Design and Manufacturing of a Rover with Transformable Wheel

Guanyu Xu, Jiawen Li, Yimin Wang, and Haobo Fang

***Abstract*—The development of autonomous vehicle systems necessitates innovative designs to navigate diverse terrains and overcome obstacles efficiently. This research focuses on the creation of a transformable wheel mechanism to enhance the adaptability and functionality of autonomous vehicles on both paved roads and sandy terrains. The primary objective is to design a wheel that can transition seamlessly between different modes, such as wheeled locomotion for paved surfaces and legged locomotion for sandy environments, thereby improving maneuverability in varied environments. To achieve this, we employed a special mechanism incorporating electromagnetic clutches and a single motor to handle both driving and transformation tasks. This design not only simplifies the control strategy but also reduces the overall weight and energy consumption. The wheel components were manufactured using SLA 3D printing for high precision and complex geometries, while the car body parts were fabricated using laser cutting of acrylic board. Lathe turning was utilized for specially designed cylindrical components, ensuring accurate dimensions and surface finish. The results demonstrate that the transformable wheel mechanism provides smooth transitions between modes, enhancing the vehicle's ability to navigate obstacles and varying terrains. The use of a Proportional-Integral-Derivative (PID) control algorithm in both straight-line and turning maneuvers ensures precise navigation and robust handling of dynamic environments. Our device weighing 1 kg with a shrinkage-to-expansion ratio of 1.5 can roll at a speed of 5 cm/s on the paved road. This research contributes to the field by offering a cost-effective and practical solution for multimodal mobility in autonomous vehicles, addressing common issues such as excessive wear and mechanical jamming through meticulous engineering and design considerations.**

***Index Terms*—Autonomous, Electromagnetic Clutches, Lightweight, Multimodal Mobility, PID Control Systems, Transformable wheel**

I. Introduction

THE development of autonomous vehicles capable of navigating complex terrains is crucial for exploring remote and inaccessible regions such as the moon, deserts, or areas beneath collapsed buildings. These vehicles are essential for data collection, exploration, and rescue operations in challenging environments. The robot must be equipped with advanced obstacle detection and navigation systems to adapt to unforeseen situations while maintaining stability and efficiency across various surfaces. Potential applications for such autonomous vehicles include extraterrestrial exploration, disaster response, and environmental monitoring, where safe and efficient access to remote or hazardous areas is essential.

This work is supported by VM3500J, UM-JI, Shanghai Jiao Tong University
Guanyu Xu, UM-JI, SJTU (email: xuguanyu@sjtu.edu.cn)
Jiawen Li, UM-JI, SJTU (email: david_lee@sjtu.edu.cn)
Yimin Wang, UM-JI, SJTU (email: wangyimin_claire@sjtu.edu.cn)
Haobo Fang, UM-JI, SJTU (email: 18297766606@sjtu.edu.cn)

Recent years have witnessed numerous designs of robots to help humans access unreachable places. For instance, the Laboratory of Intelligent Systems at EPFL proposed a miniature 7-gram jumping robot capable of overcoming obstacles more than 27 times its size, although its adaptability on various terrains remains uncertain [1]. K. Tadakuma et al. introduced a wheel-leg hybrid robot inspired by the armadillo's retractable configuration, which enhances obstacle-climbing ability but still requires optimization of the module mechanism and integration of more sensors [2]. Another innovative design is the wheel-leg hexapod robot developed at the University of California, which exhibits high-speed locomotion and agile turning but faces challenges with turning smoothness and parameter mapping [3]. The IMPASS robot showcases adaptability to different environments by changing topological structures and confirms theoretical models through experiments, yet requires further exploration into kinematics and motion planning [4]. Additionally, a deformable wheel robot utilizing a waterbomb origami pattern was developed, offering minimal mechanical parts but facing issues with deformed diameter and asymmetry [5]. Researchers at Seoul National University invented a soft-material-based deformable-wheel robot using Shape Memory Alloy (SMA) spring actuators for wheel deformation, providing high environmental adaptability but encountering complexities in manufacturing and control [6]. The Wheel Transformer, also from Seoul National University, integrates circular wheel stability with obstacle-climbing capabilities but struggles with synchronizing wheel transformations during climbing [7]. At Ohio State University, a transformable wheel robot with a passive leg was invented, offering benefits of wheeled and legged locomotion but needing improvements in long-term stability and control algorithms [8]. A 3D-printed soft pneumatic actuator with customizable bending modalities from the National University of Singapore demonstrates potential in object manipulation and rehabilitation robotics, although it requires further research into durability and long-term performance [9].

Our work proposes a novel transformable wheel design incorporating a cam mechanism and a centralized rotational transformation system, operated by a single DC motor coupled with two electromagnetic clutches. This configuration allows the wheel to switch between a compact 'wheel mode' for navigating narrow passages and an expanded 'leg mode' for overcoming obstacles such as stairs. Leveraging 3D printing technology for fabricating intricate parts ensures both precision and cost-efficiency. Our prototype demonstrates the wheel's ability to transform dynamically while in motion, facilitated by an integrated control strategy manipulating the

electromagnetic clutches based on the desired movement. The operational efficacy and robustness of the wheel's mechanical components were validated in real-world scenarios, showcasing significant improvements in mobility and adaptability over traditional designs. This innovative approach not only enhances mobile platform functionality but also reduces complexity and potential failure points associated with multiple actuator systems.

II. Design

A. Design of the Transformable wheel

The transformable wheel (Fig. 1) mainly consists of three components: 1). Six wheel rims with arc plate and guide rod. 2). A central gear with six curved slots. 3). A wheel hub with six evenly distributed guide rail. Based on these structures, the wheel can transform between two different modes - wheel mode (where the guide rod of the wheel rim is sheathed into the guide rail of the wheel hub) and leg mode (where the guide rail expands out). The transformation between these two modes is realized by a relative rotation between the central gear and the wheel hub which is concentric with the rotation of the whole wheel.

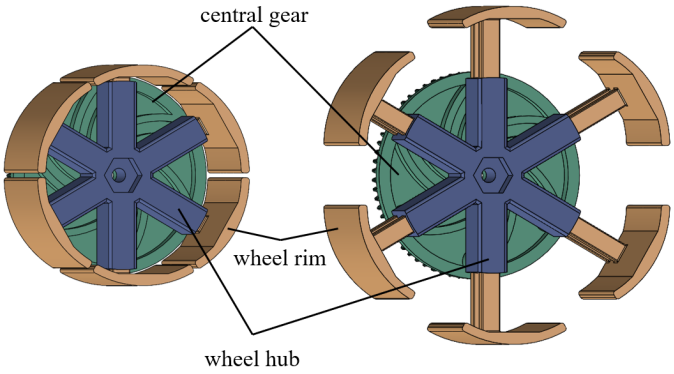


Fig. 1: Overall CAD drawing of the transformable wheel with wheel rims, central gear, and wheel hub labeled

The transformation of the wheel is based on a cam mechanism. The inside wall of the slot on central gear defines the cam profile and the pin on the wheel rim together with the guide rod and guide rail form the follower. See Fig. 2 for details. To ensure the practicability of the transformation mechanism, the slot is curved to keep the maximum pressure angle of $\phi = 22.5°$. When the central gear rotates, the guide rod will move out radially along the guide rail, enabling the wheel to transform from a minimum diameter of 76 mm to maximum diameter of 114 mm. These dimensions ensure that the vehicle can pass through a tunnel with height of 90 mm in wheel mode and climb stairs with height of 60 mm in leg mode.

Another important feature of this design is that only a single DC motor together with two electromagnetic clutches - wheel clutch and car clutch - are used to both drive the vehicle forward and transform the wheel. See Fig. 3. The output shaft of the motor is rigidly connected to the wheel hub by a hexagonal coupling, and is keyed to the inner ring of the

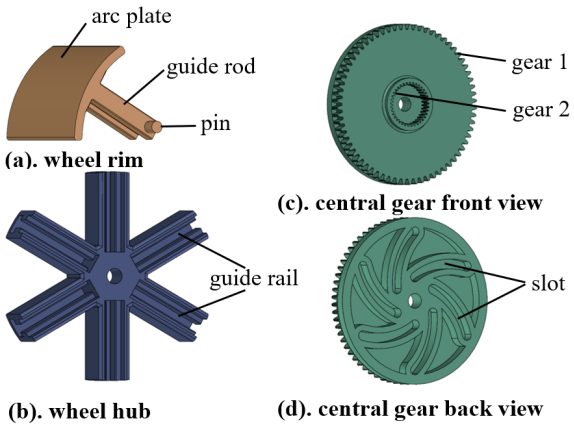


Fig. 2: Detailed CAD drawing of each components of the wheel

wheel clutch. The outer ring of the wheel clutch is rigidly attached to the central gear through a meshing with gear 2. In addition, the inner ring of the car clutch is connected to vehicle through a key on the car clutch mount. The outer ring is rigidly connected to the central gear through a meshing with gear 1. The attraction of wheel clutch prevents the relative rotation

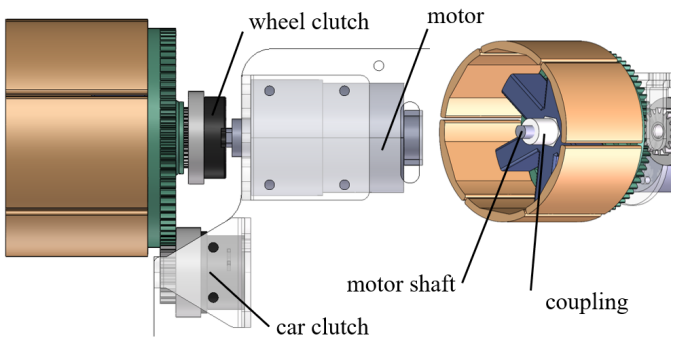


Fig. 3: The configuration of the wheel, motor, wheel clutch and car clutch

between the central gear and the wheel hub. The attraction of car clutch, on the other hand, prevents the relative rotation between central gear and the vehicle body. Therefore, when the wheel clutch is attracted and the car clutch is released, forward rotation of the motor will drive the car forward. When the wheel clutch is released and the car clutch is attracted, forward rotation of the motor will expand the wheel and backward rotation of the wheel will shrink the wheel. This design reduce the need of servo motor and replace it with clutches, which significantly reduce the cost. Table. II summarizes the behavior of motor, wheel clutch, and car clutches at four different states.

B. Originality of the Wheel Design

The creativity of the linkage design in this transformable wheel lies primarily in its unique combination of a cam mechanism with a centralized rotational transformation system. Unlike conventional transformable wheel designs that often rely on complex linkage system and multiple actuators (usually

No.	Method	Major Findings	Unresolved Issues	Refs.
L1	Others: jumping robot	The robot can overcome obstacles approximately 1.4 meters in height. The robot's leg structure allows for control over the jumping force and take-off angle.	The robot's adaptability and stability on various terrains.	[1]
L2	Wheel-leg: Integrated driving-transformation methods	Smooth transition between wheeled and legged modes; Able to move at about 155 mm/s in wheeled mode.	Needed to optimize the module's proportions, develop automated mode-switching mechanisms, and integrate sensors	[2]
L3	Wheel-leg: Integrated driving-transformation method. A six-legged robot designed with agility and efficiency.	Effective dynamic turning and actively control leg stiffness using shape memory alloy.	Needed to enhance the gait frequency and smoothness during turning	[3]
L4	Wheel-leg: Integrated driving-transformation methods. Integrated driving-transformation methods.	diverse locomotion modes, including straight-line walking, turning, obstacle climbing, and adapting to various terrains.	Kinematics and motion planning	[4]
L5	Others: Origami-wheels	Adapt to various terrains and kinematic model analysis can predict the wheel's motion	The actual deformed diameter of the wheel is larger than the theoretical value and wheel's shape is asymmetrical during deformation	[5]
L6	Others: Origami-wheels. Use of shape memory alloy (SMA) spring actuators for wheel deformation and a novel spoke structure	MHigh-speed linear movement, caterpillar-like movement through narrow gaps, and legged-wheel movement for climbing steps higher than the wheel radius	Difficulty in controlling multiple SMA actuators	[6]
L7	Wheel-leg: Integrated driving-transformation methods. Passive transformable wheel.	Achieve a speed of 2.4 body lengths per second on flat surfaces and climb over obstacles with a height of up to 130 mm.	lateral tilting of the robot body during the climbing	[7]
L8	Wheel-leg: Crank-slider mechanisms. 3D printing with ABS plastic material. Slider-crank mechanisms drive active legs, while an embedded elastic band actuates the passive leg.	The passive leg design significantly reduces the actuation force required for transformation	Not discuss the long-term stability and durability of the robot on various terrains.	[8]
L9	Others: soft pneumatic actuators.	Capable of 2D planar bending and intricate 3D helical motion. The bending angle and helical radius can be adjusted.	Durability and reliability of the actuator in practical applications and material fatigue under varying pressures	[9]
L10	Wheel-leg: Cam mechanisms. SLA 3D printing	Switch between a compact mode for narrow spaces and an expanded mode for climbing, using fewer actuators to reduce complexity and potential failures.	long-term durability of electromagnetic clutches and large torque requirement	Proposed work

TABLE I: Literature review of multimodal mobility robots.

state	car clutch	wheel clutch	motor
Forward	on	off	forward
Expansion	off	on	forward
Shrink	off	on	backward

TABLE II: Control strategy of wheel transformation using motor and two clutches

a DC motor and a servo motor) [3], [10], this design employs a single DC motor and two electromagnetic clutches to achieve both locomotion and transformation. This streamlined approach not only simplifies the mechanical structure but also enhances the reliability of the system by reducing the number of moving parts and potential points of failure. The integration of the cam mechanism, which uses a curved slot in the central gear to control the radial movement of the guide rods, ensures precise and smooth transformation between wheel and leg modes, a feature not commonly found in other designs.

Another distinctive aspect of this design is its emphasis on cost efficiency and practicability. By replacing multiple servo motors with electromagnetic clutches, the design significantly lowers production costs without compromising functionality.

The use of a single motor to handle both driving and transformation tasks further simplifies the control strategy and reduces the overall weight and energy consumption of the vehicle. Additionally, the detailed consideration of mechanical limits and operational angles, such as maintaining a maximum pressure angle of 22.5° for the cam profile, underscores the meticulous engineering behind this design. This attention to detail ensures the smooth operation and longevity of the mechanism, addressing common issues faced by other transformable wheel systems [7], such as excessive wear and mechanical jamming.

C. Pictures of prototypes and components

The pictures of prototypes and some important components are shown below.

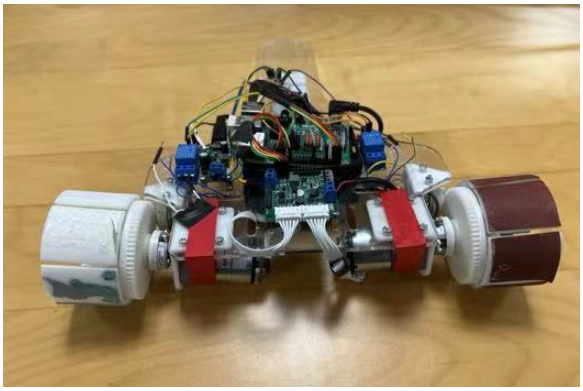
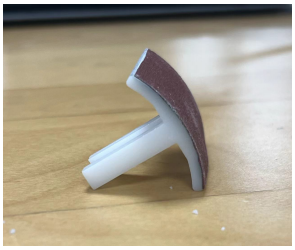
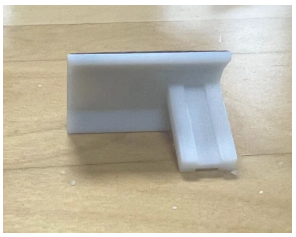


Fig. 4: The whole prototype.



(a) front view



(b) side view

Fig. 5: Picture of wheel rim.

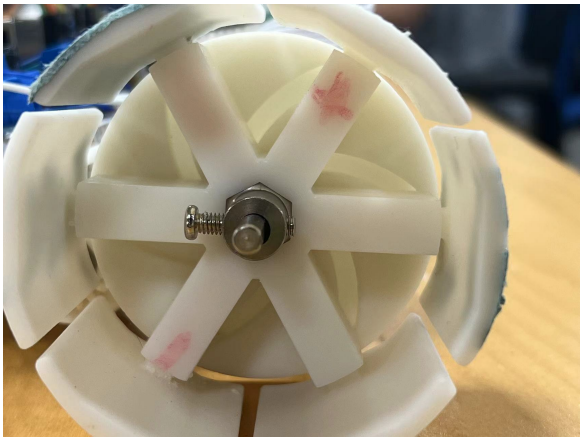


Fig. 6: The side view of the wheel.

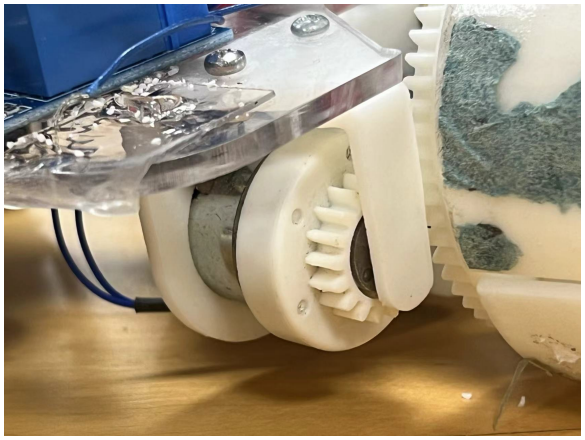


Fig. 7: The details of wheel clutch.

III. Manufacturing

A. Justification of Selection of Materials

Choosing the right materials is critical for ensuring the functionality, durability, and efficiency of the transformable wheel design. The primary considerations for material selection include strength, weight, and compatibility with manufacturing methods.

1) Part material: We selected Photosensitive Resin for the major components of the wheel (wheel rim, central gear, and wheel hub). This material, used widely in Stereolithography (SLA) 3D printing offers excellent balance between lightweight properties and strength. The material also has suitable thermal properties. It has a heat deflection temperature of 56 °C, which is adequate for our prototype that doesn't encounter high temperatures.

2) Motors and Sensors: Motors are carefully selected based on its torque and dimension of the output shaft. A unit with rated torque of 11.60 $\mathrm{Kgf \cdot cm}$ is chosen to ensure sufficient power to drive the vehicle and facilitate wheel transformation. The output shaft of the motor is 5 mm in diameter and 70 mm in length, which is compact and fits well within the design constraints.

Sensors were selected based on their ability to meet functional requirements. A 6 DOF (3 axis accelerometer and 3 axis gyroscope) inertial measurement unit (IMU) is chosen to detect the position and orientation of the vehicle [11]. These information are used to ensure the car driving through a straight as well as the judgement of the timing of expansion and shrinkage of the wheel. For recognizing the turning sign, an RGB camera is used to provide necessary visual feedback for real-time adjustments.

B. Justification of Selection of Manufacturing Methods

The choice of manufacturing methods is driven by the need for precision, availability, time-efficiency, cost, and ease of fabrication.

1) SLA 3D printing: The major components of the wheel is manufactured using SLA 3D printing. This method offers high precision and the ability to produce complex geometries, making it ideal for the intricate parts of the transformable wheel. Additionally, it is cost-effective for prototyping and small batch production.

2) Laser cutting: For the parts of the car body that mainly consists of a flat plate, we use laser cutting of acrylic board. Laser cutting provides high precision and speed, ensuring high time-efficiency and acceptable precision.

3) Lathe turning: For specially designed cylindrical components such as the motor shaft, lathe turning is used. It ensures that the motor shaft has the correct dimensions and surface finish to connect the wheel hub to the motor effectively.

C. Procedure of Manufacturing

Once the designs are finalized, the next step is the 3D printing of the wheel components using Stereolithography (SLA). Photosensitive Resin is selected for this purpose due to its strength and precision in SLA printing. After printing, the components are cleaned to remove any residual unpolymerized resin and polished to keep the smooth surface.

Parallel to the 3D printing process, the vehicle body is manufactured using acrylic board, which is cut to precise dimensions using a laser cutter. The cut parts include body panels and structural supports, which are then assembled using screws and adhesives to ensure robustness and stability of the vehicle chassis. For the mechanical parts that require high precision, such as the motor shaft and axle bearings, lathe turning is employed.

D. Procedure of Assembly

The assembly begins with the mechanical components. The wheel components, including rims, central gear, and hub, previously printed using SLA 3D printing, are first inspected for any imperfections. Each wheel rim is carefully aligned and attached to the central gear, ensuring that the cam mechanisms mesh smoothly for effective transformation between modes. The wheel hubs are then mounted onto the axles, and secured with locking collars to maintain alignment and reduce wear during operation. Meanwhile, make sure the contact between the wheel rim and wheel hub is smooth, as possible as lower friction.

Once the wheels are assembled, the DC motor is installed, connecting its output shaft to the wheel hub via a hexagonal coupling for efficient torque transfer. Following this, the electromagnetic clutches—critical for the switching mechanism between driving and wheel transformation—are integrated. The wheel clutch connects the motor to the central gear, while the car clutch links the motor to the vehicle body.

The next phase is the installation of the electronic systems and sensors that control and monitor the vehicle's environment and status. Electrical wiring is meticulously routed and connected to the motor, sensors, and clutches. Special care is taken to secure the wiring, preventing interference with mechanical components. The inertial measurement unit (IMU) is centrally mounted to accurately track the vehicle's orientation, and the RGB camera is installed at the front to assist with navigation and obstacle avoidance.

The final step is the integration of acrylic body panels, which protect the internal components while providing structural support. After the body assembly, calibration of sensors and mechanical adjustments are made to fine-tune the vehicle's operations, enhancing accuracy and responsiveness to environmental variables and control inputs.

IV. Control

A. Control Strategy and Flowchart of Raspberry Pi Code

The control strategy for the autonomous vehicle system is meticulously designed to ensure precise navigation through various scenarios, including straight-line movement, stair and sandbox expansions, as well as turning and parking maneuvers. The control system is divided into two main sections: Straight Line Control and Whole Control, each comprising specific tasks to handle different navigation challenges. The flowchart of the control algorithm is shown in Fig. 8.

In the Straight Line Control section, the system begins by initializing all necessary sensors and the camera to ensure accurate data acquisition. Once initialized, the IMU starts to acquire acceleration and gyroscope data, which are then processed to determine the vehicle's current position and orientation. A decision node assesses whether the vehicle is off-course. If the vehicle deviates from its intended path, corrective actions are implemented based on a proportional-integral-derivative (PID) control algorithm. The PID controller fine-tunes the steering angle by considering the current error, its integral over time, and its rate of change, ensuring smooth and accurate adjustments. Control signals, derived from the PID controller's output, are then sent to the motors to adjust the vehicle's direction and maintain the desired path. The vehicle proceeds to move along the corrected path, continually monitoring and adjusting its course as needed.

In the Whole Control section, similar to straight-line control, sensors and the camera are initialized, and a live video feed is read and processed to detect environmental features. A crucial part of the whole control strategy is the detection of stairs. If the distance to the stairs is less than 0.2 meters, the vehicle undergoes an expansion procedure to handle the stairs appropriately. If no stairs are detected, it continues with straight-line control. The system processes video frames to identify turning signs, and upon detecting a sign within 1.4 meters, the vehicle prepares for a turn. A closer proximity check at 0.6 meters initiates the turning control sequence, where a PID algorithm is utilized to manage the turn smoothly, ensuring accurate navigation through the turn. After executing the turn, the system checks for the finish line, and upon reaching it, the vehicle transitions into a parking maneuver, completing its navigation task.

A significant innovation within our design is the integration of the PID algorithm in both the straight-line and turning control mechanisms based on the orientation angle from the IMU. The PID controller enhances the vehicle's ability to maintain a precise course and execute turns smoothly by continuously adjusting the control inputs based on real-time feedback. This approach not only improves the accuracy of the navigation system but also ensures robustness in handling dynamic and unpredictable environments. The creative application of PID control in both linear and rotational movements distinguishes our system, providing a reliable and efficient solution for autonomous vehicle navigation.

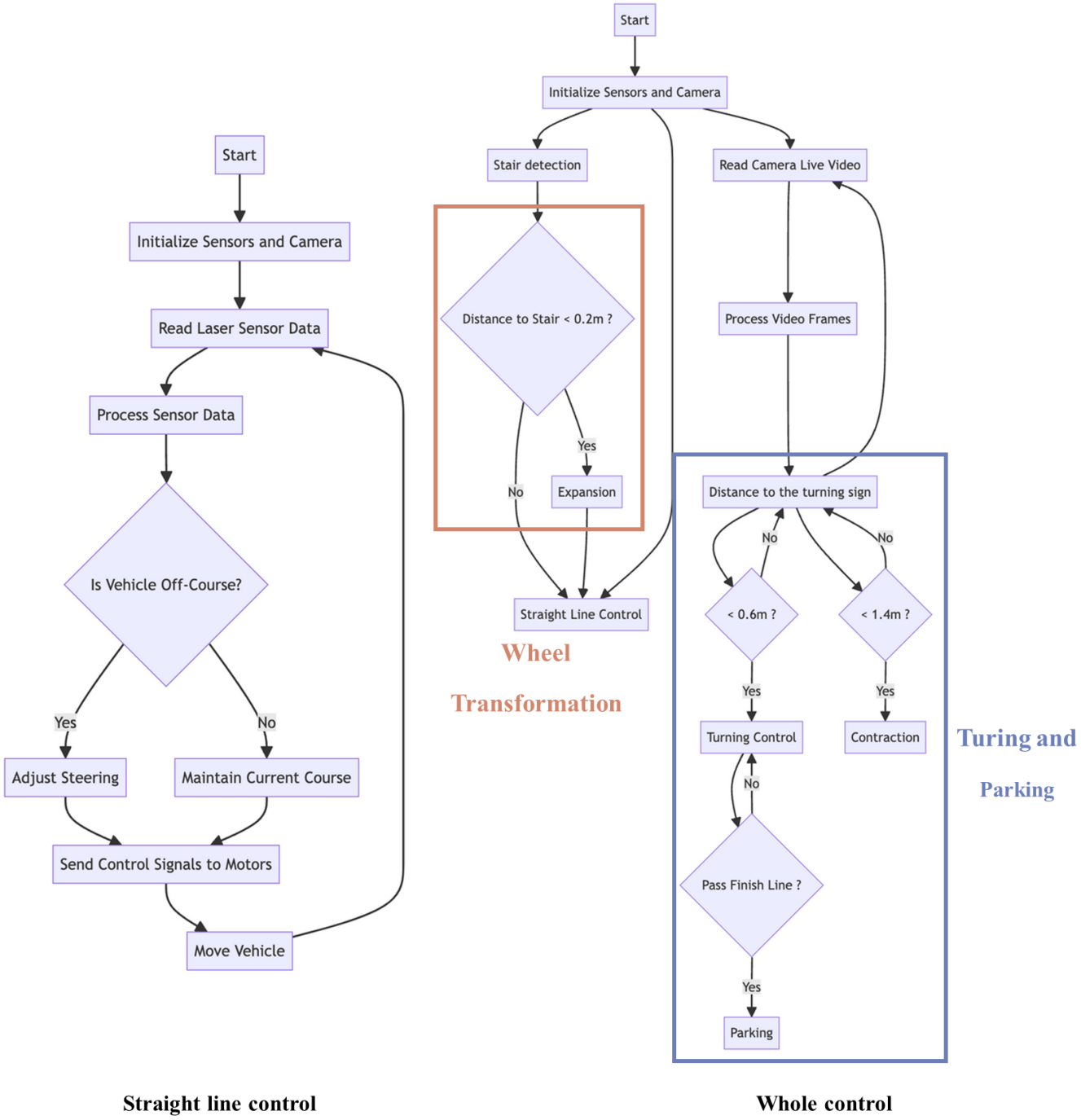


Fig. 8: The flowchart of Raspberry Pi code

B. Circuit diagram

The circuit diagram shown in Fig. 9 illustrates the control system of the autonomous vehicle using a Raspberry Pi as the central processing unit. The setup includes an IMU and a camera for obstacle detection and navigation, interfaced with the Raspberry Pi. The sensors and camera gather environmental data, which is processed by the Raspberry Pi to determine the vehicle's course. The motor control is managed by an H-bridge, which is connected to the Raspberry Pi and controls four DC motors, enabling differential drive for precise maneuvering. Power is supplied by a battery pack, which ensures the system's mobility and independence. The ultrasonic sensors are positioned to provide distance measurements for obstacle avoidance, while the camera captures live video feed for more complex navigation tasks. The control signals from the Raspberry Pi, based on processed sensor and camera data, are sent to the H-bridge, which then drives the motors accordingly. This integration allows for real-time adjustments and ensures the vehicle can navigate through various environments effectively.

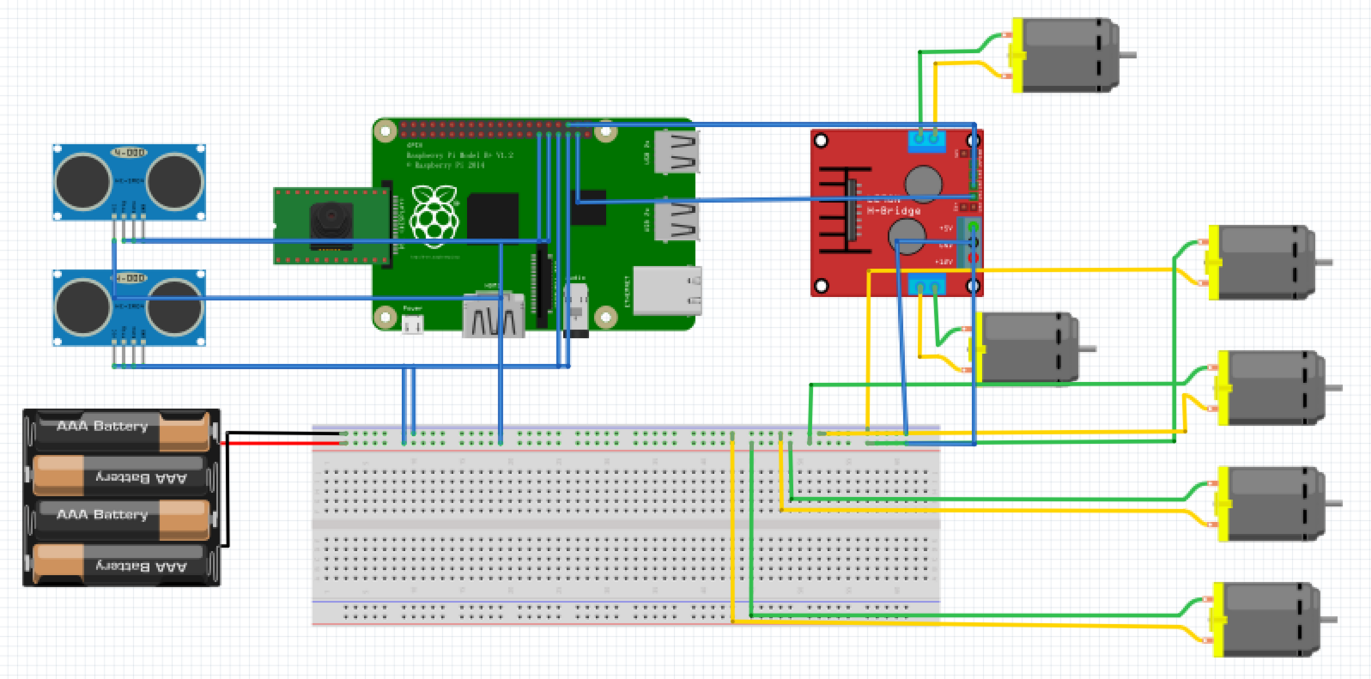


Fig. 9: The circuit block diagram of the whole system

V. Analysis

A. Classification of the Designed Linkage

The linkage diagram of the transformable wheel is shown in Fig. 10.

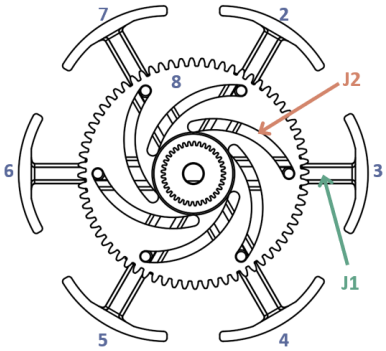


Fig. 10: The linkage diagram of the transformable wheel

From this figure, we can see that there are a total of 8 linkages ($L = 8$), 7 full joints ($J_1 = 7$), and 6 half joints ($J_2 = 6$). Then, according to Gruebler's equation, we know that the mobility is

$$M = 3(L-1) - 2J_1 - J_2 \tag{1}$$
$$= 3 \times (8-1) - 2 \times 7 - 6 \tag{2}$$
$$= 1\text{DOF} \tag{3}$$

Thus, the linkage in the transformable wheel only has one DOF. The linkage we used in the transformable wheel is a cam mechanism, which is equivalent to a four-bar linkage with variable-length links.

B. Position Analysis for Transformation

Position analysis is crucial to understanding the transformation dynamics of our system. According to Fig. 11, the position r of point A can vary between a minimum and a maximum position as follows:

$$r_{\min} = a + R = 14.21 + 24 = 38.21 \text{ mm} \tag{4}$$

$$r_{\max} = b + R = 29 + 24 = 53 \text{ mm} \tag{5}$$

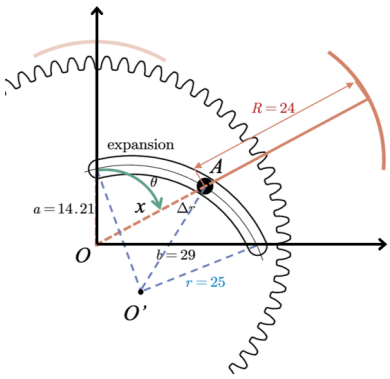


Fig. 11: Detailed geometry of the transformable wheel

The relationship between the position x and the rotation angle θ is derived geometrically. From the geometry,

$$x^2 - 23.49\cos(148.69^\circ - \theta)x - (\cos(148.69^\circ - \theta) + 487.17) = 0 \tag{6}$$

Solving for x using the quadratic formula,

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{7}$$

where

$$a = 1 \tag{8}$$

$$b = -23.49\cos(148.69^\circ - \theta) \tag{9}$$

$$c = -\cos(148.69^\circ - \theta) - 487.17 \tag{10}$$

The analysis of the position and speed was performed using MATLAB. Figure 12 shows the increase ratio versus the rotation angle θ. The maximum radius increase ratio is approximately 38%.

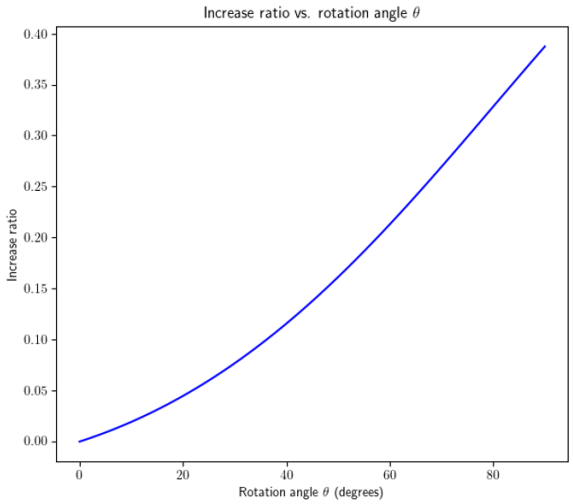


Fig. 12: Increase ratio vs. rotation angle θ

The speed v as a function of time t during the full expansion, which takes 3 seconds, is depicted in Figure 13. The speed increases rapidly at the beginning, reaching a peak before decreasing towards the end of the expansion period.

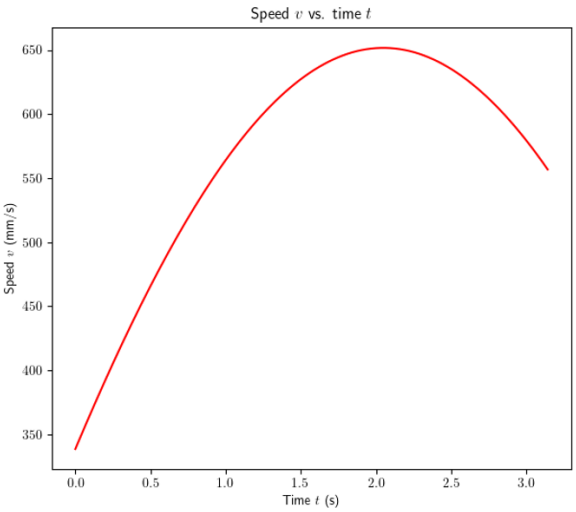


Fig. 13: Speed v vs. time t

In conclusion, the transformable wheel we designed has a minimum radius of $r_{\min} = 38.21$ mm and a maximum radius of $r_{\max} = 53$ mm. Given the rotation speed of the wheel motor, the wheel can reach full expansion in 3 seconds and achieve a maximum of 38% of radius increase. The maximum expansion speed is about 650 mm/s.

C. Force Analysis

To lift the weight of the whole robot onto the steps, the robot has a full expansion, lifted by frictional force. The schematic diagram is shown in Fig. 14.

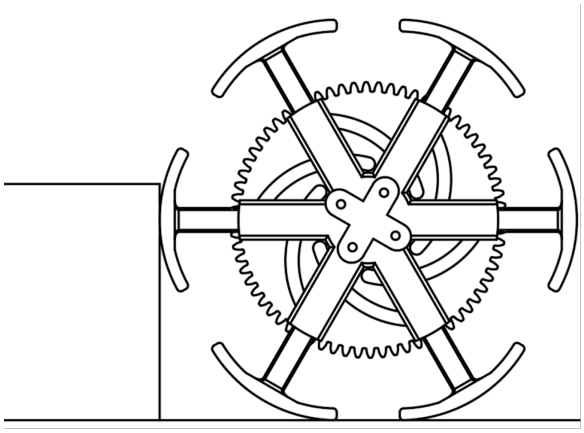


Fig. 14: Full expansion.

The free-body diagram is shown in Fig. 15.

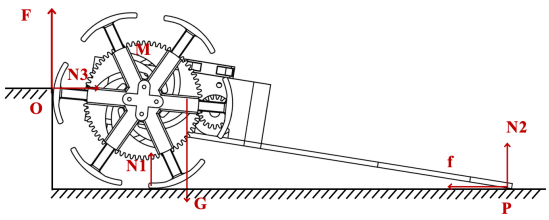


Fig. 15: Free body diagram of the robot when climbing the stairs.

Based on this free-body diagram, we can get the equations below through force analysis. The force equilibrium of the whole car gives

$$\uparrow \quad \sum F_y = 0 \qquad F + N_1 + N_2 - G = 0$$

$$\rightarrow \quad \sum F_x = 0 \qquad N_3 - f = 0$$

$$\circlearrowleft \quad \sum M_o = 0 \qquad G \cdot l_1 - N_2 \cdot l_2 = 0$$

where $F = \mu N_3$, $\mu = 0.3$ (obtained by online data), $G = 10N$ and $N_1 = 0N$. Furthermore, the moment equilibrium of the car plate gives

$$\circlearrowleft \quad \sum M_P = 0 \qquad T \cdot (l_2 - l_3) - G \cdot (l_3 - l_1) = 0$$

and the moment equilibrium of the wheel gives

$$\circlearrowleft \quad \sum M_O = 0 \qquad T \cdot l_3 - M = 0$$

The dimensions of the vehicle give

$$l_1 = 78\text{mm} \qquad l_2 = 283\text{mm} \qquad l_3 = 48\text{mm}$$

Solving the above set of equations, we know that the minimum torque provided by the wheel motor should be

$$M_{\min} = 4.36\text{Kgf} \cdot \text{cm} \tag{11}$$

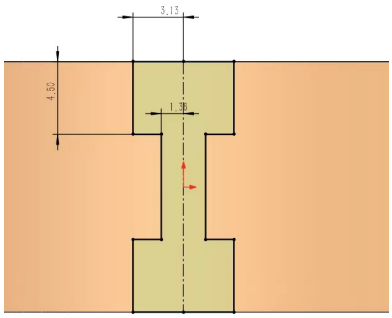


Fig. 16: Sectional view of the guide rod.

Therefore, we need to select two JGB37-520-1260 motors with a rated torque of 5.50 $\mathrm{Kgf}\cdot\mathrm{cm}$ to drive the wheel.

To ensure the material can sustain the pressure of the external force, the safety factor needs to be calculated. According to the study by Farah et al. [12], the tensile stress of photosensitive resin is 59MPa. And the actual stress can be obtained by

$$\sigma = \frac{F}{A} \tag{12}$$

where F here equals to N_3 and A equals to the cross-section area of the guide rod.

From the equations above, we can already get $N_3 = 24.65N$ and the actual cross-section is shown in Fig. 16.

From the data in the figure, we can calculate

$$\begin{aligned} A &= 6.26\,\mathrm{mm} \times 15.5\,\mathrm{mm} - 2 \times 6.5\,\mathrm{mm} \times 1.75\mathrm{mm} \\ &= 74.28\,\mathrm{mm}^2 \end{aligned}$$

Thus $\sigma = \frac{F}{A} = 0.332MPa$

$$\mathrm{SF} = \frac{\mathrm{TS}}{\sigma} \approx 177.8$$

So there is no possibility for the part to crack.

D. Rolling Speed Analysis

Rolling speed is of critical interest in analyzing the passability of the vehicle in different terrains. We expect the wheel to shrink when it passes through organized surfaces such as flat ground and expand when it encounters sand, stairs, and other uneven surfaces.

When the wheel is under wheel mode, the rolling speed depends on the rotation speed of the motor as well as how much slippage occurs between the surface and the wheel. Under the loading condition of our vehicle, the motor we use has a maximum rotation speed of 18 RPM. To integrate the straight line control algorithm, we constraint the rotation speed of the motor to be 70% of the maximum speed, which is $N_{\mathrm{actual}} = 12.6$ RPM. Then the rolling speed under wheel mode is

$$\begin{aligned} v &= r_{\min}\omega = 2\pi N_{\mathrm{actual}} r_{\min} \\ &= 2\pi \times 12.6 \times 38.21 = 50.42\mathrm{mm/s} \end{aligned}$$

When a wheel undergoes expansion, its radius increases. This leads to a larger contact area with the surface, reducing the pressure exerted per unit area. Although this could potentially decrease the rolling resistance due to the larger contact patch, it also increases the wheel's moment of inertia, making it harder to accelerate. Therefore, the net effect on rolling speed depends on the balance between these factors. In most practical scenarios, the increased inertia tends to dominate, resulting in a reduced rolling speed.

Furthermore, we need to take into account the possibility that the wheel sink into the sand. We will use Cheng et al.'s terradynamic model [13] to analyze the passability of our vehicle on the sand. See Fig. 17. In this model, we mainly

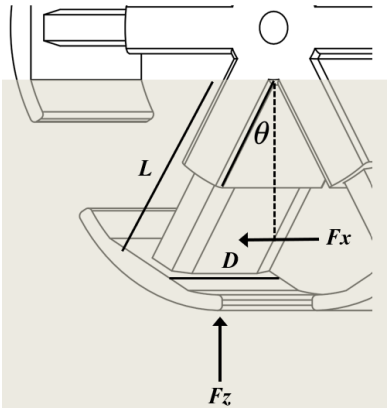


Fig. 17: The model of wheel in the sand

consider the hydrostatic-like pressure of the wheel in the sand. The hydrostatic stresses is

$$\sigma_{z,x}\left(|z|,\beta,\gamma\right) = \begin{cases} \alpha_{z,x}\left(\beta,\gamma\right)|z| & \mathrm{if} z < 0 \\ 0 & \mathrm{if} z > 0 \end{cases}$$

where β and γ are the attack angle and the intrusion angle respectively, $\alpha_{z,x}$ depends on both of them. The resistive force is the integration of stresses over the legs

$$F_{z,x} = \int_{\mathcal{S}} \sigma_{z,x}\left(\beta_s,\gamma_s,|z_s|\right)\mathrm{d}A_s = \int_{\mathcal{S}} \alpha_{z,x}\left(\beta_s,\gamma_s\right)|z_s|\,\mathrm{d}A_s$$

The value of $\alpha_{z,x}$ can be specifically determined by θ. When $\theta = 30^\circ$, we have

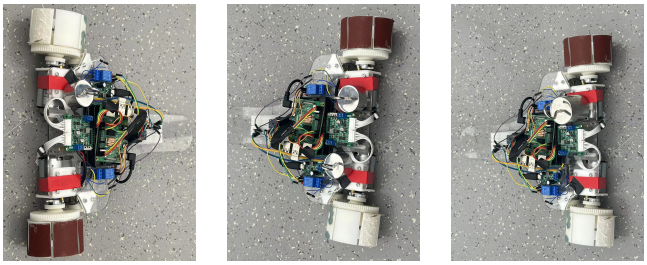
$$\begin{aligned} F_z &= \alpha_{z,x}\left(\frac{\pi}{6},\frac{\pi}{4}\right)\int_0^{L\sin\theta}\int_0^{D} y\cot\theta\sqrt{\cot^2\theta+1}\mathrm{d}x\mathrm{d}y \\ &= \alpha_{z,x}\left(\frac{\pi}{6},\frac{\pi}{4}\right)\frac{1}{2}DL^2\cos\theta \\ &= 0.157 \times 0.5 \times 4.5 \times 5^2 \times \cos\left(\frac{\pi}{3}\right) = 4.42\mathrm{N} \end{aligned}$$

Similarly, we can get $F_x = 3.37$N. Therefore, the supporting force and dragging force is enough for our wheel to drive on sand.

VI. Experiment

A. Demonstration of load-carrying capacity

By gradually increasing the extra load on the chassis base and measuring whether the wheel can rotate as usual when the motors work at full power, we can demonstrate the load-carrying capacity of our vehicle.



(a) 200g load (b) 400g load (c) 550g load

Fig. 18: Performance with 200g load, 400g load, and 550g load respectively.

We found that when load increased to around 550g, the rotation of the wheels can be neglected so 550g is the maximum load-carrying capacity of our vehicle.

Utilizing the formula in the Force analysis section, we can find when $G = 15.4N$ we have:

$$T_{\text{needed}} = 5.23\text{Kgf} \cdot \text{cm} \tag{13}$$

The relative error is around 4.91%. Within the margin of error, we considered the deviation comes from:

- The friction between hub and rims as well as slots and shafts.
- The deviation of support force due to different test field.

B. Demonstration of movement performance

1) climbing onto the sandbox: To put it frankly, our vehicle failed to climb onto the stairs in the gameday, for which we will elaborate the reasons in the following contexts: The failure mainly comes from the Force analysis section. We underestimated the weight of the vehicle and overestimated the friction ratio of the contact area, namely both G and μ differ a lot from the ideal case. As a result, we wrapped up the wheels with double-sided foam tapes and then covered them with a layer of sandpaper to increase the grip. However, the output torque of our motors still couldn't afford the power needed to climb onto the stairs. There is almost no qualified motors in the market as we require the length of the motor shaft to at least be longer than $60mm$. To put it further, we learnt the lesson that when designing our vehicle, we also needed to pay attention to what elements we can easily access off the shelf in the market instead of creating something totally based on our imagination.

2) running through the tunnel: Our vehicle can go through the tunnel without hitting any obstacles in the second round. In the first round, the height restriction was knocked down by the wires so we bound them together and fix onto the chassis base of the vehicle.

C. Measure of the rolling speed

We conducted a simulation experiment in the Tang junyuan Innovation Center and measured the rolling speed using the formula:

$$v = \frac{s}{t} = \frac{57.1\text{cm}}{12.0\text{s}} = 47.58\text{mm/s} \tag{14}$$

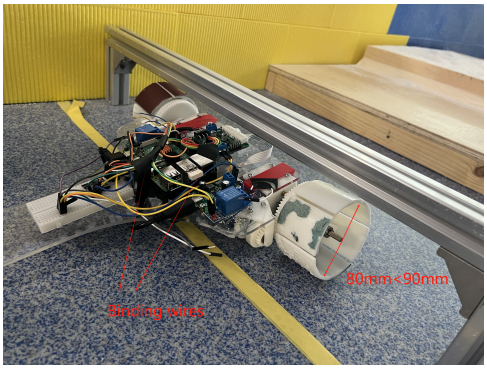


Fig. 19: Our vehicle can go through the tunnel.

The relative error is around 5.63%. Within the margin of error, we considered the deviation comes from:

- The friction ratio is different due to different test field (in the Analysis section we use 0.3 for μ).
- The use of double-sided foam tape and abrasive paper to enhance grip.
- Measuring errors.

D. Measure of the wheel's transformability and vehicle configuration

We used a 1mm-precision ruler to measure the diameter of the wheels in wheel mode and leg mode respectively. The theoretical value is captured from our CAD models and matches with the statistics in the Analysis section.

	Experimental [mm]	**Theoretical [mm]**	**Error**
Wheel mode	40	38.21	4.68%
Leg mode	55	53	3.77%

TABLE III: Measure of the wheel's transformability

There is a 2mm absolute error no matter in the wheel mode or leg mode. We consider that may come from:

- The fabrication error during the 3D-printing process.
- The use of double-sided foam tape and abrasive paper to enhance grip.
- Measuring errors.

Moreover, we also noticed that the dimensions of left wheel and right wheel differ so that the actual loads of the motors on both sides get different.

As for the vehicle configuration, we found that the weight of the circuit elements is out of expectation so that the acrylic has "collapsed" a little bit in the middle. This resulted that the motor shaft couldn't remain parallel with the chassis base, which can be viewed in Fig. 20.

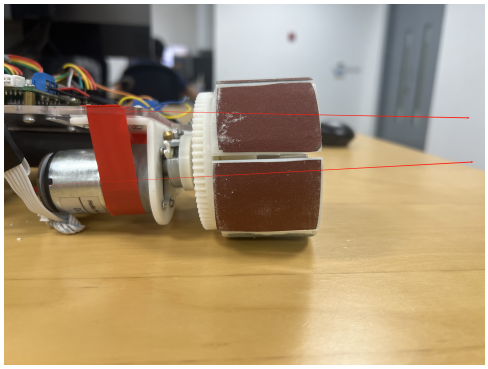


Fig. 20: The motor shaft couldn't remain parallel with the chassis base.

Under these circumstances, to eliminate of the effects on the actual performance of the motors, we fixed the motors with tapes as shown in Fig. 20.

VII. Discussion

A. Agreement and disagreement with previously published work.

Our findings show both consistence and divergences with existing literature on transformable wheel mechanisms. Similar to the work by Tadakuma et al [2]. on armadillo-inspired retractable modules, our transformable wheel demonstrated an ability to enhance mobility over obstacles. However, unlike the miniature jumping robot developed by Kovac et al [1]., which emphasized jumping height, our design focuses more on adaptability across diverse terrains. This distinction highlights different approaches to solving the challenge of mobility in robotic systems, with our work contributing to the discourse by providing a practical solution for adaptable ground mobility.

B. The Intellectual Merit of Our Work

The intellectual merit of this study lies in its innovative approach to creating a wheel that can transform its shape to optimize performance across various terrains. Our transformable wheel incorporates a unique mechanism involving adjustable spoke lengths and a flexible outer rim, which allows the wheel to expand and contract efficiently. This mechanism not only enhances the wheel's adaptability but also ensures stability and load-bearing capacity during transformations. By integrating theoretical analysis with CAD modeling and empirical testing, we provided a comprehensive examination of the mechanical properties and practical applicability of the transformable wheel. The methodology used ensures that the findings are robust and applicable to real-world scenarios, thereby advancing the field of mobile robotics. Our work adds to the body of knowledge by presenting a novel solution that combines the benefits of both traditional wheels and adaptive mechanisms.

C. Theoretical and Practical Implications

Theoretically, our study contributes to the understanding of how transformable mechanisms can be designed and implemented to enhance robotic mobility. The analysis of the wheel's minimum and maximum radii, expansion speed, and load-bearing capacity offers valuable insights into the dynamics of transformable wheels. Our wheel's mechanism, which involves adjustable spoke lengths and a flexible outer rim, provides a detailed case study of how such components can be integrated to create a versatile and adaptive wheel design. This contributes to the broader theoretical framework of transformable robotics and mechanical design.

Practically, the findings have significant implications for the development of robots capable of navigating complex and varied terrains, such as disaster response robots, planetary rovers, and autonomous vehicles for search and rescue operations. The ability of the wheel to adapt its shape in response to different environmental conditions ensures improved stability and efficiency, making it a viable option for future robotic applications. The adjustable spoke mechanism, in particular, allows for rapid and efficient transformations, enhancing the robot's operational versatility in real-time scenarios.

By addressing both theoretical frameworks and practical applications, this work bridges the gap between conceptual designs and functional prototypes, paving the way for more versatile and capable robotic systems. The broader impact of our research is evident in its potential to influence the design and deployment of next-generation robots across various industries and use cases.

VIII. Conclusion

A. Summary of Objectives and Methods

The primary objective of this project was to design and manufacture an autonomous rover equipped with transformable wheels capable of navigating various terrains, including stairs and sandy surfaces. The key methodologies involved the integration of advanced sensors, the application of a novel transformable wheel mechanism, and the implementation of a sophisticated control algorithm utilizing a Raspberry Pi.

B. Vital Findings

- **Transformable Wheel Mechanism**:
  - The innovative transformable wheel mechanism allows seamless transitions between different modes, ensuring adaptability to various terrains.
  - This mechanism significantly reduces the need for multiple actuators by integrating electromagnetic clutches, which streamline the transformation process.
- **Control System**:
  - The control system, managed by a Raspberry Pi, employs a Proportional-Integral-Derivative (PID) algorithm for precise navigation.
  - The system effectively handles straight-line movement, turning, and obstacle navigation through continuous real-time feedback.
- **Manufacturing Techniques**:
  - Utilization of SLA 3D printing for high-precision components.
  - Laser cutting for efficient production of the car body.
  - Lathe turning for accurate cylindrical components.

C. Generalized Ideas

The results of this project demonstrate that the transformable wheel mechanism, combined with advanced control strategies, offers a robust solution for autonomous navigation across challenging terrains. The key insights include:

- **Mechanism Efficiency**:
  - The integration of electromagnetic clutches reduces the complexity and potential points of failure compared to traditional multi-actuator systems. This enhances both reliability and cost-effectiveness.
- **Control Precision**:
  - The application of the PID algorithm in both linear and rotational control significantly improves the rover's ability to maintain a precise course and adapt to dynamic environments, highlighting the importance of real-time feedback in autonomous navigation.
- **Manufacturing and Material Selection**:
  - The careful selection of materials and manufacturing methods ensures the durability and functionality of the transformable wheels, making the design viable for practical applications in remote and hazardous environments.

In conclusion, the developed autonomous rover with its advanced transformable wheel mechanism and control system provides a significant step forward in the design of versatile and efficient robots capable of exploring and operating in complex terrains. The methodologies and findings from this project offer valuable insights for future research and development in the field of autonomous vehicles.

References

[1] M. Kovač, M. Fuchs, A. Guignard, J.-C. Zufferey, and D. Floreano. (2008, May) A miniature 7g jumping robot. 2008 IEEE International Conference on Robotics and Automation. Accessed: 2008-05-19. [Online]. Available: http://ieeexplore.ieee.org/document/4541218/

[2] K. TADAKUMA, R. TADAKUMA, A. MARUYAMA, E. Rohmer, K. NAGATANI, K. YOSHIDA, A. MING, S. MAKOTO, M. HIGASHIMORI, and M. KANEKO, "Armadillo-inspired wheel-leg retractable module," in *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*. IEEE, December 2009, pp. 610–615.

[3] A. M. Hoover, S. Burden, X.-Y. Fu, S. S. Sastry, and R. S. Fearing, "Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot," in *Proceedings of the 2010 3rd IEEE RAS 0026 EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE, September 2010, pp. 869–876.

[4] P. Ren and D. Hong, "Mobility analysis of a spoked walking machine with variable topologies," *Journal of Mechanisms and Robotics*, vol. 3, no. 4, pp. 041 005–1–16, November 2011.

[5] D.-Y. Lee, J.-S. Kim, S.-R. Kim, J.-S. Koh, and K.-J. Cho, "The deformable wheel robot using magic-ball origami structure," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences Computers and Information in Engineering Conference*. ASME, 2013, pp. DETC2013–13 016.

[6] D.-Y. Lee, J.-S. Koh, J.-S. Kim, S.-W. Kim, and K.-J. Cho, "Deformable-wheel robot based on soft material," *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 8, pp. 1439–1445, August 2013.

[7] Y.-S. Kim, G.-P. Jung, H. Kim, K.-J. Cho, and C.-N. Chu, "Wheel transformer: A wheel-leg hybrid robot with passive transformable wheels," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1487–1499, December 2014.

[8] Y. She, C. J. Hurd, and H.-J. Su, "A transformable wheel robot with a passive leg," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, p. 4165.

[9] L. Rosalia, B. W.-K. Ang, and R. C.-H. Yeow, "Geometry-based customization of bending modalities for 3d-printed soft pneumatic actuators," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3489–3495, October 2018.

[10] W. Zeng, G. Xu, H. Jiang, and F. Gao, "Development of a novel variable-diameter wheel," *Applied Sciences*, vol. 9, no. 21, p. 4631, 2019.

[11] H. Almabrouk, M. Kharroubi, F. Tounsi, B. Mezghani, and Y. Bernard, "Macro model analysis of a single mass 6-dof inertial measurement unit system," in *2016 11th International Design & Test Symposium (IDT)*. IEEE, 2016, pp. 290–295.

[12] S. Farah, D. G. Anderson, and R. Langer, "Physical and mechanical properties of pla, and their functions in widespread applications — a comprehensive review," *Advanced Drug Delivery Reviews*, vol. 107, pp. 367–392, 15 December 2016, available online 26 June 2016, Version of Record 7 December 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169409X16305007X

[13] C. Li, T. Zhang, and D. I. Goldman, "A terradynamics of legged locomotion on granular media," *science*, vol. 339, no. 6126, pp. 1408–1412, 2013.

Appendix A
Gantt Chart

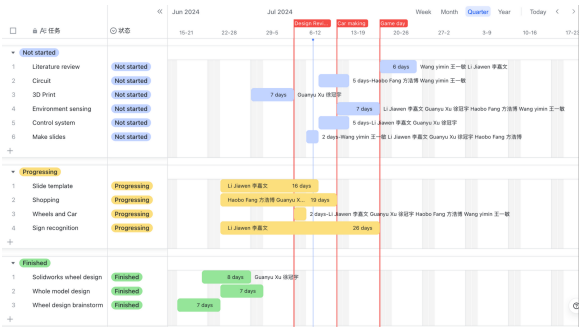


Fig. 21: Gantt chart.

Appendix B
Budget Table and Justification

The budget table is presented in Table IV. The total budget of our project is ¥642.80, which is within the budget limit.

Appendix C
Raspberry Pi Programming Code

```
import time
import RPi.GPIO as GPIO
from mpu6050 import mpu6050
from Motor import Motor
import numpy as np
import cv2
from picamera2 import Picamera2

# GPIO pin setup
LEFT_MOTOR_IN1 = 20
LEFT_MOTOR_IN2 = 21
LEFT_MOTOR_ENCODER = 13
LEFT_MOTOR_DIRECTION = 19
RIGHT_MOTOR_IN1 = 12
RIGHT_MOTOR_IN2 = 16
RIGHT_MOTOR_ENCODER = 23
RIGHT_MOTOR_DIRECTION = 24
LEFT_CLUTCH = 18
RIGHT_CLUTCH = 25

# setting up the Pi camera
picam2 = Picamera2()
dispW=1280
dispH=720
```

TABLE IV: The budget table. In this table, total price is the sum of *Number* times *Unit Price* of all items, and total budget refers to the extra item we bought in this project. The items with a (*) in the *Name* column do not count in the total budget.

Category	Name	Details	Number	Unit Price (CNY)
Power Supply	Battery	12V	1	56
	Power Module	Supports 11.1V, 3.3V, 5V output	1	19
	Power Module	11.1V-18650	1	57.8
	Relay	Supports 12V	1	11.8
	Power Cable*	Raspberry Pi power supply, Type-C	1	0
Motor	Motor	MG513 motor (Hall encoder)	2	49
	Motor Driver	AT8236, soldered pin header	1	35
Wheel Control	Wheel Clutch	(6mm center shaft)	2	3.35
	Bracket 1	Fixed wheel clutch	2	5
	Car Clutch	SLA 3D printing	2	3
	Bracket 2	Fixed car clutch	2	5
	Relay	SLA 3D printing	2	3.58
Wheels	Gear	SLA 3D printing	2	19.25
	Hub	SLA 3D printing	2	14.68
	Rim	SLA 3D printing	12	6.3
Chassis	Gyroscope	MPU6050, soldered pin header	1	9.48
	Camera	Raspberry Pi Camera	1	14
	Raspberry Pi	module 4B	1	228
	Infrared distance sensor	GP2Y0A02YK0F 20-150cm	4	108.2
	OLED screen	0.96 cun	1	7.9
	Car Chassis*	3mm thick acrylic	1	0
			Total Price (CNY)	656.4

```
picam2.preview_configuration.main.size = (dispW, dispH)
picam2.preview_configuration.main.format = "RGB888"
picam2.preview_configuration.controls.FrameRate=30
picam2.preview_configuration.align()
picam2.configure("preview")
picam2.start()

# define variables
fps=0
pos=(30,60)
font=cv2.FONT_HERSHEY_SIMPLEX
height=1.5
weight=3
myColor=(0,0,255)
GhueLow = 10
GhueHigh = 179
GsatLow = 66
GsatHigh = 255
GvalLow = 65
GvalHigh = 255
RhueLow = 0
RhueHigh = 10
RsatLow = 100
RsatHigh = 255
RvalLow = 100
RvalHigh = 255

GPIO.setmode(GPIO.BCM)

# Initialize MPU6050
sensor = mpu6050.mpu6050(0x68)
#Gyroscope calibration constants
GyroX_offset = -2.375541984732824
GyroY_offset = -7.537664122137405
GyroZ_offset = -1.764442748091603
#Accelerometer calibration constants
AccelX_k = 0.99873091
AccelX_b = -0.04609702
AccelY_k = 0.99933942
AccelY_b = 0.00491675
AccelZ_k = 0.96768578
AccelZ_b = -0.02067255

gyro_angle_x = 0
gyro_angle_y = 0
gyro_angle_z = 0

# Initialize Motors
left_motor = Motor(LEFT_MOTOR_IN1, LEFT_MOTOR_IN2, LEFT_MOTOR_ENCODER, LEFT_MOTOR_DIRECTION)
right_motor = Motor(RIGHT_MOTOR_IN1, RIGHT_MOTOR_IN2, RIGHT_MOTOR_ENCODER, RIGHT_MOTOR_DIRECTION)

# Initialize Clutches
GPIO.setup(LEFT_CLUTCH, GPIO.OUT)
GPIO.setup(RIGHT_CLUTCH, GPIO.OUT)


# PID parameters, controlling straight line and turning
Kp = 1.0
Ki = 0.0
Kd = 0.0

# Initialize PID variables

base_speed = 70
DELTA_T = 0.01

def get_front_distance():
    frame = picam2.capture_array()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    edges = cv2.Canny(blurred, 50, 150)
    lines = cv2.HoughLines(edges, 1, np.pi/180, threshold=100)
    if lines is not None:
        distance = lines.shape[0]  # You can modify this calculation based on your specific requirements
    else:
        distance = 0
    return distance

def get_yaw():
    global gyro_angle_z, gyro_angle_x, gyro_angle_y
    accel_data = sensor.get_accel_data(g=True)
    gyro_data = sensor.get_gyro_data()
    AccX = (AccelX_k * accel_data['x'] + AccelX_b)
    AccY = (AccelY_k * accel_data['y'] + AccelY_b)
    AccZ = (AccelZ_k * accel_data['z'] + AccelZ_b)
    GyroX = gyro_data['x'] - GyroX_offset
    GyroY = gyro_data['y'] - GyroY_offset
    GyroZ = gyro_data['z'] - GyroZ_offset
```

```
    acc_angle_x = np.arctan2(AccY, np.sqrt(AccX**2 + AccZ**2)) * 180 / np.pi
    acc_angle_y = np.arctan2(AccX, np.sqrt(AccY**2 + AccZ**2)) * 180 / np.pi
    gyro_angle_x += GyroX * DELTA_T
    gyro_angle_y += GyroY * DELTA_T
    gyro_angle_z += GyroZ * DELTA_T
    yaw = gyro_angle_z
    return yaw


def camera_judgement():
    front = get_front_distance()
    if front < 2:
        expansion_judgement = 1
    else:
        expansion_judgement = 0
    tStart=time.time()
    frame= picam2.capture_array()
    frameHSV=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    cv2.putText(frame,str(int(fps))+' FPS',pos,font,height,myColor,weight)
    # define my masks
    GlowerBound=np.array([GhueLow,GsatLow,GvalLow])
    GupperBound=np.array([GhueHigh,GsatHigh,GvalHigh])
    GmyMask=cv2.inRange(frameHSV,GlowerBound,GupperBound)

    RlowerBound=np.array([RhueLow,RsatLow,RvalLow])
    RupperBound=np.array([RhueHigh,RsatHigh,RvalHigh])
    RmyMask=cv2.inRange(frameHSV,RlowerBound,RupperBound)

    # define my objects
    GmyObject=cv2.bitwise_and(frame,frame, mask=GmyMask)
    RmyObject=cv2.bitwise_and(frame,frame, mask=RmyMask)

    # get the contour
    Gcontours,junk=cv2.findContours(GmyMask,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    Rcontours,junk=cv2.findContours(RmyMask,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    if len(Gcontours)>0:
        Gcontours=sorted(Gcontours,key=lambda x:cv2.contourArea(x),reverse=True)
        Gcontour=Gcontours[0]
        x1,y1,w1,h1=cv2.boundingRect(Gcontour)
        cv2.rectangle(frame,(x1,y1),(x1+w1,y1+h1),(0,255,0),3)

        if len(Rcontours)>0:
            Rcontours=sorted(Rcontours,key=lambda x:cv2.contourArea(x),reverse=True)
            Rcontour=Rcontours[0]
            x2,y2,w2,h2=cv2.boundingRect(Rcontour)
            cv2.rectangle(frame,(x2,y2),(x2+w2,y2+h2),(0,0,255),3)
            if x2+w2/2>x1+w1/2:
                print('Green is on the left')
                turning_judgement = 1
            if x2+w2/2<x1+w1/2:
                print('Green is on the right')
                turning_judgement = -1
    print()
    cv2.imshow("Camera", frame)
    tEnd=time.time()
    loopTime=tEnd-tStart
    fps=.9*fps + .1*(1/loopTime)
    return turning_judgement, expansion_judgement

def expansion(left_motor, right_motor, left_clutch, right_clutch):
    # Wheel Expansion
    GPIO.output(left_clutch, GPIO.HIGH)
    GPIO.output(right_clutch, GPIO.HIGH)
    left_motor.rotate_open(-80)
    right_motor.rotate_open(80)  # check this direction carefully
    time.sleep(0.2) # time for expansion, could be modified
    GPIO.output(left_clutch, GPIO.LOW)
    GPIO.output(right_clutch, GPIO.LOW)

def shrink(left_motor, right_motor, left_clutch, right_clutch):
    # Wheel Shrink
    GPIO.output(left_clutch, GPIO.HIGH)
    GPIO.output(right_clutch, GPIO.HIGH)
    left_motor.rotate_open(80)
    right_motor.rotate_open(-80)
    time.sleep(0.2)  # time for shrink, could be modified
    GPIO.output(left_clutch, GPIO.LOW)
    GPIO.output(right_clutch, GPIO.LOW)

target_yaw = 0  # Initial target yaw for straight line movement
yaws = []
last_yaw = 0

try:
    while True:
        yaw = get_yaw()
        print("Yaw angle:",yaw)
        yaws.append(yaw)
        last_yaw = yaw
        turning_judgement, expansion_judgement = camera_judgement()
        # Control:
        if turning_judgement == 0:
            if expansion_judgement == 0:
                # Straight line movement
                proportional = Kp * yaw
                integral = Ki * sum(yaws)
                derivative = Kd * (yaw - last_yaw)
                correction = proportional + integral + derivative
                left_motor.rotate(base_speed + correction)
                right_motor.rotate(base_speed - correction)
            else:
                expansion(left_motor, right_motor, LEFT_CLUTCH, RIGHT_CLUTCH)
                left_motor.rotate(100)
                right_motor.rotate(100)
                time.sleep(5) # time for crossing the sand
                shrink(left_motor, right_motor, LEFT_CLUTCH, RIGHT_CLUTCH)
        elif turning_judgement == 1:
            left_motor.rotate(-base_speed)
            right_motor.rotate(base_speed)
            time.sleep(0.5)
        elif turning_judgement == -1:
            left_motor.rotate(base_speed)
            right_motor.rotate(-base_speed)
            time.sleep(0.5)
        time.sleep(0.01)

except KeyboardInterrupt:
    print("Program terminated.")
finally:
    # Clean up GPIO
    GPIO.cleanup()
```