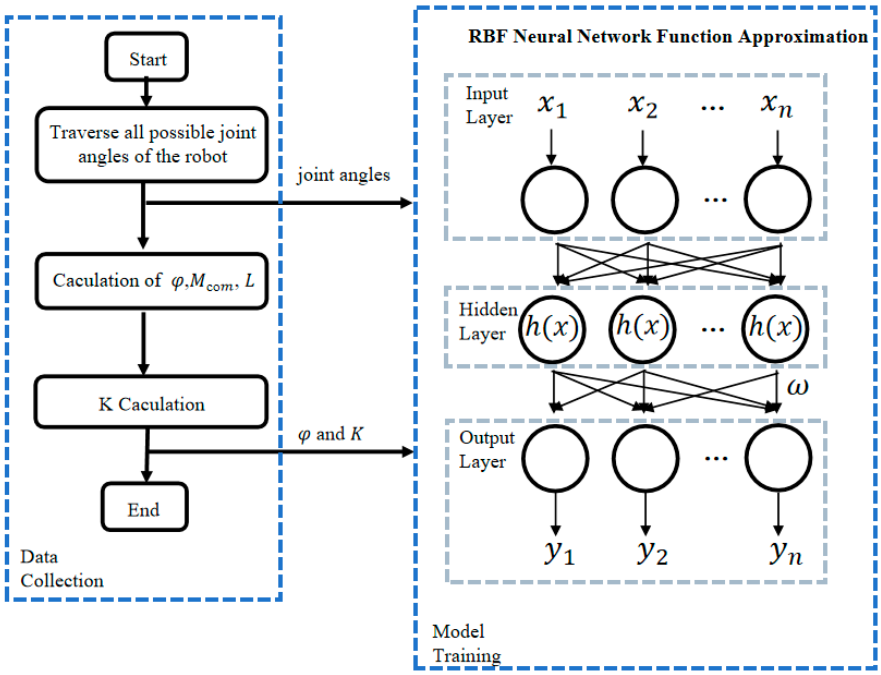


Figure 7. RBF-Integrated LQR Gain and Center-of-Mass Regulation Strategy.

3.1.2. The L1-Augmented LQR Controller

As established in Section 2.2, conventional LQR designs the feedback gain matrix K based on the system's dynamic equation (i.e., the state-space model) and the weighting matrices Q and R in the quadratic performance index, subsequently yielding the control output u. This demonstrates that LQR performance is inherently dependent on the accuracy of the system model. In practice, however, model inaccuracies—such as errors in the state-space matrices A and B—and the presence of external disturbances are often unavoidable, and these factors can significantly compromise the stability of robotic control. To address this limitation, this section integrates adaptive control principles with LQR, thereby enhancing the system's robustness against model uncertainties and external disturbances. The corresponding control block diagram is illustrated in Figure 8.

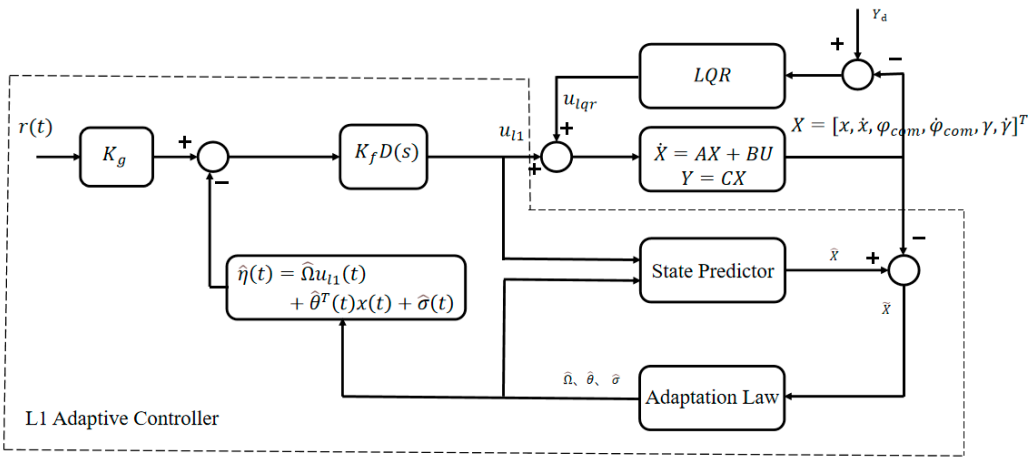


Figure 8. Block Diagram of the L1-Enhanced LQR Controller.

Define $A_m = A - BK$, then according to Formula (7) we have: $\dot{X} = A_m X$, $Y = CX$. Here, A_m is a Hurwitz matrix by design. Now, consider introducing model uncertainty terms θ and Ω, as well as an external disturbance term σ. The system dynamics can be expressed as:

$$\begin{aligned} \dot{X} &= \left(A_m + B\theta^T\right)X + B\Omega u_{l1} + B\sigma \\ Y &= CX \end{aligned} \tag{12}$$

Here, the three terms $B\Omega u_{l1}$, $B\theta^T X$, and $B\sigma$ represent the effects of uncertainty in the B matrix, uncertainty in the A matrix, and external disturbances, respectively. If the true values of θ, Ω, and σ can be accurately estimated, the system can be regarded as a precise model that accounts for both model inaccuracies and external disturbances. To this end, a state predictor is designed. Let $\hat{\theta}$, $\hat{\Omega}$, and $\hat{\sigma}$ denote the adaptive laws—i.e., the estimates of θ, Ω, and σ—and let $\hat{X}$ be the estimated system output. To ensure that $\hat{X}, \hat{\theta}$, $\hat{\Omega}$, $\hat{\sigma}$ converge to their true counterparts X, θ, Ω, σ, the following equations are formulated:

$$V(\widetilde{X}, \widetilde{\theta}, \widetilde{\sigma}, \widetilde{\Omega}) = \widetilde{X}^T P\widetilde{X} + \widetilde{\theta}^T \Gamma_\theta \widetilde{\theta} + \widetilde{\sigma}^T \Gamma_\sigma \widetilde{\sigma} + \widetilde{\Omega}^T \Gamma_\Omega \widetilde{\Omega} \tag{13}$$

Here, $\widetilde{X} = \hat{X} - X$. For the adaptive law, this paper presents the following mathematical expression:

$$\begin{aligned} \dot{\hat{\theta}} &= \mathrm{Proj}\left(\hat{\theta}, -\tfrac{1}{\Gamma_\theta} X B^T P \widetilde{X}\right), \hat{\theta}(0) = \hat{\theta}_0 \\ \dot{\hat{\sigma}} &= \mathrm{Proj}\left(\hat{\sigma}, -\tfrac{1}{\Gamma_\sigma} B^T P \widetilde{X}\right), \hat{\sigma}(0) = \hat{\sigma}_0 \\ \dot{\hat{\Omega}} &= \mathrm{Proj}\left(\hat{\Omega}, -\tfrac{1}{\Gamma_\Omega} u_2 B^T P \widetilde{X}\right), \hat{\Omega}(0) = 1 \end{aligned} \tag{14}$$

Here, Proj denotes the projection operator. Let $\Gamma_\theta > 0$, $\Gamma_\sigma > 0$, and $\Gamma_\Omega > 0$. Let P be the solution to the Lyapunov equation $A_m P + P A_m = -Q_a$, with $P = P^T > 0$ and $Q_a = Q_a^T > 0$. Based on the state predictor in Equation (13), we transform the system into the complex frequency domain (s-domain). As time approaches infinity, we have $\lim_{t\to\infty} s = 0$, and the following holds:

$$\hat{Y} = -C{A_m}^{-1} B\left(\hat{\Omega} u_{l1} + \hat{\theta}^T X + \hat{\sigma}\right) \tag{15}$$

To ensure that $\hat{Y} = r$, where r denotes the reference input, the following relationship can be derived:

$$u_{l1} = -\frac{1}{\hat{\Omega}}\left(\frac{1}{C{A_m}^{-1}B} r + \hat{\theta}^T X + \hat{\sigma}\right) \tag{16}$$

Here, $\frac{1}{C{A_m}^{-1}B} r$ constitutes the feedforward term. Then design the low-pass filter L such that the output is smoothed:

$$u_{l1} = -\frac{1}{\hat{\Omega}} L\left(\frac{1}{C{A_m}^{-1}B} r + \hat{\theta}^T X + \hat{\sigma}\right) \tag{17}$$

Define the feedforward gain $K_g = -\frac{1}{C{A_m}^{-1}B}$, $L = \frac{\hat{\Omega} K_f D}{1 + \hat{\Omega} K_f D}$, with $D = \frac{1}{s}$, and let $\hat{\eta} = \hat{\Omega} u_{l1} + \hat{\theta}^T x + \hat{\sigma}$. Define the reference command $r = [\varphi_{comd}, \dot{\varphi}_{comd}]$. Based on the expression of the predictive model, the input provided by the L1-augmented LQR to the system is given by:

$$U = u_{l1} + u_{lqr} \tag{18}$$

3.1.3. Stability Analysis

To ensure system stability, H is designed as the solution to the Lyapunov equation $A_m^T H + H A_m = -Q_a$, where $H = H^T > 0$ and $Q_a = Q_a^T > 0$. This gives $\dot{V}(\widetilde{X}, \widetilde{\theta}, \widetilde{\Omega}) \leq 0.$, enabling the adaptive law to be designed as follows:

$$\dot{\hat{\theta}} = -\frac{1}{\Gamma_\theta} X B^T H \widetilde{X}$$

$$\dot{\hat{\sigma}} = -\frac{1}{\Gamma_\sigma} B^T H \widetilde{X}$$
$$\dot{\hat{\Omega}} = -\frac{1}{\Gamma_\Omega} u_{l1} B^T H \tag{19}$$

To ensure smooth parameter adaptation while constraining the estimates within predefined bounds, the update law incorporates a projection operator [23] which enforces the required parameter limits during the adaptive process.

3.1.4. LQR Control Strategy Enhanced by RBF-L1 Synergy

Sections 3.1.1 and 3.1.2 respectively employed RBF neural networks and L1 adaptive control to enhance the LQR controller, addressing the dynamic adjustment of control parameters with robot states and the compensation for model uncertainties and external disturbances. However, in complex real-world scenarios, robots often face the dual challenge of slowly varying dynamic parameters and sudden disturbances. A single method struggles to handle all sources of uncertainty: while RBF neural networks, through offline training and online fitting, effectively adapt to gradual system parameter changes, their response to unmodeled dynamics and abrupt disturbances exhibits latency. Future work will focus on developing online RBF neural network strategies. By dynamically adjusting kernel function parameters and implementing incremental learning mechanisms, the system's real-time response capability to complex disturbances will be enhanced, thereby addressing the limitations of the current offline approach. However, this study will not optimize the aforementioned limitations and will continue to employ the existing method for subsequent research.

To address these limitations and further improve the system's robustness and adaptability in complex environments, this paper proposes a synergistic RBF-L1-enhanced LQR control strategy (RBF-L1-Adaptive-LQR). The core concept is a layered approach to uncertainty management: the RBF neural network acts as a feedforward optimizer, responsible for online identification and compensation of slowly varying dynamic parameters caused by changes in robot posture, thereby providing a more accurate baseline model for the LQR controller. Simultaneously, the L1 adaptive controller is integrated as a feedback compensator, specifically designed to handle model errors not captured by the RBF network, as well as sudden external disturbances. By working in synergy, these two components enhance the LQR controller's performance from different dimensions.

3.2. RBF-PID Based Virtual Model Control for Robotic Legs

The conventional Proportional-Integral-Derivative (PID) control algorithm is widely used in robotic control due to its simplicity and robustness. Nevertheless, its performance heavily relies on precise parameter tuning, making it difficult to adapt to dynamically changing environments. This often results in insufficient control accuracy and poor adaptability.

To enhance the performance of PID control on complex terrain, this section builds upon the foundation laid in Section 2.3 by incorporating RBF neural network for online tuning of PID parameters, thereby enabling dynamic parameter optimization. Leveraging the powerful nonlinear mapping capability and self-learning properties of the RBF neural network, the PID controller can better accommodate dynamic variations in the robot's base roll angle, significantly improving the robot's stability and adaptability on complex terrain. The theoretical foundation of the RBF neural network has been detailed in Section 3.1.1.

Based on the control requirements for robot height, roll angle, and leg compliance, a virtual "spring-damper" model is set at the leg end of the robot, as shown in Equation (9) and Figure 3. On this basis, an RBF-PID control algorithm block diagram is established,

as illustrated in Figure 9. The RBF neural network is used to approximate the Jacobian relationship $\frac{\partial y(t)}{\partial u(t)}$ between the input and output of the wheeled-legged robot system. The Jacobian information of the output with respect to the input is expressed as:

$$\frac{\partial y(t)}{\partial u(t)} \approx \frac{\partial y_m(t)}{\partial u(t)} = \sum_{j=1}^{m} w_j h_j \frac{c_{ji} - u_j}{b_j^2} \tag{20}$$

Here, the system output $y(t)$ represents the roll angle θ_{roll} of the robot base, and $u(t)$ denotes the robot control input. To achieve adaptive adjustment of the PID control parameters, the variation rates Δk_p, Δk_I, and Δk_d for the PID controller parameters are designed. These three variation rates are updated using the Jacobian information between the inputs and outputs of the RBF neural network. Given the roll angle error $e = r(t) - y(t)$, where $r(t) = \theta_{roll_d}$ is the desired angle, the tuning metric for the RBF neural network is designed as $E = \frac{1}{2}e^2$. Using the gradient descent method, the update rates for the PID controller are denoted as Δk_p, $\Delta k_i, \Delta k_d$, The updated PID gains are then given by $K_p = k_p + \Delta k_p$, $K_i = k_i + \Delta k_i$, $K_d = k_d + \Delta k_d$. Here, lr_p, lr_i, lr_d represent the learning rates for $\Delta k_p, \Delta k_i, \Delta k_d$, respectively, while k_p, k_i, k_d are the initial values assigned based on empirical experience.

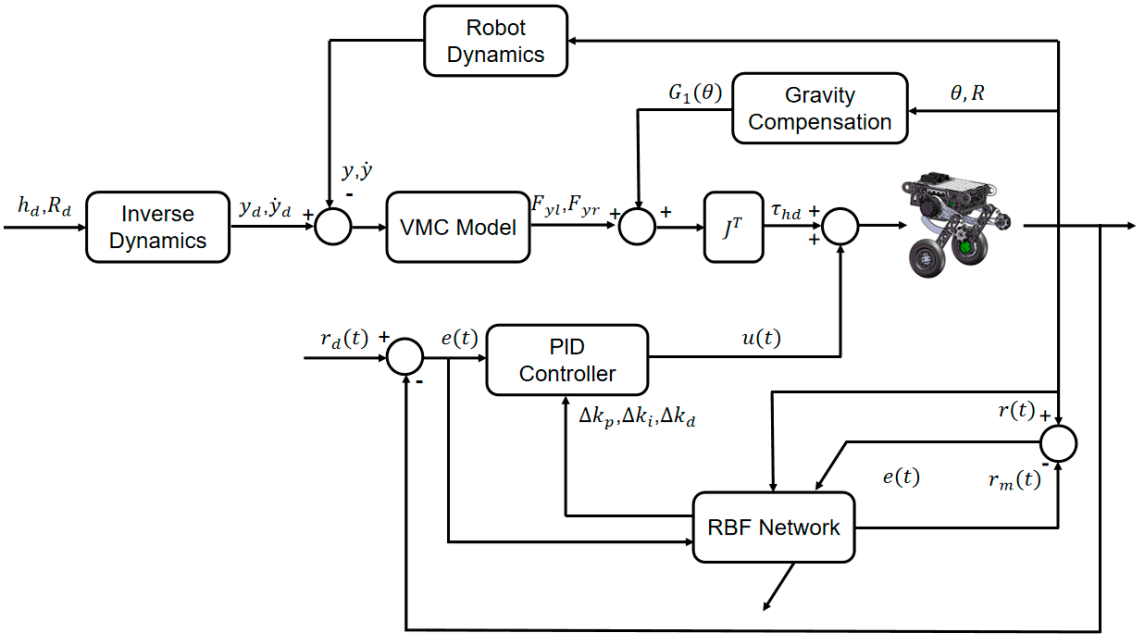


Figure 9. Block Diagram of the RBF-PID-VMC-Based Robot Leg Control System.

3.3. Based Whole-Body Control Framework

To address the coordination and optimization challenges among multiple control tasks in wheeled-legged robots, a whole-body control framework based on a hierarchical multi-task structure is developed. By formulating the whole-body motion control problem as a Quadratic Programming (QP) optimization [24], a task hierarchy capable of jointly optimizing wheel and leg control is established. The schematic diagram of the proposed framework is illustrated in Figure 10.

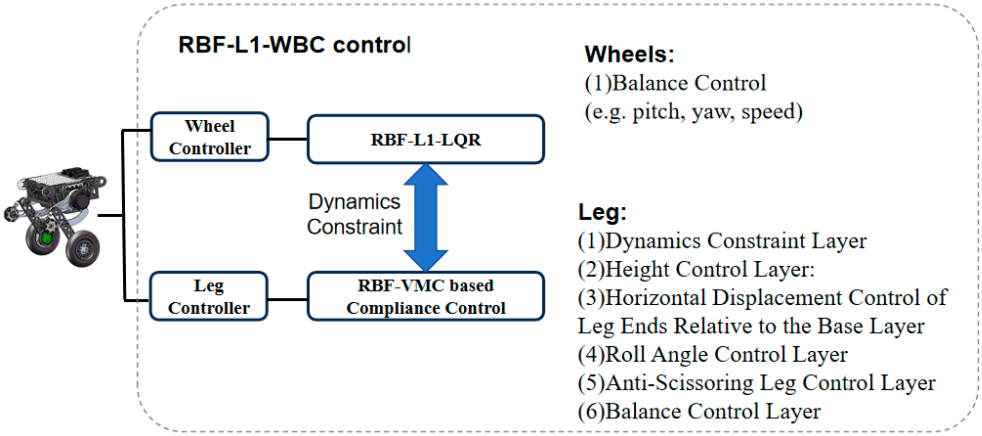


Figure 10. Block Diagram of the RBF-L1-WBC Robot Control Framework.

Define the robot's state variables as

$$x = \left[\ddot{\theta}_{l1}, \ddot{\theta}_{l2}, \ddot{\theta}_{l3}, \ddot{\theta}_{r1}, \ddot{\theta}_{r2}, \ddot{\theta}_{r3}, \tau_{l1}, \tau_{l2}, \tau_{l3}, \tau_{r1}, \tau_{r2}, \tau_{r3} \right]^T \tag{21}$$

The objective function is formulated as:

$$\underset{x}{\text{argmin}} \ f(x) = \frac{1}{2} x^T H x + f^T x \tag{22}$$

Here, H and f denote the coefficient weights of state variables. Transform each control objective of the wheeled-legged robot into the following constraint equation form:

$$\text{subject to } Ax = b, Lx \leq d \tag{23}$$

The optimization framework adopts a hierarchical structure where Layer 1 enforces strict satisfaction of the dynamic equations as a necessary feasibility condition; Layer 2 ensures core stability by directly mapping the torque outputs from the RBF-L1-LQR controller to the wheels; and Layers 3–6 sequentially address secondary tasks—height regulation, horizontal displacement control, roll angle stabilization, and anti-interference leg coordination—using weighted least squares to prioritize execution based on importance, thereby guaranteeing dynamic feasibility while meeting the primary requirement of balance stability, Specifically, the layers are structured as follows:

3.3.1. Dynamics Constraint Layer

The state variables of the wheeled-legged robot must satisfy the full-body dynamic model, with the corresponding constraint equations expressed as:

$$A_1 x = b_1 \tag{24}$$

Here, $A_1 = [M, -I_{6\times6}]$, $b_1 = [-B_t]$; M denotes the inertia matrix, B_t represents the simplification term, and I is the identity matrix.

3.3.2. Height Control Layer

Derive the acceleration-level representation for the robot's height control task:

$$H = \frac{1}{2}(l_{r2}s_{r1r2} + l_{r1}s_{r1} + l_{l2}s_{l1l2} + l_{l1}s_{l1})$$
$$\dot{H} = J_h \dot{\theta}$$

$$\ddot{H} = J_{\mathrm{h}}\ddot{\theta} + \dot{J}_{\mathrm{h}}\dot{\theta} \tag{25}$$

Here, J_{h} denotes the Jacobian matrix, and $\dot{\theta}$= [$\dot{\theta}_{l1}$, $\dot{\theta}_{l2}$, $\dot{\theta}_{l3}$, $\dot{\theta}_{r1}$, $\dot{\theta}_{r2}$, $\dot{\theta}_{r3}$]^T. Rearranging the above equation yields:

$$A_2 x = b_2 \tag{26}$$

Here, $A_2 = [\, J_{\mathrm{h}}, O_{1\times 6}\,]$ and $b_2 = \left[\, \ddot{H} - \dot{J}_{\mathrm{h}}\dot{\theta}\,\right]$. Design a PD controller to adjust H_d and $\dot{H}_d$ by controlling $\ddot{H}$:

$$\ddot{H} = k_H(H - H_d) + d_H\left(\dot{H} - \dot{H}_d\right) + \ddot{H}_d \tag{27}$$

Here, k_H and d_H represent the proportional and derivative gains of the PD controller.

3.3.3. Horizontal Displacement Control of Leg Ends Relative to the Base Layer

The relationship between the horizontal displacement X of the leg ends relative to the base and the joint angles is derived as follows:

$$X = \frac{1}{2}(l_{r2}c_{r1r2} + l_{r1}c_{r1} + l_{l2}c_{l1l2} + l_{l1}c_{l1})$$

$$X = J_X\dot{\theta}$$

$$\ddot{X} = J_X\ddot{\theta} + \dot{J}_X \tag{28}$$

Here, J_X is the Jacobian matrix for horizontal displacement. These equations can be rearranged into the form:

$$A_3 x = b_3 \tag{29}$$

where $A_3 = [\, J_X,\ O_{1\times 6}\,]$, $b_3 = [\, \ddot{X} - \dot{J}_X\dot{\theta}\,]$, A PD controller is used to regulate $\ddot{X}$.

3.3.4. Roll Angle Control Layer

Let the theoretical roll angle be defined as the rotation R of the robot base around the x-axis caused by the height difference between the two legs. Then we have:

$$R = atan\left(\frac{(l_{r2}s_{r1r2} + l_{r1}s_{r1} - l_{l2}s_{l1l2} - l_{l1}s_{l1})}{D}\right) \approx \frac{(l_{r2}s_{r1r2} + l_{r1}s_{r1} - l_{l2}s_{l1l2} - l_{l1}s_{l1})}{D}$$

$$\dot{R} = J_R\dot{\theta}$$

$$\ddot{R} = J_R\ddot{\theta} + \dot{J}_R\dot{\theta} \tag{30}$$

Rearranging yields:

$$A_4 x = b_4 \tag{31}$$

where $A_4 = [\, J_R, O_{1\times 6}\,]$ and $b_4 = \left[\, \ddot{R} - \dot{J}_R\dot{\theta}\,\right]$, A PD controller is used to regulate $\ddot{R}$.

3.3.5. Anti-Scissoring Leg Control Layer

The scissoring distance ΔX is defined as the difference in the x-axis positions of the left and right leg ends relative to the base frame:

$$\Delta X = l_{r2}c_{r1r2} + l_{r1}c_{r1} - l_{l2}c_{l1l2} - l_{l1}c_{l1}$$

$$\Delta\dot{X} = J_{\Delta X}\dot{\theta}$$

$$\Delta\ddot{X} = J_{\Delta X}\ddot{\theta} + \dot{J}_{\Delta X}\dot{\theta} \tag{32}$$

Here, $J_{\Delta X}$ is the Jacobian matrix for the scissoring distance. These equations can be rearranged into the form:

$$A_5 x = b_5 \tag{33}$$

where $A_5 = [\ J_{\Delta X},\ O_{1\times 6}\ ], b_5 = [\ \Delta\ddot{X} - \dot{J}_{\Delta X}\Delta\dot{X}\ ]$, A PD controller is used to regulate $\Delta\ddot{X}$.

3.3.6. Balance Control Layer

Define the parameters for the balance layer task as follows:

$$\begin{aligned} A_6 &= [\ O_{1\times 6},\ 0,\ 0,\ 1, 0,\ 0,\ 0\ ] \\ b_6 &= [\ u_l\ ] \end{aligned} \tag{34}$$

$$\begin{aligned} A_7 &= [\ O_{1\times 6},\ 0,\ 0,\ 0,\ 0,\ 0,\ 1\ ] \\ b_7 &= [\ u_r\ ] \end{aligned} \tag{35}$$

Here, u_l and u_r represent the outputs of the LQR controller based on RBF and L1 adaptation in Equation (18), i.e.,

$$\begin{bmatrix} u_l \\ u_r \end{bmatrix} = -K_{lqr}X_{lqr} + u_{l1} \tag{36}$$

Here, K_{lqr} is the gain feedback matrix from the LQR controller, $X_{lqr} = [\ x, \dot{x}, \varphi_{com}, \dot{\varphi}_{com}, \gamma, \dot{\gamma}\ ]^T$ is the state vector for the LQR, and u_{l1} is the output of the wheel controller L1. Integrating the aforementioned control objectives yields the parameters for the equality constraints as:

$$\begin{aligned} A &= \left[A_1,\ A_2,\ A_3,\ A_4,\ A_5,\ A_6,\ A_7\ \right]^T \\ b &= [\ b_1,\ b_2,\ b_3,\ b_4,\ b_5,\ b_6,\ b_7\ ]^T \end{aligned} \tag{37}$$

For the inequality constraints, they are defined as the robot's actuator constraint layer, which is given as follows:

$$\begin{aligned} L &= [\ O_{6\times 6},\ I_{6\times 6}\ ] \\ d &= [\ \tau_{l1_max},\ \tau_{l2_max},\ \tau_{l3_max},\ \tau_{r1_max},\ \tau_{r2_max},\ \tau_{r3_max}\ ]^T \end{aligned} \tag{38}$$

In terms of control feasibility, this framework adopts a "hierarchical hard constraints + soft constraints" strategy to ensure robust solvability. Specifically, the Layer 1 dynamics constraints are enforced as hard constraints that the system must satisfy, guaranteeing the physical realizability of control commands. In contrast, the tasks in Layers 2–6 are converted into soft constraints through weighting coefficients, which are essentially optimization objectives that are "desired to be satisfied" rather than mandatory requirements. This design effectively avoids the infeasibility issues that can arise from multi-task conflicts. The solution process of the framework is equivalent to the quadratic programming problem shown in Equation (22), where the decision variables include the torque commands for each robot joint. The optimal solution x is obtained by minimizing the objective function, and the last six elements of x, corresponding to the joint torque vector τ, are extracted as the final control input.

4. Simulation and Experimentation

Simulations and experimental evaluations were conducted to verify the effectiveness of the proposed algorithm. The simulation studies were carried out using the Gazebo platform, while the experimental tests were performed on the wheeled-legged robot shown in Figure 3. Detailed configurations and implementation setups are provided in the following sections.

4.1. Simulation Section

4.1.1. LQR Controller Based on RBF Neural Network

The control performance of the RBF-LQR method was compared against that of the conventional LQR within the Gazebo simulation environment (Figure 11) under varying

robot height conditions. During the simulation, the robot initially remained in a stable equilibrium state, and its height was altered from 0.26 m to 0.20 m at the 5 s mark.

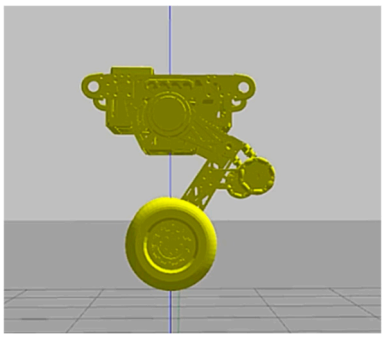


Figure 11. Gazebo Simulation.

Figure 12 presents the simulation results. Under conventional LQR control, the steady-state pitch angle converges to approximately $-1.9°$, indicating model inaccuracies that cause the center of mass to deviate from a zero-pitch configuration during balancing. In contrast, the RBF-LQR controller achieves a steady-state pitch angle near $0°$. When the robot's center-of-mass height is altered at t = 5 s, the classical LQR controller requires approximately 15 s to reestablish equilibrium, and its steady-state pitch shifts from $-1.9°$ to $-2.1°$. By comparison, the RBF-LQR controller converges within 3 s, with the equilibrium pitch shifting only slightly from $0°$ to $-0.1°$. These results demonstrate that conventional LQR incurs substantial computational delay when recalculating the feedback gain K, whereas the proposed RBF-LQR method enables rapid adaptation under changing height conditions. The shift in equilibrium pitch in both cases originates from variation in the robot's height, which alters the overall center-of-mass position.

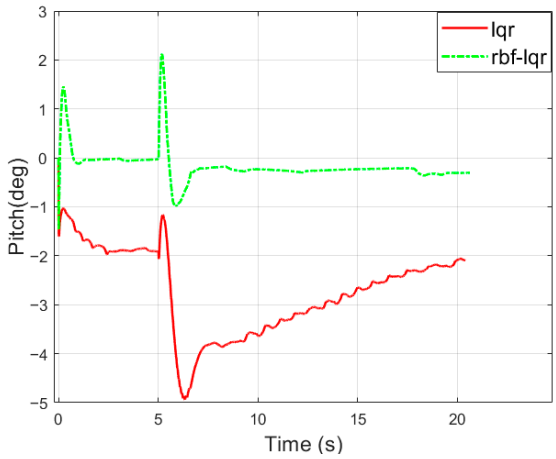
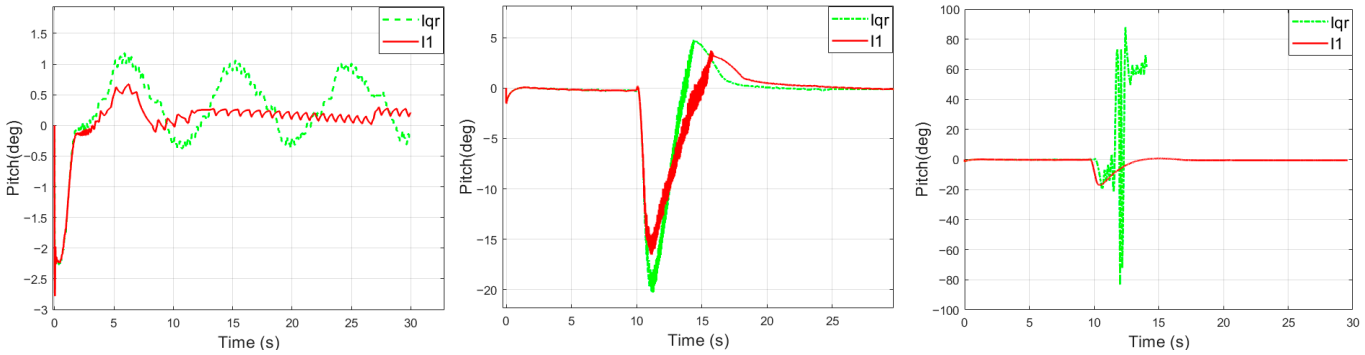


Figure 12. Robot Center of Mass Pitch Response During Height Descent.

4.1.2. Robot Balance Control Based on L1-LQR

To validate the control performance of the L1–LQR framework, four distinct operational scenarios were designed, incorporating variations in payload, robot height, ground friction coefficient, and locomotion velocity.

In Figure 13a, load disturbances are introduced by modifying the parameter M_{com} in the dynamic model. The conventional LQR controller, which relies heavily on model accuracy, exhibits pronounced oscillations with an amplitude of approximately $1.5°$. In contrast, the L1-LQR controller demonstrates rapid convergence after the initial transient response, achieving a significantly smaller steady-state oscillation amplitude of about $0.3°$.



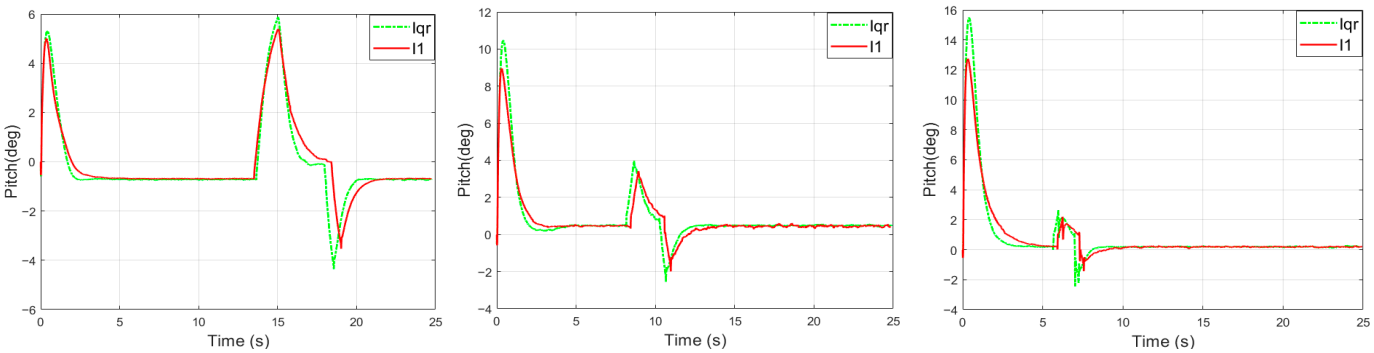
(**a**) Load Disturbance Response (**b**) External Force Impact (**c**) Friction Variation Performance

Figure 13. Performance comparison of LQR and L1-LQR.

Figure 13b depicts the comparative control efficacy of LQR and L1-LQR in response to external disturbances resulting from alterations in base height (corresponding to pendulum length L). Experimental results demonstrate that, prior to the introduction of external disturbance (i.e., before 10 s in the simulation), both robots utilizing LQR and L1-enhanced LQR control algorithms attain reliable tracking of the target value. At t = 10 s, the application of a 25 N disturbance force to the robot base revealed that the L1-enhanced LQR control algorithm outperformed the standard LQR algorithm. Under this control, the greatest pitch angle of the robot's center of mass was around 16°, indicating a decrease of nearly 4° from the maximum pitch angle (approximately 20°) recorded under LQR control.

Figure 13c presents the comparison results under variations in the ground friction coefficient. The results indicate that L1-LQR maintains superior disturbance-rejection capability and stability, while conventional LQR shows larger deviations due to its sensitivity to modeling uncertainties introduced by friction changes.

Thereafter, the control performance of both algorithms was evaluated under variations in the robot's locomotion speed. As shown in Figure 14, the L1-LQR controller reduces pitch-angle fluctuations by approximately 1° during both start-up and ramp-motion phases. Combined with the results from the base-height, payload, and friction-variation experiments, these findings demonstrate that the L1-enhanced LQR control algorithm provides superior robustness compared with the conventional LQR controller. In particular, L1-LQR exhibits enhanced capability in handling complex operating conditions characterized by model uncertainties and external disturbances.



(**a**) slope velocity: 0.5 m/s (**b**) slope velocity: 1.0 m/s (**c**) slope velocity: 1.5 m/s

Figure 14. LQR vs. L1-LQR control comparison.

Subsequently, the control performance of the two algorithms was evaluated by varying the robot's speed, as shown in Figure 13. The L1-LQR controller effectively reduced

pitch angle fluctuations during both startup and slope traversal, achieving a reduction of approximately 1 degree. Combined with the experimental results on base height, mass variation, and slope negotiation, it can be concluded that the L1-augmented LQR control algorithm demonstrates superior robustness compared to the traditional LQR controller when handling model uncertainties and external disturbances under complex conditions.

4.1.3. Simulation of Roll Angle Control for a Wheeled-Legged Robot Based on RBF-PID

A single-sided bridge terrain simulation was constructed to evaluate the effectiveness of RBF-VMC, as illustrated in Figures 15 and 16. The experiment compares the roll-angle regulation performance of conventional PID control with that of the RBF-PID controller. During the simulation, the robot's target velocity was progressively increased to 0.5, 1, 1.5, and 2 m/s. The arrows in Figure 16 indicate the robot's forward direction.

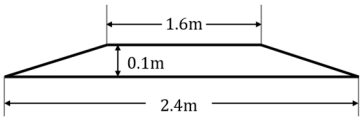


Figure 15. Schematic of Ramp Terrain Profile and Parameters.

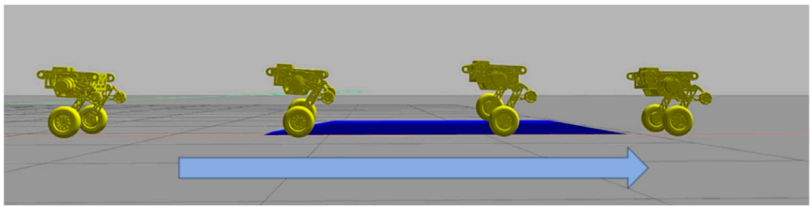


Figure 16. Schematic of Robot Traversing a One-Sided Slope (Simulation).

The experimental results presented in Table 2 and Figure 17 indicate that the RBF-PID control method demonstrates enhanced performance in roll angle control at various speeds. The maximum roll angle is considerably less than that attained by traditional PID control, without negatively impacting pitch angle or forward velocity.

Table 2. PID vs. RBF-PID control performance on one-sided slope.

Desired Velocity	Control Method	max ($\lvert\theta_{roll}\rvert$)	θ_{roll} Steady State Error	max ($\lvert\theta_{pitch}\rvert$)	θ_{pitch} Steady State Error
0.5	PID	2.48	±0.26	5.93	±0.58
	RBF-PID	1.67	±0.15	5.94	±0.57
1.0	PID	2.93	±0.05	11.02	±1.24
	RBF-PID	2.03	±0.01	11.00	±1.23
1.5	PID	3.26	±0.01	16.02	±1.42
	RBF-PID	2.17	±0.01	15.98	±1.40
2.0	PID	3.57	±0.01	21.27	±1.11
	RBF-PID	2.20	±0.01	21.17	±1.13

The experimental results, as presented in Table 2 and Figure 17, demonstrate that the RBF-PID control algorithm achieves superior performance in roll angle regulation across different speeds. Specifically, it yields a significantly smaller maximum roll angle compared to the classical PID controller, without compromising pitch angle stability or forward speed.

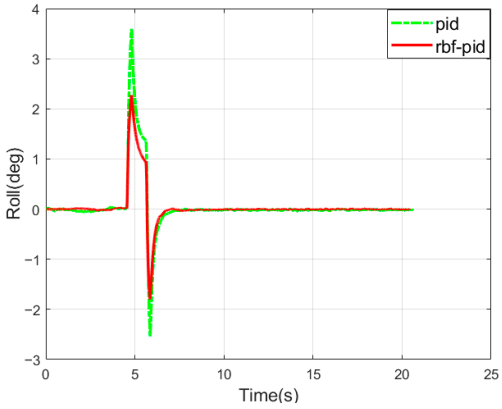


Figure 17. Robot traversing a 2.0 m/s one-sided slope (snapshot sequence).

4.1.4. Simulation of Wheeled-Legged Robot Control Based on RBF-L1-WBC

To highlight the distinct advantages of RBF neural network and WBC, this study conducts a comparative analysis of three control strategies. The first method, termed L1-VMC, adopts a hierarchical architecture: wheel balancing is handled by an L1-adaptive-enhanced LQR to compensate for model uncertainties and external disturbances, while leg compliance is managed via VMC combined with a conventional ID controller for roll angle regulation. The core idea is to decouple wheel and leg control into independent modules, achieving basic functionality through simple superposition. The second method, RBF-L1-VMC, enhances the baseline by integrating an RBF neural network for online optimization of local controllers: wheel control combines RBF-LQR with L1 adaptation to dynamically adjust LQR parameters in response to center-of-mass variations, while leg control employs RBF-PID to enhance VMC by real-time tuning of roll angle control parameters. Although RBF improves the adaptability of each module, the overall architecture remains hierarchical. The third and proposed method, RBF-L1-WBC, represents the core innovation, employing a unified optimization framework: based on a full-body dynamic model, it formulates balancing, posture, and velocity tasks as a QP problem with hierarchical priorities for global coordination, and synergistically integrates the RBF neural network for online estimation of dynamic parameters and L1 adaptive control for handling external disturbances.

This subsection presents simulation experiments designed to evaluate whole-body motion control strategies. Using the Gazebo simulation platform, a unilateral ramp scenario is constructed to replicate complex terrain conditions, with a sampling time of 2 ms. The experimental setup is consistent with that described in Section 4.1.3, enabling a comprehensive assessment of the motion performance of whole-body control algorithms under challenging terrain. The study then conducts a comparative analysis of the control effectiveness of the three methods introduced earlier.

The simulations systematically test the robot's performance at three desired forward speeds: 0.5 m/s, 1 m/s, and 1.5 m/s. For each speed setting, the motion performance of the three control strategies is compared as the robot traverses the unilateral ramp. The analysis focuses on three key response parameters: the robot's roll angle, the pitch angle of its center of mass, and its actual forward speed. These metrics are monitored to quantitatively evaluate the stability, tracking accuracy, and overall robustness of each control method under dynamic and uneven terrain conditions.

To thoroughly measure control performance, we present the Integral of Time-weighted Absolute Error (ITAE) [25] as a metric, with the following mathematical expression:

$$ITAE = \int_0^{\infty} t|e(t)|dt \tag{39}$$

Here, e(t) denotes the system error and t represents time. In contrast to traditional error measurements, ITAE emphasizes subsequent errors by multiplying them by time, hence enhancing their impact on the total evaluation. This indicates that ITAE not only accounts for the mistake amount but also highlights the system's capacity for long-term adjustment. Additionally, due to the varying dimensions and numerical ranges of different control metrics, we utilize the 'Min-Max Normalization Method' to preprocess error data. For each control method, the normalized ITAE values for roll angle error, pitch angle error, and velocity error are aggregated to provide the composite ITAE as the ultimate evaluation criterion:

$$ITAE_{total} = ITAE_{roll} + ITAE_{pitch} + ITAE_{velocity} \tag{40}$$

The simulation results for the three control methods are shown in Figure 18 and Table 3.

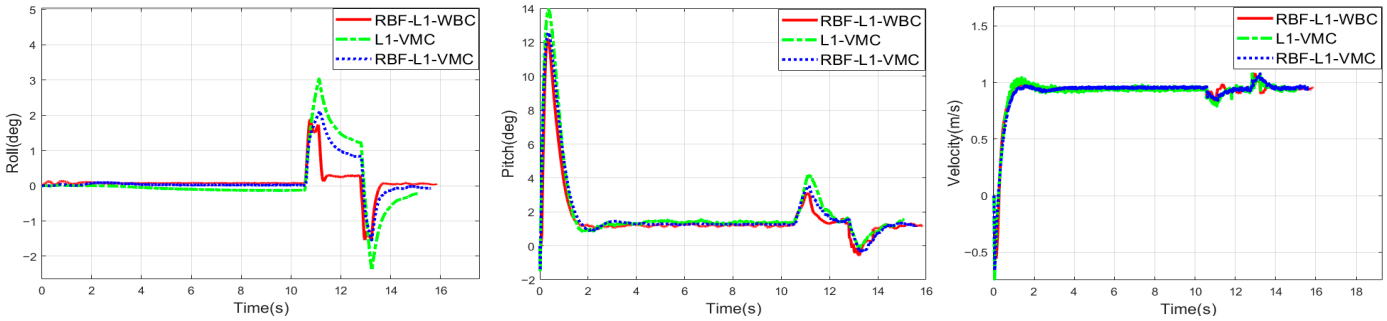


Figure 18. Simulation results: three control methods at 1.0 m/s.

Table 3. Control performance simulation: WBC vs. RBF-L1-LQR-VMC.

Desired Velocity		0.5 m/s			1.0 m/s			1.5 m/s		
Item \ Performance Metric		Over-Shoot	Steady-State Error	*ITAE* Value	Over-Shoot	Steady-State Error	*ITAE* Value	Over-Shoot	Steady-State Error	*ITAE* Value
L1-VMC	Roll	2.90	±0.45	198.89	3.03	±0.27	40.74	3.16	±0.02	14.29
	Pitch	7.10	±0.84		13.94	±1.38		19.50	±1.60	
	Velocity	0.91	±0.02		1.74	±0.07		2.39	±0.05	
	computational costs		1.95			1.96			1.98	
RBF-L1-VMC	Roll	1.57	±0.22	185.12	2.12	±0.07	38.09	2.28	±0.01	16.27
	Pitch	6.78	±0.73		12.57	±1.23		18.53	±1.54	
	Velocity	0.95	±0.01		1.66	±0.04		2.44	±0.07	
	computational costs		0.51			0.53			0.56	
RBF-L1-WBC	Roll	1.34	±0.15	171.56	1.88	±0.05	33.15	2.89	±0.08	13.08
	Pitch	6.63	±0.78		12.23	±1.25		16.72	±1.40	
	Velocity	1.08	±0.04		1.56	±0.05		2.50	±0.05	
	computational costs		1.21			1.25			1.28	

Note: Units for angles and velocities are degrees (°) and meters per second (m/s), all time-related values are expressed in milliseconds (ms).

According to the unilateral bridge traversal simulation results presented in Table 3, the proposed RBF-L1-WBC whole-body control strategy demonstrates significant superiority over both L1-VMC and RBF-L1-VMC in terms of roll angle control accuracy, dynamic response speed, and multi-task coordination performance. This fully validates the advantages of its unified optimization framework for achieving synergistic multi-task control. Specifically, under low-speed conditions where roll angle variations are more pronounced,

the RBF-L1-WBC strategy restricts the steady-state roll angle error to within ±0.15°, reducing the peak error by 53.8% and 14.6% compared to the other two strategies, respectively. Moreover, its integrated performance evaluation metric (ITAE) consistently achieves the lowest value, with speed tracking steady-state error remaining below ±0.05 m/s and a smoother transient response. In high-speed scenarios (1.5 m/s), the steady-state pitch angle error is controlled to only ±1.4°, significantly outperforming the other two methods.

Given the system's 2 ms sampling period, the computational overhead data clearly confirms that the real-time performance of all methods is fully guaranteed. The comparative data also highlights the significant advantage of the RBF neural network in enhancing computational efficiency. The RBF-L1-VMC method reduces computational overhead by over 70% compared to traditional L1-VMC—at 0.5 m/s, the computation time is drastically reduced from 1.95 ms to 0.51 ms, a reduction of 73.8%; at 1.0 m/s and 1.5 m/s, the reductions are 73.0% and 71.7%, respectively. These figures indicate that the RBF neural network compresses the single-cycle computation time to approximately one-quarter of that of traditional methods while maintaining control performance, representing a qualitative leap in computational efficiency.

Although the computational overhead of RBF-L1-WBC is slightly higher than that of L1-VMC, this precisely demonstrates the universal applicability of RBF across algorithms of varying complexity. WBC inherently requires solving complex optimization problems, resulting in a computational burden far greater than that of hierarchical wheel-leg control. By integrating the RBF neural network with WBC, the strategy not only preserves WBC's advantages in multi-objective coordinated control but also keeps the computational overhead within a reasonable range. This effectively proves the efficacy of RBF in mitigating the computational growth associated with complex algorithms.

4.1.5. Wheeled-Legged Robot Experiment

This experimental platform for a bipedal wheeled-legged robot includes both hardware and software components, addressing the physical setup of the robot and the control development utilizing the C# programming language. Figure 19 depicts that the control system for the bipedal wheeled-legged robot consists of a compact industrial computer (PC) and an Ethernet-to-CAN converter. The industrial computer employs a Beikong industrial control unit, while the actuator system has six electric motors that operate the robot's knee, hip, and wheel joints accordingly. Both the knee and hip joints utilize HT8115 disc-type brushless joint motors.

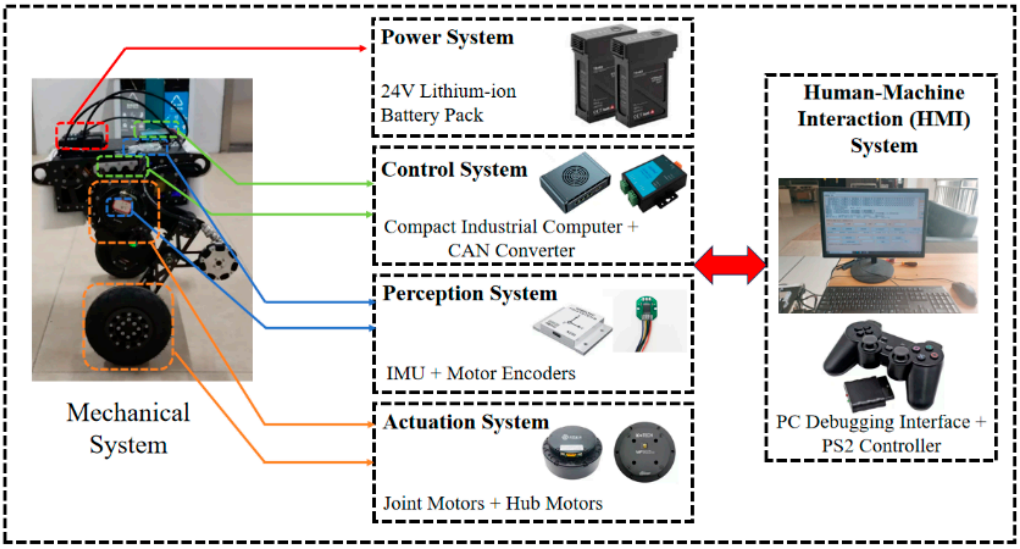


Figure 19. Hardware Architecture of the Robot Experimental Platforms.

For the whole-body motion control experiments, a unilateral ramp scenario was constructed using the slope terrain configuration illustrated in Figure 15, with a sampling time of 2 ms. This comprehensive experimental setup was used to evaluate the robot's motion performance under the whole-body control algorithm. To validate the effectiveness of the control strategies in a real-world environment, comparative tests were conducted for three different methods: L1-VMC, RBF-L1-VMC, and RBF-L1-WBC. The experiments systematically tested the robot's performance at desired speeds of 0.5 m/s, 1.0 m/s, and 1.5 m/s, comparing the motion performance of the three control strategies as the robot traversed the unilateral ramp. The experimental results are presented in Figure 20 and Table 4.

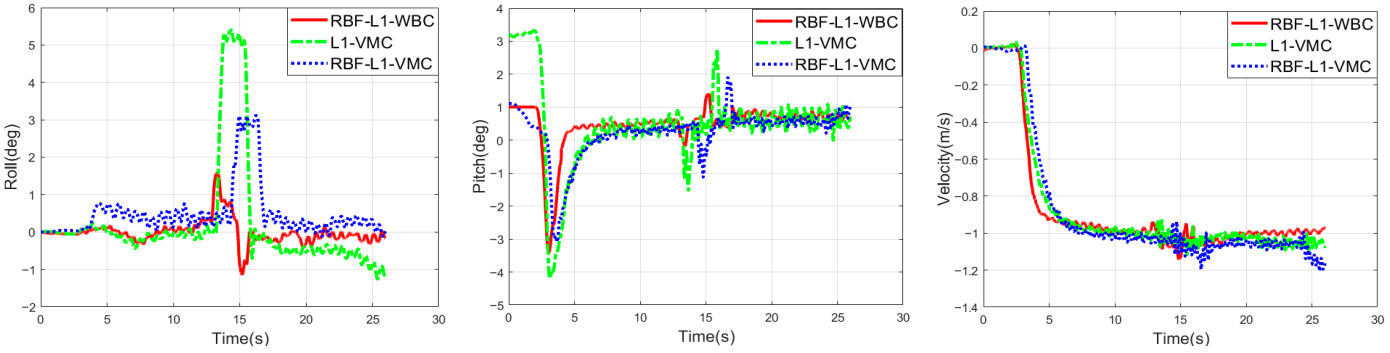


Figure 20. Three control methods compared at 1.0 m/s.

Table 4. Comparison of control performance between WBC and RBF-L1-LQR-VMC.

Desired Velocity		0.5 m/s			1.0 m/s			1.5 m/s		
Item \ Performance Metric		Over-Shoot	Steady-State Error	*ITAE* Value	Over-Shoot	Steady-State Error	*ITAE* Value	Over-Shoot	Steady-State Error	*ITAE* Value
L1-VMC	Roll	6.03	±0.46	541.06	5.41	±1.21	126.12	9.46	±0.41	67.94
	Pitch	2.46	±0.22		4.20	±0.98		5.52	±0.81	
	Velocity	0.55	±0.08		1.03	±0.07		1.56	±0.23	
RBF-L1-VMC	Roll	3.04	±0.14	364.06	3.12	±0.02	105.50	2.97	±0.22	42.70
	Pitch	2.87	±0.12		3.09	±0.64		18.53	±0.94	
	Velocity	0.52	±0.07		1.01	±0.02		1.59	±0.10	
RBF-L1-WBC	Roll	1.95	±0.23	217.59	1.60	±0.15	80.88	2.16	±0.04	30.06
	Pitch	1.44	±0.11		3.43	±0.63		6.32	±0.63	
	Velocity	0.51	±0.02		1.02	±0.03		1.51	±0.07	

Note: Units for angles and velocities are degrees (°) and meters per second (m/s), all time-related values are expressed in milliseconds (ms).

According to the comparative control performance results presented in Table 4, the proposed RBF-L1-WBC strategy demonstrates comprehensive and significant superiority across all three core performance metrics. In terms of peak error control, the strategy reduces the roll angle error at 0.5 m/s from 6.03° (L1-VMC) and 3.04° (RBF-L1-VMC) to 1.95°, achieving improvements of 67.7% and 35.9%, respectively. Regarding steady-state error, the pitch angle steady-state error at 1.5 m/s is ±0.63°, a 22.2% optimization compared to RBF-L1-VMC's ±0.81°. For comprehensive dynamic performance (ITAE value), the speed ITAE at 1.0 m/s is 80.88, a 33.3% reduction from L1-VMC's 121.30. In terms of computational overhead, although RBF-L1-WBC incurs higher complexity due to the whole-body dynamics optimization required by the WBC framework—resulting in computation times of 1.25–1.32 ms, higher than RBF-L1-VMC's 0.53–0.57 ms—it remains

significantly lower than L1-VMC's 1.96–1.98 ms and fully satisfies real-time requirements within the 2 ms sampling period. These results clearly indicate that the RBF-L1-WBC strategy achieves an excellent balance between control accuracy, dynamic response, and real-time performance by deeply integrating the adaptive compensation capability of the RBF neural network with the unified optimization framework of WBC, providing a reliable solution for high-performance motion control of wheel-legged robots in dynamic and complex terrain.

5. Conclusions

Based on a comprehensive analysis of the experimental results, the proposed RBF-L1-WBC whole-body control strategy demonstrates significant superiority over both L1-VMC and RBF-L1-VMC in terms of roll angle control accuracy, speed tracking performance, pitch angle stability, and multi-task coordination. Specifically, for roll angle control accuracy, RBF-L1-WBC achieves a 67.7% and 35.9% reduction in error at 0.5 m/s compared to the other two methods, with a steady-state error of $\pm 0.15^{\circ}$ at 1.0 m/s—an 87.6% improvement over L1-VMC's $\pm 1.21^{\circ}$. In speed tracking, the steady-state error at 1.5 m/s is ± 0.07 m/s, a 69.6% enhancement over L1-VMC's ± 0.23 m/s. For pitch angle stability, the error peak at 1.5 m/s is 6.32°, a 65.9% reduction compared to RBF-L1-VMC's 18.53°.

The fundamental advantage of this strategy lies in its unified multi-task hierarchical optimization framework, which abstracts the robot's whole-body dynamics into a quadratic programming problem and coordinates multiple coupled control objectives—such as balancing, posture adjustment, and speed tracking—by setting task priorities. This approach fundamentally avoids the goal conflicts inherent in modular methods that decouple wheel and leg control. The superiority of this framework is reflected in the integrated performance evaluation metric: at 1.5 m/s, the ITAE value for RBF-L1-WBC is 30.06, significantly lower than L1-VMC's 67.94 (a 55.8% reduction) and RBF-L1-VMC's 105.50 (a 71.5% reduction).

Simultaneously, the RBF neural network plays a critical role in the system by enabling online real-time estimation of dynamic parameters and adaptive tuning of control gains, effectively mitigating the impact of model uncertainties and external disturbances. This further enhances the controller's adaptability and robustness in complex terrain. The effectiveness of RBF is evident in the 49.6% reduction in roll angle error peak achieved by RBF-L1-VMC compared to L1-VMC at 0.5 m/s, with RBF-L1-WBC delivering additional optimization on top of this improvement. The experimental results confirm that RBF-L1-WBC not only excels in key metrics—roll angle steady-state error, speed tracking accuracy, and pitch angle error—but also consistently achieves the lowest ITAE values across all three speeds (217.59, 80.88, and 30.06 at 0.5, 1.0, and 1.5 m/s, respectively), outperforming the comparative algorithms comprehensively. This validates that the strategy achieves synergistic improvements in response speed and control accuracy through its global optimization mechanism.

In terms of computational real-time performance, RBF-L1-WBC incurs an overhead of 1.25–1.32 ms, which, while higher than RBF-L1-VMC's 0.53–0.57 ms, is significantly lower than L1-VMC's 1.96–1.98 ms. This overhead fully satisfies the real-time requirements of the 2 ms sampling period, demonstrating an excellent balance between performance and efficiency. The strategy thus provides an effective solution for high-performance motion control of wheel-legged robots in unstructured environments.

Author Contributions: Conceptualization, G.M. and S.C.; methodology, R.Z. and S.C.; investigation, X.C.; formal analysis, X.C. and H.Z.; writing—original draft preparation, H.Z.; writing—review and editing, G.M. and S.C.; visualization, Q.C.; supervision, Y.G.; project administration, R.Z. and T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Tertiary Education Scientific research project of Guangzhou Municipal Education Bureau, grant number 80, and the National Natural Science Foundation of China, grant number 62573144. The APC was funded by Renyi Zhou.

Data Availability Statement: The data involved in this study cannot be publicly shared due to privacy concerns and/or ethical restrictions. Access to the data is not provided as it is protected under privacy regulations.

Conflicts of Interest: Authors Renyi Zhou and Guangcai Ma were employed by the Advanced Technology Department, Guangzhou Automobile Group 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.

References

1. Liu, X.; Sun, Y.; Wen, S.; Cao, K.; Qi, Q.; Zhang, X.; Shen, H.; Chen, G.; Xu, J.; Ji, A. Development of wheel-legged biped robots: A review. *J. Bionic Eng.* **2024**, *21*, 607–634. [CrossRef]
2. Cui, L.; Wang, S.; Zhang, J.; Zhang, D.; Lai, J.; Zheng, Y.; Zhang, Z.; Jiang, Z.P. Learning-based balance control of wheel-legged robots. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7667–7674. [CrossRef]
3. Yu, H.; Guan, S.; Li, X.; Feng, H.; Zhang, S.; Fu, Y. Whole-body motion generation for wheeled biped robots based on hierarchical MPC. *IEEE Trans Ind Electron* **2025**, *72*, 8301–8311. [CrossRef]
4. Feng, X.; Liu, S.; Yuan, Q.; Xiao, J.; Zhao, D. Research on wheel-legged robot based on LQR and ADRC. *Sci. Rep.* **2023**, *13*, 15122. [CrossRef] [PubMed]
5. Dong, J.; Liu, R.; Guo, X.; Liu, H. LQR-based balance control of two-wheeled legged robot. In *2022 41st Chinese Control Conference (CCC)*; IEEE: New York City, NY, USA, 2022; pp. 450–455.
6. Liu, D.; Yang, F.; Liao, X.; Lyu, X. DIABLO: A 6-DoF wheeled bipedal robot composed entirely of direct-drive joints. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York City, NY, USA, 2024; pp. 3605–3612.
7. Kim, S.; Yeom, K. Development of a hand-fan-shaped arm and a model predictive controller for leg crossing, walking, and one-legged balancing of a wheeled-bipedal jumping robot. *Machines* **2024**, *12*, 284. [CrossRef]
8. Xin, Y.; Rong, X.; Li, Y.; Li, B.; Chai, H. Movements and balance control of a wheel-leg robot based on uncertainty and disturbance estimation method. *IEEE Access* **2019**, *7*, 133265–133273. [CrossRef]
9. Chen, G.; Guo, S.; Hou, B.; Wang, J. Virtual model control for quadruped robots. *IEEE Access* **2020**, *8*, 140736–140751. [CrossRef]
10. Guo, T.; Liu, J.; Liang, H.; Zhang, Y.; Chen, W.; Xia, X.; Wang, M.; Wang, Z. Design and dynamic analysis of jumping wheel-legged robot in complex terrain environment. *Front. Neurorobot.* **2022**, *16*, 987345. [CrossRef] [PubMed]
11. Xie, A.; Chen, T.; Rong, X.; Zhang, G.; Li, Y.; Fan, Y. A robust and compliant framework for legged mobile manipulators using virtual model control and whole-body control. *Robot. Auton. Syst.* **2023**, *164*, 104411. [CrossRef]
12. Bellicoso, C.; Gehring, C.; Hwangbo, J.; Fankhauser, P.; Hutter, M. Perception-less terrain adaptation through whole-body control and hierarchical optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*; IEEE: New York City, NY, USA, 2016; pp. 558–564.
13. Klemm, V.; Morra, A.; Gulich, L.; Mannhart, D.; Rohr, D.; Kamel, M.; de Viragh, Y.; Siegwart, R. LQR-assisted whole-body control of a wheeled bipedal robot with kinematic loops. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3745–3752. [CrossRef]
14. Wang, S.; Cui, L.; Zhang, J.; Lai, J.; Zhang, D.; Chen, K.; Zheng, Y.; Zhang, Z.; Jiang, Z.P. Balance control of a novel wheel-legged robot: Design and experiments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York City, NY, USA, 2021; pp. 6782–6788.
15. Xin, Y.; Chai, H.; Li, Y.; Rong, X.; Li, B.; Li, Y. Speed and acceleration control for a two wheel-leg robot based on distributed dynamic model and whole-body control. *IEEE Access* **2019**, *7*, 180630–180639. [CrossRef]
16. Xin, Y.; Li, Y.; Chai, H.; Rong, X.; Li, B.; Li, Y. Planning and execution of dynamic whole-body locomotion for a wheeled biped robot on uneven terrain. *Int. J. Control Autom. Syst.* **2024**, *22*, 1337–1348. [CrossRef]
17. Zhang, J.; Li, Z.; Wang, S.; Dai, Y.; Zhang, R.; Lai, J.; Zhang, D.; Chen, K.; Hu, J.; Gao, W.; et al. Adaptive optimal output regulation for wheel-legged robot Ollie: A data-driven approach. *Front. Neurorobot.* **2023**, *16*, 1102259. [CrossRef] [PubMed]
18. Majdisova, Z.; Skala, V. Radial basis function approximations: Comparison and applications. *Appl. Math. Model.* **2017**, *51*, 728–743. [CrossRef]
19. Jafari, S.; Ioannou, P.; Rudd, L.E. What is L1 adaptive control. In Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference, Boston, MA, USA, 19–22 August 2013; p. 4513.

20. Sakthivel, M.; Kumar, A.S.; Shankar, G.; Chittawadigi, R.G. Virtual and physical components to build modular robotic arms for effective understanding of Denavit-Hartenberg (DH) parameters. In *IFToMM World Congress on Mechanism and Machine Science*; Springer Nature: Cham, Switzerland, 2023; pp. 969–978.
21. Ahmad, N.; Ghazilla, R.A.R.; Khairi, N.M.; Kasi, V. Reviews on various inertial measurement unit (IMU) sensor applications. *Int. J. Signal Process. Syst.* **2013**, *1*, 256–262. [CrossRef]
22. Yin, J.; Chen, G.; Yang, S.; Huang, Z.; Suo, Y. Ship Autonomous Berthing Strategy Based on Improved Linear-Quadratic Regulator. *J. Mar. Sci. Eng.* **2024**, *12*, 1245. [CrossRef]
23. Lee, K.; Singh, S. Multi-input submarine control via L1 adaptive feedback despite uncertainties. *Proc. Inst. Mech. Eng. I J. Syst. Control Eng.* **2014**, *228*, 330–347.
24. Frasch, J.V.; Sager, S.; Diehl, M. A parallel quadratic programming method for dynamic optimization problems. *Math. Program. Comput.* **2015**, *7*, 289–329. [CrossRef]
25. Sanoj, K.; Ram, V. Optimal PID controller tuning for multivariable unstable systems by minimizing ITAE criteria. *IFAC-PapersOnLine* **2024**, *57*, 391–396. [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.