Department of Mechatronics and Robotics

Trajectory Planning, intelligent control and rocker-bogie Coordination of Mars Rover

Author : Yizhe.Zhu, Tianyi Xiang, Ruidan.Xu Hongyuan.Zhu

Contents

1 Introduction

The project aims to create a replica of the Mars Perseverance Rover, which is a significant undertaking that involves understanding, designing, and constructing various critical components. The rover employs a rocker-bogie suspension system to navigate the challenging Martian terrain and is equipped with independent DC motors for each wheel, allowing it to move forward and backward. The primary objective is to replicate these features and functionality, providing a practical and educational model of a real-world space exploration technology.

2 Physical Dimension Rocker-bogie Design

2.1 Mechanical Design of the Mars car

The **rocker-bogie suspension system** is a highly effective design that enables the rover to navigate smoothly over uneven terrain. This system is particularly advantageous for exploration missions on Mars due to the following features and benefits:

Enhanced Terrain Adaptability. The rocker-bogie suspension allows the rover to climb over obstacles that are up to twice the diameter of its wheels. This capability is crucial for navigating the rocky and varied Martian surface, ensuring the rover can traverse large rocks and deep craters without getting stuck. Six-Wheel Ground Contact The design ensures that all six wheels remain in contact with the ground at all times. This maximizes traction and stability, which is essential for maintaining control and preventing tipping on steep slopes or soft, shifting soils commonly found on Mars. Minimized Impact from Terrain Irregularities. The suspension system distributes the load evenly across all wheels, reducing the impact of uneven terrain on the rover's chassis. This helps in preserving the structural integrity of the rover over long-duration missions.

With independent movement of each wheel, the rover can better maneuver around obstacles and through challenging terrains. This flexibility is critical for reaching scientifically interesting sites that are otherwise difficult to access.

Increased Durability and Reliability The rocker-bogie design is mechanically simple and robust, with fewer moving parts compared to other suspension systems. This simplicity enhances the rover's durability and reliability, reducing the likelihood of mechanical failure in the harsh Martian environment.

Energy Efficiency By maintaining constant ground contact and distributing weight evenly, the rover expends less energy overcoming obstacles. This energy efficiency is vital for Mars rovers, which rely on limited power sources such as solar panels or nuclear batteries.

These features make the rocker-bogie suspension an optimal choice for Mars rovers, ensuring they can perform extensive and effective exploration of the planet's surface.

As shown in Fig. 2, the servo steering system is designed by one 25kg servo, and two bearing, connected by a M6 threaded screw. Noting M6 Tightening screw is applied to lighten the connection of servo and the tire.

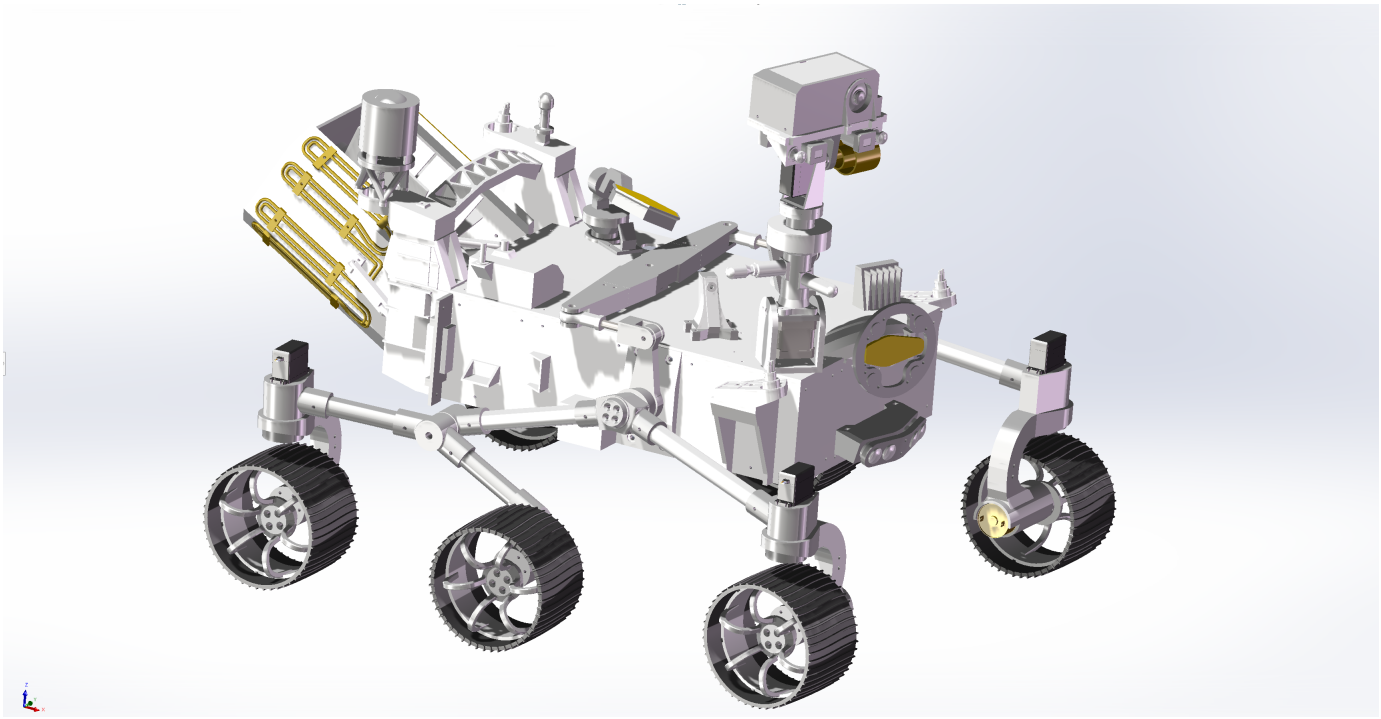

Figure 1: Overall review of the Mars perseverance rover replica

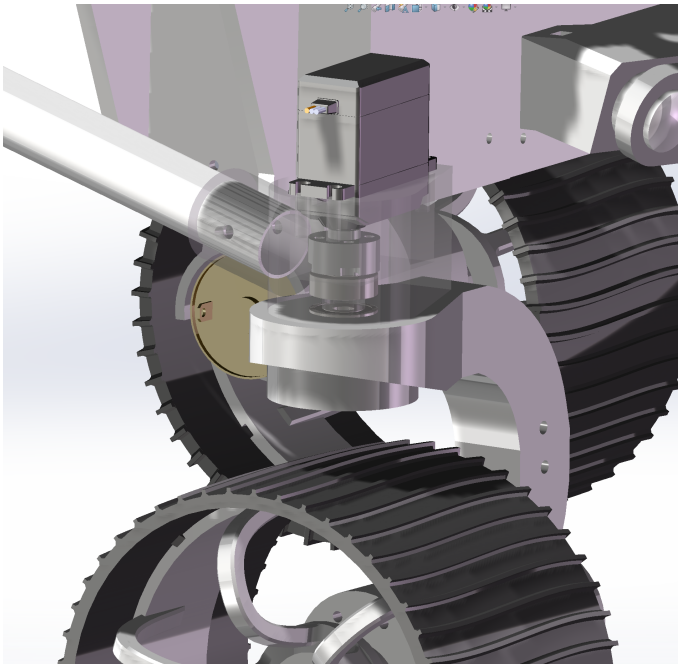

Figure 2: Screenshoting of the servo steering system design

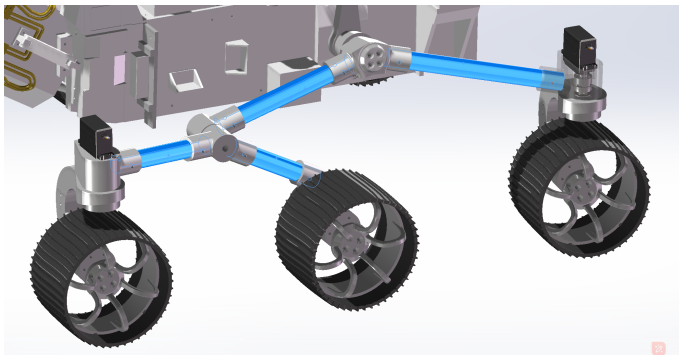

Figure 3: Double joint Suspension Mechanical Design

For connecting the differential with the rocker, we need a rod end ball joint. I'm using M8 rod end ball joint and we also need an M8 threaded rod with 50mm length. The threaded rod goes into a 3D printed part which has an M8 nut on one side, and on the other side it goes into the rod end ball joint.

The ability of stepping over the barrier in terrible terrain come from the double joint design as

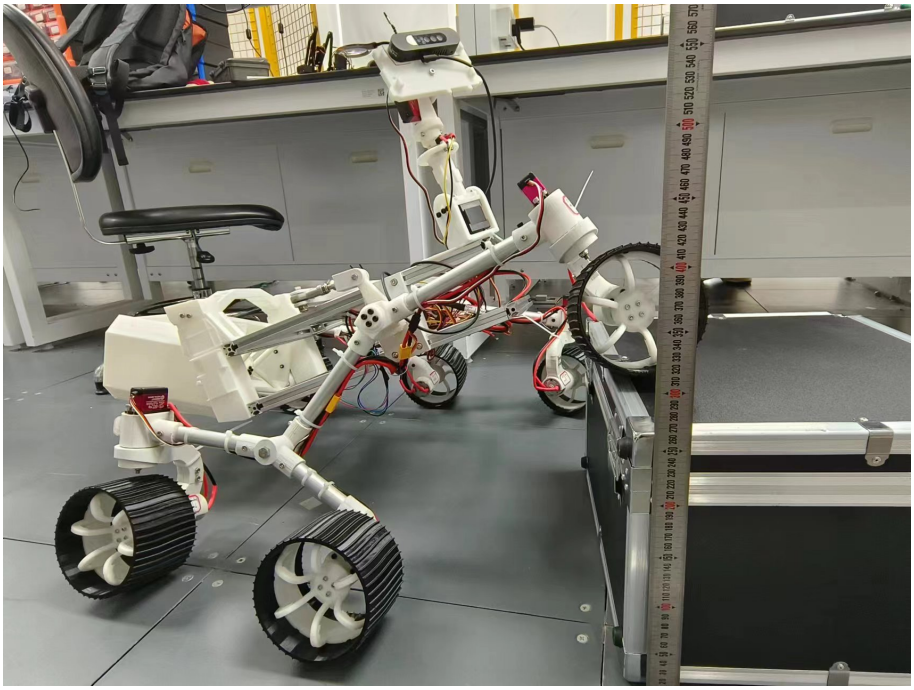

Figure 4: Suspension System Step-over the 250mm big barrier

shown in Fig. 3. Such Design enable all wheels to the ability to move independently. This provides all wheels to be in contact with the ground all the time. The chassis goes only half the motion of the leg, or the chassis has an average pitch angle of both rockers.

Together with the Differential of the rocker, our rocker-bogie suspension is completed. When one side goes up, the other goes down and vice-versa.

2.2 Result of the Mechanical Suspension System

As shown in Fig. 4, the suspension system could enable the mars car step over a huge box, in which the height is around 270 mm. Particularly worth mentioning is the chassis of the car is only 172.89 mm. The suspension system could enable the car to step over the barrier whose height is higher than the chassis. Such result validate the successful implementation of the rocker-bogie suspension design.

2.3 Electrical Dimension of the Mars Car

Figure 5 illustrates the comprehensive electrical layout of the Mars rover. This diagram details the connections between various components, including the Arduino Mega, DC motors, servos, camera system, and power supply.

- **Power Supply**: A 3S LiPo battery provides the primary power, delivering 12V, which is regulated by a buck converter to provide 6V for the servos and 5V for the other electronics.
- **Motors**: Six DC motors, controlled by DRV8871 motor drivers, are used for the rover's loco-

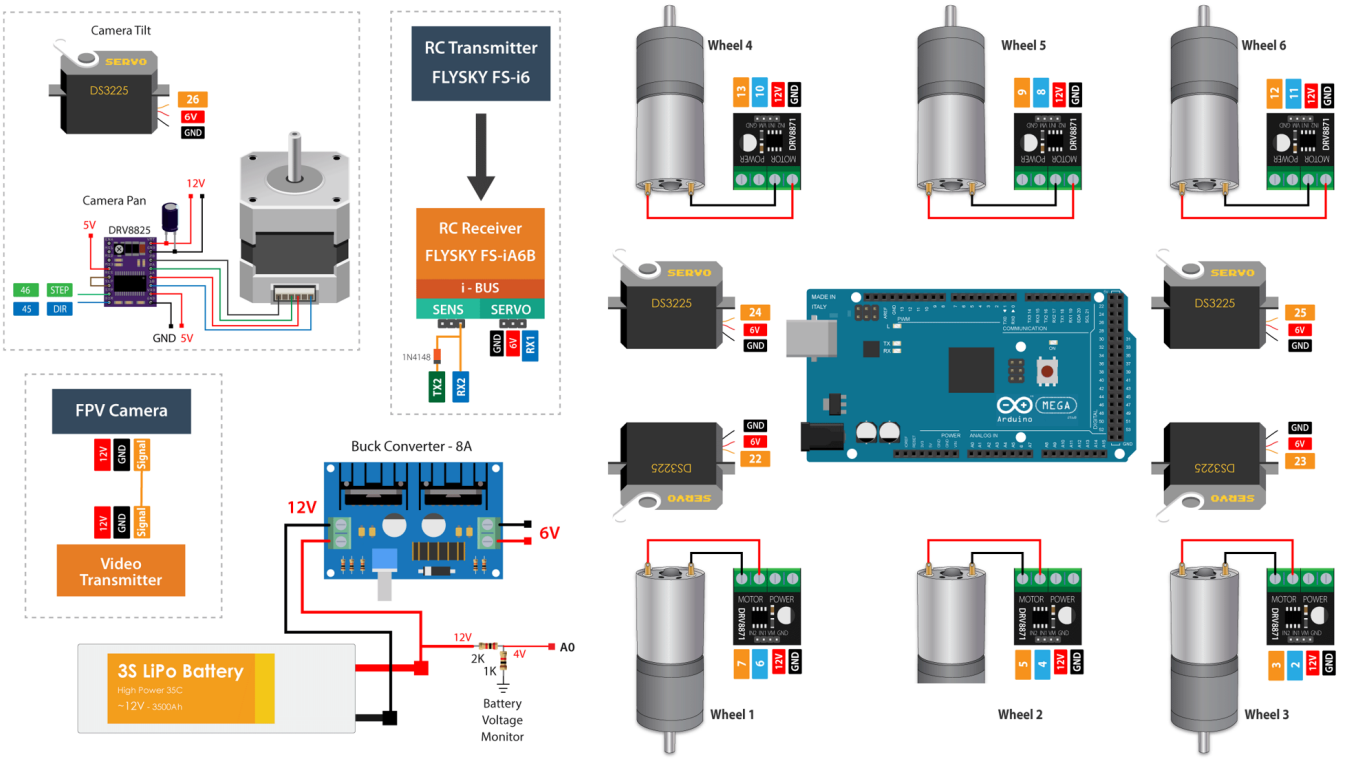

Figure 5: The circuit design of the mars car

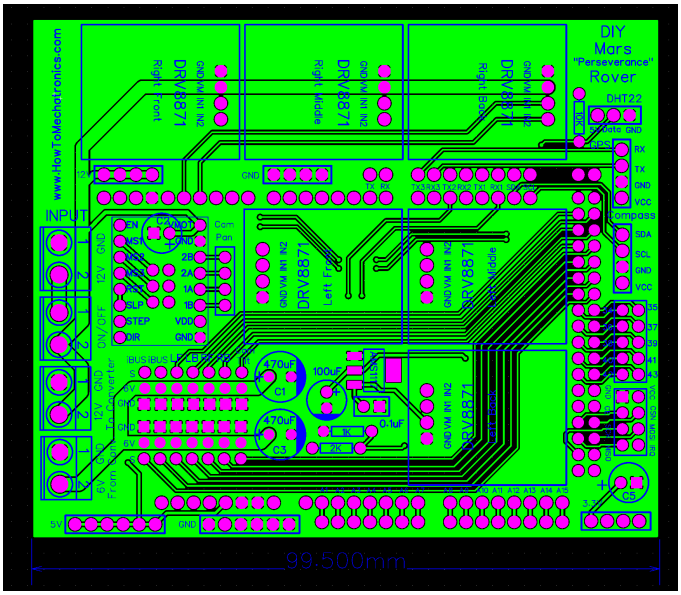

Figure 6: Mars Car PCB Design

motion. Each motor driver is connected to the Arduino Mega for control.

- **Camera System**: The camera system comprises an FPV camera for real-time video transmission and a pan-tilt mechanism. The tilt is controlled by a DS3225 servo, and the pan by a stepper motor driven by a DRV8825 driver.
- **Control and Communication**: An RC receiver (FLYSKY FS-iA6B) communicates with the RC transmitter (FLYSKY FS-i6) to control the rover. The receiver is connected to the Arduino Mega via the iBUS protocol.
- **Sensors**: Various sensors can be integrated into the system through available pins on the Arduino Mega for additional functionalities.

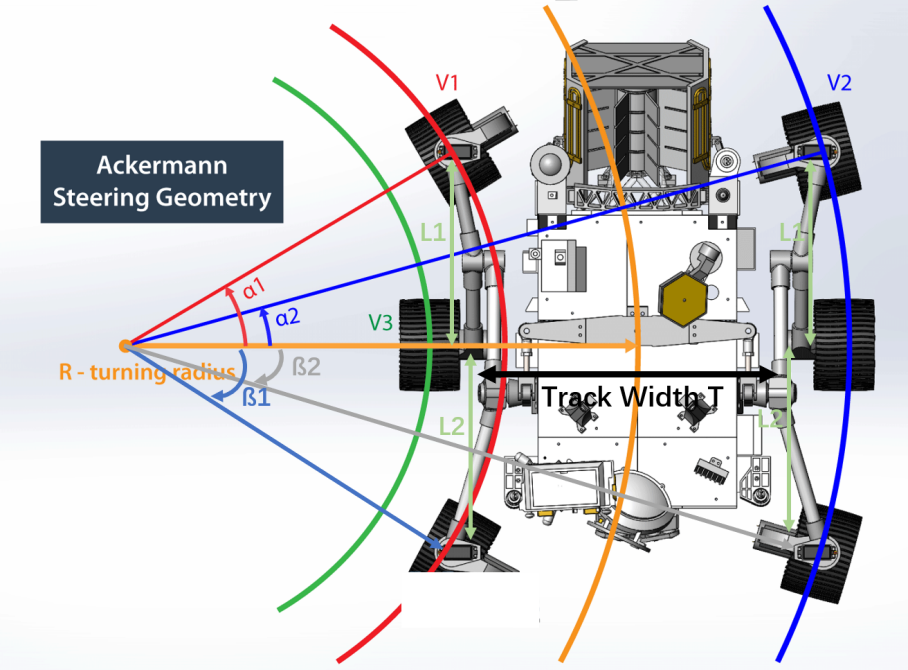

Figure 7: Ackermann-Steering-Geometry of mars car

The PCB layout for the DIY Mars "Perseverance" Rover is shown in Figure 6. This PCB integrates various components and connectors necessary for the rover's control and sensory systems.

Figure 6 provides a detailed design of the custom PCB used in the Mars rover. This PCB integrates motor drivers, power distribution, and connections for sensors and communication modules.

- **Motor Drivers**: The PCB includes multiple DRV8871 motor drivers for controlling the six DC motors. Each driver has dedicated pins for power, ground, and control signals.
- **Power Management**: Capacitors are strategically placed to smooth out voltage fluctuations. The input section handles power distribution from the buck converter and battery monitoring circuitry.
- **Communication Interfaces**: The PCB layout includes designated areas for connecting the RC receiver, GPS, compass, and other sensors. This modular design ensures ease of integration and troubleshooting.
- **Component Placement**: The layout is designed to minimize noise and interference by careful routing of traces and placement of components.

2.4 Ackermann Steering Geometry

Each wheel is powered by an independent DC motor, allowing the rover to move forward or backward. The control input use PWM signal, in which the scope is from -255 to 255.

To improve steering efficiency and minimize tire slipping during turns, the rover utilizes Ackermann steering geometry. This system allows for

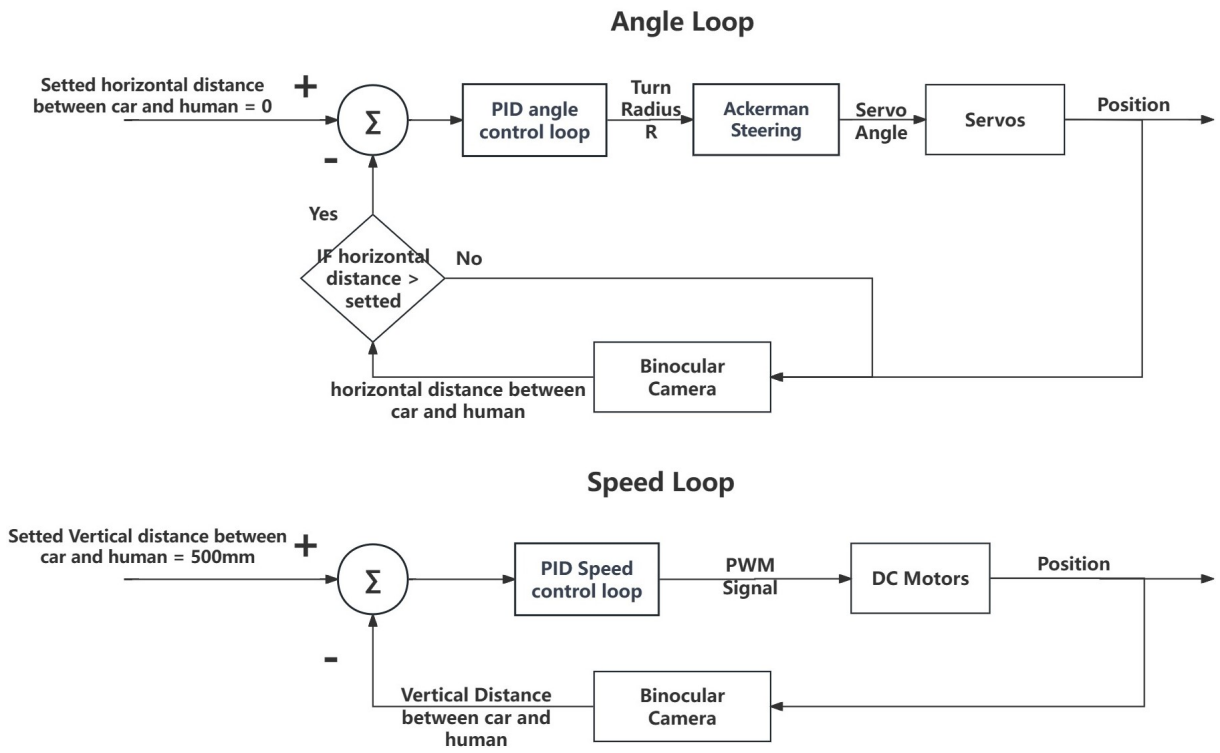

Figure 8: Dual PID Control Systems, PID Angle Loop, PID Position Loop

precise calculation of the speed and angle of each wheel based on the turning radius.

As shown in Fig. 7, the four wheels at front right, front left, back right, and back left can have independent steering angle by servo. The different curvature radius mean that to avoid sliding, the steering geometry must steer the inside front tyre at a larger angle than the outside front. Ackermann Steering refers to the geometric configuration that allows both front wheels to be steered at the appropriate angle to avoid tyre sliding.

For a given turn radius R, back wheelbase L_1, front wheelbase L_2, and track width T, engineers calculate the required front steering angles α_1, α_2 and β_1, β_2 with the following expressions:

$$\alpha_1 = \tan^{-1}\left(\frac{L}{R-\frac{T}{2}}\right) \quad \alpha_2 = \tan^{-1}\left(\frac{L}{R+\frac{T}{2}}\right) \tag{1}$$

and the back wheel steering angle yields:

$$\beta_1 = \tan^{-1}\left(\frac{L}{R-\frac{T}{2}}\right) \quad \beta_2 = \tan^{-1}\left(\frac{L}{R+\frac{T}{2}}\right) \tag{2}$$

Hence, we can set the turn radius as input to the steering system. After knowing turn radius, we could know the angle that each servo's angle.

3 Trajectory Planning, intelligent control

The control system architecture for the Mars car's trajectory planning and control system is depicted in Figure 8. The system comprises two main loops: the Angle Loop and the Speed Loop. These loops work in tandem to ensure the Mars car follows the desired path and maintains a specific distance from a human operator.

The Angle Loop is designed to maintain the correct horizontal alignment between the car and the human operator:

- **Set Horizontal Distance:** The desired horizontal distance between the car and the human is set to zero.
- **Error Calculation:** The error between the set horizontal distance and the actual horizontal distance (measured by the binocular camera) is calculated.
- **PID Angle Control Loop:** The error is fed into a PID controller, which adjusts the steering angle to minimize this error.
- **Ackerman Steering:** The PID controller output determines the turn radius using Ackerman steering principles.
- **Servo Control:** The required servo angle is then applied to the servos, adjusting the car's position to align horizontally with the human.

The Speed Loop ensures the car maintains a set vertical distance from the human operator:

- **Set Vertical Distance:** The desired vertical distance between the car and the human is set to 500mm.
- **Error Calculation:** The error between the set vertical distance and the actual vertical distance (measured by the binocular camera) is calculated.
- **PID Speed Control Loop:** This error is fed into a PID controller, which generates a PWM signal to adjust the speed of the DC motors.
- **Motor Control:** The PWM signal controls the DC motors, adjusting the car's speed to maintain the set vertical distance from the human.

4 Computer Vision Methodology

4.1 Design philosophy

The design concept of our Mars rover is to obtain the distance data between the target and the rover as well as the position data of the target in the camera frame through an external camera. Through these three data to achieve the goal of the Mars rover always follow the target movement. We use the distance data to control the forward and backward movement of the vehicle, and then use the vision algorithm to detect the distance of the target in the screen to achieve left and right turns.

4.1.1 Design Ideas for Straight Motion Programming

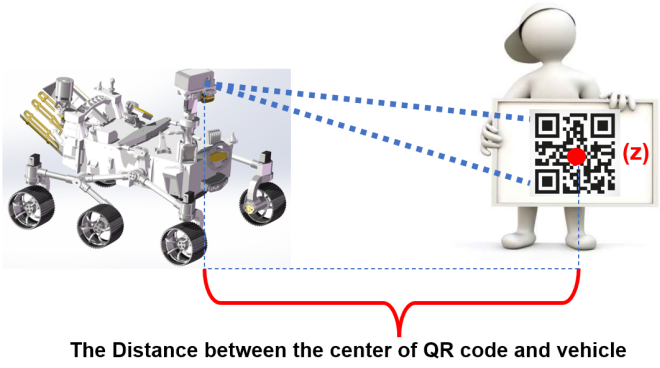

Figure 9: Design Ideas for Straight Motion

We use the binocular camera's visual detection to estimate the distance of the target from the trolley. This distance is then converted into a PWM signal for the motor, controlling the trolley's speed. If the current distance is less than the target distance, the PWM signal is negative, causing the motor to reverse and the trolley to move away. If the current distance is greater, the PWM signal is positive, moving the trolley closer to the target.

4.1.2 Design Ideas for Steering Motion Programming

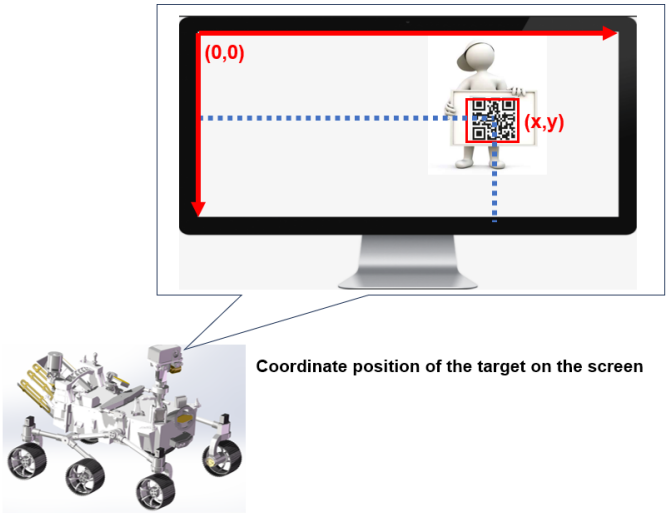

Figure 10: Design Ideas for or Steering Motion

We determine whether the cart is facing the target by detecting the coordinates of the target point on the screen. Take the centre of the screen as the reference point and get the horizontal coordinate of the centre of the screen. When the horizontal coordinate of the target point is larger than the horizontal coordinate of the centre point, the target is on the right side of the cart. When the horizontal coordinate of the target point is smaller than the horizontal coordinate of the centre point, the target is on the left side of the cart. Finally, this difference is used as an input value for the steering angle of the cart to control the size of the steering angle.

4.2 Visual detection framework

4.2.1 Visual detection model

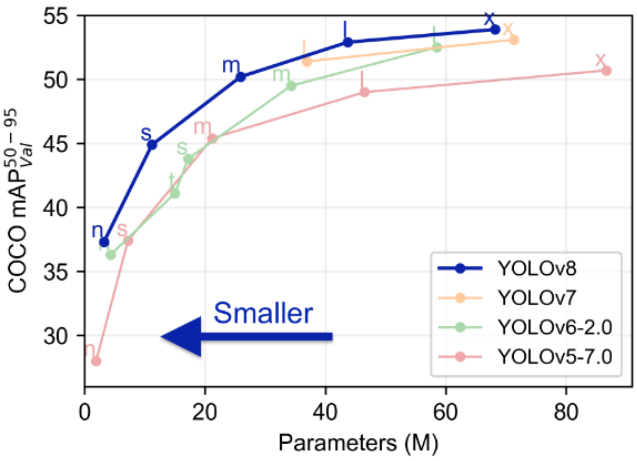

Figure 11: Comparison of YOLO model performance

The visual model framework we used is the YOLOv8 model. YOLO, as a famous model in visual detection, has been updated, among which the YOLOv8 model is a new model just released in the last few years, which refreshes several SOTA in the coco dataset, and has better detection effect compared to other models, which can meet our detection needs.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figure 12: Model selection in YOLO

YOLOv8 provides a variety of models with different parameters to choose from, and we chose the YOLOv8n model with the lowest number of parameters and the shortest detection time. The reason for this is that our main goal is to monitor in real time, so we may care more about the detection time of the model than the detection correctness.

4.2.2 Model training parameters and results

After preparing the dataset, we train the model using the following parameters:

Name	Value
Patience	100
Epoch	1000
resolution	480*640

After 1000 epochs of training, we can get the predicted loss value of the model as well as the prediction accuracy, and the final result is shown in Fig13 and Fig 14.

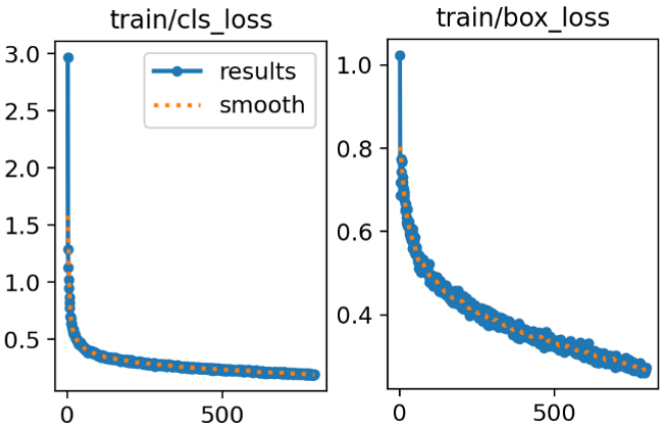

Figure 13: Model training loss

From the variation curve of the loss function, we know that after 1000 rounds of training, the rate of reduction of the loss value is already very slow and the loss value is sufficiently reduced, so we can consider 1000 rounds of training as a reasonable option.

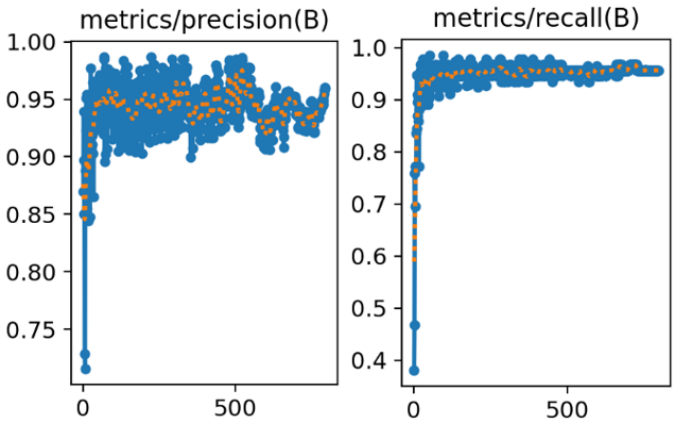

Figure 14: Model recall and presision rate

From the accuracy image and recall image, it can be learnt that after 1000 epochs of training both the accuracy and recall reached around 0.95 and remained for some time without significant improvement. Therefore, we can consider that the model has been trained at this time.

4.3 Deep Inspection Framework

4.3.1 Depth Detection Implementation Principle

A binocular camera consists of two cameras, usually placed horizontally and spaced a certain baseline distance apart. When observing the same object, due to the different positions of the two cameras, the position of the object in the images of the two cameras will have a certain difference, and this difference is called parallax. The distance from the object to the camera can be calculated from the parallax.

The depth of the object is calculated from the parallax value using the following formula:

$$Z = \frac{f \cdot B}{d} \tag{3}$$

Where, Z is the distance (depth) from the object to the camera.
f is the focal length of the camera.
B is the baseline distance between the two cameras.
d is the parallax value.

We can estimate the distance of the car from the object by using the above equation

4.3.2 Introduction to Depth Detection parameters

In the depth camera code, we define the following parameters:

Name	Value
sampling interval	0.1s
frame rate	30fps
resolution	400*640
maximum depth	10000mm
minimum depth	20mm

Using the above parameters, we can change the original image to a depth image by code, where each pixel point represents the positional distance of that point in the graph. If we use colours to differentiate between different distances, with darker colours for objects closer to us and brighter colours for objects further away from us, then our image should be like Figure 15.

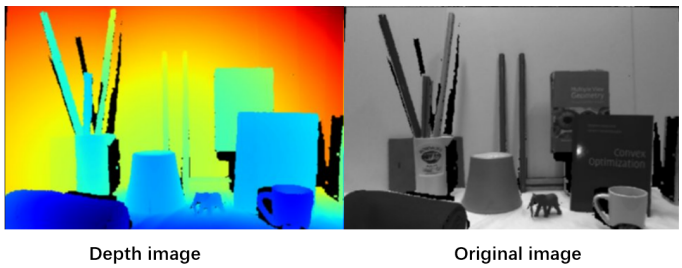

Figure 15: Converting images to depth images

4.4 Communication between documents

In this project, we have three main communication objects, which are camera, computer, and arduino development board. The underlying code for each of these three objects is different, so this provides us with a challenge for data transfer. We designed a communication system which is responsible for transferring the data of the three objects to each other, the data transfer method of this system is as follows:

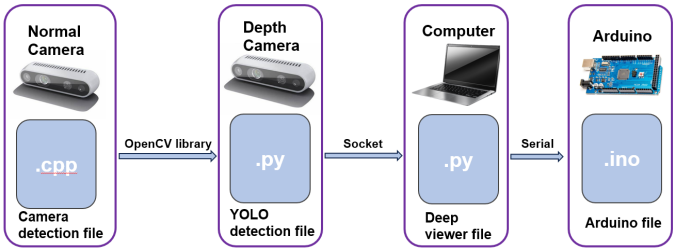

Figure 16: Diagram of the communication programs

In the whole procedure, we use the regular camera to capture the video data and use YOLO model to detect the position of the target, the result of the image processing is transferred to the depth camera via Socket to get the object depth data. The result of the depth camera will be transferred to the Arduino board through Serial communication and finally the Arduino will process the data and perform the relevant actions.

The diagram below illustrates a communication and control system designed for a Mars vehicle. The system integrates image processing, distance measurement, and PID control to navigate the vehicle towards a specified target.

A normal camera in the GeminiPro captures images containing a QR code. These images are processed by the YOLOv8 algorithm to locate the QR code and determine its coordinates (x, y) on the screen.

A deep viewer in the GeminiPro measures the distance (z) between the Mars vehicle and the target point using the QR code as a reference.

The coordinates (x, y) and distance (z) are sent to an Arduino Mega. The Arduino Mega processes these inputs to generate the appropriate control signals using a PID controller.

The PID controller computes the error between the desired and actual positions. It generates a control voltage based on proportional (K_p), integral (K_i), and derivative (K_d) terms. The controller then sends pulse-width modulation (PWM) and angle signals to adjust the vehicle's movement.

The Mars vehicle receives electronic signals from the PID controller, guiding it towards the target point based on the calculated adjustments.

A detailed flowchart can be found in the Appendix A

5 Future work

Due to the limitation of computer performance, there is still a lot of room for improvement in the sensitivity of the control of the trolley, and we can try to use a computer with more arithmetic power to improve the performance of the vehicle. In addition to this, we can also try to use models with smaller YOLO to simplify the computation and training process and improve the operation and execution efficiency.

6 Appendix

Appendix A