6568USST: Engineering Project Final report

Title: Modeling and Control of a Biped Serial-Leg Robot with Wheel Actuation

Name: Jingqi Mei
Student ID: SBC-22-1020
LJMU number: 1040153
Major: EEE

Supervisor: Gareth Lewis

Submission Date:

12 Apr 2026

Abstract

This project studies the modeling and control of a biped serial-leg robot with wheel actuation. The main aim is to achieve stable balance and forward motion with a control-oriented mechanical design and a practical control method. First, the leg mechanism is simplified and analyzed by forward kinematics and virtual model control, so that the virtual leg force can be mapped to the joint motor torques. Then, the whole robot is reduced to a wheel-leg inverted pendulum model, and the nonlinear dynamic equations are established using classical mechanics. After that, the model is linearized around the upright equilibrium point and converted into a state-space form. Based on this model, an LQR controller is designed for balance stabilization, while a PID controller is used in the outer loop for speed control under remote operation. To improve adaptability to different leg lengths, gain scheduling is introduced by fitting the LQR feedback matrix as a function of virtual leg length. Finally, a Simscape Multibody model is built for verification. The simulation results show that the robot can recover from landing impact, regulate leg posture, suppress body oscillation, and maintain stable straight-line locomotion.

CONTENTS

1. Introduction 3

2 Method........ 5

2.1 Balance and Vertical Control 5

2.1.1 Modeling 5

2.1.2 VMC........ 7

2.1.3 VMC Jacobi Matrix........ 8

2.2 Inverted Pendulum Model 10

2.2.1 System Modeling........ 10

2.2.2 Mechanical Analysis 11

2.2.3 State Space Model 14

2.3 Controller Design 15

2.3.1 LQR........ 15

2.3.2 PID........ 16

2.4 Simulation 18

3. Conclusion........ 23

Modeling and Control of a Biped Serial-Leg Robot with Wheel Actuation

1. Introduction

Mobile robots are increasingly required to operate in complex, unstructured, and dangerous environments, such as disaster response, radiation inspection, fire detection, and search-and-rescue missions. In these scenarios, robotic platforms must maintain stability while responding rapidly to external disturbances and varying terrain conditions. These demands place strict requirements on both the mechanical structure and control strategy of mobile robots.

Traditional wheeled robots are highly efficient and capable of fast motion on flat ground; however, their adaptability to uneven or discontinuous terrain is limited. In contrast, legged robots offer superior terrain adaptability but often suffer from high mechanical complexity, increased energy consumption, and demanding control requirements. To combine the advantages of both locomotion modes, wheeled–legged robots have attracted significant research interest. By integrating wheels with leg mechanisms, wheeled–legged robots aim to achieve efficient locomotion while retaining the ability to adjust posture and maintain balance in complex environments.

Despite these advantages, wheeled–legged robots are inherently nonlinear, underactuated, and strongly coupled systems, which makes their modelling and control particularly challenging. Among the various research topics in this field, balance control remains a fundamental problem. External disturbances, coupling between wheels and leg joints, and uncertainties in ground contact all contribute to dynamic instability, making balance control a critical issue for practical deployment.

The mechanical topology of wheeled–legged robots play a crucial role in determining both motion capability and control complexity. In recent years, a wide variety of robot structures have been proposed, ranging from complex multi-link leg mechanisms to simplified hybrid designs, such as Swiss Zurich team's two-wheeled *ANYmal* robot [1][2] and *Ascento* robot [3][4]. While elaborate leg structures can enhance terrain adaptability, they often increase the number of degrees of freedom and introduce strong dynamic coupling, which significantly complicates system modelling and control.

In contrast to closed-chain or multi-link leg mechanisms, serial-leg structures consist of leg segments connected in a serial kinematic chain. Serial-leg configurations are widely used in classical legged robotics due to their structural simplicity and intuitive kinematic interpretation. When combined with wheel actuation, serial-leg mechanisms provide a natural means of posture adjustment and vertical compliance while maintaining efficient wheeled locomotion. Importantly, serial-leg designs reduce mechanical redundancy and allow for clearer dynamic modelling, making them particularly attractive for control-oriented robot design.

However, serial-leg wheeled robots remain underactuated systems, as not all degrees of freedom can be directly controlled. The interaction between serial leg joints and wheel actuation introduces coupled dynamics that resemble an extended inverted pendulum system. As a result, achieving stable balance under disturbances remains a challenging control problem, especially when the system is required to operate on resource-constrained embedded platforms.

Various control approaches have been proposed to address balance control in wheeled–legged robots. In the *Ascento* robot paper [4], an advanced optimization-based method called Whole-Body Control (WBC) demonstrated strong performance in high-degree-of-freedom robotic systems. Nevertheless, WBC typically requires accurate dynamic models and substantial computational resources, making it unsuitable for embedded platforms. Consequently, such methods are not aligned with the objectives of this project.

Alternatively, model-based linear control methods, particularly the Linear Quadratic Regulator (LQR) [4], provide an effective balance between control performance and computational efficiency. By linearizing the system dynamics around an upright equilibrium configuration, LQR enables systematic controller design for underactuated balance systems. Furthermore, the simplicity of serial-leg structures facilitates reduced-order modelling, which improves controller robustness and eases real-time implementation.

Motivated by these considerations, this project focuses on the modeling and control of a biped serial-leg robot with wheel actuation. The robot consists of two symmetric serial legs, each terminating in a wheel that provides ground contact and actuation. The serial-leg structure enables posture adjustment through coordinated joint motion, while wheel actuation ensures efficient locomotion and direct control over the system's base dynamics.

In this project, a simplified dynamic model is derived to capture the dominant balance-related behavior of the serial-leg wheeled robot. Based on this model, an LQR-based balance controller is designed to stabilize the robot around its upright equilibrium. The control framework emphasizes robustness, real-time performance, and suitability for embedded implementation, rather than full-body optimization or complex trajectory planning.

The primary objective of this work is to demonstrate that stable balance control of a wheeled–legged robot can be achieved through a control-oriented serial-leg design combined with simplified modelling and linear control techniques. This approach seeks to integrate the structural advantages of serial-leg mechanisms into an inverted-pendulum-like balancing system, providing a practical and robust solution for wheeled–legged robot balance control.

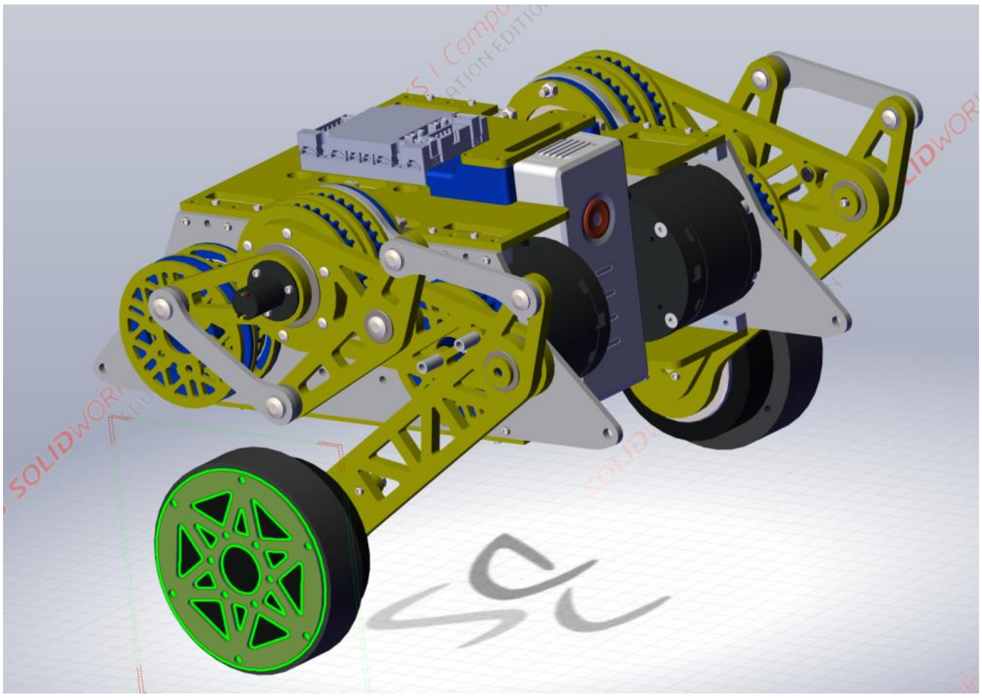

Fig. 1 Whole robot model

2 Method

2.1 Balance and Vertical Control

2.1.1 Modeling

Because the two legs of the robot are symmetrical, VMC can be applied to one leg.

In the figure below, the black lines represent the leg structure.

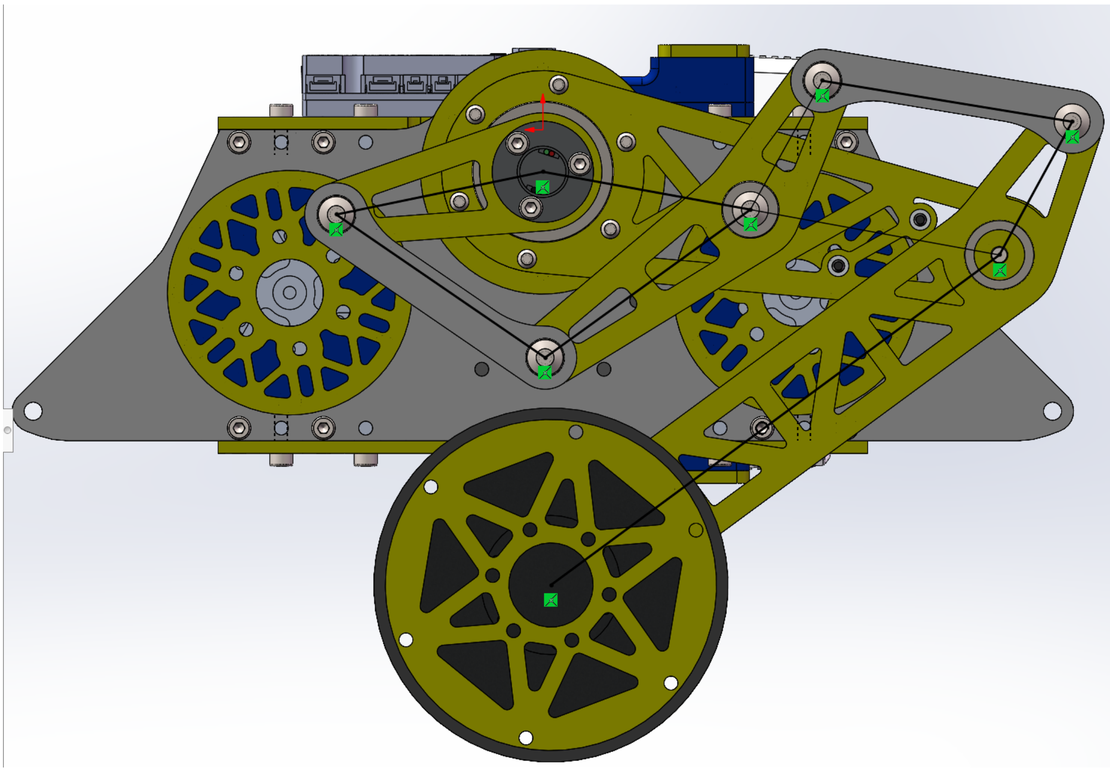

Fig. 2 Side View of the robot

For ease of modeling, it is idealized in Fig. 3. For convenience, the structure is optimized as follows: $AB = AC, BD = CD$, and quadrilateral CGHE is a parallelogram. According to the principle of virtual work, the complex serial-leg model can be transformed into a simple four-link model: AEFI.

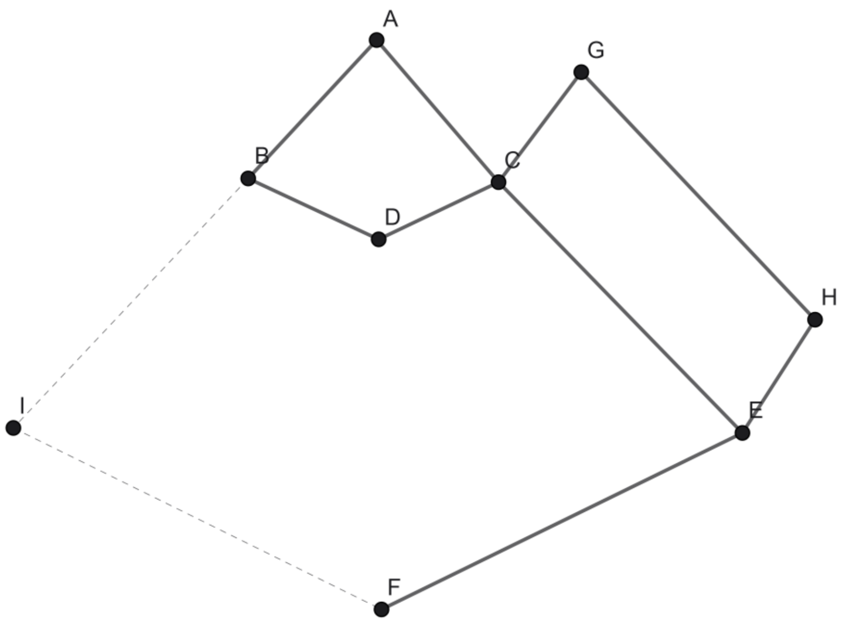

Fig. 3 Simplified model diagram

2.1.2 VMC

To obtain the robot's leg length L_0, forward kinematics calculations are needed for the five-bar linkage of the robot's leg. In the wheel-leg inverted pendulum model, T_p requires the application of virtual model control (VMC) to obtain the output torque of the two joint motors based on T_p. The parameters of the four-link model are defined as shown in the figure.

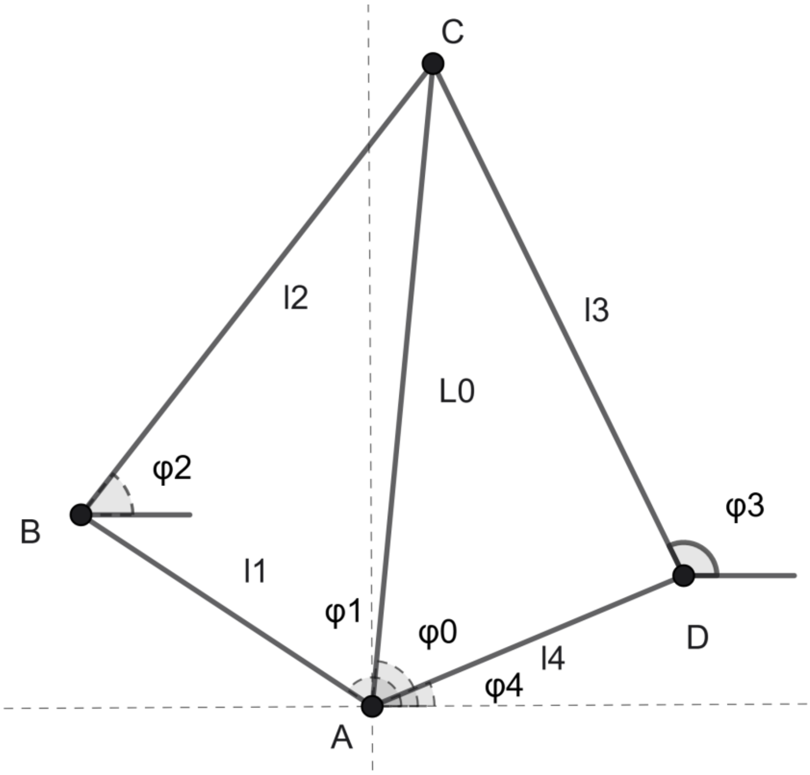

Fig. 4 Four-link model

Using the coordinates of point C as the equality relation, it's easy to obtain from the figure that:

$$\begin{cases} x_B + l_2 cos\varphi_2 = x_D + l_3 cos\varphi_3 \\ y_B + l_2 sin\varphi_2 = y_D + l_3 sin\varphi_3 \end{cases} \tag{1.1}$$

Solve the equation:

$$\Phi_2 = 2\arctan\left(\frac{B_0 + \sqrt{A_0^2 + B_0^2 - C_0^2}}{A_0 + C_0}\right) \tag{1.2}$$

Where：

$$\begin{cases} A_0 = 2l_2(x_D - x_B) \\ B_0 = 2l_2(y_D - y_B) \\ C_0 = l_2^2 + l_{BD}^2 - l_3^2 \end{cases} \tag{1.3}$$

$$l_{BD} = \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} \tag{1.4}$$

Substitute Φ_2 into the rectangular coordinates of point C: $(x_C, \ y_C)$

$$\begin{cases} x_C = l_1 cos\varphi_1 + \ l_2 cos\varphi_2 \\ y_C = l_1 sin\varphi_2 + l_2 sin\varphi_2 \end{cases} \tag{1.5}$$

Which leads to the polar coordinate:

$$\begin{cases} L_0 = \sqrt{{x_C}^2 + y_c^2} \\ \Phi_0 = arctan\dfrac{y_C}{x_C} \end{cases} \tag{1.6}$$

2.1.3 VMC Jacobi Matrix

The four-link model is converted into an inverted pendulum model using VMC. Appropriate virtual components are constructed for each controllable degree of freedom to generate suitable virtual forces. These virtual forces are then mapped to the forces and torques output to the joints.

It is necessary to map the force F at the end of the four-link model along the leg and the torque T_p along the central axis to the torques T_1 and T_2 of rotations AB and AD. Define $x = [L_0 \quad \varphi_0]$, $q = [\varphi_1 \quad \varphi_4]$, and its normal kinematic model is $x = f(q)$. Taking its total differential gives that:

$$\begin{cases} \delta L_0 = \dfrac{\partial L_0}{\partial \varphi_1}\delta\varphi_1 + \dfrac{\partial L_0}{\partial \varphi_4}\delta\varphi_4 \\ \delta\varphi_0 = \dfrac{\partial \varphi_0}{\partial \varphi_1}\delta\varphi_1 + \dfrac{\partial \varphi_0}{\partial \varphi_4}\delta\varphi_4 \end{cases} \tag{1.7}$$

Therefore, the total differential expression of the forward motion model $x = f(q)$ can be written as:

$$\delta x = \begin{bmatrix} \dfrac{\partial L_0}{\partial \varphi_1} & \dfrac{\partial L_0}{\partial \varphi_4} \\ \dfrac{\partial \varphi_0}{\partial \varphi_1} & \dfrac{\partial \varphi_0}{\partial \varphi_4} \end{bmatrix} \delta q \tag{1.8}$$

Define the Jacobi Matrix as:

$$J = \begin{bmatrix} \dfrac{\partial L_0}{\partial \varphi_1} & \dfrac{\partial L_0}{\partial \varphi_4} \\ \dfrac{\partial \varphi_0}{\partial \varphi_1} & \dfrac{\partial \varphi_0}{\partial \varphi_4} \end{bmatrix} \tag{1.9}$$

Therefore:

$$\delta x = J\delta q \tag{1.10}$$

According to the principle of virtual work:

$$T^T\delta q + (-F)^T\delta x = 0 \tag{1.11}$$

Where $T = [T_1 \quad T_2]$, $F = [F \quad T_p]$. Substitute $\delta x = J\delta q$ into the equation:

$$T = J^T F \tag{1.12}$$

There is another way to get the matrix of *T*. It's easy to find that Jacobi Matrix *J* above is obtained from total differential in polar coordinates. However, it can also be obtained from mapping different velocities of joints.

The velocity of point C:

$$\begin{cases} \dot{x}_C = -l_1\dot{\varphi}_1 sin\varphi_1 - l_2\dot{\varphi}_2 sin\varphi_2 \\ \dot{y}_C = -l_1\dot{\varphi}_1 cos\varphi_1 + l_2\dot{\varphi}_2 cos\varphi_2 \end{cases} \tag{1.13}$$

Differentiate with respect to (2.1):

$$\begin{cases} \dot{x}_B - l_2\dot{\varphi}_2 sin\varphi_2 = \dot{x}_D - l_3\dot{\varphi}_3 sin\varphi_3 \\ \dot{y}_B + l_2\dot{\varphi}_2 cos\varphi_2 = \dot{y}_D + l_3\dot{\varphi}_3 cos\varphi_3 \end{cases} \tag{1.14}$$

Where:

$$\dot{\varphi}_2 = \frac{(\dot{x}_D - \dot{x}_B)cos\varphi_3 + (\dot{y}_D - \dot{y}_B)sin\varphi_3}{l_2 sin(\varphi_3 - \varphi_2)} \tag{1.15}$$

$$\begin{cases} \dot{x}_B = -l_1\dot{\varphi}_1 sin\varphi_1 \\ \dot{y}_B = -l_1\dot{\varphi}_1 cos\varphi_1 \\ \dot{x}_D = -l_4\dot{\varphi}_4 sin\varphi_4 \\ \dot{y}_D = -l_4\dot{\varphi}_4 cos\varphi_4 \end{cases} \tag{1.16}$$

Substitute $\dot{\varphi}_2$ into equation (2.13):

$$\begin{cases} \dot{x}_C = -l_1\dot{\varphi}_1 sin\varphi_1 - l_2\left(\dfrac{(\dot{x}_D - \dot{x}_B)cos\varphi_3 + (\dot{y}_D - \dot{y}_B)sin\varphi_3}{l_2 sin(\varphi_3 - \varphi_2)}\right)sin\varphi_2 \\ \dot{y}_C = -l_1\dot{\varphi}_1 cos\varphi_1 + l_2\left(\dfrac{(\dot{x}_D - \dot{x}_B)cos\varphi_3 + (\dot{y}_D - \dot{y}_B)sin\varphi_3}{l_2 sin(\varphi_3 - \varphi_2)}\right)cos\varphi_2 \end{cases} \tag{1.17}$$

Then calculate the Jacobi Matrix *J:*

$$J = \begin{bmatrix} -\frac{l_1 sin(\varphi_0 - \varphi_3) sin(\varphi_1 - \varphi_2)}{sin(\varphi_2 - \varphi_3)} & -\frac{l_1 cos(\varphi_0 - \varphi_3) sin(\varphi_1 - \varphi_2)}{l_0 sin(\varphi_2 - \varphi_3)} \\ -\frac{l_1 sin(\varphi_0 - \varphi_2) sin(\varphi_3 - \varphi_4)}{sin(\varphi_2 - \varphi_3)} & -\frac{l_1 cos(\varphi_0 - \varphi_2) sin(\varphi_3 - \varphi_4)}{l_0 sin(\varphi_2 - \varphi_3)} \end{bmatrix} \tag{1.18}$$

$$\begin{bmatrix} \dot{x}_C \\ \dot{y}_C \end{bmatrix} = J \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_4 \end{bmatrix} \tag{1.19}$$

As a result, the relationship of the torque on the joints and the virtual force can be displayed as:

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = J^T \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{1.20}$$

Use the rotation matrix *R* to unify the virtual force and polar coordinate direction:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} cos\left(\varphi_0 - \frac{\pi}{2}\right) & -sin\left(\varphi_0 - \frac{\pi}{2}\right) \\ sin\left(\varphi_0 - \frac{\pi}{2}\right) & cos\left(\varphi_0 - \frac{\pi}{2}\right) \end{bmatrix} \begin{bmatrix} F_c \\ F_t \end{bmatrix} \tag{1.21}$$

Where F_t and F_c are the virtual forces along L_0 and perpendicular to L_0, respectively, and can be mapped to the final virtual force using the transformation matrix *M*.

$$\begin{bmatrix} F_t \\ F_c \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L_0} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F \\ T_p \end{bmatrix} \tag{1.22}$$

The final torque of the joints can be obtained from the equations above:

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} -\frac{l_1 sin(\varphi_0 - \varphi_3) sin(\varphi_1 - \varphi_2)}{sin(\varphi_2 - \varphi_3)} & -\frac{l_1 cos(\varphi_0 - \varphi_3) sin(\varphi_1 - \varphi_2)}{l_0 sin(\varphi_2 - \varphi_3)} \\ -\frac{l_1 sin(\varphi_0 - \varphi_2) sin(\varphi_3 - \varphi_4)}{sin(\varphi_2 - \varphi_3)} & -\frac{l_1 cos(\varphi_0 - \varphi_2) sin(\varphi_3 - \varphi_4)}{l_0 sin(\varphi_2 - \varphi_3)} \end{bmatrix} \begin{bmatrix} F \\ T_p \end{bmatrix} \tag{1.23}$$

However, this method is too complex to run on embedded development boards, as it takes several minutes to complete even in MATLAB on a PC.

2.2 Inverted Pendulum Model

2.2.1 System Modeling

After obtaining the virtual leg length L_0 using the above method, we can focus on the robot's upper mechanism, the posture of the leg, and the movement of the drive wheel, while ignoring the changes in the robot's leg length and only considering the posture of the leg, that is, the angle of the line connecting the center of the drive wheel axle and the center of the motor shaft of the two joints of the leg relative to the inertial frame.

2.2.2 Mechanical Analysis

This inverted pendulum model consists of a main body, a support rod, and a drive wheel. The model diagram and parameter definition are displayed in the figure below.

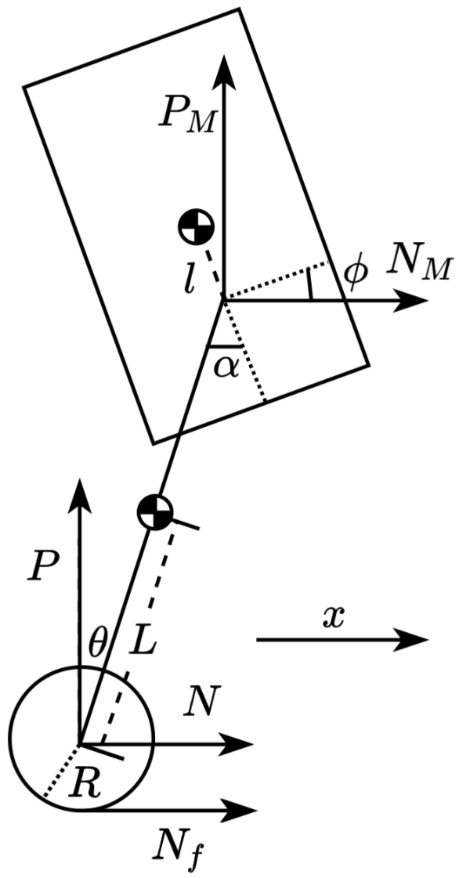

Fig. 5 Inverted pendulum mode

Table 1 Vector Definition Table

Symbol	Meaning	Positive Direction	Unit
θ	Angle between the pendulum rod and the vertical direction	As shown in the figure	rad
x	Displacement of the driving wheel	Direction of the arrow	m

Symbol	Meaning	Positive Direction	Unit
φ	Angle between the body and the horizontal direction	As shown in the figure	rad
T	Output torque of the driving wheel	Same as θ	N·m
T_p	Output torque of the hip joint	Same as α	N·m
N	Horizontal component of the force exerted by the driving wheel on the pendulum rod	Direction of the arrow	N
P	Vertical component of the force exerted by the driving wheel on the pendulum rod	Direction of the arrow	N
N_m	Horizontal component of the force exerted by the pendulum rod on the body	Direction of the arrow	N
P_m	Vertical component of the force exerted by the pendulum rod on the body	Direction of the arrow	N
N_f	Friction force exerted by the ground on the driving wheel	Direction of the arrow	N

Table 2 Scalar Definition Table

Symbol	Meaning	Unit
R	Driving wheel radius	m

L	Distance from pendulum center of mass to driving wheel axle	m
LM	Distance from pendulum center of mass to body rotation axis	m
l	Distance from body center of mass to its rotation axis	m
m_w	Driving wheel rotor mass	kg
m_p	Pendulum mass	kg
M	Body mass	kg
I_w	Driving wheel rotor moment of inertia	kg·m²
I_p	Pendulum moment of inertia about its center of mass	kg·m²
I_M	Body moment of inertia about its center of mass	kg·m²

For drive wheels, the translational and rotational dynamic equations can be written as:

$$m_w\ddot{x} = N_f - N \tag{2.1}$$

$$I_w\frac{\ddot{x}}{R} = T - N_f R \tag{2.2}$$

The acceleration of the drive wheel was obtained after calculation:

$$\ddot{x} = \frac{T - NR}{\frac{I_w}{R} + m_w R} \tag{2.3}$$

For the leg, the horizontal force balance, vertical force balance, and rotational dynamic equation are given by:

$$N - N_M = m_p \frac{\partial^2}{\partial t^2}(x + Lsin\theta) \tag{2.4}$$

$$P - P_M - m_p g = m_p \frac{\partial^2}{\partial t^2}(Lcos\theta) \tag{2.5}$$

$$I_p \ddot{\theta} = (PL + P_M L_M)sin\theta - (NL + N_M L_M)cos\theta - T + T_p \tag{2.6}$$

For the body, the dynamic equations in horizontal direction, vertical direction, and rotational motion are expressed as:

$$N_M = M \frac{\partial^2}{\partial t^2}(x + (L + L_M)sin\theta - lsin\varphi) \tag{2.7}$$

$$P_M - Mg = M \frac{\partial^2}{\partial t^2}\big((L + L_M)cos\theta + lcos\varphi\big) \tag{2.8}$$

$$I_M \ddot{\varphi} = T_p + N_M lcos\varphi + P_M lsin\varphi \tag{2.9}$$

Based on the above equations, the coupled dynamic model of the wheel-leg-body system can be established. This model provides the foundation for subsequent linearization and controller design.

2.2.3 State Space Model

In this model, robot posture is controlled by controlling motor torque. In this model, robot posture is controlled by inputting motor torque. Therefore, the state vector x and control vector u are defined as follows:

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix}, u = \begin{bmatrix} T \\ T_p \end{bmatrix} \tag{2.10}$$

Define the nonlinear model of the system:

$$\dot{x} = f(x, u) \tag{2.11}$$

Because there are many intermediate quantities in the derivation of classical mechanics, such as N_m, P_m, P, N, etc. These intermediate quantities need to be converted into state quantities in the state vector x. However, the LQR controller is only applicable to linear systems. So, the model needs to be transferred to a linear model to get the Jacobi Matrix:

$$A = \frac{\partial f}{\partial x}(\bar{x}, \bar{u}), B = \frac{\partial f}{\partial u}(\bar{x}, \bar{u}) \tag{2.12}$$

In these equations, $(\bar{x}, \bar{u})$ is the system equilibrium point, which is also the solution of $f(\bar{x}, \bar{u}) = 0$. At the system equilibrium point, the robot is in a static, upright position, where $\theta = 0, \varphi = 0$.

$$\bar{x} = \begin{bmatrix} 0 \\ 0 \\ x \\ 0 \\ 0 \\ 0 \end{bmatrix}, \bar{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.13}$$

Performing Jacobian linearization near the equilibrium point yields (Complex expressions are represented by A and B):

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ A_1 & 0 & 0 & 0 & A_2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ A_3 & 0 & 0 & 0 & A_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ A_5 & 0 & 0 & 0 & A_6 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ B_1 & B_2 \\ 0 & 0 \\ B_3 & B_4 \\ 0 & 0 \\ B_5 & B_6 \end{bmatrix} u \tag{2.14}$$

Taking the output matrix as the identity matrix:

$$y = I_6 x$$

I_6 is a six-dimensional matrix, where the quantities can be directly measured or obtained by sensors.

2.3 Controller Design

Maintaining the machine's balance requires LQR, while controlling the speed of the drive wheels requires PID.

2.3.1 LQR

The core objective of LQR is to design a state feedback control law for this linear system to stabilize the system while balancing state error and control energy:

$$u = -Kx = -\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \tag{3.1}$$

To obtain the gain value K, a standard needs to be defined. Define the cost function J:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{3.2}$$

Where $x^T Q x$ is the punishment state deviation and $u^T R u$ is the punishment for excessive input.

To minimize the cost function, the input should be satisfied:

$$K = R^{-1} B^T P \tag{3.3}$$

In this step, the optimal feedback gain *K* is found by finding the optimal matrix *P*.

P satisfies the algebraic Riccati equation:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \tag{3.4}$$

Its origin lies in the Hamilton-Jacobi-Bellman equations of optimal control theory.

For linear systems and quadratic cost functions, it can ultimately be simplified to this matrix equation. In actual calculations, MATLAB will solve it directly.

The above method can achieve system stability near the linearized equilibrium point. To enable the robot to track the trajectory, a reference input needs to be added to the system input:

$$u = K(x_d - x) \tag{3.5}$$

The reference input x_d is composed of the robot's expected position *x*.

$$x_d = \begin{bmatrix} 0 \\ 0 \\ x \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

To account for different robot leg lengths, the system model is linearized every 10mm within the leg length range, and its feedback gain matrix *K* is solved. A polynomial equation K_{ij} is fitted to each element of the matrix as a function of leg length $L_0 = L + L_M$ to obtain:

$$K_{ij}(L_0) = p_{0ij} + p_{1ij} L_0 + p_{2ij} L_0^2 + p_{3ij} L_0^3 \tag{3.6}$$

To sum up, the longitudinal motion control law of this robot is:

$$u = K(L_0)(x_d - x) \tag{3.7}$$

2.3.2 PID

In order to control the robot's movement using a remote control, a PID controller is also needed to control the drive wheels. Based on the speed tracking and attitude adjustment requirements of robots in actual operation, a proportional-integral-derivative (PID) controller can be introduced to assist in the system adjustment in addition to state feedback control.

Let the system reference input be $r(t)$, and the actual output be $y(t)$, then the control error is defined as:

$$e(t) = r(t) - y(t) \tag{3.8}$$

The continuous-time expression for a PID controller is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{3.9}$$

K_p is the proportional coefficient, used to improve the system response speed; K_i is the integral coefficient, used to reduce steady-state error; K_d is the derivative coefficient, used to suppress error variation trends and improve overshoot.

In robot speed control, the error between the reference speed v_d and the actual speed v can be defined as:

$$e_v(t) = v_d(t) - v(t) \tag{3.10}$$

The speed loop PID controller can then be written as:

$$u_v(t) = K_{pv} e_v(t) + K_{iv} \int_0^t e_v(\tau)d\tau + K_{dv} \frac{de_v(t)}{dt} \tag{3.11}$$

Furthermore, the PID controller can also be written in discrete form to facilitate implementation by a digital controller. Let the sampling period be T_s, and the control error at time k be:

$$e(k) = r(k) - y(k) \tag{3.12}$$

Therefore, the discrete PID control law can then be expressed as:

$$u(k) = K_p e(k) + K_i \sum_{i=0}^{k} e(i) T_s + K_d \frac{e(k) - e(k-1)}{T_s} \tag{3.13}$$

In practical implementations, to avoid the derivative term being overly sensitive to noise, it is usually filtered, or only PI control is used. For the outer loop of the wheeled robot's speed control, the integral term can be used to reduce steady-state speed error, while the proportional term provides the main adjustment capability. Therefore, in many cases, speed control can be simplified to:

$$u_v(t) = K_{pv} e_v(t) + K_{iv} \int_0^t e_v(\tau)d\tau \tag{3.14}$$

To sum up, PID controllers can be used to construct the outer loop structure of robot motion control and work in conjunction with the lower-level balance controller. Its final control form can be expressed as:

$$u = K_p e + K_i \int e dt + K_d \dot{e} \tag{3.15}$$

By properly tuning parameters K_p, K_d and K_i, the system can achieve good tracking performance and anti-interference capability while ensuring stability.

2.4 Simulation

The feasibility of the algorithm described above is verified by building a simple model using Simscape Multibody.

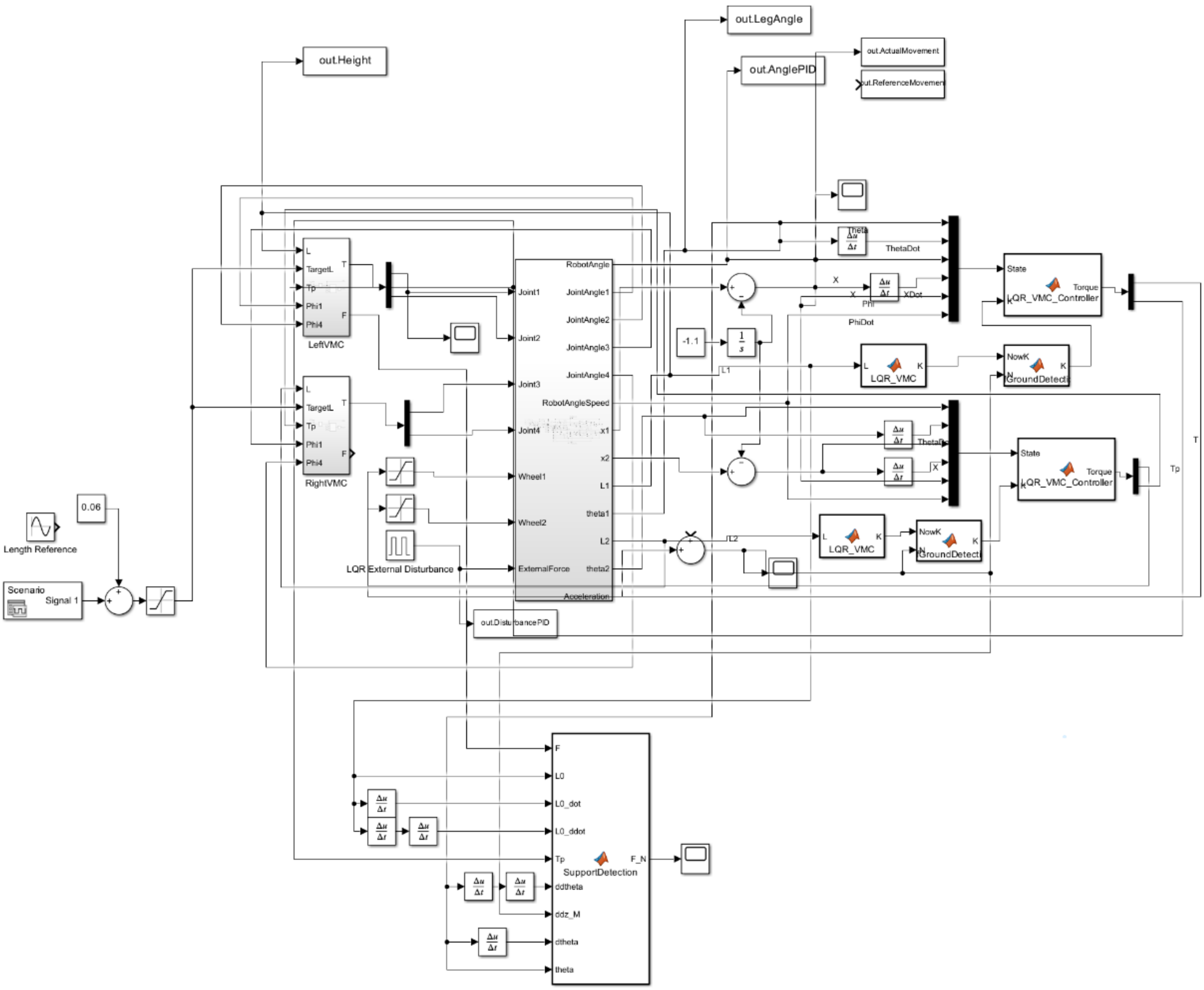

Fig. 6 Simulation of LQR Control with VMC

In the LQR controller design, the weighting matrices Q and R were selected as:

$$Q = \begin{pmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, R = \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix} \tag{4.1}$$

The matrix Q determines the penalty imposed on the state error, while R determines the penalty imposed on the control input. By tuning these two matrices, the closed-loop dynamic response can be adjusted to satisfy the stability and motion requirements of the robot.

In this work, the state vector is defined as

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \tag{4.2}$$

where θ represents the pendulum angle, $x$represents the wheel displacement, and ϕ represents the body pitch angle. A relatively large weight was assigned to the body angle ϕ, namely 5000, because maintaining body posture is one of the most critical objectives for the wheel-legged robot. If the body pitch deviates excessively, the system may quickly lose balance. Therefore, the controller is required to suppress the body angle error with high priority.

The displacement x was also assigned a relatively large weight of 500, indicating that position deviation should be effectively constrained during locomotion. Compared with displacement, the velocity $\dot{x}$ was given a smaller weight of 100, since moderate velocity variation is acceptable during dynamic adjustment. The pendulum angle θ and angular velocity $\dot{\theta}$ were both assigned weights of 10, which means that these states are regulated, but their priority is lower than that of the body pitch angle and displacement. In addition, the body angular velocity $\dot{\phi}$ was assigned to a weight of 1, which mainly provides auxiliary damping and prevents excessive oscillation.

For the input weighting matrix R, the two control inputs are the wheel torque $T$and the hip torque T_p. The corresponding weights were chosen as $1$and 0.25, respectively. This indicates that the wheel torque is penalized more strongly, while the hip torque is allowed to vary more actively in order to improve posture regulation. Such a choice reflects the practical consideration that the hip torque plays a more direct role in body stabilization, whereas excessive wheel torque may lead to aggressive translational motion or unnecessary energy consumption.

Overall, the selected $Q$and Rmatrices were determined empirically through repeated simulation tuning. The adopted values provide a compromise between balance stability, position regulation, and control smoothness, enabling the robot to maintain stable motion under the designed operating conditions.

Substituting the value of Q and R, we can solve the equation to obtain the gain matrix K:

$$K = \begin{bmatrix} -0.5337 & -0.0614 & -0.3144 & -0.2561 & 2.7902 & 0.0566 \\ -0.0024 & 0.0004 & 0.0134 & 0.0047 & 6.6656 & 0.1282 \end{bmatrix} \tag{4.3}$$

In the simulation, the robot falls from a certain height and moves forward in a straight line.

The virtual leg length L_0, virtual leg inclination ϕ_0, angles between the virtual link and the horizontal direction ϕ_1 and ϕ_4, robot body pitch angle and leg posture angle ϕ(blue) and θ(red) are displayed in following:

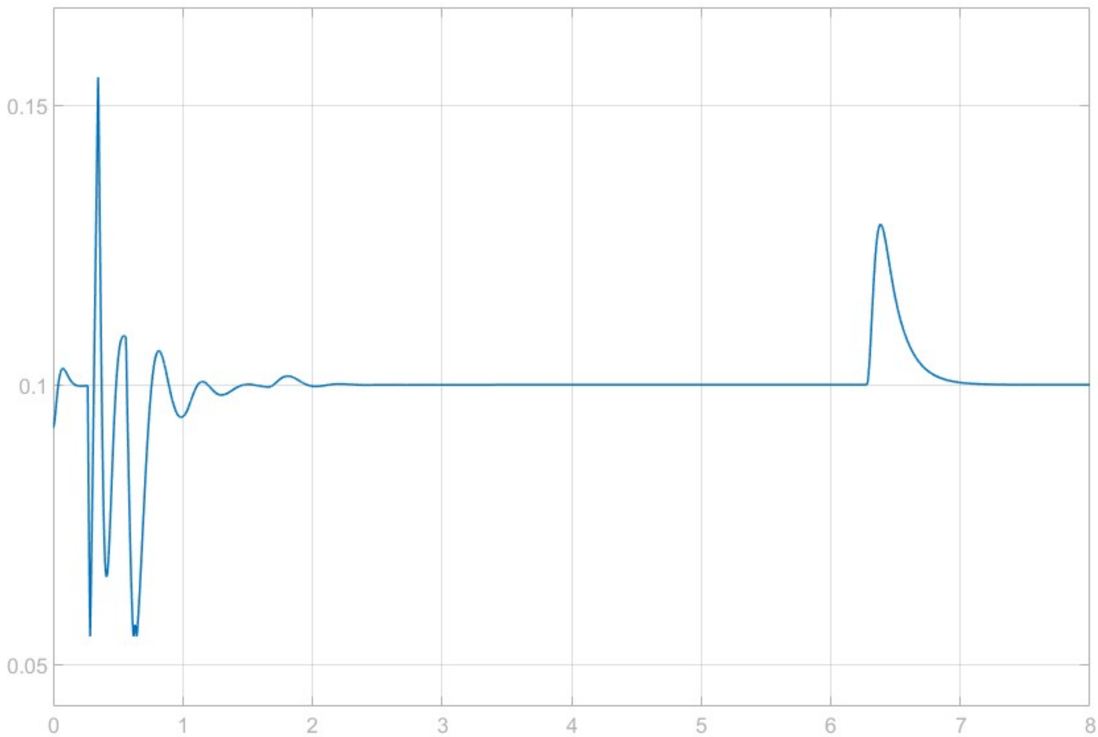

Fig. 7 L_0 changes over time

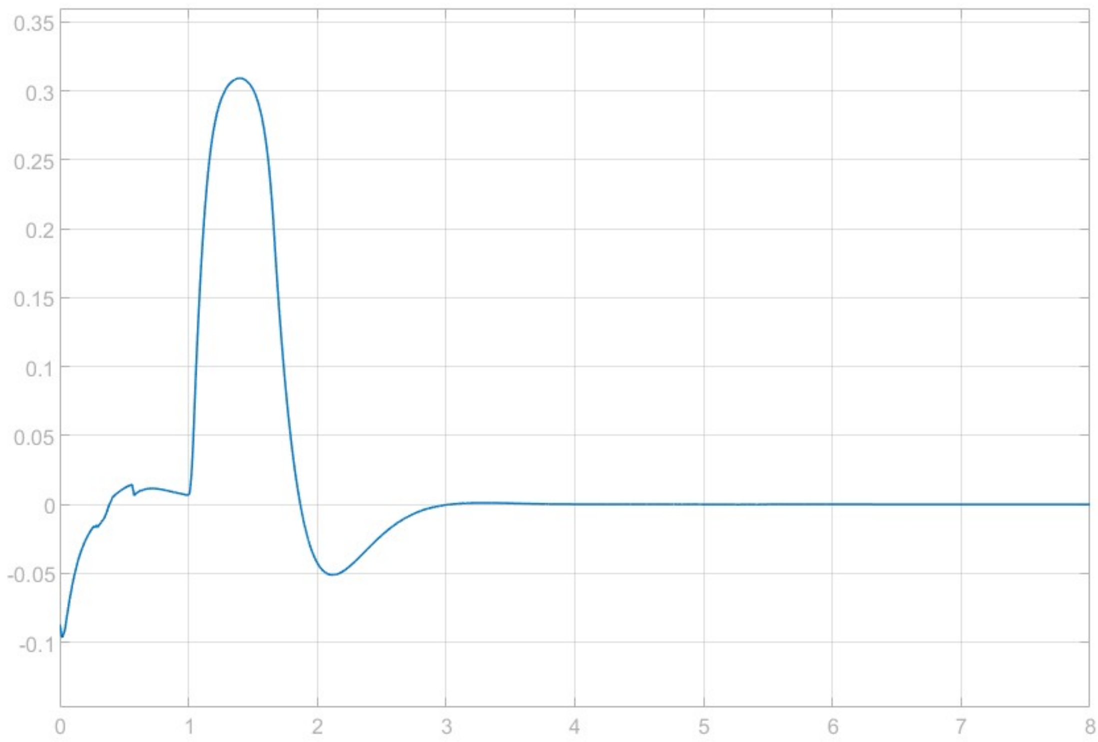

Fig. 8 ϕ_0 changes over time

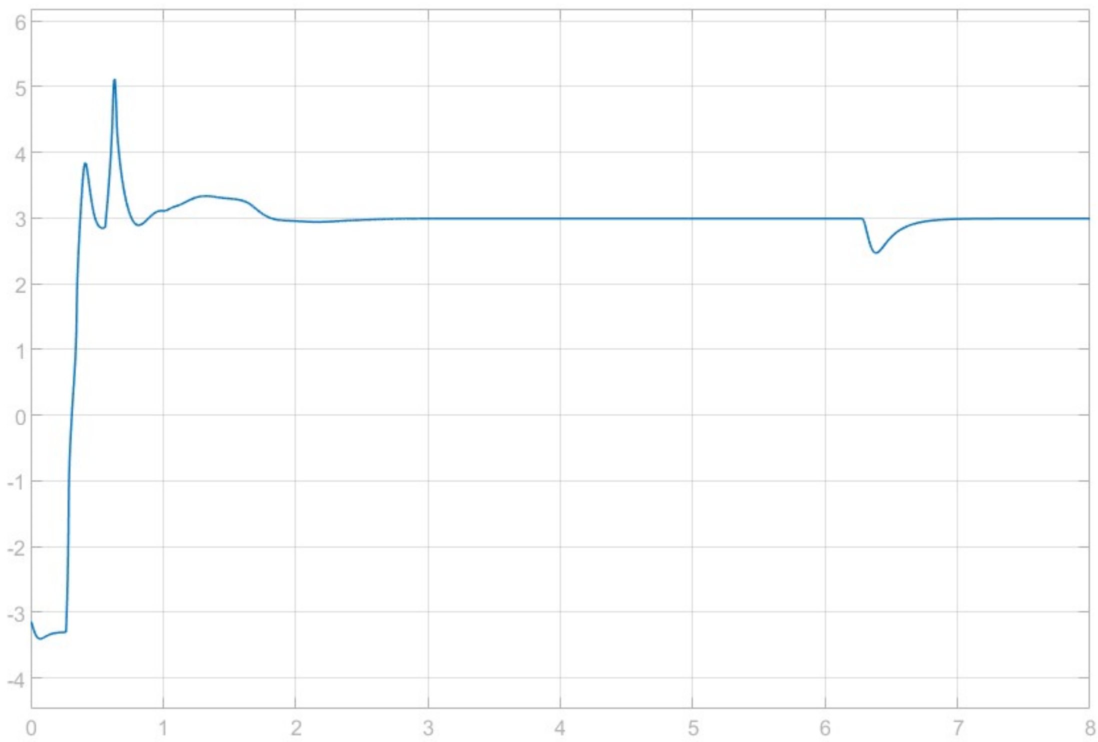

Fig. 9 ϕ_1 changes over time

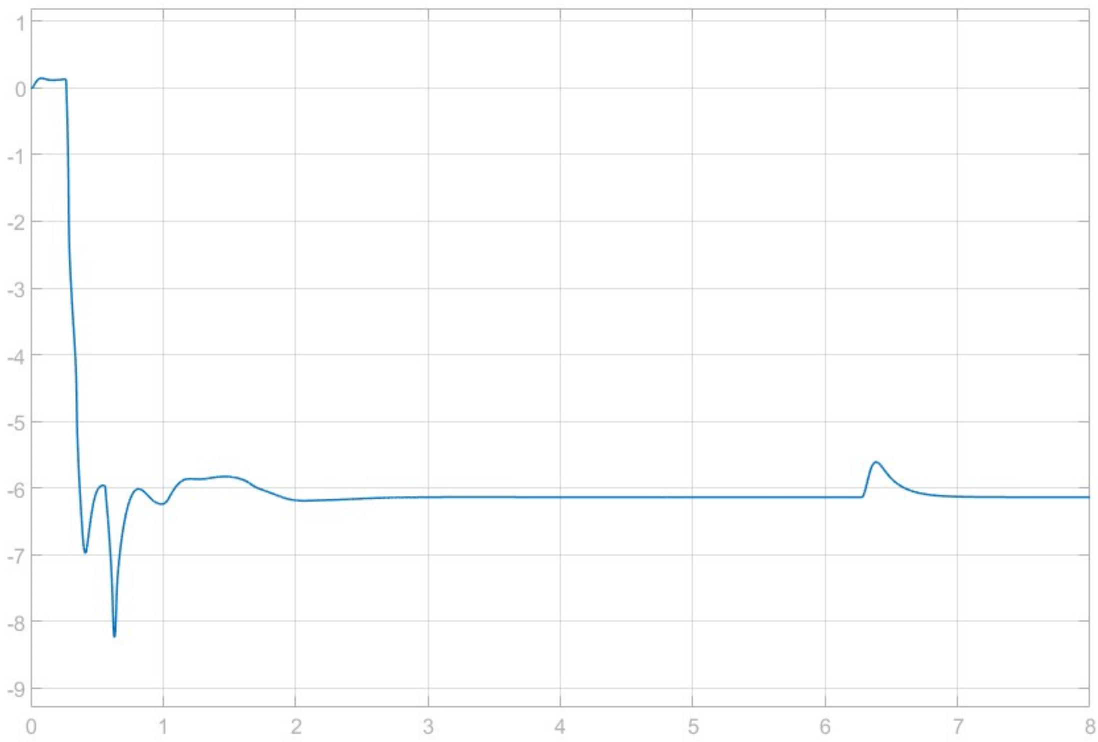

Fig. 10 ϕ_4 changes over time

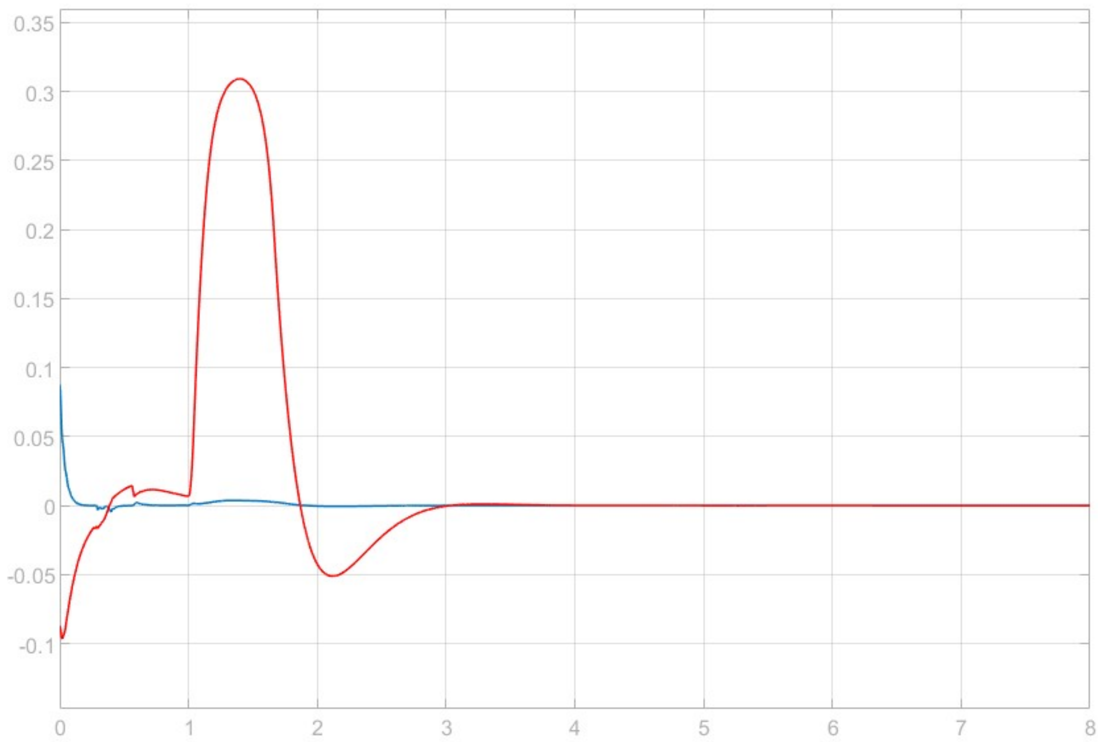

Fig. 11 ϕ(blue) and θ(red) change over time

As illustrated in Fig. 7, the virtual leg length L_0experiences an obvious transient fluctuation immediately after landing due to the impact with the ground, and then gradually converges to a steady value around 0.10 m. A small disturbance can also be observed at about $t = 6.3$ s, after which the leg length quickly returns to equilibrium, indicating good recovery performance. Fig. 8 shows the variation of the virtual leg inclination ϕ_0. A relatively large transient response appears at the initial stage, but the angle is subsequently stabilized around a constant value, which demonstrates that the leg posture can be effectively regulated during locomotion. The responses of the two joint angles ϕ_1 and $\phi_4$are presented in Fig. 9 and Fig. 10. Both angles exhibit short-term oscillations at the beginning and then converge smoothly to their respective steady-state values, which indicates that the joint motions remain bounded and coordinated.

The body pitch angle $\phi$and the leg posture angle θ are plotted together on Fig. 11. It can be seen that both variables undergo transient oscillations during the falling and landing process. In particular, θ shows a relatively larger overshoot because it is directly affected by the initial impact and rapid posture adjustment. However, both ϕ and θ converge to values close to zero after the transient response, which means that the robot is able to recover its balance and maintain a stable forward motion. Overall, the simulation results verify that the proposed controller can effectively suppress landing disturbances, regulate the virtual leg configuration, and ensure stable locomotion of the robot after impact.

3. Conclusion

This project studied the modeling and control of a biped serial-leg robot with wheel actuation. A simplified control-oriented framework was developed to address the strong coupling and balance problem of the wheel-legged system. The leg mechanism was first modeled through kinematic analysis, and the virtual leg force was mapped to the joint torques by using the Jacobian matrix and the principle of virtual work. Then, the upper structure of the robot was simplified as an inverted pendulum system, and the dynamic equations of the drive wheel, leg, and body were derived through classical mechanics. After converting the nonlinear model into a linear state-space form around the upright equilibrium point, an LQR controller was designed for balance stabilization. In addition, a PID controller was introduced as the outer speed loop to support motion command tracking.

The simulation results verify the effectiveness of the proposed method. The robot is able to withstand the landing disturbance after falling from a certain height, and the virtual leg length, virtual leg inclination, joint angles, body pitch angle, and leg posture angle all converge to stable values after transient oscillation. These results indicate that the controller can suppress external disturbance, coordinate the leg motion, and maintain stable forward locomotion. The fitted gain matrix under different virtual leg lengths also shows that gain scheduling is a feasible way to improve controller adaptability when the leg configuration changes.

Although the proposed model is simplified and some assumptions are introduced to reduce complexity, the overall control framework is effective and suitable for real-time implementation. Therefore, this work provides a feasible solution for the balance control of serial-leg wheel-legged robots. Future work should focus on hardware implementation, sensor fusion, remote-control experiments, and controller optimization under more complex terrain and stronger disturbances, so that the proposed approach can be further validated on a physical prototype.

References

[1]. Marko Bjelonic *et al.*, "Keep Rollin'—Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots," *IEEE robotics and automation letters*, vol. 4, no. 2, pp. 2116–2123, Apr. 2019, doi: https://doi.org/10.1109/lra.2019.2899750.

[2]. M. Bjelonic, P. K. Sankar, C. D. Bellicoso, H. Vallery, and M. Hutter, "Rolling in the Deep – Hybrid Locomotion for Wheeled-Legged Robots Using Online Trajectory Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3626–3633, Apr. 2020, doi: https://doi.org/10.1109/lra.2020.2979661.

[3]. V. Klemm *et al.*, "Ascento: A Two-Wheeled Jumping Robot," *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, doi: https://doi.org/10.1109/icra.2019.8793792.

[4]. V. Klemm *et al.*, "LQR-Assisted Whole-Body Control of a Wheeled Bipedal Robot With Kinematic Loops," *IEEE robotics & automation letters*, vol. 5, no. 2, pp. 3745–3752, Apr. 2020, doi: https://doi.org/10.1109/lra.2020.2979625.

[5]. Y. CHEN, H. WANG, and L. ZHANG, "Control of Wheel-legged Balancing Robot[J]," *INFORMATION AND CONTROL*, vol. 52, no. 5, pp. 648–659, Mar. 2023, doi: https://doi.org/10.13976/j.cnki.xk.2023.2533.

[6]. S. Wang *et al.*, "Balance Control of a Novel Wheel-legged Robot: Design and Experiments," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6782–6788, May 2021, doi: https://doi.org/10.1109/icra48506.2021.9561579.

[7]. B. Zi, J. Cao, and Z. Zhu, "Dynamic Simulation of Hybrid-Driven Planar Five-Bar Parallel Mechanism Based on SimMechanics and Tracking Control," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, p. 37, Sep. 2011, doi: https://doi.org/10.5772/45683.

[8]. X. Liu *et al.*, "Development of Wheel-Legged Biped Robots: A Review," *Journal of Bionic Engineering*, vol. 21, no. 2, pp. 607–634, Feb. 2024, doi: https://doi.org/10.1007/s42235-023-00468-1.

[9]. X. Feng, S. Liu, Q. Yuan, J. Xiao, and D. Zhao, "Research on wheel-legged robot based on LQR and ADRC," *Scientific Reports*, vol. 13, no. 1, p. 15122, Sep. 2023, doi: https://doi.org/10.1038/s41598-023-41462-1.

[10]. A. Zhang, R. Zhou, T. Zhang, J. Zheng, and S. Chen, "Balance Control Method for Bipedal Wheel-Legged Robots Based on Friction Feedforward Linear Quadratic Regulator," *Sensors*, vol. 25, no. 4, pp. 1056–1056, Feb. 2025, doi: https://doi.org/10.3390/s25041056.

[11]. Y. Xu, Z. Wang, and C. Lu, "Design and Control of a Wheeled Bipedal Robot Based on Hybrid Linear Quadratic Regulator and Proportional-Derivative Control," *Sensors*, vol. 25, no. 17, p. 5398, Sep. 2025, doi: https://doi.org/10.3390/s25175398.

[12]. L. Wang, Z. Zhang, Z. Shao, and X. Tang, "Analysis and optimization of a novel planar 5R parallel mechanism with variable actuation modes," *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 178–190, Oct. 2018, doi: https://doi.org/10.1016/j.rcim.2018.09.010.

[13]. H. Zhao, L. Yu, S. Qin, G. Jin, and Y. Chen, "Design and Control of a Bio-Inspired Wheeled Bipedal Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 4, pp. 2461–2472, Aug. 2025, doi: https://doi.org/10.1109/tmech.2024.3449397.

[14]. M. D. Ratolikar and P. Kumar R., "Optimized design of 5R planar parallel mechanism aimed at gait-cycle of quadruped robots," *Journal of Vibroengineering*, vol. 24, no. 1, pp. 104–115, Nov. 2021, doi: https://doi.org/10.21595/jve.2021.22131.

[15]. G. Xin, B. Jin, C. Liu, and M. Jiang, "A Unified Control Framework for Self-Balancing Robots: Addressing Model Variations in Wheel-Legged Platforms and Human-Carrying Wheelchairs," *Sensors*, vol. 25, no. 23, p. 7144, Nov. 2025, doi: https://doi.org/10.3390/s25237144.

[16]. T. Wang *et al.*, "Head Stabilization for Wheeled Bipedal Robots via Force-Estimation-Based Admittance Control," *arXiv.org*, 2025. https://arxiv.org/abs/2511.18712 (accessed Apr. 12, 2026).

[17]. C. Wen *et al.*, "Whole-Body Control With Terrain Estimation of A 6-DoF Wheeled Bipedal Robot," *arXiv.org*, 2025. https://arxiv.org/abs/2511.06397 (accessed Apr. 12, 2026).

[18]. D. Liu, F. Yang, X. Liao, and X. Lyu, "DIABLO: A 6-DoF Wheeled Bipedal Robot Composed Entirely of Direct-Drive Joints," *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3605–3612, Oct. 2024, doi: https://doi.org/10.1109/iros58592.2024.10801943.

[19]. J. Gao, R. Fan, H. Yang, H. Pang, and H. Tian, "Research on Motion Control Method of Wheel-Legged Robot in Unstructured Terrain Based on Improved Central Pattern Generator (CPG) and Biological Reflex Mechanism," *Applied Sciences*, vol. 15, no. 15, pp. 8715–8715, Aug. 2025, doi: https://doi.org/10.3390/app15158715.

[20]. B. Wang, Y. Xin, C. Chen, Z. Song, B. Sun, and T. Guo, "Whole-Body Control with Uneven Terrain Adaptability Strategy for Wheeled-Bipedal Robots," *Electronics*, vol. 14, no. 1, p. 198, Jan. 2025, doi: https://doi.org/10.3390/electronics14010198.

[21]. CHEN Yang, WANG Hongxi, and ZHANG Lanyong, "Control of Wheel-legged Balancing Robot," *Ebsco.com*, vol. 52, no. 5, p. 648, Sep. 2023, doi: https://doi.org/10.13976/j.cnki.xk.2023.2533.