Adaptive and Robust Wheel-Legged Biped Robot for Semi-Structured Community Tasks

Bowen Lan, Hailong Huang, *Senior Member, IEEE*

***Abstract*—This paper presents the development and optimization of a wheel-legged biped robot designed for community inspection tasks. The robot combines the efficiency of wheeled robots with the adaptability of legged systems, making it suitable for navigating diverse terrains in semi-structured environments. The main innovations of this research include the development of an integrated control system that combines balance control, state estimation, and terrain adaptation using multi-sensor fusion. The robot's mechanical structure is designed to withstand jumps and falls, while its electronic hardware and software architecture ensure real-time control and robust performance. Experiments demonstrate the robot's capabilities including tracking velocity as high as 2 m/s, resisting severe disturbance or slippage, and traversing steps over 10 cm automatically. The results show that the proposed control methods and hardware design effectively ensure the robot's robustness and adaptability, validating its potential for practical community inspection applications. This work contributes to the growing field of autonomous robots in public service and provides a foundation for future research and development.**

***Index Terms*—wheel-legged biped robot, linear quadratic regulator, multi-sensor fusion, feedforward control**

I. Introduction

In recent years, the deployment of autonomous robots for community-based tasks has gained significant attention due to their potential to improve efficiency, safety, and accessibility in public and private spaces [1]–[4]. Among these, wheel-legged biped robots stand out as a promising platform for applications such as community inspection and monitoring. Combining their stability and maneuverability with the adaptability and versatility of legged systems, these robots are uniquely suited for navigating the diverse and often unpredictable environments found in community settings [5], [6].

Community inspection tasks, such as monitoring infrastructure and detecting hazards, often require robots to operate in environments that are highly dynamic and contain a mix of structured and unstructured terrains [1]. For example, a robot may need to traverse smooth sidewalks, ascend ramps, cross uneven pavements, or navigate narrow corridors within buildings. Traditional wheeled robots, while efficient on flat surfaces, struggle with uneven terrains or obstacles. Conversely, legged robots excel in challenging terrains but often suffer from lower energy efficiency and slower speeds. Wheel-legged biped robots offer an optimal middle ground, leveraging their compact and energy-efficient design while maintaining the ability to handle moderate terrain variations.

The integration of wheel-legged biped robots into community inspection workflows presents several key advantages. First, their compact form factor and high agility make them well-suited for navigating crowded or confined spaces, such as office buildings, residential complexes, or urban streets. Second, their ability to balance dynamically on two wheels enables them to quickly change direction or adapt to obstacles, ensuring efficient and safe operation in environments shared with pedestrians. Third, these robots can be equipped with advanced sensing and perception systems, such as LiDAR, cameras, and environmental sensors, to autonomously identify and respond to potential issues, such as structural damage, fire hazards, or environmental anomalies.

Beyond their functional capabilities, deploying wheel-legged biped robots for community inspection also holds societal significance. These robots can reduce the reliance on human inspectors for repetitive or hazardous tasks, minimizing risks and freeing up human resources for more complex decision-making roles. Additionally, their presence in communities can enhance public trust in technology, as they provide visible, tangible benefits in maintaining safety, cleanliness, and functionality within shared spaces.

In this paper, we focus on the development and optimization of a wheel-legged biped robot system specifically tailored for community inspection tasks. By addressing the challenges of perception and locomotion in semi-structured environments, we aim to demonstrate the feasibility and practicality of these robots as a reliable solution for improving community infrastructure and safety. Through this work, we contribute to the growing body of research on autonomous robots in public service, paving the way for their broader adoption in real-world applications.

We address the stability and terrain traversal capability of a wheel-legged biped robot equipped with two-degree-of-freedom legs in semi-structured urban environments by significantly enhancing its perception and control algorithms. The main contributions are as follows:

1) Building upon the conventional linear control method, we design a Kalman filter to fuse IMU and wheel odometry measurements. By detecting and isolating abnormal data induced by wheel slippage through measurement residual analysis, we substantially improve the accuracy of velocity estimation as well as the robot's control stability in complex terrains.
2) We propose a feedforward control method based on the wheel–ground contact angle, addressing the limitations

This work was supported by the Research Centre for Unmanned Autonomous Systems (RCUAS), the Hong Kong Polytechnic University. The authors are with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hung Hom, Hong Kong. (e-mail: bowen.lan@connect.polyu.hk; hailong.huang@polyu.edu.hk)

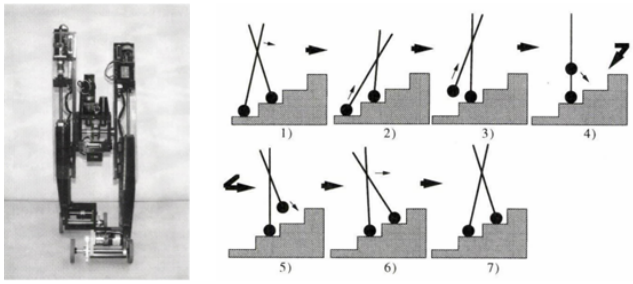
of existing approaches and enhancing control precision on uneven terrain.
3) The surrounding terrain is mapped using LiDAR and depth cameras, for which we design a 2.5D trajectory prediction and slope estimation algorithm.
4) We introduce a step detection method based on the predicted trajectory, along with a corresponding autonomous jumping algorithm, which significantly enhances the robot's capability to traverse challenging environments.
5) We engineer a complete robotic hardware and software system, aiming to facilitate and accelerate the deployment of this robotic configuration in real-world applications.

The remainder of this paper is organized as follows. Section II presents the related work. Section III details the modelling of the robot and the control methods. Section IV shows the experiment results. Section V summarizes this paper.

II. Related Work

A. Development History of Wheel-legged Biped Robot

In 1998, Professor Osamu Matsumoto and his team in Japan developed a wheel-legged biped robot, as shown in Fig. 1a [7]. This robot features ball screw structures on its legs for extension and retraction, and wheels at the ends of the legs driven by DC motors. The height of the robot ranges from 0.575 to 0.74 meters, with a weight of 10 kg and a load capacity of 2 kg. The robot is capable of ascending steps by swinging its legs, as illustrated in Fig. 1b.



(a) Biped type robot (b) Climbing stairs

Fig. 1: Biped type robot and its stair climbing process [7].

In 2015, the DRC-Hubo robot from the Korea Advanced Institute of Science and Technology (KAIST) participated in and won the DARPA Robotics Challenge. Its primary configuration is a bipedal humanoid robot, but it features two drive wheels added at the knee joints and two caster wheels at the toes. This design allows the robot to transition into a kneeling position to enter a wheeled mode of movement, as shown in Fig. 2. This configuration significantly enhances its speed on flat surfaces while maintaining its obstacle-crossing capabilities [8].

In 2017, Boston Dynamics unveiled the wheel-legged robot Handle, as shown in Fig. 3a This robot stands 1.9 meters tall, weighs approximately 105 kg, and has a maximum load capacity of 45 kg. It can achieve a top speed of 24 m/s. The wheels are driven by motors, while the upper limbs

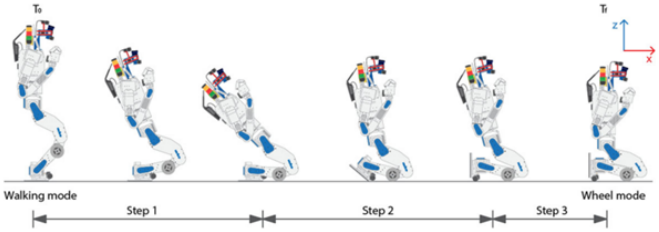
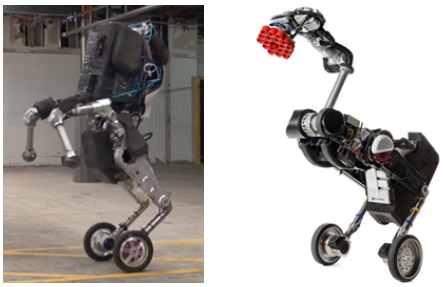


Fig. 2: DRC-Hubo transitions from standing to kneeling [8].

are hydraulically powered. In a video released by Boston Dynamics [9], Handle demonstrated its remarkable maneuverability, including high-speed movement, descending stairs, limb stabilization, jumping, and off-road capabilities. In 2019, Boston Dynamics introduced the second generation of the Handle robot, designed for warehouse handling tasks. This version features an additional robotic arm on its upper body for picking up goods, as shown in Fig. 3b.



(a) Handle ver. 1 (b) Handle ver. 2

Fig. 3: The Handle robot [9].

In 2019, researchers at ETH Zurich in Switzerland developed a wheel-legged biped robot named Ascento [10], as shown in Fig. 4. This robot underwent a topology optimization design and utilized lightweight nylon 3D printing, resulting in a total weight of only 10.4 kg. The leg linkages of Ascento are specially designed so that the hip joint motors can drive the wheels to move vertically along the body, achieving decoupled control of balance and leg length. This design reduces the complexity of the control system. The robot's height can be adjusted between 31 cm and 66 cm, and it is capable of performing actions such as jumping onto steps [11].

Fig. 4: The Ascento robot [10].

In 2021, Tencent's Robotics X Lab introduced a wheel-legged biped robot named Ollie [12], as depicted in Fig. 5. Ollie employs an innovative pentagonal linkage as its leg structure, with each leg using two joint motors to control the position of the wheels. Compared to traditional wheel-legged designs, this configuration adds an extra degree of

freedom, providing more flexible movement capabilities. In a subsequent video released by the lab [13], the Ollie robot demonstrated its exceptional maneuverability, including the ability to traverse inclined terrains smoothly and perform actions such as jumping and backflips.

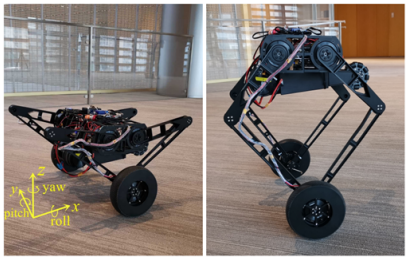


Fig. 5: The Ollie robot [12].

B. Control Methods

When examining the control algorithms for wheel-legged biped robots, it is evident that current research has made significant progress across various control strategies, effectively enhancing the robots' balance and maneuverability.

Take the Ascento robot as an example, its control system employs a Linear Quadratic Regulator (LQR) control strategy. This approach introduces a set of state variables to describe the robot's dynamic configuration, including tilt angle, linear velocity, and angular velocity, enabling precise state feedback control. Additionally, Ascento integrates a Kalman filter for state estimation, further improving the system's accuracy and robustness. This combination allows the robot to effectively perform various actions, such as autonomous jumping and balance recovery, even in the presence of external disturbances.

On the other hand, the Ollie robot utilizes a linear feedback controller and an Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) method. This approach enhances system stability under uncertainties and disturbances by applying principles of energy shaping and damping injection. The application of IDA-PBC enables Ollie to maintain effective control despite nonlinear characteristics. Furthermore, Ollie is equipped with an observer designed to estimate system states, which further enhances the robot's adaptability.

In addition, various control methods have emerged that integrate techniques such as fuzzy control [14], model predictive control [15], and reinforcement learning [16] algorithms with LQR. These approaches have further improved the control robustness of wheel-legged biped robots to varying degrees. However, due to the complexity of these methods, they have yet to be widely adopted in practical applications.

In summary, the existing control methods for wheel-legged biped robots have effectively achieved balance control. These algorithms, by integrating advanced control theories and state estimation techniques, ensure that robots maintain stability and robustness in dynamic environments. Although there is still room for improvement, the current control strategies have reached a practical level, adequately meeting the balance control demands in real-world applications.

However, there are still some critical research gaps in current studies. Firstly, most existing control strategies primarily rely on proprioceptive sensors, such as Inertial Measurement Units (IMUs) and encoders. While these devices provide information about the robot's own motion state, they lack integration with external terrain data. This limitation may lead to insufficient robustness in motion control when facing unstructured terrains, thereby restricting the robot's adaptability in complex environments. Secondly, the failure to fully utilize external terrain data means that current control algorithms may not achieve optimal path planning and obstacle avoidance on dynamic and irregular terrains. This not only affects the robot's maneuverability but also limits its practicality in diverse application scenarios. Therefore, developing control algorithms that can integrate proprioceptive and external terrain information is crucial for enhancing the performance of wheel-legged biped robots.

The primary objective of this research is to develop a set of locomotion control algorithms suitable for wheel-legged biped robots. By integrating multi-source sensory information, the aim is to improve the robot's motion robustness in unstructured terrains. This research will contribute to expanding the robot's application capabilities in complex environments, enhancing task execution efficiency and safety, and providing new directions and possibilities for the future development of robotic technology.

III. Methodology

A. Modeling

Due to the excessive complexity of performing a complete 3D modeling, we assume that the left and right legs move synchronously and conduct the modeling solely within a 2D plane when considering balance control.

Based on the structure illustrated in Fig. 6, we begin by simplifying the leg structure, reducing the five-bar linkage mechanism to a single variable-length straight pole that actuated by a revolute joint at ϕ_0 and a prismatic joint at L_0, which is hereafter referred to as the "virtual leg". Referring to the conclusions presented in [17], we can derive the following equation through static force analysis:

$$T_1 = -\frac{l_1 \sin(\phi_4 - \phi_3)\left(T_p \cos(\phi_0 - \phi_2) + F L_0 \sin(\phi_0 - \phi_2)\right)}{L_0 \sin(\phi_3 - \phi_2)} \tag{1}$$

$$T_2 = -\frac{l_4 \sin(\phi_2 - \phi_1)\left(T_p \cos(\phi_0 - \phi_3) + F L_0 \sin(\phi_0 - \phi_3)\right)}{L_0 \sin(\phi_3 - \phi_2)} \tag{2}$$

This equation allows the torque T_p generated by the virtual leg at angle ϕ_0, as well as the thrust force F along the virtual leg at L_0, to be translated into the joint motor torques T_1 and T_2 acting at positions ϕ_1 and ϕ_4, respectively.

The simplified structure of the robot is illustrated in Fig. 7, comprising three main components: the body, the wheels, and a virtual leg. This configuration can be interpreted as the addition of a rigid body to the upper section of the classical wheeled inverted pendulum (WLP). Following the

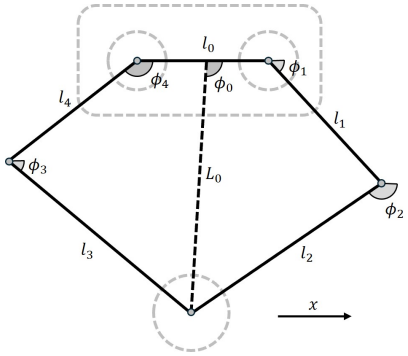


Fig. 6: The pentagonal linkage of the legs.

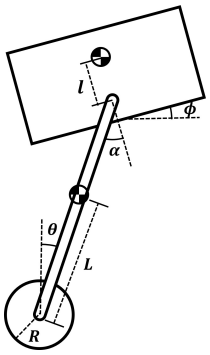


Fig. 7: The simplified physical model.

methodology outlined in [17], we assume that the length of the virtual leg remains constant and define the state variables as

$$\boldsymbol{x} = \begin{bmatrix} \theta & \dot{\theta} & x_b & \dot{x_b} & \phi & \dot{\phi} \end{bmatrix}^T, \tag{3}$$

where θ represents the deviation angle of the virtual leg, x_b is the displacement of the body in the forward-backward direction, and ϕ is the pitch angle of the body. By employing classical mechanics analysis stated in [17] for system modeling, we derive the system's nonlinear model as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{4}$$

where $\boldsymbol{u} = \begin{bmatrix} T_p & T \end{bmatrix}^T$, in which T_p is the torque generated by the virtual leg, and T is the torque produced by the wheels.

Next, we linearize the system around the equilibrium point corresponding to the upright posture. This process yields the system's state-space equation

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}. \tag{5}$$

By discretizing this continuous-time system, we obtain the discrete-time state-space equation

$$\boldsymbol{x}_{k+1} = \boldsymbol{G}\boldsymbol{x}_k + \boldsymbol{H}\boldsymbol{u}_k. \tag{6}$$

B. Balance Control

To achieve balanced control of the system using LQR, we need to solve the discrete-time Riccati equation:

$$\boldsymbol{G}^T\boldsymbol{P}\boldsymbol{G} - \boldsymbol{P} + \boldsymbol{Q} - \boldsymbol{G}^T\boldsymbol{P}\boldsymbol{H}\left(\boldsymbol{R} + \boldsymbol{H}^T\boldsymbol{P}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{P}\boldsymbol{G}, \tag{7}$$

where Q and R are the weighting matrices for the state variables and control inputs, respectively, and P is the unknown matrix to be solved. Once P is obtained, the optimal feedback gain matrix K can be calculated as:

$$\boldsymbol{K} = \left(\boldsymbol{H}^T\boldsymbol{P}\boldsymbol{H} + \boldsymbol{R}\right)^{-1}\boldsymbol{H}^T\boldsymbol{P}\boldsymbol{G}. \tag{8}$$

Similar to the methods described in [10] and [17], for virtual legs of varying lengths, the corresponding state-space equations are formulated individually, and the respective feedback gain matrices K are subsequently computed. By performing polynomial fitting on each element of K (L_0 ranging from 0.1 m to 0.35 m with step size 0.01 m, and the polynomial order is 3 in out case), we can obtain a feedback gain matrix $K(L_0)$ that varies as a function of the virtual leg length L_0. Consequently, the system's feedback control law is expressed as:

$$\boldsymbol{u} = -\boldsymbol{K}(L_0)(\boldsymbol{x}_d - \boldsymbol{x}), \tag{9}$$

where $\boldsymbol{x}_d = \begin{bmatrix} 0 & 0 & x_d & \dot{x_d} & 0 & 0 \end{bmatrix}^T$, in which x_d is the desired displacement of the body.

C. State Variables Estimation

The aforementioned modeling considers only the robot's motion in a two-dimensional plane, without accounting for the actual bipedal structure. Therefore, it is necessary to design an appropriate state observer to estimate the required state variables in Eq. 3. The observer is designed as

$$\boldsymbol{x} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x_b \\ \dot{x}_b \\ \phi \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \alpha - \frac{\pi}{2} - \phi_{imu} \\ \dot{\alpha} - \dot{\phi}_{imu} \\ R\theta_w + L_0 \sin\theta \\ R\dot{\theta}_w + L_0\dot{\theta}\cos\theta + \dot{L}_0 \sin\theta \\ \phi_{imu} \\ \dot{\phi}_{imu} \end{bmatrix}, \tag{10}$$

where ϕ_{imu} and $\dot{\phi}_{imu}$ are the pitch angle and its changing rate, respectively, measured by the body-mounted IMU. $\theta_w = \frac{1}{2}(\theta_w^l + \theta_w^r)$, $\dot{\theta}_w = \frac{1}{2}(\dot{\theta}_w^l + \dot{\theta}_w^r)$ represent the average cumulative rotational angle and angular velocity of the left and right wheels obtained from wheel feedback. $\alpha = \frac{1}{2}(\phi_0^l + \phi_0^r)$, $\dot{\alpha} = \frac{1}{2}(\dot{\phi}_0^l + \dot{\phi}_0^r)$ are the average angles and angular velocity of left and right virtual legs, calculated through forward kinematics of the legs.

D. Basic Control Framework

In addition to the balance controller, there are PD controllers designed for controlling height, roll, yaw, and synchronization of the two legs. As shown in Fig. 8, $[\dot{x}^{\text{ref}}, h^{\text{ref}}, \phi_{\text{roll}}^{\text{ref}}, \phi_{\text{yaw}}^{\text{ref}}]$ are the reference command of state variables, body height, body roll angle, and body yaw angle respectively. $\mathbf{x}$ is the state variables vector. l_0 is the average length of left and right virtual legs. ϕ_0^{diff} is the difference angle between two virtual legs. ϕ_{roll} and ϕ_{yaw} are the roll and yaw angles of the body, respectively. $F_p^{\text{feedforward}} = m_b g$ is the feedforward force for the leg length controller, where m_b is the mass of the body. These PD controllers work in conjunction with the LQR to achieve comprehensive motion control of the entire robot.

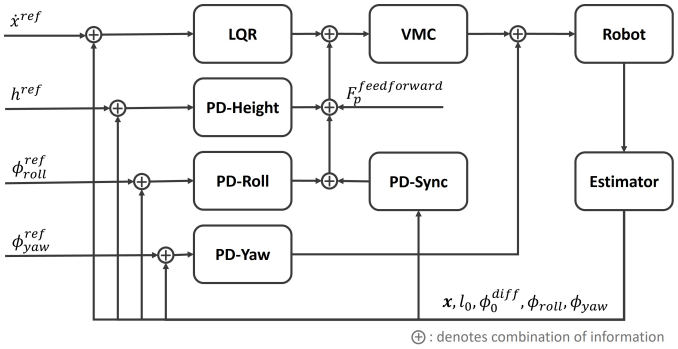


Fig. 8: The complete control framework incorporates LQR and PD controllers.

E. Velocity Estimation

The aforementioned model assumes perfect rolling without any slippage between the wheels and the ground, which does not accurately reflect real-world conditions. According to the observation method described above, when wheel slippage occurs, both θ_w and $\dot{\theta}_w$ exhibit significant fluctuations. These fluctuations can cause the controller to generate excessively large torque outputs, which may further aggravate the slippage and ultimately lead to system instability and failure.

In order to address this issue, we employ a Kalman filter to fuse data from the left and right wheel encoders as well as the acceleration and angular velocity measurements from the IMU. The fusion process explicitly accounts for the uncertainties associated with different sensor sources. During the update step, residuals are monitored in real time to identify and isolate anomalous sensor data. This approach enables us to obtain a more accurate estimation of the robot's body velocity, which significantly enhances the control robustness of the robot when operating on uneven terrain.

1) System State-Space Model: To simplify the problem, we model the robot as a two-wheel differential drive system during the filtering process. The state vector of the filter is defined as follows:

The state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} v \\ a \\ \omega \end{bmatrix}, \tag{11}$$

where v denotes the linear velocity of the robot body along the forward direction, a represents the linear acceleration along the same direction, and ω is the yaw rate about the vertical axis.

The state transition model is given by:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k, \tag{12}$$

where $\mathbf{F}$ is the state transition matrix:

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{13}$$

and $\mathbf{w}_k$ is the process noise, assumed to be zero-mean Gaussian with covariance matrix $\mathbf{Q}$.

2) Measurement Model: The measurement vector is defined as:

$$\mathbf{z} = \begin{bmatrix} \omega_{\text{right}} \\ \omega_{\text{left}} \\ a_{\text{imu}} \\ \omega_{\text{imu}} \end{bmatrix}, \tag{14}$$

where ω_{right} and ω_{left} represent the rotational angular velocities of the right and left wheels measured by the wheel encoders, respectively. a_{imu} denotes the linear acceleration of the robot's body along the forward direction as measured by the IMU, and w_{imu} is the yaw angular velocity about the vertical axis measured by the IMU.

The measurement equation is given by:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \tag{15}$$

where $\mathbf{H}$ is the measurement matrix with wheel radius R and the distance between the wheels L:

$$\mathbf{H} = \begin{bmatrix} \frac{1}{R} & 0 & \frac{L}{2R} \\ \frac{1}{R} & 0 & -\frac{L}{2R} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

and $\mathbf{v}_k$ is the measurement noise, assumed to be zero-mean Gaussian with covariance matrix $\mathbf{R}$.

3) Slippage Detection: Between the prediction and update steps of the Kalman filter, we detect wheel slippage by monitoring the measurement residuals. The measurement residual $\widetilde{\mathbf{z}}_k$ is defined as:

$$\widetilde{\mathbf{z}}_k = \begin{bmatrix} \widetilde{\omega}_{\text{right},k} \\ \widetilde{\omega}_{\text{left},k} \\ \widetilde{a}_{\text{imu},k} \\ \widetilde{\omega}_{\text{imu},k} \end{bmatrix} = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \tag{17}$$

where $\hat{x}_{k|k-1}$ is the predicted state vector.

If the measurement residual for the wheel rotational speed exceeds a certain threshold ($\widetilde{\omega}_{\max}$), it is determined that the corresponding wheel is experiencing slip. In this case, the observation for that wheel is corrected by removing the residual, meaning that the predicted value is directly used:

$$\hat{\omega}_{i,k} = \omega_{i,k} - \widetilde{\omega}_{i,k}, \quad (i = \text{left, right}). \tag{18}$$

The corrected measurement vector is used for the update step.

$$\hat{\mathbf{z}}_k = \begin{bmatrix} \hat{\omega}_{\text{right},k} \\ \hat{\omega}_{\text{left},k} \\ a_{\text{imu},k} \\ \omega_{\text{imu},k} \end{bmatrix}. \tag{19}$$

4) Kalman Iteration:

- State prediction:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}. \tag{20}$$

- Error covariance prediction:

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}. \tag{21}$$

- Kalman gain calculation:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1}. \tag{22}$$

- State update using corrected measurement vector:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\hat{\mathbf{z}}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}). \tag{23}$$

- Error covariance update:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}. \tag{24}$$

Through experimental testing, it was found that neglecting the leg dynamics and directly using the linear velocity v obtained via Kalman filtering, along with its integral, as the state variables for balance control does not lead to significant adverse effects. Specifically,

$$\begin{cases} \dot{x}_b = v \\ x_b = \int v\, dt. \end{cases} \tag{25}$$

5) Summary: The proposed multi-sensor fusion observer leverages a Kalman filter to estimate the linear velocity and yaw rate of the robot using measurements from wheel encoders and the IMU. By observing the measurement residual, the observer can effectively detect and handle wheel slippage, thereby acquiring highly accurate estimations of the state variables.

F. Jumping Control

Active jumping is a crucial capability for wheel-legged biped robots, primarily achieved by altering the length of the virtual leg. As shown in Fig. 9, a single jump motion can be described in four stages:

- Phase 1: The robot initially reduces the leg length to l_{low}.
- Phase 2: The robot then exerts a powerful thrust against the ground until the leg reaches its predetermined maximum length l_{high}.
- Phase 3: The robot shortens the leg length again to achieve the maximum possible height off the ground.
- Phase 4: The robot extends the leg once more until it makes contact with the ground.

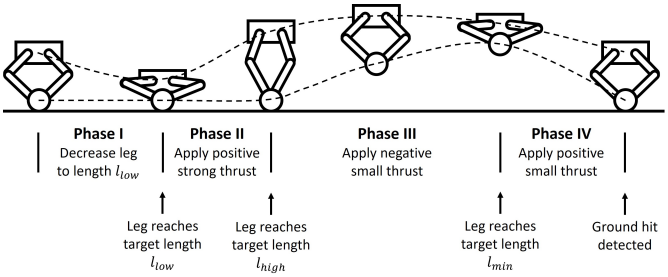


Fig. 9: The jumping process.

During the flight phase, when the wheels are not in contact with the ground, the torques of the wheels are set to zero, and the LQR balance controller is deactivated. Instead, a PD controller applied to T_p is used to maintain the virtual leg perpendicular to the ground, preparing for ground contact and cushioning the landing.

In addition, our experiments reveal that the maximum jumping height can be effectively controlled by adjusting l_{high}. This approach will be utilized in subsequent tasks where the robot is required to jump over known terrain.

G. Terrain Feedforward Control

Previous studies have primarily focused on robot motion on flat terrain. However, in real-world scenarios, terrain undulations have a significant impact on the stability of the robot. Incorporating terrain considerations into the controller can greatly enhance the robot's performance on complex terrains. To address this, we have designed a trajectory prediction and feedforward control method described below.

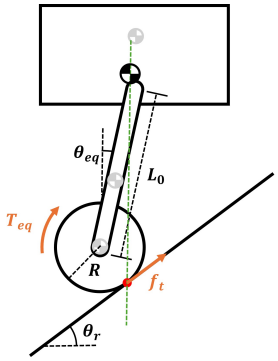


Fig. 10: Static analysis of the robot on a slope.

1) Feedforward Control: The feedforward terms for the robot were primarily derived through a static mechanical analysis when the robot remains stationary on a slope, as illustrated in Fig. 10. It can be observed that the main influence of uneven terrain on the robot arises from the contact angle θ_r between the wheels and the ground. This alters both the equilibrium pole angle θ and the wheel output torque T compared to the case of a level surface.

Through analysis, it is found that the robot achieves static equilibrium on a slope when its overall center of mass is located vertically above the wheel contact point. Assuming the robot body remains horizontal and taking the wheel center as the origin, the horizontal position of the overall center of mass is calculated and set equal to the contact point:

$$\frac{m_w \cdot 0 + m_p L \sin(\theta_{\text{eq}}) + m_b L_0 \sin(\theta_{\text{eq}})}{M} = R\sin(\theta_r), \tag{26}$$

where m_w, m_p, and m_b denote the masses of the wheel, pole, and body, respectively, $M = m_w + m_p + m_b$ is the total mass, θ_{eq} is the equilibrium leg angle, and $L = kL_0$ is the position of the pole's center of mass along itself.

For the robot as a whole, it is straightforward to obtain the force balance in the direction along the slope:

$$Mg\sin(\theta_r) = f_t, \tag{27}$$

where f_t is the friction force between the ground and the wheel. From the torque equilibrium of the wheel, we have:

$$T_{\text{eq}} = f_t R, \tag{28}$$

where T_{eq} is the equilibrium output torque of the wheel's motor.

Combining the above equations, we obtain:

$$\begin{cases} \theta_{\text{eq}} = \arcsin\left(\dfrac{MR\sin(\theta_r)}{(km_p + m_b)L_0}\right) \\ T_{\text{eq}} = MgR\sin(\theta_r) \end{cases}. \tag{29}$$

Here, θ_{eq} is used as the reference value for the leg angle state θ in the LQR balance controller, and T_{eq} is superimposed on the torque output calculated by the LQR controller for the wheel motors.

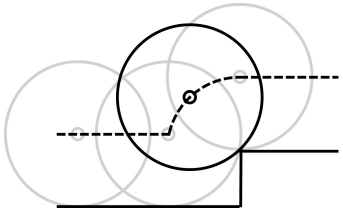


Fig. 11: The wheel center trajectory when climbing a stair.

2) Trajectory Prediction and Slope Calculation: We can notice that the contact angle θ_r of the wheels is equal to the slope angle of the trajectory of the wheel center, as illustrated in Fig. 11. Thus, we design the following method to obtain the contact angle.

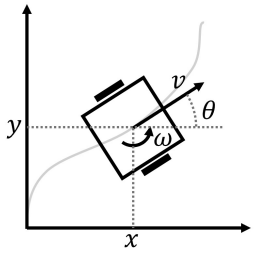


Fig. 12: The differential driving model.

For simplicity, the wheel-legged biped robot can be modeled as a differential drive system with two independently actuated wheels. As shown in Fig. 12, its configuration is described by its position (x, y) in the global coordinate frame and its orientation θ, which represents the robot's heading angle. The kinematic model relates the robot's control inputs, linear velocity v and angular velocity ω, to its motion in the planar workspace.

The continuous-time kinematics of the robot are expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{30}$$

where $\dot{x}$ and $\dot{y}$ are the translational velocities in the x and y directions, respectively, and $\dot{\theta}$ is the rate of change of the robot's orientation. The matrix transformation accounts for the robot's orientation in the global frame.

For implementation in digital systems, the continuous model is discretized using Euler integration:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \Delta t \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}, \tag{31}$$

where k and $k+1$ denote the current and next time steps, and Δt is the sampling time.

Next, assuming the robot perfectly tracks the given linear velocity v_{ref} and angular velocity ω_{ref}, the predicted 2D body trajectory can be computed by propagating forward for N steps using Equ. (31) as

$$S_{\text{2d}} = \{(x_i, y_i) | i = 1, 2, \ldots, N\}. \tag{32}$$

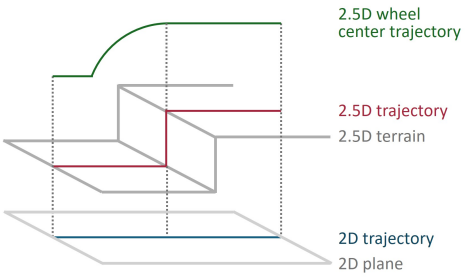


Fig. 13: 2.5D trajectory calculation.

Subsequently, we construct the elevation map of the terrain with LiDAR using the elevation mapping algorithm [18] [19] and localization algorithm Point-LIO [20] that is especially suitable for IMU saturation during impacts. As shown in Fig. 13, by performing bilinear interpolation on the elevation map, the projected trajectory can be derived:

$$S_{\text{proj}} = \{(s_i^c, z_i^c) | i = 1, 2, \ldots, N\}, \tag{33}$$

where s_i^c is the length of the trajectory that travels on the X-Y plane:

$$s_i^c = \sum_{j=1}^{i-1} \|(x_{j+1}, y_{j+1}) - (x_j, y_j)\|_2, \tag{34}$$

and z_i^c is the interpolation height of point (x_i, y_i) on the elevation map.

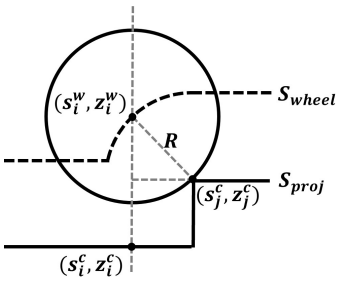


Fig. 14: Wheel center trajectory calculation.

Then, we calculate the trajectory of the wheel center as shown in Fig. 14. For each point (s_i^c, z_i^c) in S_{proj}, we compute its corresponding wheel center position (s_i^w, z_i^w) as $s_i^w = s_i^c$ and

$$z_i^w = \max_{-R \le s_j^c - s_i^c \le R} \left(z_j^c + \sqrt{R^2 - (s_j^c - s_i^c)^2} \right), \tag{35}$$

which yields the wheel center trajectory:

$$S_{\text{wheel}} = \{(s_i^w, z_i^w) | i = 1, 2, \ldots, N\}. \tag{36}$$

Finally, we can calculate the contact angle at each sampling point as follows:

$$\theta_r(s_i^w) = \arctan\left(\frac{z_{i+1}^w - z_i^w}{s_{i+1}^w - s_i^w} \right). \tag{37}$$

H. Automatic Jumping on Known Terrain

If at any point on the predicted trajectory the wheel contact angle exceeds a certain range, the wheels will not be able to provide sufficient traction to traverse that section using normal driving. In urban road environments, this typically occurs at steps, so we can detect steps along the trajectory and then cross these terrains with a jumping motion.

First, it is necessary to accurately predict and control the jumping action, including calculating the jump timing and controlling the jump height. Since the duration of the jump is very short and there are many influencing factors, using a purely dynamics-based approach is complex and might differ significantly from reality. Therefore, we directly conduct multiple experiments in the real world, and fit the formula between jumping parameters and control quantities as:

$$\begin{cases} l_{\text{high}} = g_l(h_{\text{air}}) \\ t_{\text{delay}} = g_t(h_{\text{air}}) \end{cases} \tag{38}$$

where h_{air} is the maximum height the wheel leaves the ground, t_{delay} is the delay from triggering the jump to the wheel reaching this height, and l_{high} is the maximum leg extension during the jump, as shown in Fig. 9.

Next, we need to analyze the step height and the distance from the robot's current position to the step on the predicted trajectory. This can be achieved by analyzing the contact angle along the trajectory. We search for the nearest segment in the predicted trajectory where the contact angle exceeds $\theta_r^{\max}$ over a sequence of consecutive sampling points, and treat this as the rising part of the step. We then calculate the height difference of this segment as the step height h_{step}. The trajectory length between its highest point and the current position, s_{step}, is used to calculate the jump trigger lead time. When the robot's state satisfies:

$$\begin{cases} h_{\text{step}} + h_{\text{add}} \le h_{\text{air}}^{\max} \\ |s_{\text{step}} - v\, g_t(h_{\text{step}} + h_{\text{add}})| \le \epsilon_s \end{cases}, \tag{39}$$

the jump is triggered, where the jump control input is given by

$$l_{\text{high}} = g_l(h_{\text{step}} + h_{\text{add}}). \tag{40}$$

Here, h_{add} is the jump height margin, $h_{\text{air}}^{\max}$ is the robot's maximum jump height, v is the current linear velocity of the robot, and ϵ_s is the jump window distance tolerance.

IV. Experiment

A. Real-world Robot Design

(a) Main supporting framework (b) Complete mechanical structure

Fig. 15: Mechanical structure of the real-world robot.

1) Mechanical Structure Design: In this experiment, we designed and constructed a prototype of a wheel-legged biped robot to validate the improved locomotion method. The focus of the mechanical structure design was to ensure the robot's structural strength and stability during jumps or falls. We used high-strength aluminum profiles to build the main support frame (see Fig. 15a), which not only provides robust support but also effectively prevents damage from external forces. Other structural components were made from carbon fiber, featuring a hollow design to maximize weight reduction while maintaining strength and aesthetics.

To enhance the robot's ability to perceive terrain ahead, we installed the LiDAR sensor with a downward tilt (see Fig. 15b). This design increases the utilization of the LiDAR point cloud and improves the accuracy of terrain perception.

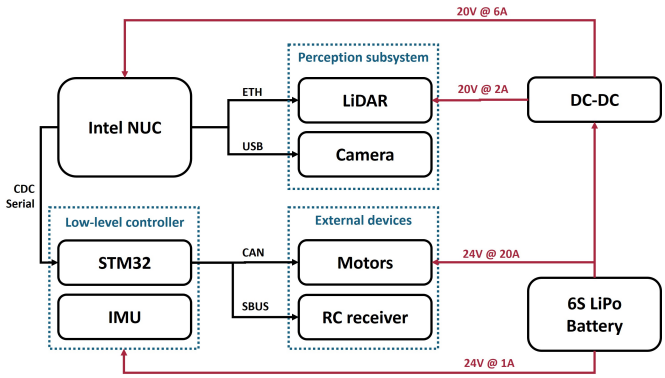


Fig. 16: Electronic design of the real-world robot.

2) Electronic Hardware Design: For the electronic hardware, we selected a high-performance 13th generation i7 Intel NUC as the main computing device to ensure sufficient computational power. An STM32 microcontroller was used as the low-level controller, responsible for hardware driving and IMU attitude calculation. The lower-level system communicates with the main computer via a USB CDC serial port, transmitting data and receiving commands for execution. The entire system is powered by a 6s LiPo battery, with a step-down device supplying power to the main computer (see Fig. 16).

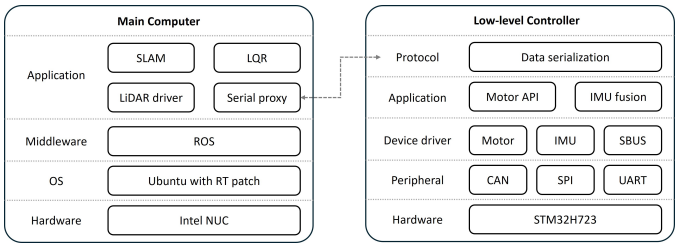


Fig. 17: Software architecture of the real-world robot.

3) Software Architecture: In terms of software architecture, all computations are performed entirely on the robot's onboard system without reliance on any external computing device. We deployed almost all major programs on the main computer, including the core computations for motion control. To ensure real-time control, the main computer operates on a ROS system with a real-time kernel patch, and multiple control

nodes are constructed, such as data transmission nodes, motion control nodes, and SLAM nodes (see Fig. 17).

The lower-level micro-controller program is divided into multiple layers: the peripheral layer handles driving of CAN, UART, and other low-level communication protocols; the device driver layer includes drivers for different devices; the application layer provides high-level encapsulation and algorithms; and the protocol layer manages custom protocol encapsulation between the upper and lower systems. Through this multi-layer encapsulation, the upper computer can control different devices via a unified interface, achieving a complete control chain from low-level device communication to top-level serialization protocol.

B. Locomotion Experiments

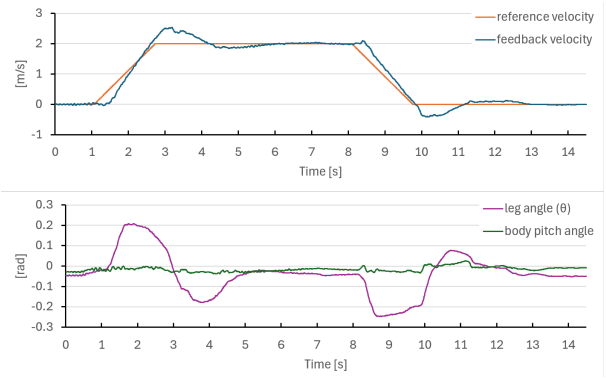


Fig. 18: Robot follows a trapezoidal velocity reference.

1) Velocity Tracking: In this experiment, we maintain a reference virtual leg length of $L_0^{ref} = 0.2\,\text{m}$ and set the reference yaw and roll angles as $\phi_{yaw}^{ref} = \phi_{roll}^{ref} = 0$. The reference velocity $\dot{x}_d$ described in Eq. 9 starts from 0 and increases at a constant speed of $1.2\,\text{m/s}^2$ until reaching 2 m/s. The robot's tracking performance is shown in Fig. 18.

As observed, the robot successfully follows the reference velocity and displacement trajectory, while the pitch angle of the upper body remains highly stable throughout the motion. This behavior aligns well with the expected outcomes, demonstrating the effectiveness of the control strategy.

2) Velocity Estimation and Slippage Detection: In this experiment, we manually lifted the right wheel of the robot and then released it from the air while spinning it at a high speed to simulate severe slippage. This was performed twice under different conditions: with or without the Kalman sensor fusion method. As shown in Figs. 19 and 20, when slippage happens without sensor fusion, the dramatic change in velocity causes failure in the balance control, while the abnormal wheel measurement is almost ignored using the sensor fusion method. The experimental result demonstrates that the proposed method effectively suppresses the inaccuracies in observation caused by slippage. This significantly enhances the robustness of the system.

3) Disturbance Resistance: In this experiment, the robot was subjected to an external kick. As shown in Figs. 21 and 22, the robot was momentarily accelerated to a speed of about

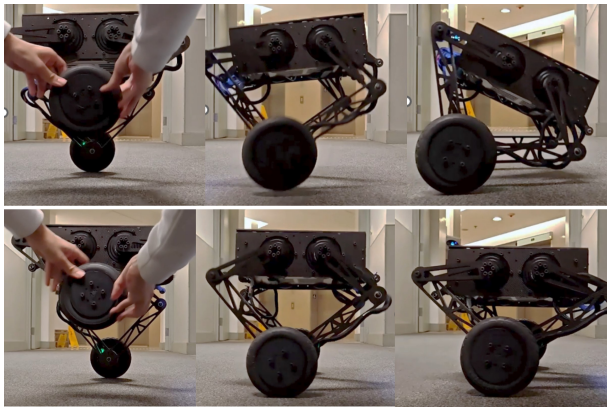


Fig. 19: Without sensor fusion, the robot slips and falls (top); with sensor fusion, it recovers from slipping (bottom).

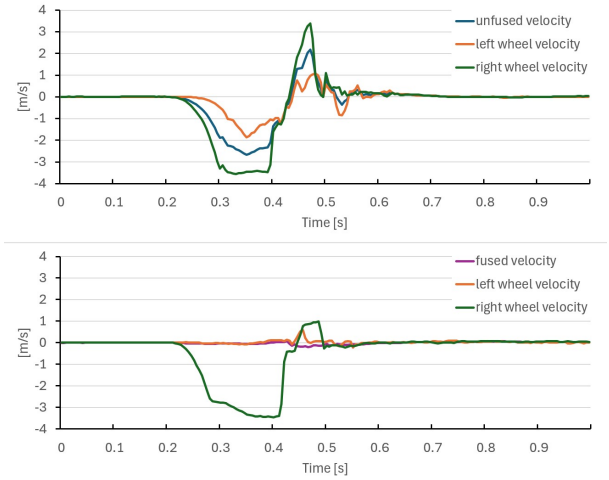


Fig. 20: Without sensor fusion, velocity is severely affected by slippage (top); with sensor fusion, velocity estimation is accurate (bottom).

2 m/s, resulting in deviations in the body attitude angles of approximately $\phi^e = 0.118\,\text{rad}$, $\phi^e_{\text{roll}} = 0.128\,\text{rad}$, and $\phi^e_{\text{yaw}} = 0.627\,\text{rad}$. Subsequently, the controller quickly corrected these deviations, restoring the robot to its normal, stationary upright posture within 2 seconds.

4) Terrain Feedforward: In this experiment, the robot was positioned stationary on a slope with a controllable inclination angle. We measured the tracking performance of key state variables under different inclination angles, both with and without feedforward control enabled. Among all state variables, the body pitch angle ϕ exhibited the most significant change. As illustrated in Fig. 23, when the wheel contact angle increases, the pitch angle ϕ gradually deviates from

Fig. 21: Robot recovers after being kicked.

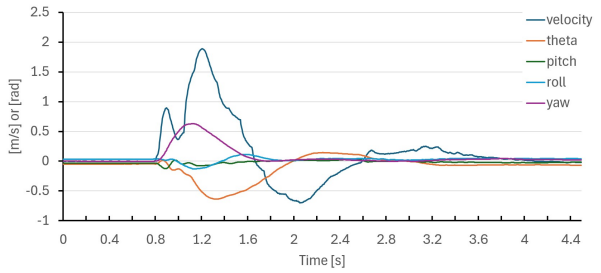


Fig. 22: Robot posture when recovering from kicking.

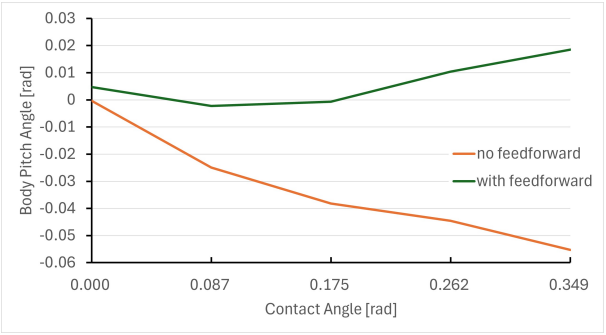


Fig. 23: Body pitch angle ϕ changes with contact angle.

the horizontal reference when feedforward control is disabled. In contrast, enabling feedforward control substantially reduces this deviation.

Upon further analysis, one primary reason that the deviation does not fully vanish is the presence of a dead zone in the joint motors, leading to imprecise torque execution. Further improvements at the hardware level may be required to address this issue.

Fig. 24: Robot traverses a 3 cm speed bump.

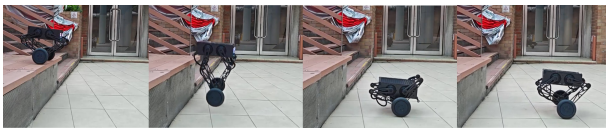


Fig. 25: Robot jumps down from a 40 cm platform.

5) Traversability: When traversing a 3 cm high speed bump, the robot passed smoothly (Fig. 24), and its ability to jump down from a 40 cm high platform and land smoothly (Fig. 25) demonstrates the controller's effective utilization of the robot's leg structure. By adjusting joint compliance, the robot absorbed the impact of landing, ensuring a safe touchdown.

6) Automatic Step Crossing: As shown in Fig. 26, we set up a step with a height of 10 cm. The robot utilized LiDAR for localization and employed the depth camera to acquire

Fig. 26: Robot automatically jumps onto a 10 cm step.

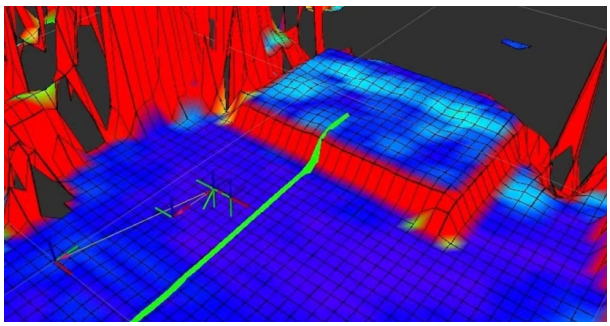


Fig. 27: Trajectory prediction on the elevation map.

precise terrain information, which was then used to construct an elevation map as in Fig. 27. Using the previously described methods, the robot performed trajectory prediction and step detection to estimate the step height and distance. At the appropriate moment, the jump process was triggered with the calculated jumping parameters, enabling the robot to successfully cross the step. This result demonstrates the feasibility of the proposed series of algorithms and significantly enhances the robot's terrain traversal capability.

C. Summary

These experimental results indicate that the locomotion control method designed for handling complex terrain and dynamic disturbances exhibits commendable robustness and flexibility. Furthermore, they affirm the rationality of the hardware and software design of the experimental platform.

V. Conclusion

The development and implementation of the wheel-legged biped robot presented in this paper demonstrate significant advancements in the field of autonomous robotics for community inspection and monitoring. Through a comprehensive approach that includes mechanical design, electronic hardware, software architecture, and advanced control algorithms, we have successfully addressed the challenges of perception and locomotion in semi-structured environments.

Our mechanical structure, crafted with high-strength materials and a focus on stability, has proven capable of withstanding the rigors of real-world operation, including jumps and falls. The electronic hardware, powered by a high-performance computing device, ensures the necessary computational power for real-time control and processing. The software architecture,

built on an ROS system with real-time capabilities, supports the complex control nodes required for seamless operation.

In terms of locomotion, the robot has exhibited commendable performance in velocity tracking, disturbance resistance, slippage compensation, and traversability. The integration of a Kalman filter for state estimation has significantly mitigated the effects of wheel slippage, while the feedforward controller improved the performance on unstructured terrains, both enhancing the system's robustness. The robot's ability to jump and traverse various terrains validates the efficacy of our jumping control method and the overall control strategy.

In conclusion, the wheel-legged biped robot system developed in this research has demonstrated itself to be a reliable and practical solution for community inspection tasks. It can not only improve efficiency and safety in public spaces but also contribute significantly to the expanding body of research on autonomous robots in public service. The successful integration of multisource sensory information and the development of a locomotion control method suitable for unstructured terrains have enhanced the robot's operational capabilities in complex environments. This work lays the groundwork for the broader adoption of such robots in real-world applications and provides a solid foundation for future research and development in the field of autonomous robotics.

There are still limitations in this work that require further investigation. A primary issue is the omission of potential differences in the motion of the two legs, which may cause problems in scenarios such as traversing steps at an angle. Future work may explore more sophisticated leg motion planning strategies to enable even greater terrain traversal capabilities.

REFERENCES

[1] D. Lattanzi and G. Miller, "Review of robotic infrastructure inspection systems," *Journal of Infrastructure Systems*, vol. 23, no. 3, p. 04017004, 2017.

[2] M. Fisher, R. C. Cardoso, E. C. Collins, C. Dadswell, L. A. Dennis, C. Dixon, M. Farrell, A. Ferrando, X. Huang, M. Jump *et al.*, "An overview of verification and validation challenges for inspection robots," *Robotics*, vol. 10, no. 2, p. 67, 2021.

[3] F.-H. Tseng and Y.-J. Hsieh, "Delanalty minimization with reinforcement learning in uav-aided mobile network," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 1991–2001, 2023.

[4] S. Teng, Y. Wang, X. Wang, J. Li, Y. Li, X. Zhang, L. Li, L. Chen, and F.-Y. Wang, "The parallelworkforce: A framework for synergistic collaboration in digital, robotic, and biological workers of industry 5.0," *IEEE Transactions on Computational Social Systems*, 2024.

[5] X. Liu, Y. Sun, S. Wen, K. Cao, Q. Qi, X. Zhang, H. Shen, G. Chen, J. Xu, and A. Ji, "Development of wheel-legged biped robots: A review," *Journal of Bionic Engineering*, vol. 21, no. 2, pp. 607–634, 2024.

[6] M. Bjelonic, V. Klemm, J. Lee, and M. Hutter, "A survey of wheeled-legged robots," in *Climbing and walking robots conference*. Springer, 2022, pp. 83–94.

[7] O. Matsumoto, S. Kajita, M. Saigo, and K. Tani, "Dynamic trajectory control of passing over stairs by a biped type leg-wheeled robot with nominal reference of static gait," *Journal of the Robotics Society of Japan*, vol. 16, no. 6, pp. 868–875, 1998.

[8] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin *et al.*, "Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals," *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829, 2017.

[9] B. Dynamics, "Introducing handle," https://www.youtube.com/watch?v=-7xvqQeoA8c, 2017, accessed: 2024-12-15.

[10] V. Klemm, A. Morra, C. Salzmann, F. Tschopp, K. Bodie, L. Gulich, N. Küng, D. Mannhart, C. Pfister, M. Vierneisel *et al.*, "Ascento: A two-wheeled jumping robot," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7515–7521.

[11] Ascento, "Ascento: A two-wheeled jumping robot (icra 2019)," https://www.youtube.com/watch?v=U8bIsUPX1ZU, 2019, accessed: 2025-02-28.

[12] S. Wang, L. Cui, J. Zhang, J. Lai, D. Zhang, K. Chen, Y. Zheng, Z. Zhang, and Z.-P. Jiang, "Balance control of a novel wheel-legged robot: Design and experiments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6782–6788.

[13] BotJunkie, "Tencent ollie," https://www.youtube.com/watch?v=ZtlRVO8fpE0, 2021, accessed: 2024-12-15.

[14] T. Guo, J. Liu, H. Liang, Y. Zhang, W. Chen, X. Xia, M. Wang, and Z. Wang, "Design and dynamic analysis of jumping wheel-legged robot in complex terrain environment," *Frontiers in neurorobotics*, vol. 16, p. 1066714, 2022.

[15] Z. Cui, Y. Xin, S. Liu, X. Rong, and Y. Li, "Modeling and control of a wheeled biped robot," *Micromachines*, vol. 13, no. 5, p. 747, 2022.

[16] L. Cui, S. Wang, J. Zhang, D. Zhang, J. Lai, Y. Zheng, Z. Zhang, and Z.-P. Jiang, "Learning-based balance control of wheel-legged robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7667–7674, 2021.

[17] Y. Chen, H. Wang, and L. Zhang, "Control of a wheel-legged balance robot," *Information & Control*, vol. 52, no. 5, 2023.

[18] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[19] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.

[20] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200459, 2023.

Bowen Lan received his bachelor degree in Automation from Jilin University, Jilin, China, in 2023. He is now an MPhil student in Aeronautical and Aviation Engineering at the Hong Kong Polytechnic University, Hong Kong. His current research interests include legged robots locomotion, optimal control and reinforcement learning.

Hailong Huang (Senior Member, IEEE) received his Ph.D degree in Systems and Control from the University of New South Wales, Sydney, Australia, in 2018. He is now an Assistant Professor with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong. His current research interests include guidance, navigation, and control of UAVs and mobile robots. He is an Associate Editor of IEEE Transactions on Automation Science and Engineering, Journal of Field Robotics, IEEE Transactions on Vehicular Technology and IEEE Transactions on Intelligent Vehicles.