Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 1, 2026

Optimal LQR Controller Design for Wheel-Legged Robots Using Genetic Algorithm

Zhiqiu Xia

School of Physics and Optoelectronic Engineering, Guangdong University of Technology, Guangzhou, Guangdong, 510006, China

Abstract: Wheel-legged robots are typical nonlinear underactuated systems, whose balance control is commonly achieved by a Linear Quadratic Regulator (LQR). However, the performance of the LQR controller for such robots heavily depends on the selection of the state weighting matrix **Q** and the control weighting matrix **R**, which in practice relies highly on expert heuristics and often fails to achieve an efficient trade-off among multiple performance objectives. To address this issue, this paper proposes a systematic automatic tuning method for LQR controllers of wheel-legged robots. By embedding engineering constraints into the fitness function in the form of penalty terms, the proposed method can stably generate feasible controllers that achieve favorable multi-performance trade-offs without manual trial-and-error. On this basis, for variable leg-length conditions, a grid-based offline solution of the Riccati equation combined with polynomial fitting is employed to extend the optimal weightings obtained at the nominal leg length to the entire leg-length range via gain scheduling, enabling online continuous adaptation of the feedback gain with respect to leg length. Simulation results on a nonlinear model demonstrate that the controller automatically optimized by the genetic algorithm (GA) exhibits good dynamic response and robustness under various scenarios, including balancing control, fall recovery, and terrain adaptation, thereby validating the effectiveness of the proposed method.

Keywords: Wheel-legged Robot; Linear Quadratic Regulator (LQR) Parameter Tuning; Genetic Algorithm and Gain Scheduling.

1. Introduction

In recent years, robotics has witnessed significant advancements. Legged robots have become a major research focus due to their superior obstacle-climbing ability and environmental adaptability. However, their locomotion efficiency on long, flat terrains remain incomparable to that of traditional wheeled chassis. As an innovative hybrid mobile platform, the wheel-legged robot cleverly integrates the high-speed and high-efficiency characteristics of wheeled mechanisms with the terrain adaptability of legged mechanisms, demonstrating substantial application potential in unstructured scenarios such as inspection, logistics, and rescue operations [1].

The wheel-legged robot system exhibits nonlinearity, strong coupling, and inherent underactuation, posing challenges for the design of its autonomous balance controller. To date, various control methods have been applied to such systems, including whole-body control (WBC) [2], [3], model predictive control (MPC) [4], [5], and the linear quadratic regulator (LQR) [6], [7]. Compared with MPC, which requires online rolling optimization at each control cycle, LQR obtains an analytical optimal state feedback law by solving the algebraic Riccati equation offline. This approach significantly reduces real-time computational overhead while maintaining control quality, making it particularly suitable for balance tasks of wheel-legged robots that demand high control frequencies. Studies have shown that the LQR controller exhibits high reliability and robustness in robot balance tasks[8].

LQR provides a complete mathematical framework for optimal control of linear systems, where the core lies in the design of the state weighting matrix **Q** and the control weighting matrix **R**. These matrices directly determine how the controller trades off regulation performance against control energy consumption. However, the LQR theory itself does not offer a systematic method for selecting the weighting matrices [9]. In engineering practice, **Q** and **R** are typically determined by manual iteration. A more fundamental difficulty is that balance control of wheel-legged robots is inherently a multi-dimensional performance trade-off problem: designers must simultaneously consider multiple objectives such as attitude stability, disturbance rejection capability, speed tracking accuracy, energy efficiency, and control signal smoothness [10]. Manual trial-and-error can hardly navigate this high-dimensional parameter space to quickly obtain a weighting combination that yields a high-performance controller satisfying the design preferences.

With the maturation of evolutionary computation theory, intelligent algorithms have been progressively introduced into the automatic tuning of controller parameters, emerging as an effective solution to the parameter trial-and-error problem in traditional control systems [11]. For example, in DC motor speed control, researchers have combined particle swarm optimization (PSO) with genetic algorithm (GA) to determine the weighting matrices of an integral LQR (ILQR), significantly improving speed regulation performance and disturbance rejection capability [12]. In permanent magnet linear synchronous motor (PMLSM) control, an improved PSO algorithm incorporating the crossover and mutation mechanisms of GA has been employed to tune the parameters of a linear active disturbance rejection controller (LADRC), effectively avoiding premature convergence and achieving faster dynamic response and stronger robustness under no-load start-up and load disturbance conditions [13]. In UAV flight control, GA has been used to optimize the weighting matrices of a robust LQR servomechanism (RSLQR); through a multi-objective fitness function defined via nonlinear simulation, the control system attains better response speed and steady-state accuracy during complex maneuvers involving varying altitude and heading [14]. Similarly, for quadrotor UAVs, a PSO algorithm enhanced

with companion point set initialization and the golden sine mutation operator has been successfully applied to tune the parameters of a sliding mode active disturbance rejection controller, yielding improved attitude accuracy and faster convergence under external disturbances [15].

The above studies demonstrate that intelligent algorithms can effectively solve the problem of automatic controller parameter tuning across various complexities and domains, confirming their broad applicability in handling nonlinear, strongly coupled, and other challenging systems. However, beyond the aforementioned challenges, wheel-legged robots introduce a critical new dimension: the system dynamics parameters actively change with leg extension and retraction, causing the state matrix and control matrix to be continuously time-varying. Most existing intelligent-algorithm-based tuning works assume fixed model parameters, making them difficult to directly extend to operating conditions with large parameter variations. To address this issue, this study designs a control system for a five-link wheel-legged robot that can adapt to variable leg-length conditions. Specifically, the double inverted pendulum (DIP) paradigm [16] is adopted to establish the dynamic model of the robot, which significantly reduces the complexity of model-based control for such complex hardware [17]. Unlike modeling approaches that treat the equivalent inverted pendulum length as a fixed constant [18], [19], this work defines the pendulum length as a variable and introduces the idea of gain scheduling [20]. However, conventional gain scheduling typically requires redesigning the weighting matrices for different operating points, whereas this study determines the LQR weighting matrices by solving an optimization problem using a genetic algorithm (GA), and the resulting fixed weighting matrices can be directly extended to variable leg-length conditions. The core idea of this method is to reformulate LQR weight tuning as a constrained scalar parameter optimization problem solved by a genetic algorithm. Taking the state-space model and the performance preferences expressed via cost function weights as inputs, the method outputs an LQR controller that achieves a favorable trade-off among multiple performance objectives while satisfying physical feasibility. The main contributions of this article can be summarized in the following aspects:

1)An automatic LQR parameter optimization scheme based on the state-space equation is proposed. The diagonal elements of the state weighting matrix $\mathbf{Q}\in R^{6\times6}$ and the control weighting matrix $\mathbf{R}\in R^{2\times2}$ are taken as optimization variables. A weighted scalar cost function is designed, integrating overshoot, control energy, steady-state error, torque variation rate, and velocity tracking error. Meanwhile, physical priors such as control gain sign constraints and system divergence penalties are encoded as constraints to ensure that the GA search results always lie within a physically feasible design space.

2)A deployment framework combining offline GA global optimization and online gain scheduling is constructed. The optimal weighting matrices obtained by GA at the nominal leg length are adopted as the invariant design criterion. By grid-based offline solution of the Riccati equation and cubic polynomial fitting, the fixed weighting matrices at the nominal leg length are extended to the entire leg-length range, achieving online continuous adaptation of the feedback gain to leg length. This framework demonstrates that the automatically optimized results from GA can be generalized to parameter-varying intervals, avoiding the high computational cost of re-running GA for each operating point.

The remainder of this paper is organized as follows. Section 2 establishes the dynamic model of the wheel-legged robot using the double inverted pendulum (DIP) paradigm and maps the simplified model to actual joint torques via virtual model control (VMC). Section 3 elaborates on the proposed automatic design method for the LQR controller, including the weighted cost function design, the GA optimization procedure, and the gain scheduling strategy. Section 4 validates the effectiveness of the proposed method through three types of simulation experiments: balancing control, fall recovery, and terrain adaptation. Section 5 concludes the paper.

2. Dynamic Modeling

The Newton-Euler method is adopted to establish a reduced-order double inverted pendulum (DIP) dynamics model of the wheeled self-balancing robot. This model provides a unified description of the hybrid nonlinear and underactuated characteristics resulting from the dual motion mechanisms of the wheeled base and the articulated legs. By ignoring the high-frequency dynamics of leg bending, applying the rigid contact rolling constraint, and introducing virtual model control (VMC), the complexity of modeling for real-time motion control is effectively reduced. To facilitate controller design and analysis, multiple coordinate frames are established on the robot, as shown in Fig. 1(b). Under this modeling framework, the DIP approximation method achieves effective integration of the controller while keeping the body tilt angle within ±0.6 rad, which is the key operational range for preventing falls.The reference is set based on the robot's stationary equilibrium posture in the world coordinate system $\{W\}=[x_W,y_W,z_W]$ .The body coordinate system $\{B\}=[x_B,y_B,z_B]$ is obtained by rotating around the z_W-axis of the world coordinate system.The body reference coordinate system $\{BB\}=[x_{BB},y_{BB},z_{BB}]$ aligns with the robot's azimuth.These coordinate systems are interrelated through transformation matrices, and the velocity vectors of different parts of the robot are expressed in these frames. For detailed expressions, please refer to 2.1 in the supplementary materials. The parameters defining the primary models of the system are listed in Table 1.

Table 1. Parameter definition

Symbol	Definitions
m_ω,m_l,m_c	Wheel, pendulum and body mass
$r,L_0,L_C L_w,l$	Wheel radius, pendulum length, position of mass of the chassis, position of mass of the pole, the distance from the COM of the chassis to the hinge point of the virtual swing pole
I_ω,I_L,I_c	Wheel moment of inertia, pendulum moment of inertia, chassis moment of inertia
$\varphi,\dot{\varphi},\theta,\dot{\theta}$	Body's pitch and its rate, leg's pitch and its rate
T_p,T	Virtual leg torque, drive wheel torque
H,V	Horizontal force, vertical force
L_W ,L_C	Distances from the leg COM to the wheel and body joints respectively
f_s	The frictional force of the ground on the wheel

2.1. DIP Model

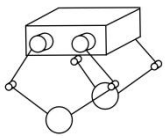

Fig 1. Wheel-leg robot model.

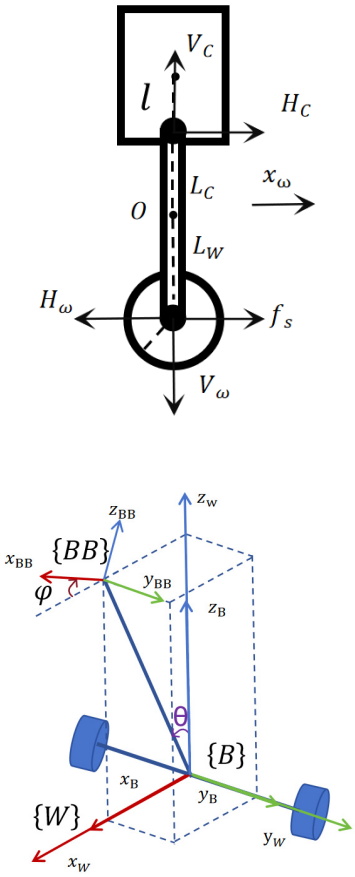

Fig 2. (a)DIP Model. (b)Description of the coordinate systems.

The wheel-legged robot is modeled as a double inverted pendulum (DIP) using the Newton-Euler method, as illustrated in Fig. 1(a). The system has two rotational degrees of freedom: the leg angle θ,measured from the vertical,and the body pitch angle φ,measured from the leg. The wheel is assumed to roll without slipping on the ground. The parameters are defined in Table 1, where L denotes the equivalent pendulum length,the distance from the wheel axis to the hinge point of the leg-body assembly. Directions shown in Fig.1. are positive.

The dynamic equations of rotation and translation of the driving wheel:

$$m_\omega \ddot{x_\omega} = f_s - H_\omega \tag{1}$$

$$I_\omega \frac{\ddot{x_\omega}}{r} = T - f_s r \tag{2}$$

$$H_W - H_c = m_l(\ddot{x_\omega} + \frac{d^2}{dt^2}(L_W sin\theta)) \tag{3}$$

$$V_W - V_C - m_l g = m_l \frac{d^2}{dt^2}(L_W cos\theta) \tag{4}$$

The leg is treated as a rigid pendulum of length L. The horizontal and vertical force balance give:

$$I_L\ddot{\theta} = (V_W L_W + V_C L_C)sin\theta + T_p - (H_W L_W + H_C L_C)cos\theta - T \tag{5}$$

For the upper body, the force and torque equations are:

$$H_C = m_C(\ddot{x_c} - \frac{d^2}{dt^2}(l\sin\varphi)) \tag{6}$$

$$V_c - m_c g = m_c \frac{d^2}{dt^2}(L_0\cos\theta + l\cos\varphi) \tag{7}$$

$$I_c\ddot{\varphi} = T_p + H_c l\cos\varphi + V_c l\sin\varphi \tag{8}$$

Select the state variable $\boldsymbol{x}=[\theta \quad \dot{\theta} \quad x_\omega \quad \dot{x_\omega} \quad \varphi \quad \dot{\varphi}]$, and the inputs $\boldsymbol{u}=[T \quad T_p]^T$ .The nonlinear dynamics can be written as $\dot{x}=f(x,u)$ .The equilibrium point for a constant forward speed v_0 is $\boldsymbol{x_0}=[0 \quad 0 \quad 0 \quad v_0 \quad 0 \quad 0]$ with $u_0=[0 \quad 0]^T$,established the state-space equations:

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \tag{9}$$

Where **A** is a 6×6 matrix,**B** is a 6×2 matrix.

Where the Jacobian matrices are evaluated symbolically. For the nominal leg length $L = 0.08m$ and with the physical parameters listed in Table 2, the numerical matrices are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 328.1099 & 0 & 0 & 0 & 7.8354 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -15.7116 & 0 & 0 & 0 & -0.0042 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 57.8751 & 0 & 0 & 0 & 57.2459 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ -513.4020 & 336.6343 \\ 0 & 0 \\ 31.8009 & -13.7297 \\ 0 & 0 \\ -50.9218 & 419.2517 \end{bmatrix}$$

The output matrix is chosen as y=**C**x, **C**=$\mathbf{I}_{6\times6}$for full-state feedback. The controllability matrix$C$=[**B**,**AB**,...,$\mathbf{A}^5$**B**]has full rank(6),so the pair(**A**,**B**)is completely controllable, satisfying a prerequisite for LQR design[21].

Table 2. Parameters of wheel leg robot

Parameter	Values
Chassis mass m_c/kg	1.44
Radius of wheel R_1/m	0.0618
Wheel mass m_ω/kg	0.6
The moment of inertia of chassis around the COM $I_C/kg{\cdot}m^2$	0.00271
The moment of inertia of wheel $I_\omega/kg{\cdot}m^2$	0.00109
Leg length L/m	0.08

2.2. VMC and FK of the Five-Link Mechanism

The robot employs a symmetric five-link mechanism. To address the coupling between the virtual pendulum motion in the wheeled inverted pendulum model and the actual joint motion of the five-link structure, virtual model control (VMC) is introduced as a decoupling approach. Specifically, the control objectives are decomposed into two independent tasks: maintaining attitude balance via the virtual pendulum angle θ and body pitch angle φ, and adapting to terrain by adjusting the five-link configuration $\mathbf{q} = [\varphi_1, \varphi_4]^T$. The core of VMC lies in using the principle of virtual work to establish a mapping from the virtual task space, which includes the pendulum thrust F and the virtual hip torque T_p, to the actual joint space described by the motor torques T_1 and T_2 , i.e., $\mathbf{T}=\mathbf{J}^T\mathbf{F}$ .This mapping, realized through the kinematic Jacobian matrix **J**, decouples the attitude control commands from the leg motion commands, allowing the high-level attitude controller to operate independently of the detailed kinematics of the five-link mechanism. Consequently, the computational complexity of real-time control is significantly

reduced.

The parameters of the five-link mechanism are shown in Fig. 2, where B and D are the two passive joints, l_2 and $l_3$are the link lengths, and φ_2 , φ_3 are the angles of the corresponding links with respect to the horizontal direction. The joint space is defined as $\mathbf{q} = [\varphi_1, \varphi_4]^{\mathrm{T}}$ and the task space as $\mathbf{s} = [\mathrm{L}_0, \varphi_0]^{\mathrm{T}}$. The closed-loop constraint equations are:

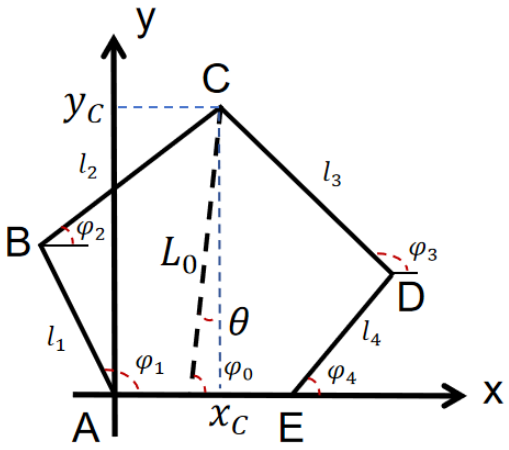

Fig 3. Motion trajectory diagram of five-link structure.

$$\begin{cases} x_B + l_2 \cos\varphi_2 = x_D + l_3 \cos\varphi_3 \\ y_B + l_2 \sin\varphi_2 = y_D + l_3 \sin\varphi_3 \end{cases} \tag{10}$$

From these constraints,the forward kinematics $s=f(q)$ is obtained as:

$$\begin{cases} L_0 = f_1(\varphi_1, \varphi_4) \\ \varphi_0 = f_2(\varphi_1, \varphi_4) \end{cases} \tag{11}$$

The Jacobian matrix is obtained by total differential on both sides of the equation,and the mapping relationship between the joint torque and output of the end-effector of the mechanism is derived based on the principle of virtual work.Total differentiation of the equations:

$$\begin{bmatrix} \delta L_0 \\ \delta \varphi_0 \end{bmatrix} = \begin{bmatrix} \frac{\delta f_1}{\delta \varphi_1} & \frac{\delta f_1}{\delta \varphi_4} \\ \frac{\delta f_2}{\delta \varphi_1} & \frac{\delta f_2}{\delta \varphi_4} \end{bmatrix} \begin{bmatrix} \delta \varphi_1 \\ \delta \varphi_4 \end{bmatrix} \tag{12}$$

Simply expressed as:

$$\delta \mathbf{s} = \mathbf{J} \delta \mathbf{q} \tag{13}$$

According to the principle of virtual work,the actual joint torques $\mathbf{T}=[T_1 \quad T_2]^{\mathrm{T}}$ and the virtual torques $\mathbf{F}=[F \quad T_p]^{\mathrm{T}}$ satisfy:

$$\mathbf{T}^{\mathrm{T}} \delta \mathbf{q} = \mathbf{F}^{\mathrm{T}} \delta \mathbf{s} \tag{14}$$

Substituting $\delta \mathbf{s} = \mathbf{J} \delta \mathbf{q}$ and eliminating δq yields the mapping:

$$\mathbf{T} = \mathbf{J}^{\mathrm{T}} \mathbf{F} \tag{15}$$

3. Controller Design

3.1. LQR Controller

For the robot state-space equation $\dot{x}=\mathbf{A}x+\mathbf{B}u$ derived in Section 2, the objective of LQR control is to find a full-state feedback control law $u(t)=-Kx(t)$ that minimizes the following quadratic cost function:

$$min(J) = \int_0^{\infty} (\mathbf{x}^{\mathrm{T}} \mathbf{Q} \mathbf{x} + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u}) dt \tag{16}$$

where the state weighting matrix $\mathbf{Q} \in \mathrm{R}^{6\times6}$ is positive semi-definite and reflects the importance of state deviations from the equilibrium point, and the control weighting matrix $\mathbf{R} \in \mathrm{R}^{2\times2}$ is positive definite and penalizes the control energy consumption. The essence of this optimization is to strike an optimal balance between regulation performance and control effort.The optimal state feedback gain matrix $\mathbf{K}$ is obtained by solving the following algebraic Riccati equation:

$$\mathbf{A}^{\mathbf{T}}\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{P} + \mathbf{Q} = 0 \tag{17}$$

and the feedback gain is given by

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{P} \tag{18}$$

Where $\mathbf{P}$ is the unique positive-definite solution of the Riccati equation. It is evident that the choice of $\mathbf{Q}$ and $\mathbf{R}$ directly determines the closed-loop performance. However, LQR theory itself does not provide a systematic guideline for selecting these weighting matrices. In the standard LQR formulation, the trade-off among multiple performance indices is abstracted into a set of diagonal entries in $\mathbf{Q}$ and $\mathbf{R}$, which lack intuitive physical meaning. The main difficulties are twofold: 1) the multi-objective coordination dilemma relying heavily on expert heuristics, and 2) the implicit constraints of engineering feasibility.

Therefore, the key idea of this paper is to retain the theoretical framework of LQR for automatically computing $\mathbf{K}$, while employing a genetic algorithm (GA) to perform the search from performance requirements to appropriate weighting matrices. This closes the overall controller design loop into an automated procedure. Within this tuning framework, optimal control can be realized by selecting suitable weighting matrices and then computing the corresponding state feedback gain matrix.

3.2. Optimal Tuning Based on Genetic Algorithm

In conventional LQR synthesis, the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are often chosen through an iterative, element-by-element process, which is time-consuming. More importantly, the resulting control law may exhibit degraded performance when applied to the accurate nonlinear model of the wheel-legged robot or in real experiments, necessitating additional tuning effort. To overcome this problem, this section develops a genetic algorithm (GA) based procedure for automatically tuning the weighting matrices. The resulting LQR controller minimizes a cost function that combines stability and performance metrics evaluated via nonlinear simulation.

3.2.1. Modeling of Optimization Problems

The genetic algorithm is a global stochastic search method inspired by natural evolution, suitable for complex optimization problems with non-convex, non-differentiable, or multi-modal objective functions. In the proposed automatic LQR design framework, GA searches the predefined parameter space according to quantitative performance feedback, seeking a combination of weighting matrices that yields a small scalar performance index J. Hence, the controller design requirements must be formalized as a constrained optimization problem, including decision variables, constraints, and an objective function.

The decision variables are taken as the diagonal entries of the state weighting matrix$\mathbf{Q} \in \mathrm{R}^{6\times6}$ and the control weighting matrix $\mathbf{R} \in \mathrm{R}^{2\times2}$. For the 6-state, 2-input system, let ξ denote the parameter vector to be optimized:

$$\xi = [q_1, q_2, \ldots, q_n, r_1, r_2]^T \in \mathbb{R}^8$$

with $\mathbf{Q}$=diag(q1,q2,...,q6) and $\mathbf{R}$=diag(r1, r2) . The corresponding LQR state-feedback control law is

$$\mathbf{u}(t) = -\mathbf{K}(\xi)\mathbf{x}(t) \tag{19}$$

To evaluate the control performance of a candidate parameter set ξ a scalar cost function $J(\xi)$ is defined (see Section 3.2.2), which is evaluated through nonlinear simulation. The LQR weighting matrix tuning problem is then cast as the following constrained parameter minimization:

$$\min_{\xi \in \mathcal{D}} J(\xi) \tag{20}$$

where the feasible domain D is defined by:

Positive definiteness constraints:$q_i \geq 0(1,...,6), r_j > 0(j=1,2)$;

Search range constraints:$q_i \in [10^{-3}, 10^4]$, $r_j \in [10^{-3}, 100]$;

Physical feasibility constraints: sign conditions on the control gains: $(\mathbf{K}(\xi))_{1,5} > 0, (\mathbf{K}(\xi))_{1,6} > 0, (\mathbf{K}(\xi))_{2,5} > 0, (\mathbf{K}(\xi))_{2,6} > 0$;and a divergence penalty that set $J=10^{10}$whenever the tilt angle exceeds 30°.

3.2.2. Design of Fitness Function

Balancing control for a wheel-legged robot is not about optimizing a single index, but rather a trade-off among several competing performance aspects. Considering dynamic response, control energy, steady-state accuracy, control smoothness, and speed tracking, the fitness function J is defined as the following weighted sum:

$$J = \omega_1 M_p + \omega_2 E + \omega_3 e_{ss} + \omega_4 E_{rate} + \omega_5 E_{vel} \tag{21}$$

The individual terms are defined as follows.

Maximum attitude deviation M_p: the largest absolute value of the body pitch angle φ and the leg angle θ over the entire simulation, i.e.

$$M_p(\xi) = \max\{(\max_t|\varphi|, \max_t|\theta|)\},$$

directly measures the dynamic stability after an initial disturbance or command change; a smaller M_p implies smoother attitude transitions.

Control energy E: the time integral of the squared sum of the two control inputs, namely the wheel torque T and the virtual hip torque T_p:

$$E = \int (T^2 + T_p^2)\,\mathrm{d}t$$

representing the energy cost paid by the controller.

Steady-state attitude error e_{ss}:the sum of the absolute mean values of the body pitch angle and leg angle during the steady-state phase:

$$e_{ss}(\xi) = \frac{1}{N_{end}} \sum_{k \in \mathcal{K}_{end}} (|\varphi(t_k)| + |\theta(t_k)|),$$

where $\mathcal{K}_{end}$ is the set of time indices in the steady-state phase. This metric quantifies the attitude holding accuracy after transients have decayed.

Control oscillation penalty E_{rate} :the integral of the squared torque rates:

$$E_{rate} = \int_0^T (\dot{T}^2 + \dot{T}_p^2)dt,$$

Where $\dot{T}^2$ and $\dot{T}_p^2$ are approximated by numerical differences. A smallerE_{rate} corresponds to smoother torque outputs and helps avoid actuator saturation.

·Velocity tracking error ISE_{vel}:the integral of the squared deviation between the desired velocity v_{ref} and the actual velocity v:

$$ISE_{vel} = \int_0^T (v - v_{ref})^2 dt.$$

The weighting coefficients ω_i are set according to the magnitude of each term and the design priorities. In this paper, we use ω_1=0.5,ω_2=0.01,ω_3=0.1,ω_4=0.01,ω_5=10.0 . The largest weight on velocity tracking error reflects the fundamental requirement of the wheel-legged robot as a mobile platform. The second largest weight on overshoot ensures system stability, while the smaller weights on energy and torque rate strike a balance between energy consumption and control smoothness.

3.2.3. Optimization via Genetic Algorithm

The algorithm takes as input the linearized system matrices and the nonlinear model, with the initial population already containing several heuristic test points. Through evolution iterations driven by simulation-based performance feedback, it gradually converges toward the optimal weighting matrices. Three levels of engineering constraints are explicitly embedded in the procedure—positive definiteness, gain sign conditions, and a divergence penalty. Any individual violating these physical priors is assigned an extremely large fitness value, thereby being naturally eliminated during evolution and ensuring that the search remains confined to a feasible design space. The resulting LQR gain matrix $\mathbf{K}^*$is ready for direct controller deployment, while the weighting matrices $\mathbf{Q}^*$ and $\mathbf{R}^*$ are subsequently reused in the gain-scheduling framework of Section 3.3.

Algorithm 1: LQR Parameter Optimization via Genetic Algorithm

Input: System matrices A, B, model M, leg length l, simulation time T_S,initial state x_0

Output: Optimal weighting matrices $\mathbf{Q}^*$,$\mathbf{R}^*$,optimal feedback gain K^*

1: **Linearization and controllability check:** [**A** **B**] ←*get_AB*(*l*)

if rank(ctrb[**A** **B**])<dim(**A**) then error

2:**Set optimization dimensions:***n*←*dim*(**A**), *m*←(**B**)

3:**Define search bounds**: **lb**←$\left[10^{-3} l_{l\times(n+m)}\right]$, **ub**←$\left[10^{4} l_{l\times(n+m)}\right]$

4: **Configure genetic algorithm:** Population size N_p←30, max generations G_{max}←40, stall gens G_{max}←5

Enable parallel computing

5:**Define objective**:*J*(*x*)←*CostFunction*(*x*,**A**,**B**,M,T_S,x_0)

x^*←ga(*J*,*n*+*m*,**lb**,**ub**,*opts*), $\mathbf{Q}^*$←*diag*(x_1^*,…,x_n^*), $\mathbf{R}^*$←*diag*(x_{n+1}^*,...,x_{n+m}^*), K^*←lqr(**A**, **B**, $\mathbf{Q}^*$,$\mathbf{R}^*$)

3.3. Leg-Length-Dependent Gain-Scheduled LQR

In the wheel-legged robot model established in Section 2, the equivalent inverted pendulum length L corresponds to the actual distance from the robot's centre of mass to the wheel axle. When the robot actively extends or retracts its legs to adapt to different terrains, this parameter varies continuously

over the interval $[L_{min},L_{max}]$ (taken as 0.08 m to 0.19 m in this paper). If a fixed-gain LQR controller designed at a nominal leg length L_0 is used, the closed-loop poles will shift when the actual leg length L deviates from L_0, potentially degrading control performance or even causing instability.

From the dynamic modelling in Section 2, both the state matrix $\mathbf{A}$ and the control matrix $\mathbf{B}$ are analytic functions of the leg length L. Consequently, the solution $\boldsymbol{P}(L)$ of the algebraic Riccati equation and the corresponding feedback gain matrix.

$$\mathbf{K}(L) = \mathbf{R}^{-1}\mathbf{B}(L)^{\mathrm{T}}\mathbf{P}(L)$$

are also continuous functions of L. This property makes it possible to apply gain-scheduling techniques to maintain system stability and performance over the entire leg-length range.

The fixed weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ obtained via the genetic algorithm optimization described in Section 3.2 can be directly extended to the full leg-length interval without rerunning the optimization for each leg length. This property stems from the comprehensive trade-off among multiple performance indices in the fitness function, which endows the obtained weighting matrices with a certain inherent robustness to leg-length variations. The fixed weighting matrices used in this paper are

$$\mathbf{Q} = diag(2.0,0.07,10.0,5.0,299,0.60), \mathbf{R} = diag(20,40.9453)$$

which are obtained from a single offline optimization. The subsequent gain-scheduling procedure relies only on offline solution of the Riccati equation and polynomial fitting, thus avoiding time-consuming GA searches for each leg length. Fixing $\mathbf{Q}$and $\mathbf{R}$also ensures that the physical meaning of the control objectives remains consistent across different leg lengths.

Over the leg-length interval $[L_{min},L_{max}]$, a finite set of operating points L_i is selected with a uniform step size of 0.01 m. For each L_i, the linearised models $\mathbf{A}(L_i)$ and $\mathbf{B}(L_i)$ are obtained via symbolic derivation, and the corresponding LQR gain matrix $\mathbf{K}(L_i) \in \mathbb{R}^{2\times 6}$ is computed by solving the Riccati equation with the fixed $\mathbf{Q}$ and $\mathbf{R}$. Each element $k_{\mathrm{ij}}(L)$ of the gain matrix (with i=1,2 and j=1,...,6) is treated as a function of the leg length L and fitted by a cubic polynomial:

$$k_{pq}(L) = \alpha_{pq,3}L^3 + \alpha_{pq,2}L^2 + \alpha_{pq,1}L + \alpha_{pq,0}.$$

The coefficients $\alpha_{\mathrm{ij}}$are determined via least-squares fitting. Over the compact interval $[L_{min},L_{max}]$, the cubic polynomial provides sufficient accuracy for all gain elements, as confirmed by the small residuals obtained during the fitting process.

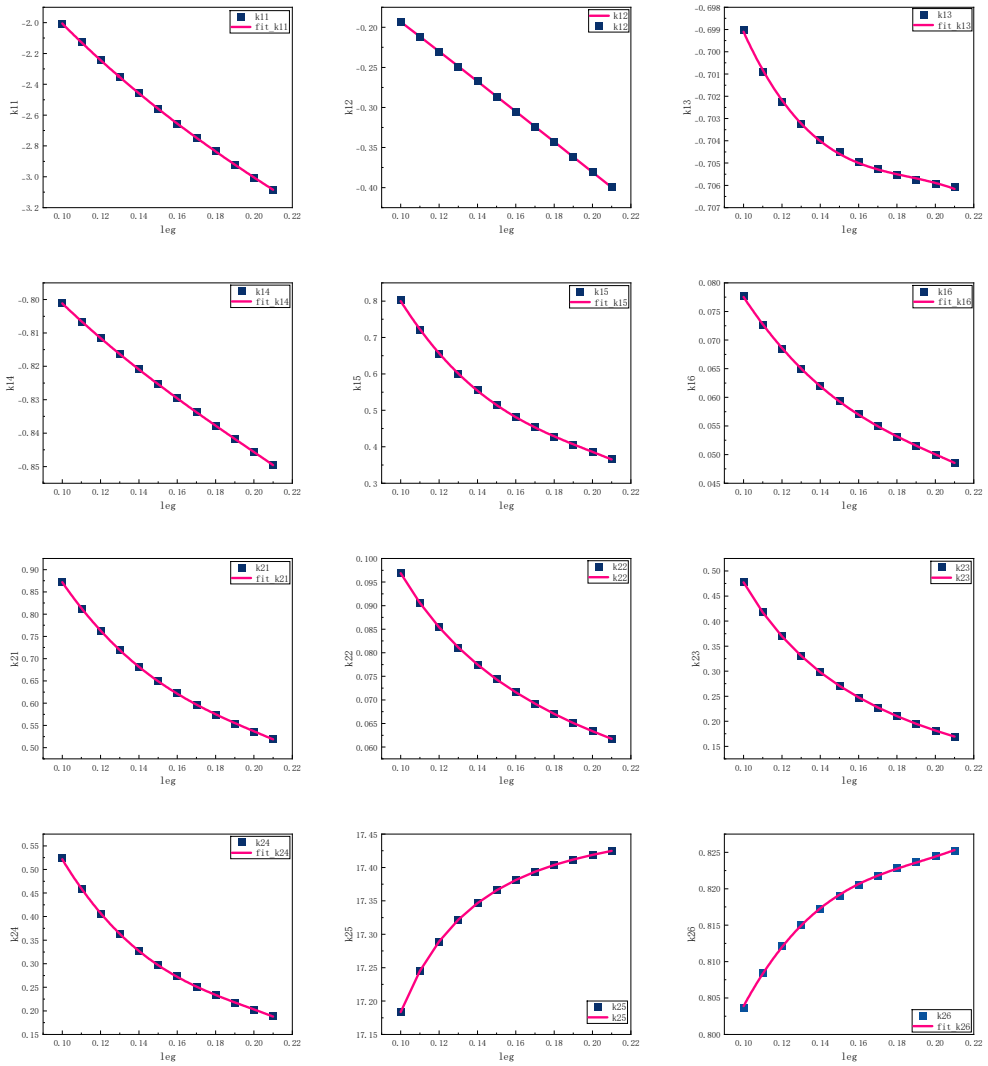

Fig 4. K Matrix Cubic Polynomial Fitting.

4. Simulation Experiment

The GA parameters listed in Table 3 are chosen based on the characteristics of the problem. The optimization involves only eight decision variables, a low-dimensional space. A population size of 30 maintains diversity while keeping the per-generation simulation time moderate. The maximum generation count of 40 is derived from pre-experiments: the fitness value typically saturates around the 30th generation, and further iterations yield negligible improvement. A crossover fraction of 0.8 effectively recombines favourable genes from different weight combinations—e.g., combining a good angle penalty from one candidate with a proper speed-tracking term from another—thereby accelerating the search for a multi-objective trade-off. The adaptive feasible mutation function (initial step size 0.01) actively avoids generating infeasible individuals that violate positive definiteness or gain-sign constraints, preventing the search from stalling due to excessive penalization. Stochastic uniform selection together with an elite count of 2 balances population diversity and convergence monotonicity. A stall generation limit of 5 terminates iterations that show no significant improvement, avoiding unnecessary computational overhead.

To validate the proposed GA-based LQR controller, three typical scenarios are simulated: fall recovery and balancing, velocity tracking, and terrain adaptation. These cover a complete verification chain from transient disturbance rejection to mobile control and robustness against parameter variations. In all experiments, the controller uses the optimal weighting matrices obtained in Section 3.2, i.e.

$$\mathbf{Q}_{\text{opt}} = diag(2.0,0.07,10.0,5.0,299,0.60), \mathbf{R}_{\text{opt}} = diag(20,40.9453)$$

and the gain-scheduled LQR described in Section 3.3 covers the full leg-length range. To emulate physical actuator limits, the wheel torque and the hip torque are saturated at $\pm 3N \cdot m$ and $\pm 9N \cdot m$ respectively. This parameter configuration consistently converged to a set of weighting matrices that pass all typical test scenarios in preliminary experiments, confirming its suitability for the wheel-legged robot LQR tuning problem.

Table 3. Genetic algorithm parameters

Parameter	Value
Population	30
Maximum generations	40
Crossover fraction	0.8
Mutation function	Adaptive feasible
Selection method	Stochastic uniform
Elite count	2
Stall generation limit	5

4.1. Fall Recovery and Balancing

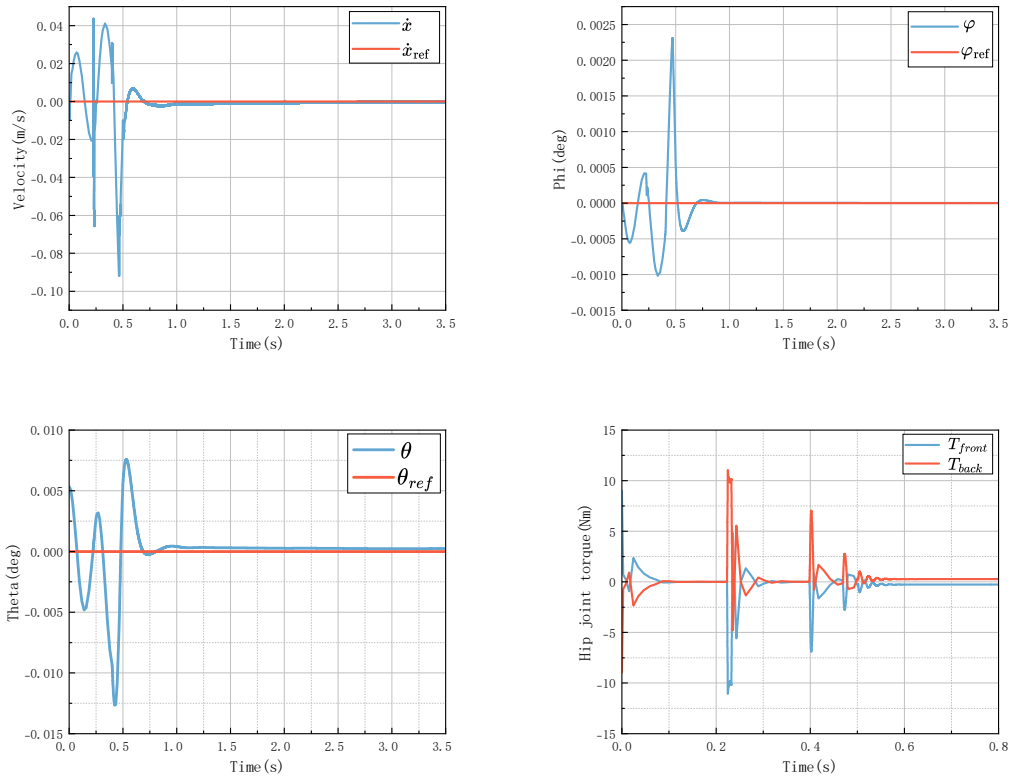

Fig 5. The simulation results of the robots in the fall and balancing experiment

This experiment aims to verify the robot's ability to recover dynamic balance after a severe initial impact and to maintain steady-state attitude accuracy. The robot is dropped freely from a height of 20 cm to the ground, simulating an emergency recovery scenario after encountering a step or a hole. The simulation results are shown in Fig. 5.

Under the action of the GA-LQR controller, the body pitch angle φ and the leg angle θ converge rapidly to near zero within approximately 1.2 s after ground contact. The maximum attitude deviation is 0.43°, and the steady-state attitude error e_{ss} is nearly zero, indicating that the system can quickly and smoothly re-establish upright balance. The control energy consumption is E=0.0019, which is extremely low.

4.2. Velocity Tracking

The velocity tracking experiment evaluates the GA-LQR controller's performance under a mobile control objective. In this experiment, the robot starts from rest and tracks a reference velocity profile v_{ref} over a simulation period of 10 s, which includes acceleration and constant-speed phases, thereby emulating typical speed conditions in practical tasks. Figure 5 shows the velocity tracking and attitude response results obtained with the controller. Throughout the speed transitions, the actual velocity v closely follows the reference trajectory. The integral squared error (ISE) of velocity tracking is 0.1480 ,indicating a low average deviation between the actual and desired speeds. In terms of attitude control, the maximum peak of the body pitch angle φ is 10.17° , occurring at the beginning of the acceleration phase. The mean pitch angle during the steady-state phase is nearly zero, and the leg angle θ also converges to zero, demonstrating that the GA-based LQR controller enables the robot to maintain a stable upright posture during constant-speed travel.

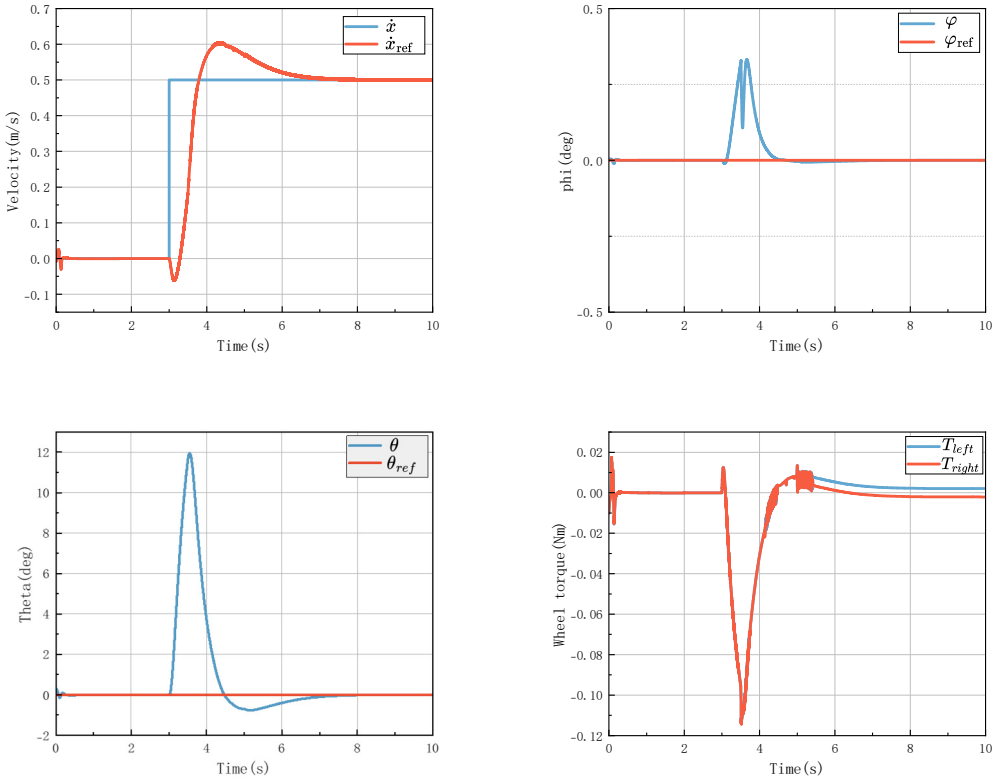

Fig 6. The simulation results of the robots in velocity tracking experiment

4.3. Terrain Adaptation

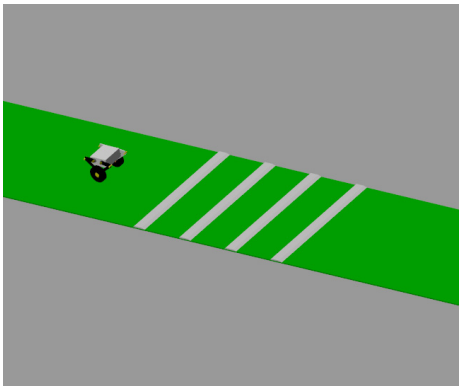

Fig 7. Crossing terrain experiment.

The terrain adaptation experiment evaluates the continuous self-adaptive capability of the gain-scheduled LQR controller under varying leg-length conditions. The robot traverses a series of trapezoidal obstacles with a height of 3 cm and a spacing of 0.2 m at a desired speed of 0.5 m/s. Under the effect of the gain scheduling method, the five-link mechanism enables the equivalent pendulum length L to vary continuously, thereby keeping the chassis level.The feedback gains are updated online using the cubic-polynomial fits of $K(L)$ obtained offline. As shown in Fig.8, throughout the entire terrain crossing process, the peak leg angle is approximately 10° while the robot's body pitch angle remains within ±1°. This highlights the advantage of the decoupled attitude-leg control model, which enables the robot to maintain stable body posture through joint-space

adjustments.The quantitative performance metrics are summarised in Table 4: the maximum attitude deviation is 10.10°, the steady-state error is 1.01°, and the velocity tracking ISE_{vel} is $0.1667(m/s)^2 \cdot s$. The control energy E=0.0167 is moderately higher than in the velocity-tracking experiment due to the additional work required for leg actuation. These results confirm that the proposed gain-scheduled LQR maintains stability and smooth control across the entire leg-length range without introducing extra chattering.

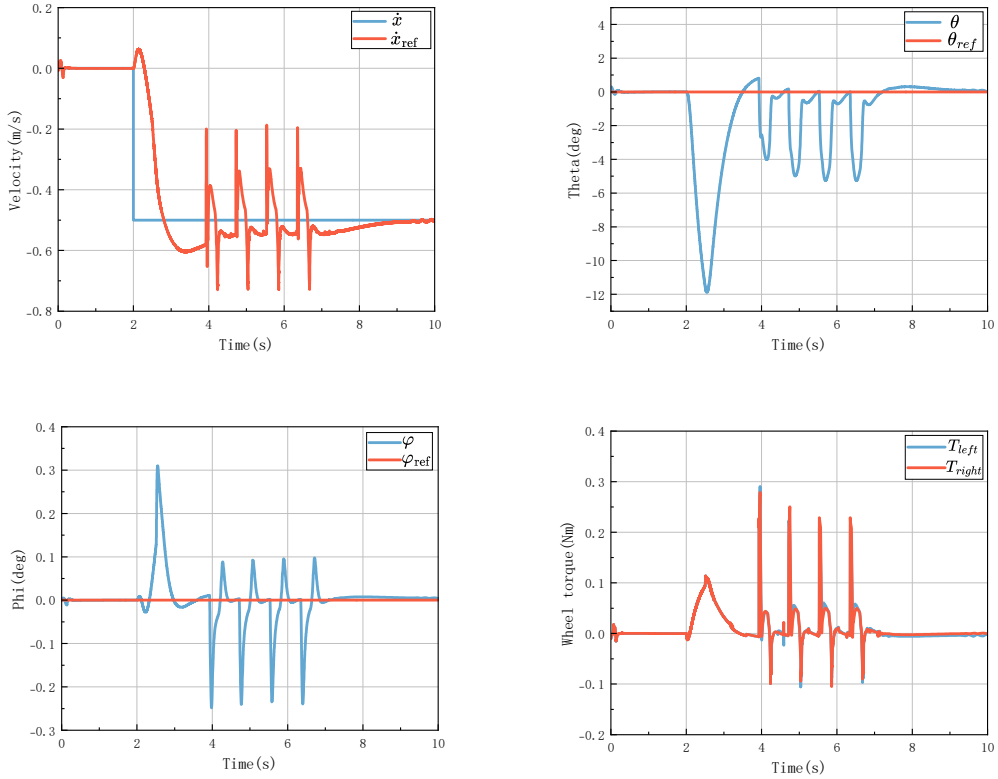

Fig 8. The simulation results of the robots in the terrain adaptation experiment.

Table 4. Performance comparison under three experimental scenarios

Scenario	M_p(deg)	E	E_{rate}	e_{ss}(deg)	ISE_{vel}
Fall recovery	0.43°	0.0019	98.5012	≈ 0°	0.0002
Velocity tracking	10.16°	0.0039	69.5115	0.01°	0.1477
Terrain adaption	10.10°	0.0167	97.84	1.01°	0.1667

5. Conclusion

This paper presents a genetic algorithm (GA) based automatic tuning method for LQR controllers of wheel-legged robots. By formulating the diagonal entries of **Q** and **R** as optimization variables and embedding physical constraints into a weighted cost function, the proposed approach effectively balances multiple performance criteria without manual trial-and-error.

To cope with varying leg lengths, a gain-scheduling strategy is developed that extends the fixed optimal weighting matrices obtained at the nominal leg length to the entire leg-length range via offline Riccati solutions and cubic polynomial fitting. This enables continuous online adaptation of the feedback gain while avoiding repeated optimization.

Nonlinear simulation results confirm that the GA-optimized LQR controller achieves fast balance recovery, accurate speed tracking, and smooth attitude maintenance under terrain variations. The gain-scheduled LQR maintains stable control across the leg-length range with low computational overhead.Future work includes hardware-in-the-loop validation, consideration of sensor noise and actuator dynamics, and the use of multi-objective evolutionary algorithms for Pareto-front exploration.

References

[1] Wang, S., et al. (2021). Balance control of a novel wheel-legged robot: Design and experiments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6782–6788). Xi'an, China. https://doi.org/ 10.1109/ ICRA 48506. 2021.9561474.

[2] Bjelonic, M., et al. (2019). Keep rollin'—Whole-body motion control and planning for wheeled quadrupedal robots. *IEEE Robotics and Automation Letters*, 4(2), 2116–2123. https://doi. org/ 10.1109/LRA.2019.2900926.

[3] Klemm, V., et al. (2020). LQR-assisted whole-body control of a wheeled bipedal robot with kinematic loops. *IEEE Robotics and Automation Letters*, 5(2), 3745–3752. https://doi.org/ 10. 1109/ LRA.2020.2979641.

[4] Cao, H., Lu, B., Liu, H., Liu, R., & Guo, X. (2022). Modeling and MPC-based balance control for a wheeled bipedal robot. In *2022 41st Chinese Control Conference (CCC)* (pp. 420–425). Hefei, China. https://doi.org/10.23919/ CCC55837.2022. 984 9614.

[5] Pan, Z., Li, B., Jing, H., Niu, Z., & Wang, R. (2023). Wheel-leg collaborative control for wheel-legged robots based on MPC with preview. In *2023 IEEE International Automated*

Vehicle Validation Conference (IAVVC) (pp. 1–8). Austin, TX, USA. https://doi.org/10.1109/IAVVC57639.2023.10262256.

[6] Klemm, V., et al. (2019). Ascento: A two-wheeled jumping robot. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 7515–7521). Montreal, QC, Canada. https://doi.org/10.1109/ICRA.2019.8793693.

[7] Dong, J., Liu, R., Lu, B., Guo, X., & Liu, H. (2022). LQR-based balance control of two-wheeled legged robot. In *2022 41st Chinese Control Conference (CCC)* (pp. 450–455). Hefei, China. https://doi.org/10.23919/CCC55837.2022.9849496.

[8] Xin, G., et al. (2021). Robust footstep planning and LQR control for dynamic quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 6(3), 4488–4495. https://doi.org/10. 1109/ LRA.2021.3069025.

[9] Lavretsky, K. W. E. (2013). *Robust and adaptive control with aerospace applications* (1st ed.). Springer-Verlag.

[10] Li, Z., Yang, C., & Fan, L. (2012). *Advanced control of wheeled inverted pendulum systems*. Springer Publishing Company.

[11] Liu, J. K. (2017). *Intelligent control* (4th ed.). Publishing House of Electronics Industry. (In Chinese)

[12] Aremu, M. B., Abdel-Nasser, M., Alyazidi, N. M., & El-Ferik, A. S. (2024). Disturbance observer-based bio-inspired LQR optimization for DC motor speed control. *IEEE Access*, 12, 152418–152429. https://doi.org/10. 1109/ACCESS. 2024. 345 2677.

[13] Ma, R., Tan, Q., Liu, X., et al. (2026). Active disturbance rejection control for permanent magnet linear synchronous motors using an enhanced particle swarm optimization algorithm. *Electric Machines and Control*, 30(2), 36–48.

[14] D'Antuono, V., De Matteis, G., Trotta, D., & Zavoli, A. (2023). Optimization of UAV robust control using genetic algorithm. *IEEE Access*, 11, 122252–122272. https://doi.org/ 10.1109/ ACCESS. 2023.3329494.

[15] Wang, Y., Xiong, G., Fu, G., et al. (2026). Research on sliding mode active disturbance rejection control for quadrotor UAVs based on an improved particle swarm optimization algorithm. *Advances in Aeronautical Science and Engineering*. https://link.cnki.net/urlid/61.1479.V.20260206.0930.002.

[16] Khatoon, S., Chaturvedi, D. K., Hasan, N., & Istiyaque, M. (2017). Optimal control of a double inverted pendulum by linearization technique. In *2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)* (pp. 123–127). https://doi.org/ 10. 1109/ MSPCT.2017.8363932.

[17] Gao, J., Jin, H., Gao, L., Zhu, Y., Zhao, J., & Cai, H. (2025). Jump control based on nonlinear wheel-spring-loaded inverted pendulum model: Validation of a wheeled-bipedal robot with single-degree-of-freedom legs. *Biomimetics*, 10, 246. https:// doi. org/10.3390/biomimetics10040246.

[18] Chen, J., He, N., Xu, Z., Dou, M., & He, L. (2025). Design and implementation of a novel two-wheeled composite self-balancing robot for stationary operations in unknown terrain. *IEEE Access*, 13, 86032–86045. https://doi.org/ 10.1109/ ACCESS. 2025.3499457.

[19] Chen, M., Ma, H., & Wang, J. (2026). Stabilization of an 8-DoF wheeled bipedal robot via whole-body control with integrated LQR and MPC. In *2026 2nd International Conference on Electrical Automation and Artificial Intelligence (ICEAAI)* (pp. 555–560). Guangzhou, China.

[20] Wang, S., et al. (2020). Gain scheduled controller design for balancing an autonomous bicycle. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7595–7600). Las Vegas, NV, USA. https://doi. org/ 10.1109/IROS45743.2020.9340956.

[21] Kalman, R. (1959). On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3), 110. https://doi. org/10.1109/TAC.1959.1104866.