

# Self Balancing Performance of Wheel-legged Robots under Disturbance and Faulty Conditions

Zhenyi Yang\*  
 School of Engineering  
 The University of Manchester  
 Manchester, UK  
 zhenyi.yang-4@postgrad.manchester.ac.uk

Chenyi Li  
 School of Engineering  
 The University of Manchester  
 Manchester, UK  
 chenyi.li@manchester.ac.uk

Long Zhang  
 School of Engineering  
 The University of Manchester  
 Manchester, UK  
 long.zhang@manchester.ac.uk

**Abstract**—This study investigates the self-balancing performance of a bipedal wheel-legged robot under a range of disturbances and faulty conditions. The disturbances include external kicks, additional weight loads, wind forces, trailer carrying, and collisions. Additionally, the paper evaluates the robot's resilience under various working and fault conditions including balancing on soft and rotatable surfaces and working with wheel malfunctions. The cascaded PID and adaptive control method are employed for the robot's balance and movement. Comprehensive simulation and experimental results demonstrate the effectiveness and robustness of the wheel-legged robot control.

**Index Terms**—*Self-balance, Disturbance rejection, Robotic manipulator*

## I. INTRODUCTION

Wheel-legged robots combine the advantages of wheeled and legged locomotion, offering versatility in maneuvering over diverse terrains, allowing them to operate in a high energy efficiency under a versatile environment. In recent years, these wheel-legged robots have attracted significant research attentions and they have potential very wide applications such as search and rescue missions, exploration, and service robotics [1]. There are many research and prototypes of wheel-legged robots. One of the major milestones of wheel-legged robots is the Boston Dynamics Handle, developed in 2019. Later, ETH proposed and published a paper on a new bipedal wheel-legged robot called Ascento in 2019 [2]. And more currently, the wheel-legged robot NeZha designed by Hua Chen et al. [3] and the robot Ollie, developed by Shuai Wang et al in 2021 [4] attract the attention of more scholars, more and more people realize the advantages of this kind of robot with wheel-legged structure.

Due to these different applications and working conditions, maintaining balance is crucial for their operation, especially under varying disturbances like additional loads, wind, impacts, and uneven surfaces. Furthermore, the robots' internal faults such as wheel malfunctions and battery degradation, which can also impact the performance of the wheel-legged robots. However, very little work has been done for examining the balancing performance under both disturbance rejection and faulty conditions in real systems [1].

\*Corresponding author: zhenyi.yang-4@postgrad.manchester.ac.uk

In this paper, a five-link type bipedal wheel-legged robot is used [5]. To the best knowledge of the authors, it is the first time to evaluate the balancing performance of the low-cost wheel-legged robot under more than ten different scenarios of disturbances and faulty conditions.

## II. WHEEL-LEGGED ROBOT INTRODUCTION

The robot's 3D model and the physical image are shown in figure 1. The overall structure of the robot was made using 3D printing material PLA, consisting of a main body, two legs, and two actively controlled wheels, they work in groups to achieve the mobility and stability of the wheel-legged robot.

The main body of the robot is a crucial structural element that houses the central hardware, including the controller board, battery, Inertial Measurement Unit (IMU), and servo joint motors. These components are essential for the robot's operation, as they facilitate the control and stabilization of the entire system. The body is suspended above the ground, maintained at a specific height by the two legs, which act as both support structures and manipulators. This elevated configuration allows the robot to navigate various terrains while keeping the main body secure from direct contact with the ground.

The overall height of the robot is adjustable between 10 cm and 25 cm, controlled by 8120MG servos, referred to as the joint motors in this paper, positioned at points A and E in figure 1. The movement of the robot is driven by PM3010 brushless motors, designated as the wheel motors, located at point C in figure 1. The hardware controllers of the robot are two micro control units (MCU) called ESP32, which has powerful floating-point operation ability and Bluetooth communication function, which enables users to remote control the robot through mobile phones. Figure 2 shows four different statuses of the wheel-legged robot [5].

- 1) Initial position. The robot starts in a lying down position on the ground.
- 2) Balance maintenance. The robot maintains balance at its minimum height of 10cm.
- 3) Height adjustment. While standing still, the robot adjusts its leg height, changing from 5cm to 20cm.
- 4) Roll angle stabilization. The robot adjusts the length of its two legs while maintaining roll angle balance.

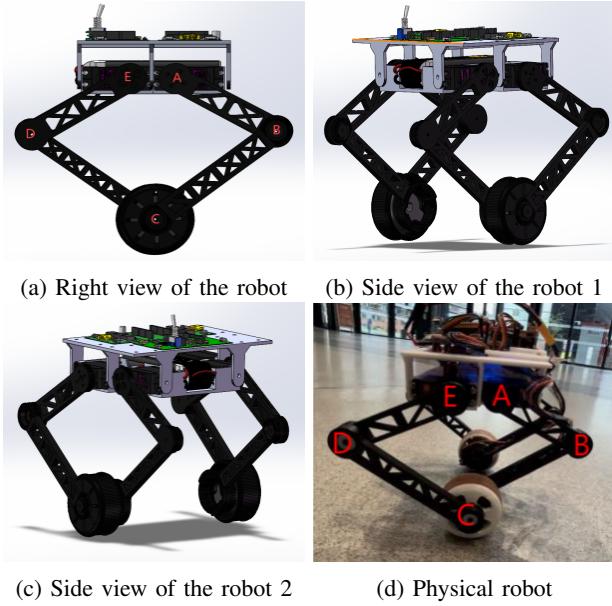


Fig. 1: The 3D model of the robot and the physical robot image

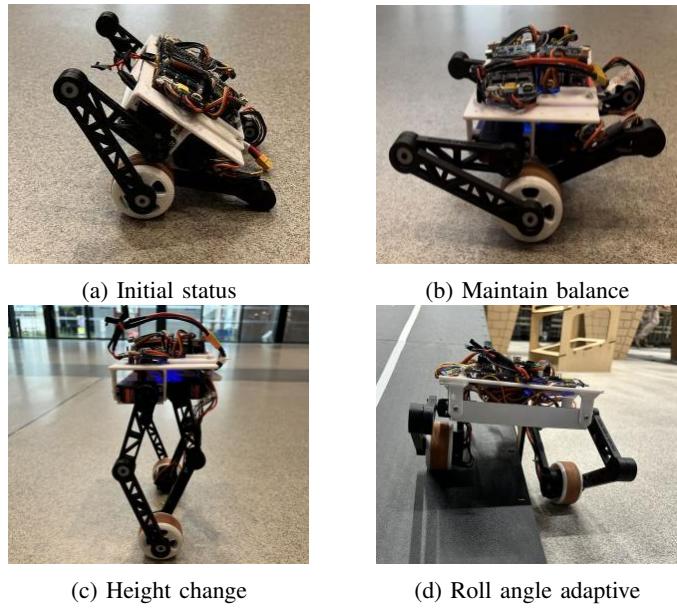


Fig. 2: Robot status

### III. ROBOT DYNAMIC MODEL

This section introduces the dynamic model of a wheel-legged robot with a focus on balance control. The dynamic model of the wheel-legged robot can be divided into two components: a Modified Inverted Pendulum on a Cart model and a five-link robotic manipulator model.

#### A. Inverted Pendulum on a Cart model

The Inverted Pendulum on a Cart (IPC) model is a model for the wheel-legged robot. In the IPC model, the wheels of the robot function as the cart, and the main body of the robot

acts as the mass at the end of the inverted pendulum. The leg structure can be simplified to a link connecting the cart and the mass end of the inverted pendulum after the five-link manipulator is modeled. Figure 3 illustrates a simplified diagram of this system, where  $M$  is the mass of the cart,  $m$  is

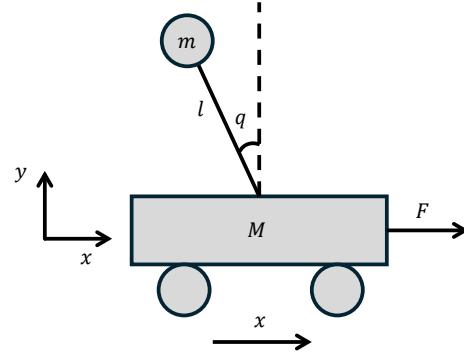


Fig. 3: The simplified diagram of IPC

the mass at the end of the inverted pendulum,  $l$  is the length of the pendulum,  $q$  is the angle between the connecting link and the vertical position,  $F$  is the external force acting on the cart,  $x$  is the displacement of the cart in the horizontal direction, and  $y$  is the displacement of the cart in the vertical direction.

However, in the wheel-legged robot model, its driving mechanism is no longer a cart but two wheels directly connected to the link. So the cart driving the pendulum should be replaced by a motor directly connected to the pendulum, as shown in figure 4 [6].

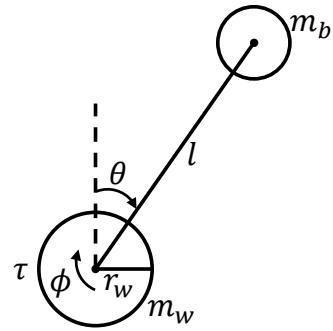


Fig. 4: The modified IPC model [6]

As the cart is replaced by a motor, the dynamic equations need to be modified. According to the research results of Huang Jian et al. [7], the dynamic equations of the modified model are given by

$$\begin{cases} m_{11}\ddot{\phi} + m_{12}\cos(\theta)\ddot{\theta} = m_{12}\sin(\theta)\dot{\theta}^2 + \tau \\ m_{12}\cos(\theta)\ddot{\phi} + m_{22}\ddot{\theta} = G\sin(\theta) - \tau \end{cases} \quad (1)$$

where

$$m_{11} = (m_b + m_w)r_w^2 + I_w, \quad m_{12} = m_b l r_w$$

$$m_{22} = m_b l^2 + I_b, \quad G = m_b g l.$$

In detail,  $\phi$  and  $\dot{\phi}$  represent the rotational angle and angular velocity of the wheel, respectively.  $\theta$  denotes the pitch angle of the robot, and  $\tau$  signifies the torque applied to the wheel. The masses of the float-based body and the wheel are denoted by  $m_b$  and  $m_w$ , respectively. Similarly, the moments of inertia of the float-based body (main body) and the wheel are represented by  $I_b$  and  $I_w$ , respectively. The radius of the wheel is  $r_w$ ,  $l$  is the height of the body and  $\dot{x} = \dot{\phi}r_w$  denotes the linear velocity of the robot. To build the state space of the robot, defining [6]

$$\mathbf{x} = [\theta \quad \dot{\theta} \quad \dot{x} - \dot{x}_d]^T,$$

where  $\dot{x}_d$  is the reference velocity.

Linearizing Eq. 1 at  $\theta = 0$  yields

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{B}\tau \quad (2)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{Gm_{11}}{m_{11}m_{22}-m_{12}^2} & 0 & 0 \\ -\frac{r_wGm_{12}}{m_{11}m_{22}-m_{12}^2} & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{m_{22}+m_{11}}{m_{11}m_{22}-m_{12}^2} \\ \frac{(m_{22}+m_{11})r_w}{m_{11}m_{22}-m_{12}^2} \end{bmatrix}$$

### B. Five-link robotic manipulator

For the wheel-legged robot model, the mass end and the driving wheel at the bottom are not directly connected by a single link but instead use a five-link robotic manipulator. Due to the legged structure, the link length  $l$  of the wheel-legged robot varies, making  $l$  an unknown variable. However, the rotation angles of the joint motors are known. Therefore, to establish the relationship between the joint motor rotation angles and the link length  $l$ , we need to model the leg structure (the five-link robotic manipulator).

In the five-link robotic manipulator model, the two joint motors can be used as the actuator, and the rotation of the two joint motors will drive the end-effector. The simplified diagram of this system is shown in figure 5, where the  $q_1$  and  $q_4$  are respectively the rotation angles of the two joint motors,  $q_2$  and  $q_3$  are the angles of the driven joints in the middle of the connecting link, and  $L_0$  and  $q_0$  are respectively the distance and angle of the end-effector to the center of the base at point O, which can be understood as the polar coordinates. The  $l_e$  is a vector from the position of the joint motor 1 to the position of the end-effector, which is used for the inverse kinematic analysis of the robotic manipulator.

1) *Forward dynamic:* By analyzing the geometric relationship between the left and right sides of the end-effector [8], we can have:

$$\begin{cases} x_B + l_2 \cos(q_2) = x_D + l_3 \cos(q_3) \\ y_B + l_2 \sin(q_2) = y_D + l_3 \sin(q_3) \end{cases} \quad (3)$$

where  $x_B$  is the horizontal coordinate of point B in the figure 5, and  $y_B$  is the vertical coordinate of point B.  $x_D$  and  $y_D$  are similar.

When eliminating  $q_3$ , we can get the expression for  $q_2$ :

$$q_2 = 2 \tan^{-1} \left( \frac{B_0 + \sqrt{A_0^2 + B_0^2 - C_0^2}}{A_0 + C_0} \right) \quad (4)$$

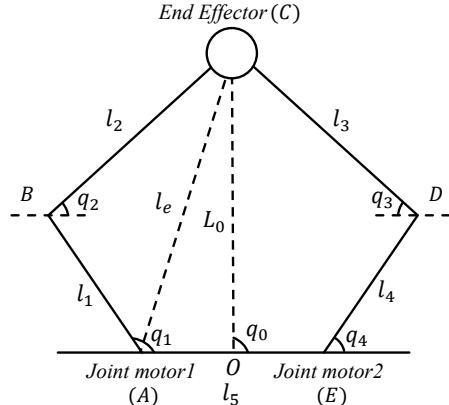


Fig. 5: The 5-link robotic manipulator

where

$$\begin{cases} A_0 = 2l_2(x_D - x_B) \\ B_0 = 2l_2(y_D - y_B) \\ C_0 = l_2^2 + l_{BD}^2 - l_3^2 \\ l_{BD}^2 = (x_D - x_B)^2 + (y_D - y_B)^2 \end{cases} \quad (5)$$

Similarly, we can have  $q_3$ , which is given by

$$q_3 = 2 \tan^{-1} \left( \frac{B_1 + \sqrt{A_1^2 + B_1^2 - C_1^2}}{A_1 + C_1} \right) \quad (6)$$

where

$$\begin{cases} A_1 = 2l_3(x_B - x_D) \\ B_1 = 2l_3(y_B - y_D) \\ C_1 = l_3^2 + l_{BD}^2 - l_2^2 \\ l_{BD}^2 = (x_D - x_B)^2 + (y_D - y_B)^2 \end{cases} \quad (7)$$

Then the coordinates of the end-effector ( $x_c, y_c$ ) can be obtained from  $q_1$  and  $q_2$  (or  $q_3$  and  $q_4$ ):

$$\begin{cases} x_c = -\frac{l_5}{2} + l_1 \cos(q_1) + l_2 \cos(q_2) \\ y_c = l_1 \sin(q_1) + l_2 \sin(q_2) \end{cases} \quad (8)$$

or

$$\begin{cases} x_c = \frac{l_5}{2} + l_4 \cos(q_4) + l_3 \cos(q_3) \\ y_c = l_4 \sin(q_4) + l_3 \sin(q_3) \end{cases} \quad (9)$$

Then we can get  $L_0$  and  $q_0$  from  $x_c$  and  $y_c$ :

$$\begin{cases} q_0 = \tan^{-1} \left( \frac{y_c}{x_c} \right) \\ L_0 = \sqrt{x_c^2 + y_c^2} \end{cases} \quad (10)$$

Two joint motors are installed on each side of the robot's main body. In the five-link robotic manipulator model, these motors serve as the driving components, and their rotation ( $q_1$  and  $q_4$ ) drives the end-effector. The position of the end-effector ( $x_c$  and  $y_c$ ) corresponds to the position of the driving wheel. According to Eq. 3 - Eq. 10, when  $q_1$  and  $q_4$  are known servo angles,  $q_2$  and  $q_3$  can be derived from  $q_1$  and  $q_4$ , then the leg length and angle ( $L_0$  and  $q_0$ ) can be obtained. Therefore, the relative position of the main body and the chassis driving

wheel can be determined by calculating the relative position between the base of the five-link robotic manipulator and the end-effector though using the forward dynamic equations.

2) *Inverse dynamic*: To actively reduce the angular deviation of the main body by adjusting the leg angle  $q_0$ . Therefore, we need to calculate the target angles of the joint motors based on the leg angle  $q_0$ , which requires inverse kinematics modeling of the five-link robotic manipulator.

The coordinate equation of the end-effector is easily obtained by means of geometric relations which is given by

$$\vec{l}_e = \vec{l}_1 + \vec{l}_2 \quad (11)$$

From this vector relationship, Eq. 12 can be obtained

$$\begin{cases} X_e^2 = l_1^2 \cos^2 \varphi_{51} + 2l_1l_2 \cos \varphi_{51} \cos(\varphi_{51} + \varphi_{12}) \\ \quad + l_2^2 \cos^2(\varphi_{51} + \varphi_{12}) \\ Y_e^2 = l_1^2 \sin^2 \varphi_{51} + 2l_1l_2 \sin \varphi_{51} \sin(\varphi_{51} + \varphi_{12}) \\ \quad + l_2^2 \sin^2(\varphi_{51} + \varphi_{12}) \\ X_e^2 + Y_e^2 = l_1^2 + 2l_1l_2 \cos \varphi_{12} + l_2^2 \end{cases} \quad (12)$$

where  $X_e$  is the horizontal coordinate of the end-effector, and  $Y_e$  is the vertical coordinate of the end-effector.  $\varphi_{51}$  is the angle between link  $l_5$  and link  $l_1$ , with the rest of the expressions having similar meanings. So we can get the angle data by

$$\begin{cases} \cos \varphi_{51} = \frac{l_2 Y_e \sin \varphi_{12} + X_e(l_2 \cos \varphi_{12} + l_1)}{(l_2 \cos \varphi_{12} + l_1)^2 + l_2^2 \sin^2 \varphi_{12}} \\ \cos \varphi_{12} = \frac{X_e^2 + Y_e^2 - l_1^2 - l_2^2}{2l_1l_2} \end{cases} \quad (13)$$

Similarly, the vector  $\vec{l}_e$  has a vector relationship represented by Eq. 14.

$$\vec{l}_e = \vec{l}_5 - \vec{l}_4 - \vec{l}_3 \quad (14)$$

The angular data expressed in Eq. 15 can be obtained by Eq. 14. The inverse kinematics analysis is complete. After the robot has calculated the desired end-effector position, the angle of the joint motor can be obtained through inverse kinematics.

$$\begin{cases} \cos \varphi_{34} = \frac{(X_e - l_5)^2 + Y_e^2 - l_3^2 - l_4^2}{2l_3l_4} \\ \cos \varphi_{45} = \frac{l_5 + l_3 \sin \varphi_{45} \sin \varphi_{34} - X_e}{l_3 \cos \varphi_{34} + l_4} \end{cases} \quad (15)$$

Eq. 15 is an implicit equation, and we can solve it using *solve()* function in MATLAB and obtain a C language expression for real-time computation.

Through the inverse kinematics equations, we can calculate the required servo rotation angles  $\varphi_{51}$  and  $\varphi_{45}$  based on the desired leg angle deviation  $q_0$ .

#### IV. CONTROL STRATEGY

##### A. Main controller design

The robot is controlled by a Proportional-Integral-Derivative (PID) controller which is a widely used control strategy that adjusts system output by considering the current error (proportional), the accumulation of past errors (integral), and the prediction of future errors based on the error's rate of change (derivative). The proportional term addresses the present error,

the integral term eliminates steady-state error, and the derivative term reduces overshoot and oscillations. By combining these three elements, a PID controller provides precise and stable control, making it effective in various applications, such as temperature regulation, motor speed control, and robotic motion.

This study implements a cascade PID controller to achieve balance control and movement control for the robot, comprising speed, angle and movement controllers. The inner loop controller is the speed controller, it is responsible for regulating the wheel motor speed by comparing the desired speed with the actual motor speed, which is obtained via encoders in the physical robot. Meanwhile, the outer loop controller is the angle controller, it governs the robot's angle by adjusting the wheel motor speed based on the target angle and real-time angle data which is acquired from the IMU in the physical robot. In addition, the robot must not only be able to restore balance when subjected to external disturbances but also return to its original position after being subjected to the displacement caused by the disturbances. Therefore, an additional movement loop controller was designed to control the robot's displacement. The movement commands issued by the user will be applied to the movement loop controller, which will then control the robot's movement accordingly. The comprehensive system architecture is depicted in figure 6.

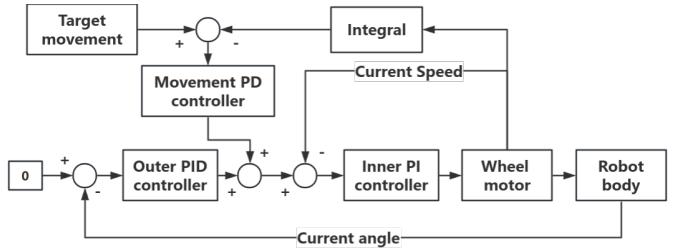


Fig. 6: The Cascade PID control

For the robot balance control, the wheel-legged robot operates at an unstable equilibrium point, making the robot prone to tipping under gravitational forces. Maintaining balance requires continuous and accurate adjustments to the motor actuation. A Proportional-Integral (PI) control strategy is employed to counteract these destabilizing forces and maintain balance. However, the robot may also be subjected to random external disturbances, such as uneven terrain or external forces. To mitigate these disturbances, the pitch angle control system must exhibit rapid response times with minimal overshoot to ensure stability. Excessive overshoot can compromise balance, thus necessitating the use of a PID controller, where the derivative component plays a crucial role in suppressing overshoot. Additionally, the tolerance for overshoot in the robot's movement control is much lower than for slow response (it is appropriate to sacrifice some response speed to achieve smoother movement). Therefore, a PD controller is used as the movement loop controller for the robot. This hierarchical control structure is essential for achieving accurate balance control.

### B. Adaptive control

According to Eq. 1, there is a mathematical relationship between the torque  $\tau$  required for the robot to maintain balance and the link length  $l$ . Therefore, when  $l$  changes, the PID parameters also need to be adjusted. By detecting the robot's height through the forward kinematics, the parameters of outer loops of the cascade PID controller is adjusted dynamically, ensuring stable control of the robot at different heights. To achieve this, the system is linearized at every 10mm interval within the leg length range, and the relatively good parameter gain matrix  $K = [K_p, K_i, K_d]$  for PID controller is calculated [9]. The gain matrix  $K$  calculated for different leg lengths is fitted to a polynomial function, resulting in a mathematical relationship between the gain matrix  $K$  and the leg length, denoted as  $K(L)$ . The polynomial equation that fits each element of the matrix as the leg length  $L = L_0 + L_M$  changes is given by:

$$K_i(L) = p_{0|i} + p_{1|i}L + p_{2|i}L^2 + p_{3|i}L^3 \quad (16)$$

where  $p_{j|i}$  is the constant obtained through polynomial fitting.

When controlling the robot, the leg length can be determined using the forward kinematics of the five-link manipulator, and the corresponding gain matrix  $K$  can then be calculated to achieve adaptive control under varying leg lengths.

Because the robot may operate under varying load conditions, and the robot may experience faults that lead to changes in its mechanical structure, its center of gravity can shift. To manage this issue, the Center of Gravity Adaptive Algorithm is implemented to help the robot maintain balance. When the robot is at the equilibrium point, the rate of change in its angle is minimal under identical conditions. Utilizing this characteristic, the error between the current angle and the equilibrium point is integrated numerically for each operational period and then applied to the equilibrium point adjustment, as described by

$$EP_k = EP_{k-1} - e_k \cdot K_i \quad (17)$$

where the  $EP_k$  is the equilibrium point at the  $k$ th period. The  $e_k$  is the error between the current angle and equilibrium point at the  $k$ th period. The  $K_i$  is a user-defined gain.

### C. Wheel-leg coordination control for balancing

The angle between the robot's leg and its main body is adjustable, enabling feedforward control of the robot's posture. By modeling the leg's inverse kinematics which is shown in Eq. 13 and Eq. 15, we can calculate the required joint motor angles (i.e.,  $q_1$  and  $q_4$  in figure 5) based on the desired leg length and angle (i.e.,  $L_0$  and  $q_0$  in figure 5). When the robot detects an angular deviation, it responds by adjusting the leg angle in the opposite direction, thus mitigating the main body's angular deviation.

The compensation angle is determined by the current speed of the wheel motor. As the robot's angle fluctuates, the balance control loop modulates the motor speed to counteract the angle change. Simultaneously, the legs generate a corrective deviation angle based on the motor speed. This corrective

angle translates a portion of the main body's angular deviation (relative to the world coordinate system) into the leg's coordinate system, thereby improving the robot's overall stability.

Furthermore, the leg angle can be adjusted faster than the main body's. Consequently, adjusting the leg angle facilitates quicker recovery of the robot to its equilibrium point.

## V. RESULTS

To test the robot's performance more efficiently and safely, we also built a simulation model of the robot. Both the simulation model and the physical robot have been evaluated under different kinds of disturbance and faulty conditions.

### A. Simulation environment

For the simulation model, a comprehensive set of experiments designed to evaluate the performance and robustness of various control strategies implemented on the robot is presented. These experiments aim to assess the robot's ability to maintain stability, track reference movements, and adapt to different challenging scenarios. The experiments include evaluating the robot's behavior during normal standing, reference movement tracking, and leg height adjustment. Additionally, we explore the robot's response to more extreme conditions such as high-altitude falls, the addition of non-coaxial external loads, and the application of external disturbances. By systematically analyzing the outcomes of these tests, we aim to demonstrate the effectiveness and limitations of our control method under varying operational conditions. The results are shown in figure 7

**Normal Standing:** The first experiment assessed the robot's ability to maintain a stable posture while standing still. The PID controller effectively minimized the pitch angle deviation, keeping the robot upright with minimal oscillations. The results demonstrated the controller's capacity to maintain balance under static conditions, a fundamental requirement for subsequent dynamic tests. The outcome is illustrated in figure 7b.

**External Disturbances:** In the second experiment, the robot's response to an external force of 3N applied horizontally at the 2-second mark was evaluated. The PID controller's task was to resist these disturbances and maintain the robot's balance. The results showed that the controller could effectively counteract the applied forces, restoring the robot to its original position with very small overshoot. This experiment demonstrated the PID controller's robustness against unexpected external disturbance, as illustrated in figure 7d.

**Reference Movement Tracking:** In the third experiment, the PID controller was tested for its ability to track a reference movement trajectory. The robot was commanded to move along a specific path, and the controller's performance was evaluated based on its accuracy in following this path. The results indicated that the PID controller managed to track the reference trajectory closely, with only minor deviations. This demonstrates the controller's effectiveness in handling dynamic tasks and real-world applications such as precise

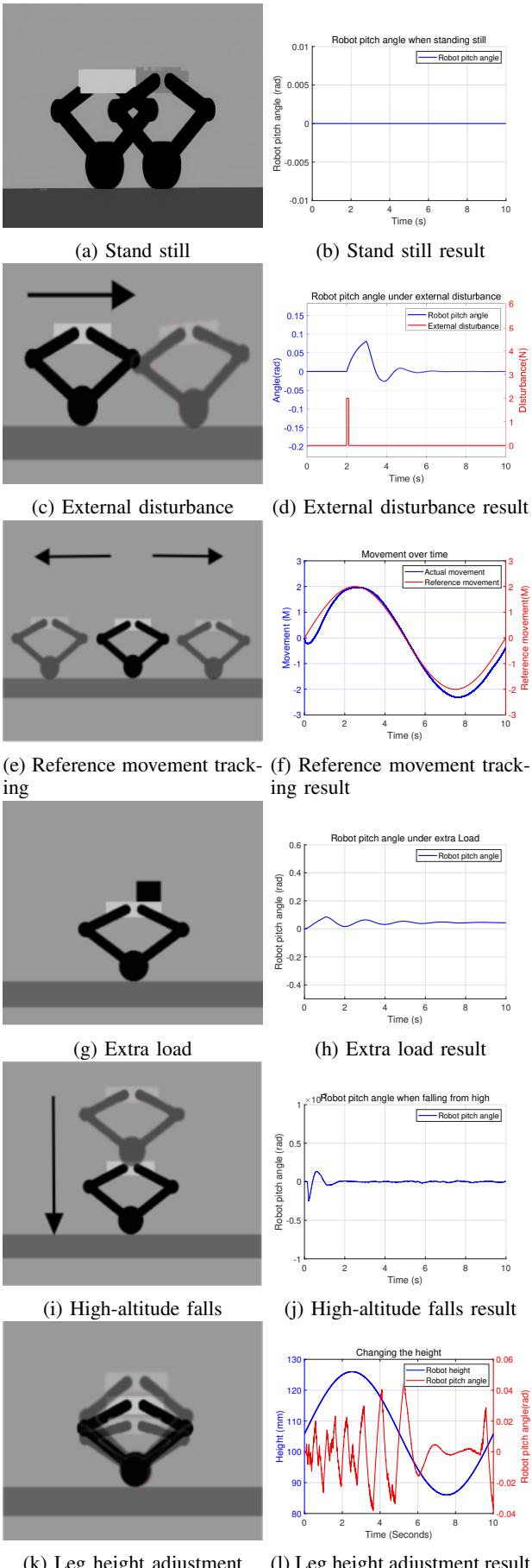


Fig. 7: The result of PID control in simulation environment

navigation in autonomous delivery robots. The results are shown in figure 7f.

**Handling Additional Non-Coaxial Loads:** In the fourth experiment, the robot was subjected to additional 0.1kg non-coaxial loads, which altered its center of gravity. It can be observed that although the robot's angle experienced slight oscillations after adding an extra load, the adaptive algorithm enabled the robot to quickly find a new equilibrium point, and successfully converge the angle to this new equilibrium point. The corresponding results are shown in figure 7h.

**Falling from a Height:** The fifth experiment simulated a scenario where the robot falls from a significant height (8cm). The focus was on observing the controller's ability to stabilize the robot quickly after impact. The PID control responded effectively by rapidly correcting the pitch angle and restoring balance within a short time frame. The controller's performance in this test underscores its robustness in handling sudden, high-impact falling disturbances, making it suitable for applications such as navigating uneven or bumpy roads. The results are displayed in figure 7j.

**Leg Height Adjustment:** The final experiment involved changing the height of the robot's legs. It can be observed that despite the use of the height adaptive algorithm, the robot's angle experienced some oscillations when its height changed. This indicates that the effectiveness of the PID controller is limited when adjusting the robot's height. The reason for this phenomenon may be that the relationship between the robot's height and the changes in PID parameters is not completely linear, which limits the effectiveness of the adaptive algorithm. The result is depicted in figure 7l.

Overall, these experiments illustrate the effectiveness and limitations of the PID control strategy in maintaining stability and handling dynamic tasks for the wheel-legged robot in various simulated conditions. Under various disturbance conditions, the control algorithm demonstrates good robustness, successfully keeping the robot's balance.

### B. Physical robot

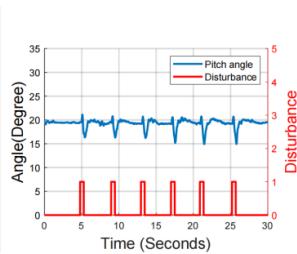
According to the simulation results, the controller has a relatively good performance then it is implemented on the physical robot. The physical robot's performance under different disturbances or faulty conditions such as cart movement, additional weight load, wind force, collisions or kicks, and varying road conditions were evaluated respectively. The results are shown in figure 8 - figure 16.

**External Disturbance:** External kick-type force is applied to the robot while maintaining balance. The results are shown in figure 8. It can be seen that the robot can still successfully recover the equilibrium point in a short time after being disturbed by the external force in the horizontal direction.

**Center of Gravity Adaptive:** An extra load is added to the robot while maintaining balance, the results are shown in figure 9. When additional load is added to one side of the robot, the pitch angle corresponding to the equilibrium point will change, and the robot will maintain the new equilibrium point. However, due to mechanical assembly deviations and

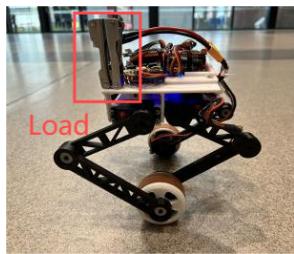


(a) Kick disturbance

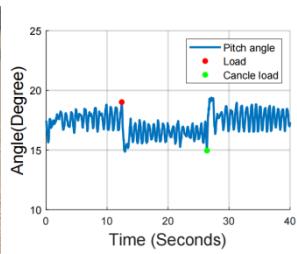


(b) Kick disturbance rejection

Fig. 8: Robot performance under kick disturbance

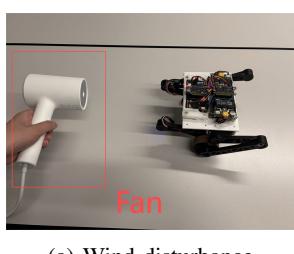


(a) Additive load

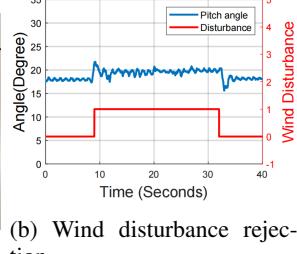


(b) Load accommodation

Fig. 9: Robot performance under load disturbance



(a) Wind disturbance

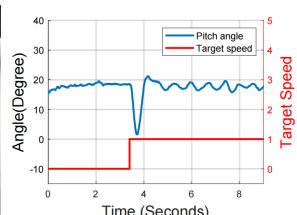


(b) Wind disturbance rejection

Fig. 10: Robot performance under wind disturbance

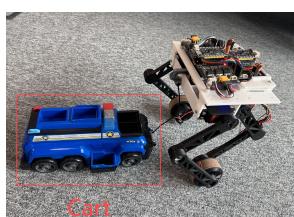


(a) Faulty wheel

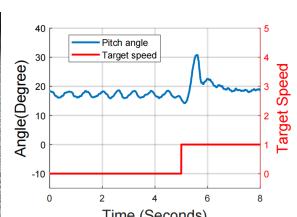


(b) Wheel fault tolerance

Fig. 11: Robot performance with additive attachment

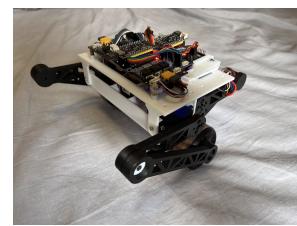


(a) Trailer load

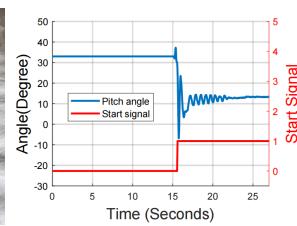


(b) Moving with trailer

Fig. 12: Robot performance under trailer load disturbance

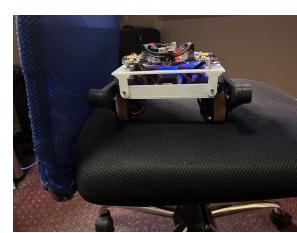


(a) Soft surface

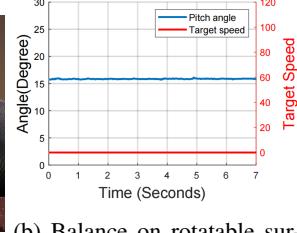


(b) Balance on soft surface

Fig. 13: Robot performance on the soft surface



(a) Rotatable surface



(b) Balance on rotatable surface

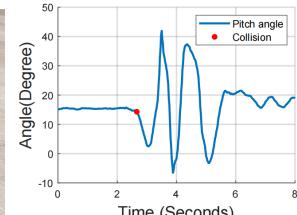
Fig. 14: Robot performance on the rotatable surface

the limitations of sensor accuracy and sampling speed, the robot will experience small oscillations near the equilibrium point. Nevertheless, these oscillations do not affect the robot's stability.

**Wind Disturbance:** The robot is affected by strong winds while maintaining balance, the results are shown in figure 10. Due to the influence of wind disturbance, the robot will generate a speed along the direction of the wind, and the pitch angle of the robot will also change. As the pitch angle changes, this speed will converge to zero, creating a new equilibrium

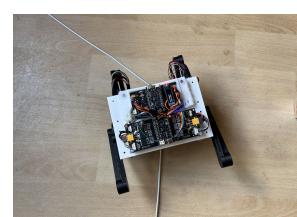


(a) Collision

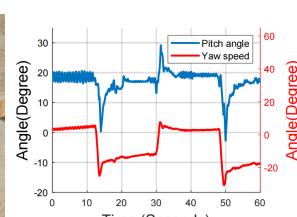


(b) Balance after collision

Fig. 15: Robot performance under collision disturbance



(a) Robot turning



(b) Turning while balancing

Fig. 16: Robot turning performance

point, and the robot is eventually able to stabilize at this new equilibrium point. Once the external wind disturbance is removed, the robot successfully returns to and stabilizes at the original equilibrium point.

**Faulty Wheel Condition:** The robot's wheels are affected by additional sticky attachments to simulate the effects of different wheel conditions on the robot, the results are shown in figure 11. When the robot remains stationary, the additive attachment on the wheels do not significantly affect its performance. However, once the robot starts moving, the pitch angle will experience larger oscillations, but this does not impact the robot's stability, and it can still maintain its balanced state.

**Trailer Load:** The robot's performance when carrying an additional cart is tested to verify the resistance of the robot to continuous disturbance in the horizontal direction, the results are shown in figure 12. After adding extra trailer load to the robot, it can find a new equilibrium point and keep in balance quickly, demonstrating the robustness to continuous external disturbances in the horizontal direction and the high response speed.

**Soft Surface:** The performance of the robot on soft surfaces (Simmons) is tested, the results are shown in figure 13. On a soft surface, the robot can still successfully stand up from a naturally fallen state and maintain balance. This demonstrates that the robot is capable of operating on soft surfaces.

**Rotatable(non-fixed) Surface:** The performance of the robot on rotatable surfaces (Swivel chair) is tested, the results are shown in figure 14. On a rotatable (non-fixed) surface, the robot can still maintain stability. Combined with its performance on soft surfaces, this demonstrates that the robot is capable of operating in various complex external environments.

**Collision:** The performance of the robot after it hits the obstacle, the results are shown in figure 15. When the robot collides with an obstacle (wall), its pitch angle will inevitably experience large oscillations. However, these oscillations do not compromise the robot's stability, and the robot can quickly converge the pitch angle back to the equilibrium point. This demonstrates the robot's ability to handle collision disturbances.

**Turning with Balance:** The robot is tested for the performance during steering (changing the Yaw axis angle), the results are shown in figure 16. When the robot rotates, its pitch angle changes along with the yaw angle, indicating that the system is coupled. However, this coupling does not make the robot unstable.

Overall, for both simulation and physical experiments, the robot performs well in maintaining stable under various external disturbances and fault conditions, as well as adapting to complex operating environments. Comprehensive result demonstrates that our control scheme can handle multiple disturbances and faults while adapting to different operating conditions.

## VI. CONCLUSION

This paper presents a bipedal wheel-legged robot with a comprehensive analysis of its modelling and controller design, and its performance under some external interference and faulty conditions. A cascaded PID control scheme coupled with the adaptive control algorithms, was implemented for the robot's balance and movement control. The robot's performance was evaluated under various disturbance and fault conditions for both the simulated and the physical model. The results demonstrate the robot's ability to effectively withstand disturbances, including external impacts, additional loads, wind forces, trailer loads, and collisions. Furthermore, the robot demonstrates robust performance under different operational environments including soft and non-fixed surfaces, and fault conditions like wheel malfunctions. Future work could consider dynamic control using advanced brushless motors instead of the current kinematic control to enhance control performance further.

## REFERENCES

- [1] M. A. Alobaidy, D. J. M. Abdul-Jabbar, and S. Z. Al-khayyt, "Faults diagnosis in robot systems: A review," *Al-Rafidain Engineering Journal (AREJ)*, vol. 25, no. 2, pp. 164–175, 2020.
- [2] V. Klemm, A. Morra, C. Salzmann, F. Tschopp, K. Bodie, L. Gulich, N. Küng, D. Mannhart, C. Pfister, M. Vierneisel, F. Weber, R. Deuber, and R. Siegwart, "Ascento: A two-wheeled jumping robot," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7515–7521.
- [3] H. Chen, B. Wang, Z. Hong, C. Shen, P. M. Wensing, and W. Zhang, "Underactuated motion planning and control for jumping with wheeled-bipedal robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 747–754, 2021.
- [4] 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)*, 2021, pp. 6782–6788.
- [5] Z. Yang and L. Zhang, "Balancing control of low-cost wheel-leg robotics for control and robotics education," in *The 29th International Conference on Automation and Computing (ICAC 2024)*, Jul. 2024.
- [6] 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.
- [7] J. Huang, Z.-H. Guan, T. Matsuno, T. Fukuda, and K. Sekiyama, "Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 750–758, 2010.
- [8] G. Rigatos and M. Abbaszadeh, "Nonlinear optimal control for a five-link parallel robotic manipulator," *Journal of Vibration and Control*, vol. 29, no. 3–4, pp. 714–735, 2023. [Online]. Available: <https://doi.org/10.1177/10775463211051449>
- [9] Y. Chen, H. Wang, and L. Zhang, "Control of wheel-legged balancing robot," *Information and Control*, vol. 52, no. 5, pp. 648–659, 2023.