# Balancing Control of Low-Cost Wheel-leg Robotics for Control and Robotics Education

**DOI:**
10.1109/ICAC61394.2024.10718853

**Document Version**
Accepted author manuscript

# Balancing Control of Low-Cost Wheel-leg Robotics for Control and Robotics Education

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

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

**Abstract—The experiments for control courses play an essential role in enriching the student learning experience in control engineering and robotics education. However, most of existing popular control facilities, such as inverted pendulum, mobile robots, manipulators and flying vehicles, are often not portable devices for flexible use, such as take-home use. In this paper, we show that it is feasible to use a low cost and portable wheel-leg robot for controller design purpose. The wheel-leg robot is reconfigurable from simple to complex systems and can be used for both introductory and advanced levels of control courses. Specifically, the fundamental control method, PID, together with advanced techniques including Kalman filter, adaptive balancing, wheel-leg coordinating control are employed for balancing control in this study. Extensive results show that the wheel-leg robot can be an exemplary platform for demonstrating the core control concepts, such as stability (via balancing control), robustness to uncertainties (e.g. body weight changes), and disturbance handling (against load changes or external forces).**

***Keywords—Wheel-legged Robot, Self-balancing, Control and Robotics education.***

## I. INTRODUCTION

The disciplines of control engineering and robotics are inherently application-oriented, with an expectation that students are able to address real-world engineering challenges upon graduation. Therefore, it is crucial for academic institutions to offer students control oriented application platforms. However, commonly available platforms like inverted pendulums, mobile robots, and flying devices are usually restricted to on-campus use with limited access. A cost-effective platform that students could use more flexibly, including at home, would greatly enhance their learning experience. Yet, the options available on the market tend to cater to applications that are stable and have a low degree of complexity due to their limited degree of freedom (DoF). In contrast, applications that are more complex, unstable, and have a high DoF are typically costly and are often for research purposes. In our work, we introduce a low-cost wheel-leg robot, featuring five links and adjustable height, designed for both control engineering and robotics applications. This development emphasizes controller design for maintaining balance control amid disturbances and adjusting to changes in the robot height.

Wheel-legged robots enjoys both advantages of wheel driven mobile vehicles and legged robots, allowing them to work in a high energy efficiency under uneven surface and obstacles. One of the major milestones of wheel-legged robots is the Boston Dynamics Handle, developed in 2019, without detailed technical information available to public. More recently, the wheel-leg robots including ETH Ascento [1], bipedal wheeled robots [2-5] and some open-source bipedal robots [6] have emerged as popular research topics but most of them are expensive and not suitable for education purpose.

In this paper, we are inspired by the open-source plan [6] and we further use low-cost four servo motors (instead of expensive brushless motors) for the leg joint motors. This made it possible for a low-cost education platform. The wheel-leg robot is reconfigurable from simple to complex systems for controller design, which provides an ideally unified platform for both entrance and advanced levels of control course application. More specifically, when only controlling the two wheels jointly without control of the legs, it is an inverted pendulum car and it can be used as a single-input and single-output control application. If the two wheels are controlled separately, it is a two-input, two-output systems. Further, if the legs are also controlled, it is a more complex multiple-input, multiple-output system with higher DoFs. Meanwhile, it can be set as a time-varying system with adjustable body heights and weighs. Third, it is a nonlinear system, and both linear and nonlinear controllers can be employed. Finally, it is an unstable system and it is also subject to different disturbances (e.g. external force) and uncertainties (e.g. loads changes due to different body weights), the central concepts of control engineering, stabilizing (also referred to as balancing here), robustness to uncertainties, and disturbance rejection, can be well tested.

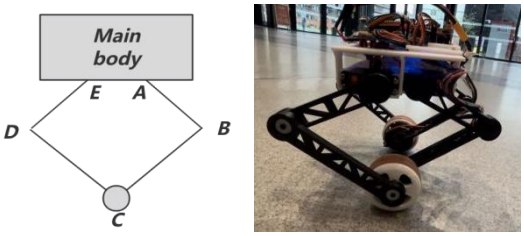
In this paper, the dynamic models of the wheel-leg robot is first introduced and then the PID cascade control is used for balance control under different height and disturbance. The Kalman filter and sliding weighted filter are used to make data smoother and more reliable. Adaptive control algorithm is developed to balance the robot under different condition. Extensive results are presented to show the effectiveness of these control methods and the wheel-leg robot potentially offers a good platform for control and robotics education.

## II. ROBOT AND DYNAMIC MODEL

### A. Robot introduction

The simplified diagram and a physical photo of the wheel-legged robot are shown in Fig 1. The wheel-legged robot comprises a main body, two legs, and two active wheels. The main body carries the hardware of the controller board, the Inertial Measurement Unit (IMU), and the joint motors. The body is suspended above the ground, supported by the two legs. Each leg of the robot resembles a five-link robotic manipulator, connected to the main body through two joint motors. The end effector of each leg is equipped with an controllable wheel, driven by a brushless motor with an encoder. The encoder provides information on the rotation speed and angle of the active wheel.

As shown in the diagram in Fig. 1, Point A and point E denote the locations of the joint motors, while point C represents the end effector of the controllable wheel. Points B and D indicate the two driven joints capable of rotation. This five-link configuration enables the position of the active wheel, situated at point C, to be determined by the rotational angles of the two motors located at points A and E. Consequently, we can manipulate the relative position between the main body and the active wheel by controlling these joint motors. Using this capability, we can achieve functionalities beyond those of conventional wheeled robots, such as roll axis compensation and center of gravity shifting. Moreover, this robot exhibits smoother movement compared to typical legged robots, resulting in higher energy efficiency and a simpler system structure.




(a)Simplified diagram (b)Physical picture

Fig. 1. The diagram of the wheel-legged robot

## B. Dynamic model

This paper mainly analyzes the dynamic model of wheel-legged robot in terms of balance control. The dynamic model of the wheel-legged robot can be decomposed into an Inverted Pendulum on a Cart (IPC) model and a five-link robotic manipulator model.

### 1) IPC model

In the IPC model, the robot's wheels represent the cart, while the main body of the robot serves as the mass on top of the inverted pendulum. The leg structure can be simplified as the link connecting the cart and the mass end of the inverted pendulum. Fig. 2 shows the simplified diagram of this system. *M* is the mass of the cart, *m* is the mass of on top 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, and *x* is the displacement of the cart in the horizontal direction.

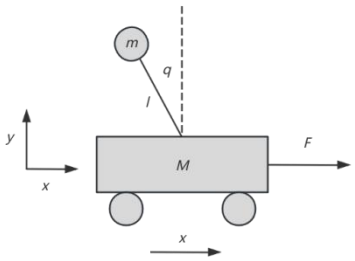



Fig. 2. The simplified diagram of IPC

The center of mass of the inverted pendulum (relative to the center ( $x, y$) of mass of the cart) is given by

$$\begin{cases} x_m = x - l\sin(q) \\ y_m = y + \frac{h_{cart}}{2} + l\cos(q) \end{cases} \tag{1}$$

Where $x_{\mathrm{m}}$ , $y_{\mathrm{m}}$ are x-coordinate and y-coordinate of the center of mass of $m$, respectively. $\mathrm{h_{cart}}$ is the height of the cart. And the speed of the center of mass of the inverted pendulum can be given by

$$\begin{cases} \dot{x_m} = \dot{x} - l\cos(q)\,\dot{q} \\ \dot{y_m} = -l\sin(q)\,\dot{q} \end{cases} \tag{2}$$

Then the Lagrangian of the system using kinetic and potential energy is given by

$$L = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}ml^2\dot{q}^2 - mlcos(q)\dot{q}\dot{x} - mlcos(q) \tag{3}$$

Using the Euler-Lagrange equation, we can derive the dynamics of the system, which is given by

$$\begin{cases} (M+m)\ddot{x}^2 + mlsin(q)\dot{q}^2 - mlcos(q)\ddot{q} = F \\ ml^2\ddot{q} - mlcos(q)\ddot{x} - mglsin(q) = 0 \end{cases} \tag{4}$$

Due to trigonometric functions, the system is nonlinear. Now we will linearize it. When $q$ is small, the linearization estimation is given by

$$sin(q) \approx q,\ cos(q) \approx 1 \tag{5}$$

So the dynamic of the system can be represented as

$$\begin{cases} (M+m)\ddot{x}^2 - ml\ddot{q} = F \\ ml^2\ddot{q} - ml\ddot{x} - mglq = 0 \end{cases} \tag{6}$$

### 2) Five-link robotic manipulator model

In the five-link robotic manipulator model, two joint motors serve as the actuators. The rotation of these joint motors drives the end effector. A simplified diagram of this system is shown in Fig. 3, where $q_1$and $q_4$ are 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 the distance and angle of the end effector to the center of the base at point *O*, respectively, which can be understood as the polar coordinates.

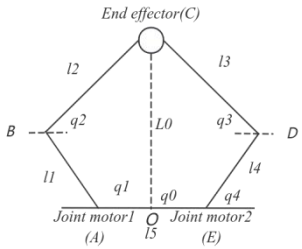



Fig. 3. The five-link robotic manipulator

By analyzing the geometric relationship between the left and right sides of the end-effector[9], we can have

$$\begin{cases} x_B + l_2cos(q_2) = x_D + l_3cos(q_3) \\ y_B + l_2sin(q_2) = y_D + l_3sin(q_3) \end{cases} \tag{7}$$

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

$$q_2 = 2tan^{-1}\left(\frac{B_0 + \sqrt{{A_0}^2 + {B_0}^2 - {C_0}^2}}{A_0 + C_0}\right) \tag{8}$$

where

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

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

$$q_3 = 2tan^{-1}\left(\frac{B_1+\sqrt{{A_1}^2+{B_1}^2-{C_1}^2}}{A_1+C_1}\right) \quad (9)$$

where

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

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

$$\begin{cases} x_c = -\frac{l_5}{2} + l_1 cos(q_1) + l_2 cos(q_2) \\ y_c = l_1 sin(q_1) + l_2 sin(q_2) \end{cases} \quad (10)$$

$$\begin{cases} x_c = \frac{l_5}{2} + l_4 cos(q_4) + l_3 cos(q_3) \\ y_c = l_4 cos(q_4) + l_3 cos(q_3) \end{cases} \quad (11)$$

Then we can get $L_0$ and $q_0$, from $x_c$ and $y_c$.

$$\begin{cases} q_0 = tan^{-1}\left(\frac{y_c}{x_c}\right) \\ L_0 = \sqrt{{x_c}^2 + {y_c}^2} \end{cases} \quad (12)$$

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 drives the end effector. The position of the end effector corresponds to the position of the driving wheel. 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.

## III. Control Algorithms

### A. Cascade PID control

We employ the classical cascade PID control scheme for the robot's balance control, which includes inner and outer loop controllers. The inner loop regulates the wheel motor speed by comparing the target speed with the current motor speed obtained from encoders. The outer loop manages the robot's angle by using target and real-time angle data from the IMU to control the wheel motor speed. The overall structure of the system is shown in Fig. 4. The wheel-leg robot is inherently unstable and can fall over due to its body's gravitational force. Therefore, motor control is essential for maintaining balance against this force. We use Proportional Integral (PI) control to achieve this balance. However, the robot may also encounter random external disturbances, such as uneven road surfaces or external forces. To reject these disturbances, the pitch angle control must respond rapidly and with minimal overshoot to maintain stability. Excessive overshoot can result in a loss of balance, so we employ a PID controller, where the derivative component helps to suppress the overshoot. This hierarchical structure ensures precise balance control. In addition, two different filters are used to smooth the feedback signal, which is introduced below.

### B. Kalman filter and sliding weighted filter

The real-time data acquisition of pitch angle is required for closed-loop control. However, the collected data is subject to noise. Therefore, it may be useful to consider to filter the data before using it for feedback control. Here, the Kalman filter is used to smooth the data.

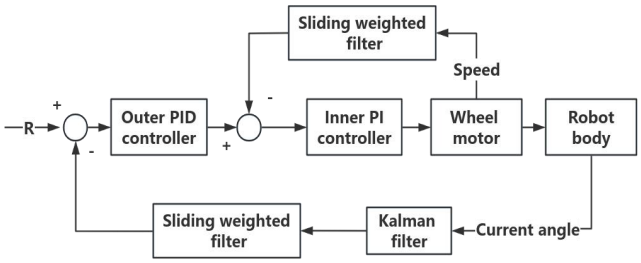



Fig. 4. Cascade PID structure

A sudden change in pitch angle or an impulse-type disturbance can result in large motor control actions due to the sensitivity of the PID controller, particularly the derivative component, which is undesirable. To mitigate these large control actions, we employ a sliding weighted filter for the pitch angle data. The sliding weighted filtering algorithm computes the mean value of data from preceding operational periods to represent the current period's data. Notably, data closer to the current period are given higher weights, ensuring both real-time data fidelity and reducing the disruptive impact of abrupt changes. A rolling filter is used to ensure that the weighted filter considers only the most recent data. This approach maintains the balance of the robot by smoothing out sudden variations in the pitch angle and minimizing excessive control actions.

Empirical testing demonstrates the efficacy of the sliding weighted filtering algorithm in mitigating the disruptive effects of abrupt data caused by pulse disturbances and user directives, thereby promoting smoother robot operations. However, it is important to note that this algorithm may introduce a trade-off in the robot's responsiveness. Therefore, adjusting the filtering length and weighting scheme to suit specific operational requirements is necessary.

### C. Adaptive balance

Since the robot's height is adjustable and it may operate under varying load conditions, its center of gravity can change. To address this, we implemented the Center of Gravity Adaptive Algorithm to help the robot maintain balance. As per Eq. 6, the external force required by the robot to maintain equilibrium is influenced by the length of the link *l*. An increase in *l* leads to a corresponding increase in the force needed to counteract the main body's angle deviation. Consequently, the coefficient of the outer loop of the balance controller is directly proportional to the robot's height. Therefore, by detecting the robot's height, we dynamically adjust the parameters of the inner and outer loops of the cascade PID controller to ensure stable control of the robot at different heights.

When the robot is at the equilibrium point, its angle changes at the slowest rate under the same conditions. Leveraging this characteristic, we numerically integrate the error between the current angle and the equilibrium point for

each operational period and apply it to the equilibrium point adjustment, it is given by

$$EP_k = EP_{k-1} - e_k * K_i \quad (13)$$

where the $EP_k$ is the equilibrium point at $k_{th}$ period. The $e_k$ is the error between the current angle and equilibrium point at $k_{th}$ period. The $K_i$ is a user defined gain. Through rigorous testing, the algorithm has proven effective in dynamically recalibrating the equilibrium point and facilitating balance maintenance even under load.

### *D. Wheel-leg coordination control for balancing*

The angle between the robot's leg and its main body is adjustable, allowing for feedforward control of the robot's angle. By modeling the forward kinematics of the leg, we can control the polar coordinates of the end actuator of the five-link manipulator through servo angle adjustments. When the robot detects a change in its angle, it adjusts the leg's polar angle in the opposite direction to counteract the change, thus reducing the angle deviation of the main body.

The compensation angle is determined by the current speed of the robot's wheel motor. As the robot's angle fluctuates, the balance control loop adjusts the motor speed to offset the angle change. Simultaneously, the legs generate a deviation angle based on the motor speed. This deviation angle effectively translates a portion of the angle deviation from the main body's coordinate system relative to the world coordinate system into the main body's coordinate system relative to the robot leg coordinate system, enhancing the robot's stability.

Additionally, the leg's angle can be adjusted more rapidly than the main body's. Therefore, modifying the leg's angle allows for quicker restoration of the robot to its equilibrium point.

## IV. EXPERIMENTS RESULTS

### *A. Robot introduction and balancing performance*

The overall height of the robot can vary between 15 cm and 30 cm. The 8120MG servo is used for the joint motor, and the PM3010 brushless motor is used for the wheel motor. The robot's hardware controller consists of two ESP32 microcontroller units (MCUs), which have powerful floating-point operation capabilities and Bluetooth communication, allowing users to remotely control the robot via mobile phones.

The robot has two controllers: an arithmetic controller and a drive controller. These controllers exchange data through serial ports. The operation period of the robot is set to 1 ms. In each period, the controller reads sensor data, calculates the target output, and drives the actuators using time-slice rotation. The arithmetic controller handles the reading of angle sensor data, driving the servo, and calculating the target output. The drive controller is responsible for reading encoder data and driving the wheel brushless motor. Six different experiments were conducted on the robot, and the pictures of these experiments are shown in Fig. 5.

1. Initial status. The robot lays down on the ground.

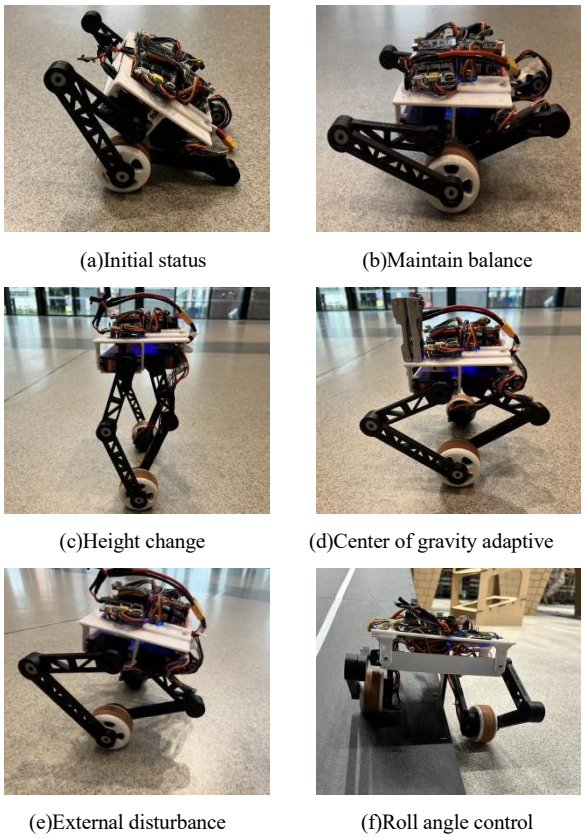
2. Maintain balance. The robot is balanced at minimum height (5cm).

3. Height change. The robot keeps standing still while changing the height of its legs to adjust height from 5cm to 20cm.

4. Center of gravity change. Add extra load to the robot while maintaining balance.

5. External disturbance. External kick type force is applied to the robot while maintaining balance.

6. Roll angle control. Change the height difference between two motors of the two wheels of the robot, and meanwhile maintain the balance of the roll angle. A demonstration video has been created and uploaded to the YouTube platform, showcasing the performance of our designed robot in different scenarios. Demonstration video link: https://youtu.be/DnpVfhcj2kk.



(a)Initial status (b)Maintain balance

(c)Height change (d)Center of gravity adaptive

(e)External disturbance (f)Roll angle control

Fig. 5. Experiments results

### *B. Kalman filter*

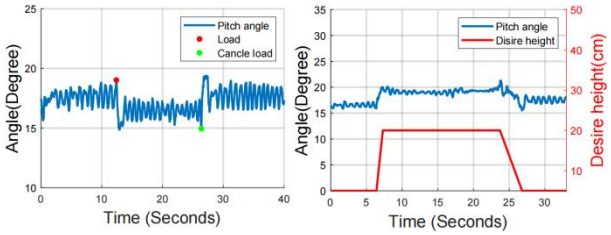
A comparative analysis of the robot's pitch angle control is conducted both with and without the integration of the Kalman filter. Random inputs are introduced to the robot, and subsequent changes in the pitch angle are recorded. Following several experimental trials, the variances of the angle data under both conditions are computed. The resultant dataset presented in Table 1 vividly illustrates the efficacy of the Kalman filter in reducing data variance and engendering smoother data dynamics.

***Table 1. Data variance comparison (move to results)***

| Test | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Filtered** | 13.63 | 57.35 | 42.76 | 72.32 |
| **Unfiltered** | 14.06 | 59.01 | 44.75 | 73.83 |

### C. *Adaptive balance*

When the height of the robot changes, the pitch angle trend is shown in Fig. 6. At the red marker, a small object weighing approximately 100g (about 10% of the robot's weight) is placed on one side, shifting the robot's center of gravity. At the green marker, the object is removed, returning the center of gravity to its original position. The figure illustrates that even with the center of gravity shift, the robot can find a new equilibrium point and remain stable. Additionally, during height adjustments, the robot's balance angle changes, but it still maintains overall balance.

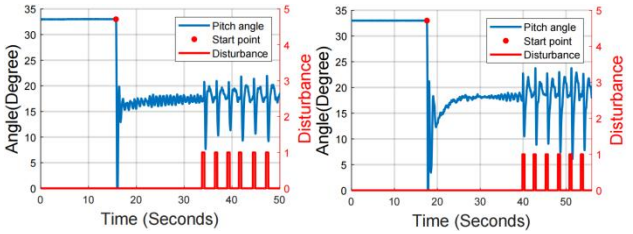



(a) Center of gravity adaptive (b) Height adaptive

Fig. 6. Adaptive balance

### D. *Wheel-leg coordination balance*

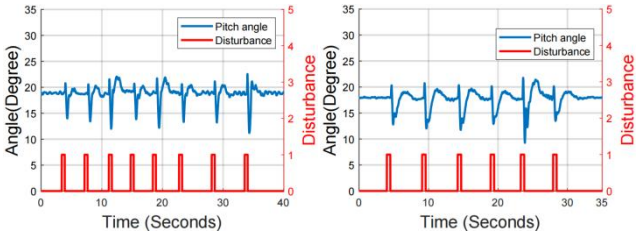
The same disturbance is applied to the robot at different heights, comparing the changes in the robot's pitch angle with and without wheel-leg cooperative control. At the lowest height, the robot starts from a static state and is subjected to disturbance forces of fixed direction and intensity. The trends of the robot's pitch angle, with and without wheel-leg coordination control, are shown in Fig. 7. Since the disturbance is not measured directly, the indicated disturbance signals represent the times when the disturbance forces are applied and are for reference only.




(a) With wheel-leg coordination (b) Without wheel-leg coordination

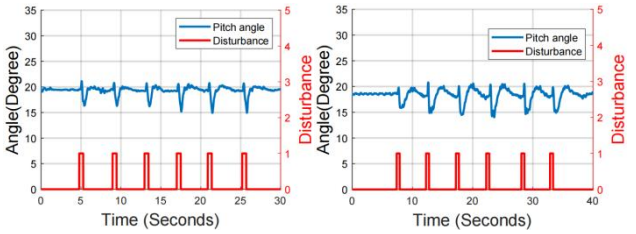
Fig. 7. Lowest height balance control

At medium and maximum heights, the trends of the robot's pitch angle with and without wheel-leg coordination control are shown in Fig. 8 and Fig. 9, respectively. By comparison, we can observe that with the addition of wheel-leg coordination control, there are improvements in start-up speed, balance stability, and recovery speed when the robot is disturbed. These enhancements are evident to varying degrees across different heights.




(a) With wheel-leg coordination (b) Without wheel-leg coordination

Fig. 8. Medium height balance control




(a)With wheel-leg coordination (b)Without wheel-leg coordination

Fig. 9. Highest height balance control

## V. Conclusion

This paper presents a control scheme of a low-cost wheel-legged robot. The wheel-legged robot uses four low-cost servo motors as leg joint motors and two brushless motors as two wheel actuators and they are used to control the wheel and leg simultaneously. The robot is controlled by cascade PID, equipped with adaptive control algorithm and wheel-leg coordination control algorithm, it can run stably. The extensive results demonstrate the low-cost wheel-leg system can be used a powerful control and robotic education platform for controller design.As the platform is a nonlinear, unstable and reconfigurable, more complex controllers, such as nonlinear, optimal, H2, H-infinity, Fuzzy will be employed in the future work.

## VI. Acknowledgments

We would like to acknowledge Guoxuan Hu for his support in hardware implementation and guidance.

## References

[1] Klemm, Victor, et al. "Ascento: A two-wheeled jumping robot." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

[2] Wang, Shuai, et al. "Balance control of a novel wheel-legged robot: Design and experiments." 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.

[3] Feng, Xujiong, et al. "Research on wheel-legged robot based on LQR and ADRC." Scientific reports 13.1 (2023): 15122.

[4] Liu, Xuefei, et al. "Development of Wheel-Legged Biped Robots: A Review." Journal of Bionic Engineering (2024): 1-28

[5] Cui, Leilei et al. “Learning-Based Balance Control of Wheel-Legged Robots.” IEEE robotics and automation letters 6.4 (2021): 7667–7674. Web.

[6] Skythinker616. (2023, July, 11). FOC Wheel-legged Robot. [Online]. GitHub. Available: https://github.com/Skythinker616/foc-wheel-legged-robot.git

[7] GuoXuan Hu. (2022). Hyun. [Online]. GitHub. Available: https://github.com/HuGuoXuang/Hyun

[8] Junheng Li. (2023, June, 27). Cart_Pole_Simulink. [Online]. GitHub. Available: https://github.com/junhengl/Cart_Pole_Simulink.git

[9] Rigatos, G. and Abbaszadeh, M. (2023). Nonlinear optimal control for a five-link parallel robotic manipulator. Journal of vibration and control, 29(3–4), pp.714–735.