

Article

Model Predictive Control of a Novel Wheeled–Legged Planetary Rover for Trajectory Tracking

Jun He *, Yanlong Sun, Limin Yang and Feng Gao

State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; sunyanlong@sjtu.edu.cn (Y.S.); ylm20159@sjtu.edu.cn (L.Y.); fengg@sjtu.edu.cn (F.G.)

* Correspondence: jhe@sjtu.edu.cn

Abstract: Amid increasing demands for planetary exploration, wide-range autonomous exploration is still a great challenge for existing planetary rovers, which calls for new planetary rovers with novel locomotive mechanisms and corresponding control strategies. This paper proposes a novel wheeled–legged mechanism for the design of planetary rovers. The leg suspension utilizes a rigid–flexible coupling mechanism with a hybrid serial–parallel topology. First, the kinematic model is derived. Then, a control strategy for the wheeled–legged rover that includes a trajectory tracking module based on the model predictive control, the steering strategy, and the wheel speed allocation algorithm is proposed. After that, three groups of cosimulations with different trajectories and speeds, and experiments are carried out. Results of both the simulations and experiments validate the proposed control method.

Keywords: mobile robot; advanced intelligent control; wheeled–legged; trajectory tracking; model predictive control



Citation: He, J.; Sun, Y.; Yang, L.; Gao, F. Model Predictive Control of a Novel Wheeled–Legged Planetary Rover for Trajectory Tracking. *Sensors* **2022**, *22*, 4164. <https://doi.org/10.3390/s22114164>

Academic Editors: Luige Vladareanu, Hongnian Yu, Hongbo Wang and Yongfei Feng

Received: 19 April 2022

Accepted: 28 May 2022

Published: 30 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The planetary rovers that were deployed for the exploration of the moon and Mars, such as Curiosity and Perseverance, are purely wheeled robotic systems [1,2]. They adopt the passive rocker–bogie suspension configuration. There are two identical linkage mechanisms on each side of the rover, which consist of a rocker and a bogie. A differential mechanism is adopted to connect the two linkage mechanisms. One wheel is fixed at one end of the rocker, while the bogie has two wheels that are mounted on the other end of the rocker. Recently, China's Zhurong Mars rover adopted an active rocker–bogie suspension. There was a novel angle-adjusting mechanism between the two rockers to generate a wheel-step motion that could help the rover avoid wheel slip sinkage [3]. Although many remarkable achievements have been made in the field of planetary exploration, the capability of wide-range autonomous exploration is still a great challenge for planetary rovers.

The hybrid leg–wheel mechanism can be used in the design of planetary rovers. Legged–wheeled robots have the merits of being both wheeled and legged robots. They can robustly deal with uncertainties or disturbances caused by the unstructured discontinuous terrain encountered during planetary exploration. Moreover, they have a relatively high locomotion efficiency. There are three categories of leg–wheel robotic systems that differ according to leg morphology [4]. The first one is the serial leg configuration. For example, the Jet Propulsion Laboratory (JPL) developed an articulated–wheeled lunar robot called ATHLETE [5]. Each leg was a 6R (rotational joint) serial mechanism with six degrees of freedom (DOFs). It could roll over flat smooth terrain on rotating wheels and could also use the wheels as feet to walk over irregular and steep terrain. Grand et al. [6] addressed a wheeled–legged robot called Hylos, which had 16 actively actuated DOFs, with each leg combining a two-DOF leg and the steering and rotation DOFs in the wheel. Smith et al. [7] presented the PAW, a four-legged vehicle with a T-shaped body and compliant

legs. Each leg had two DOFs, including a passive prismatic joint. The second one was a leg with a parallel topology. Xu et al. [8] proposed a parallel legged–wheeled robotic system called BIT-NAZA, which had four parallel platforms with six DOFs and a 6-UPU (universal–prismatic–universal joints) configuration. There were four active wheels that were mounted on the feet of the parallel legs. Compared to their serial counterparts, parallel legged–wheeled systems usually have more payload and stiffness [9]. In addition, the actuators of the parallel leg system can be installed on the body, and the inertia of the moving part of the leg can be reduced. The electric devices comprising the actuator, such as the encoder and the torque sensor, can be easily protected [10]. Finally, there is the third category of hybrid wheeled–legged robots, namely transformable wheeled–legged vehicles, such as the Whegs series [11,12], Quattroped [13], Wheel Transformer [14], TurboQuad [15], and STEP [16]. For these robots, the wheel and leg morphology can be switched via the active joints. Transformable leg–wheel robots often adopt simple mechanical structures to simplify the control strategy. Thus, the stability and maneuverability of hybrid robots are inevitably sacrificed [17].

In spite of the excellent kinematic characteristics of the leg mechanisms discussed above, they cannot be directly applied to extraterrestrial exploration rovers because of the existence of special requirements such as maneuverability and security. For example, planetary rovers such as Curiosity [18] have additional wheels and legs to maintain a high level of security to overcome the tough terrains of the outer planet. Exploration rovers must have a fault tolerance feature to ensure the safety of the vehicle. These rovers can continue to move and carry out exploration missions even if one or multiple actuators are not working. For instance, the wheeled–legged rover ShearpTT [19] adopted self-locking gears in the actuator design for the suspension. There are two benefits to this: The first one is that the rover would not fall down when the actuator in the knee joint is invalid. The other is that the rover can support its own weight through the self-locking mechanism without the need for additional electrical energy due to the motor brake.

Legged suspension can change the center of gravity, the body posture, the distribution of contact forces, and even raise wheels to negotiate obstacles. Hence, controlling wheeled–legged robots is more complex than traditional exploration rovers with rocker–bogie suspension, especially in terms of trajectory tracking. Lamon et al. [20] proposed a control method for three-dimensional trajectory tracking. Furthermore, feedback control based on stereo vision efficiently improved the accuracy of trajectory tracking [21,22]. For rovers with independent front and rear steering and four wheels that are driven independently, path tracking becomes more complicated. Krid et al. [23] developed a dynamics-based tracking controller on a horizontal plane using a linear quadratic regulator (LQR). LQR controllers are able to track the line trajectory with quite a good accuracy. However, there is an obvious decrease in the accuracy with regard to steering. In contrast, model predictive control (MPC) can handle complex trajectories [24]. The control algorithm based on MPC can be derived in a recursive form, which is computationally more efficient than the other methods. The computing efficiency is a key evaluation index for planetary rovers because there are very limited computing resources in space. Though the MPC method has been applied to wheeled robots [25,26], there are some differences for wheeled–legged robots. First, wheeled–legged robots possess terrain-adaptive capabilities [27]. The leg length can be adjusted by changing the knee joint angle even if the wheel is always under the hip joint. The attitude angles of the robot can be controlled through changing the leg lengths. Thus, the terrain-adaptive capability needs to be involved in the trajectory tracking when the robot runs across irregular terrains. Second, having wheels that drive independently can lead to an uneven speed distribution, resulting in the occurrence of wheel slip [28]. For wheeled–legged rovers, the pose of the body, the wheel–soil contact force, and the height of the gravitational center can all be adjusted by coordinating the motion of the hip and knee joints when the wheeled–legged robot moves over rough terrain [29]. Both the motion of the hip and knee joints affect the motion characteristics of the wheel, resulting in slippage.

Therefore, it is necessary to provide a suitable speed for each wheel that is based on the motion characteristics of the whole robot on rough terrain.

In this paper, a novel wheeled–legged mechanism called TAWL is proposed for the design of planetary rovers. The leg suspension utilizes a rigid–flexible coupling mechanism with a hybrid serial–parallel topology. A kinematic model is derived first. Then, a control strategy for a wheeled–legged rover is proposed that includes a trajectory-tracking module based on MPC, the steering module, and the wheel speed allocation module. After that, a cosimulation model is established in both NX/Motion and Simulink software to verify the control strategy. Finally, experiments are also carried out to validate the proposed control method.

The remainder of the paper is organized as follows: Section 2 reports the hardware design and the kinematics of the rover; Section 3 details the control strategy; Section 4 presents the simulations, experiments, and the discussion of the results; and finally, Section 5 offers the conclusions.

2. Hardware and Kinematics of the Rover

2.1. Mechanical Structure

There are two aspects that need to be considered for leg design: First, the leg inertia must be as low as possible. Each leg has four DOFs, as illustrated in Figure 1, namely the hip abduction/adduction (HAA) joint, the hip flexion/extension (HFE) joint, the hip endo/exorotation (HEE), and the knee flexion/extension (KFE) joint. The HEE joint can also be used to steer the wheels. To reduce the rotational inertia of the robot’s legs, the actuators of the HFE and KFE joints are coaxially located at the hip. The KFE joint is actuated by a pantograph mechanism. In addition, to increase the driving torques of the HFE and KFE joints, a gear reducer stage was adopted at each of their output shafts. Second, compliant mechanisms are necessary for legged robotic systems to handle uncertainties or disturbances such as ground contact collisions. A telescopic structure with a passive damped spring was used for the leg design. In addition, there are two spring ball plungers that trigger the spring–damper mechanism. When the impact force from the ground exceeds the threshold value of the spring ball plunger, the spring–damper mechanism works to dissipate the impact energy. After that, the lower leg returns to its original length with the restoring force of the spring.

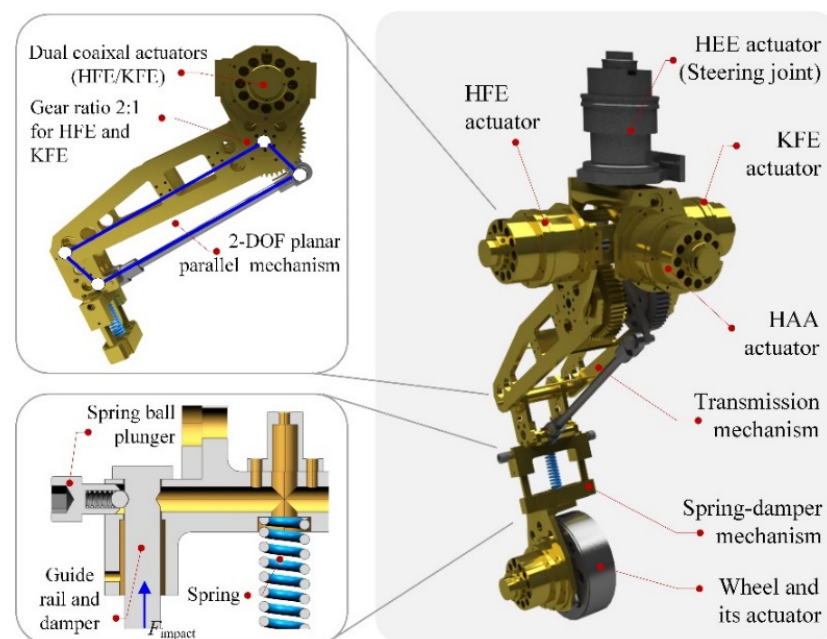


Figure 1. The leg–wheel structure.

The TAWL robot has four identical legs with wheels. The four legs are mounted to the torso in an axially symmetric distribution, as depicted in Figure 2. Its four hip joints are located on a circle with a diameter of 1.2 m. From a biological view, to obtain highly dynamic characteristics in the longitudinal direction, the ratio of the length to the width of the torso should be more than 1. However, the wheeled mode is the primary motion mode for the TAWL robot. An axially symmetrical arrangement was adopted for the robot design so that the robot would have all-directional locomotion capability in both the wheeled and legged modes (walking or trotting). Furthermore, this arrangement also increases the number of legged locomotion modes. There are at least three leg configurations for legged locomotion, i.e., the M-configuration, O-configuration, and X-configuration.



Figure 2. The TAWL robot.

2.2. Perception and Control System

There are two types of sensors: proprioceptive and exteroceptive sensors. Proprioceptive sensors contain the joint encoder, the joint torque sensor, and the inertial measurement unit (IMU), as seen in Figure 3. All of the joint angles are precisely measured by absolute encoders. Since the angle measures of each motor are absolute, the robot does not have to be homed at startup. The IMU sensor is mounted on the body and is responsible for the poses of the robot's torso. Exteroceptive sensors include visual and nonvisual sensors, which are employed to measure environmental information such as the geometrical parameters of the terrain and ground contact forces. Here, a stereo vision system was attached to the front part of the main body. Furthermore, an independent computer was implemented to deal with the vision algorithms. To improve the reliability, we did not assemble force sensors for ground contact force measurements to the end of each leg. We established a distribution measurement model and then evaluated the ground contact forces using the measurement data from the joint torque sensors.

An onboard main controller was used to run the entire control program. The main controller communicates with 20 servo drives and 16 joint torque sensors in real time via the EtherCAT industrial network protocol (Bechhoff, Verl, Germany). The measurement data from the IMU sensor are transferred into the main controller according to the RS-485 serial data standard. The main controller communicates with the visual controller by means of the ADS (automation device specifications) protocol. The TAWL robot's control software was developed using the TwinCAT software platform (Bechhoff, Verl, Germany), a real-time PC-based control system. In addition, there are two on-board lithium batteries that the robot can use to run for about 1.5 h.

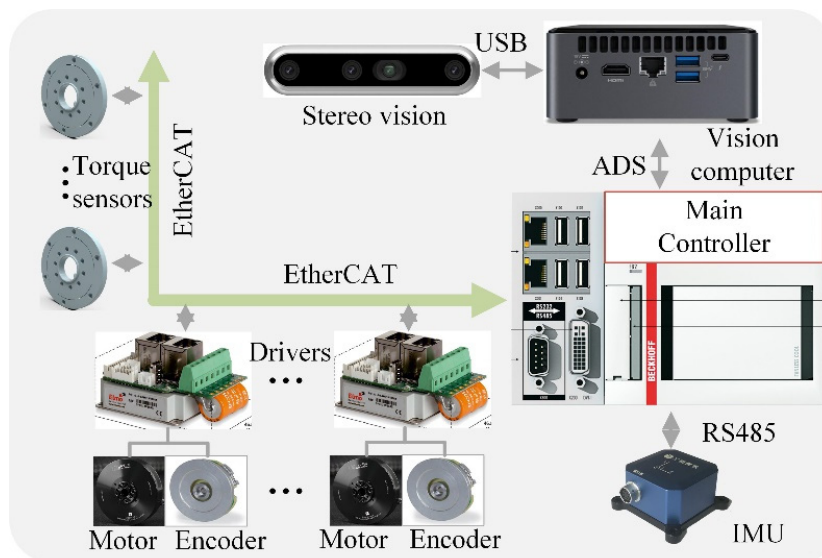


Figure 3. The perception and control system.

2.3. Kinematics of the Rover

The body frame $\{O_B-X_B Y_B Z_B\}$ of the whole robot is located at the center of the plane and is composed of the centers of four hip joints. We established the D-H coordinate systems in Figure 4 for each leg. Because each leg has the same kinematic structure, the D-H parameters of the four legs are also the same. $\{O_0-x_0 y_0 z_0\}$ is the base frame of each leg (i.e., the leg frame), which is located at the center of the hip joint. $\{O_4-x_4 y_4 z_4\}$ is the wheel frame of each leg. The D-H parameters are shown in Table 1. Therefore, the transformation matrix from frame $i - 1$ to i for the i th limb can be written as

$${}_{i-1}^i T = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where s and c denote the sine and cosine functions.

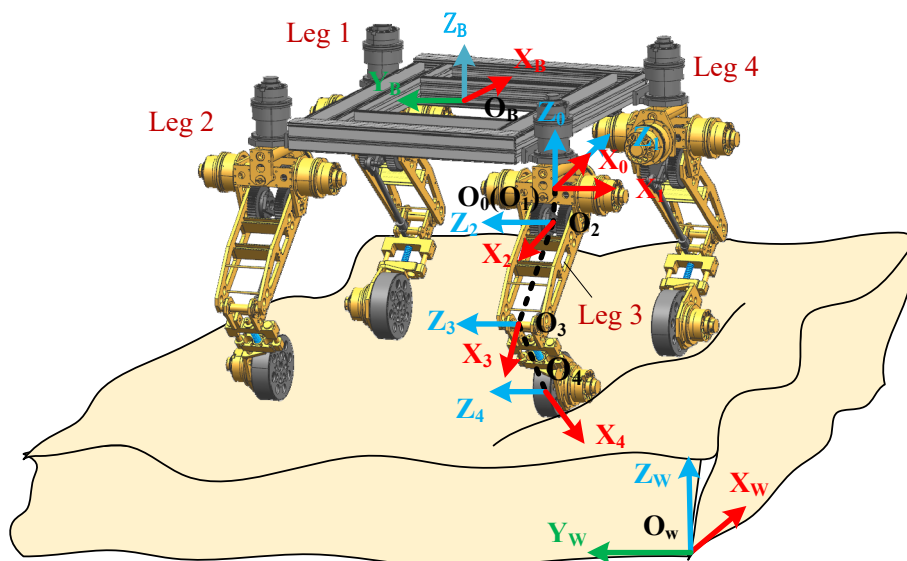


Figure 4. The coordinate systems of the rover.

Table 1. The D-H parameters.

	θ_i	d_i	α_i	a_i
1	θ_1	0	$\pi/2$	0
2	θ_2	0	$-\pi/2$	$L_1 = 87.5$
3	θ_3	0	0	$L_2 = 350$
4	θ_4	0	0	$L_3 = 310$

For the i th leg, the transformation matrix from the wheel frame to the leg frame is written as

$${}^0T^i = {}^0T^1 {}^1T^2 {}^2T^3 {}^3T^4 {}^4T^i = \begin{pmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} = \begin{pmatrix} -s_{34} & -c_{34} & 0 & -L_3 s_{34} - L_2 s_3 \\ c_{34} c_2 & -s_{34} c_2 & -s_2 & c_2 (L_1 + L_3 c_{34} + L_2 c_3) \\ c_{34} s_2 & -s_{34} s_2 & c_2 & s_2 (L_1 + L_3 c_{34} + L_2 c_3) \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2)$$

where $s_{34} = \sin(\theta_3 + \theta_4)$, $c_{34} = \cos(\theta_3 + \theta_4)$, $s_2 = \sin\theta_2$, $c_2 = \cos\theta_2$, $s_3 = \sin\theta_3$, and $c_3 = \cos\theta_3$. Here, $\theta_1 = \pi/2$ and $\mathbf{P} = (P_x, P_y, P_z)^T$ are the positions of the wheel center with respect to the leg frame.

When \mathbf{P} is given, the rotational angle of each joint, θ_2 , θ_3 , and θ_4 , can be obtained as

$$\theta_2 = \text{atan}(P_z/P_y), \quad (3)$$

$$\theta_3 = -\text{atan}\left(\frac{C}{A^2 + B^2 + C^2}\right) - \text{atan}\left(\frac{B}{A}\right), \quad (4)$$

$$\theta_4 = -\text{atan}\left(\frac{G}{\sqrt{E^2 + F^2 - G^2}}\right) - \text{atan}\left(\frac{F}{E}\right) - \theta_3 \quad (5)$$

where

$$\begin{aligned} A &= 2P_x L_2 \\ B &= -2(\sqrt{P_y^2 + P_z^2} - L_1)L_2 \\ C &= (\sqrt{P_y^2 + P_z^2} - L_1)^2 + L_2^2 + P_x^2 - L_3^2 \\ E &= 2P_x L_3 \\ F &= -2L_3(\sqrt{P_y^2 + P_z^2} - L_1) \\ G &= (\sqrt{P_y^2 + P_z^2} - L_1)^2 + P_x^2 + L_3^2 - L_2^2 \end{aligned}$$

3. Control Strategy

In this section, a control architecture for the wheeled-legged rover is proposed, as depicted in Figure 5. The control strategy consists of a planning layer, a controller layer, and a physical layer. First, the planner layer generates the reference trajectory. A planned path is generally composed of discrete points that come from the operator or the planner, which is based on a vision system. Using these points, a Bezier curve was adopted to produce a reference trajectory that included the time information. Thus, the derivation of the reference trajectory yielded the reference velocity. Second, the controller layer includes an MPC module, a steering module, and a wheel speed allocation module. The MPC module calculates the optimal control inputs through the last control inputs and the current state variables. The state variables can be estimated by a data fusion algorithm such as a Kalman filter and a particle filter, which is based on the proprioceptive and exteroceptive

sensors in the robotic system. Considering that the main purpose was to verify the trajectory tracking algorithm, the state variables, including the position and velocity of the robot, were measured by the vision motion capture system directly in the present experimental study. In addition, to eliminate the accumulated errors, a PID control method was added in the loop. Then, the steering module provided the speed and the steering angle of each wheel. After that, the wheel speed allocation algorithm was presented to avoid a wheel slip. Next, the leg joint angles were obtained through the inverse kinematics of the rover. Third, the physical layer received the steering angles, the wheel speeds, and the leg joint angles and sent these orders to servo drives.

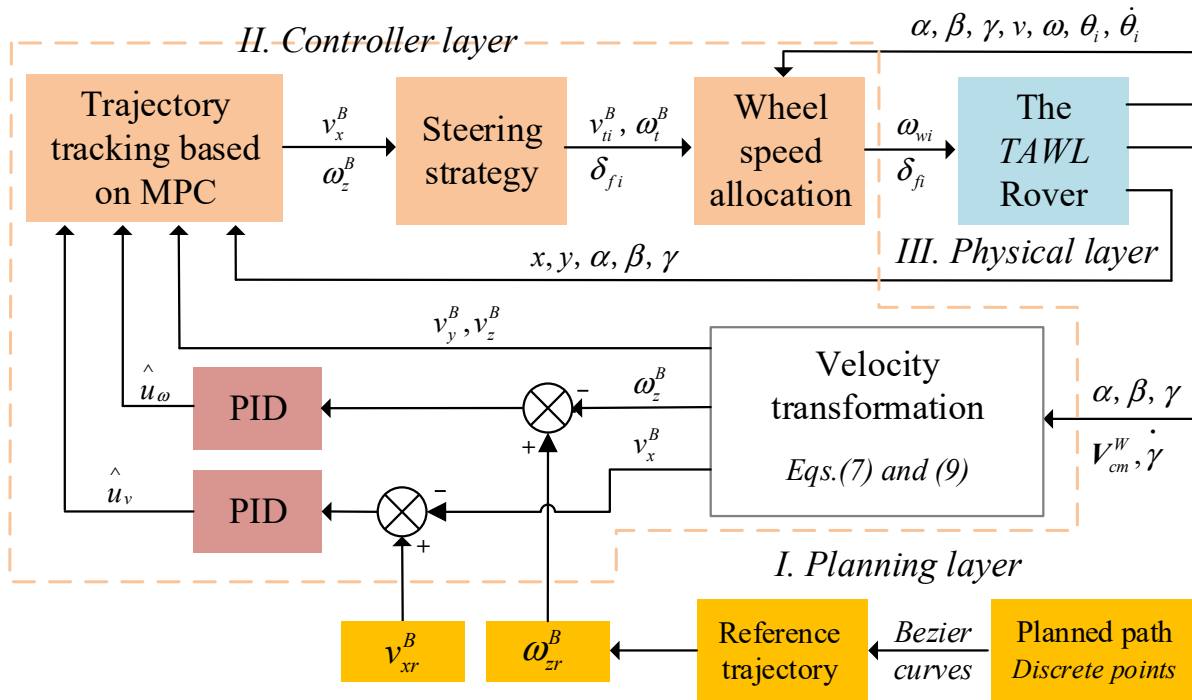


Figure 5. The rover’s control strategy.

3.1. Locomotive Equations

The rotation matrix from the body frame {B} to the world frame {W} is written as

$${}^W R_B(\alpha, \beta, \gamma) = \begin{pmatrix} c\gamma c\beta & c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ s\gamma c\beta & s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha \\ -s\beta & c\beta s\alpha & c\beta c\alpha \end{pmatrix}, \tag{6}$$

where $\alpha, \beta,$ and γ are the fixed rotational angles with respect to the $x, y,$ and z axes of the world frame, respectively.

Furthermore, the velocity of the centroid of the robot can be denoted by

$$\mathbf{V}_{cm}^W = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} c\gamma c\beta & c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ s\gamma c\beta & s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha \\ -s\beta & c\beta s\alpha & c\beta c\alpha \end{pmatrix} \mathbf{V}_{cm}^B, \tag{7}$$

where \mathbf{V}_{cm}^W and \mathbf{V}_{cm}^B are the velocities of the body centroid for frames {W} and {B}, and $\mathbf{V}_{cm}^B = (v_x^B, v_y^B, v_z^B)^T$.

The yaw angle of the robot with respect to the body frame, γ^B , can be written as

$$\tan \gamma = \tan \gamma^B c\alpha / c\beta \tag{8}$$

where γ is the yaw angle in frame {W}.

Thus, we have

$$\dot{\gamma} = \omega_z^B c\alpha / c\beta. \quad (9)$$

Accordingly, the kinematic equation for the trajectory tracking is obtained as

$$\dot{\mathbf{X}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} c\gamma c\beta & 0 \\ s\gamma c\beta & 0 \\ 0 & \frac{c\alpha}{c\beta} \end{pmatrix} \begin{pmatrix} v_x^B \\ \omega_z^B \end{pmatrix} + \begin{pmatrix} c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_y^B \\ v_z^B \end{pmatrix} \quad (10)$$

where $\mathbf{X} = (x, y, \gamma)^T$, $\mathbf{u} = (v_x^B, \omega_z^B)^T$, and $\dot{\mathbf{X}} = (\dot{x}, \dot{y}, \dot{\gamma})^T = f(\mathbf{X}, \mathbf{u}, t)$. At a reference point on the trajectory, we have $\mathbf{X}_r = (x_r, y_r, \gamma_r)^T$, $\dot{\mathbf{X}}_r = f(\mathbf{X}_r, \mathbf{u}_r, t) = (\dot{x}_r, \dot{y}_r, \dot{\gamma}_r)^T$, and $\mathbf{u}_r = (v_{xr}^B, \omega_{zr}^B)^T$.

Therefore, expanding Equation (10) in the Taylor series around the reference point $(\mathbf{X}_r, \mathbf{u}_r)$ and discarding the high order terms yields

$$\dot{\mathbf{X}} = f(\mathbf{X}_r, \mathbf{u}_r, t) + \frac{\partial f(\mathbf{X}, \mathbf{u}, t)}{\partial \mathbf{X}} \Big|_{(\mathbf{X}_r, \mathbf{u}_r)} (\mathbf{X} - \mathbf{X}_r) + \frac{\partial f(\mathbf{X}, \mathbf{u}, t)}{\partial \mathbf{u}} \Big|_{(\mathbf{X}_r, \mathbf{u}_r)} (\mathbf{u} - \mathbf{u}_r) \quad (11)$$

where

$$\begin{aligned} \frac{\partial f(\mathbf{X}, \mathbf{u}, t)}{\partial \mathbf{X}} \Big|_{(\mathbf{X}_r, \mathbf{u}_r)} &= \begin{pmatrix} \frac{\partial f_1(\mathbf{X}, \mathbf{u}, t)}{\partial x} & \frac{\partial f_1(\mathbf{X}, \mathbf{u}, t)}{\partial y} & \frac{\partial f_1(\mathbf{X}, \mathbf{u}, t)}{\partial \gamma} \\ \frac{\partial f_2(\mathbf{X}, \mathbf{u}, t)}{\partial x} & \frac{\partial f_2(\mathbf{X}, \mathbf{u}, t)}{\partial y} & \frac{\partial f_2(\mathbf{X}, \mathbf{u}, t)}{\partial \gamma} \\ \frac{\partial f_3(\mathbf{X}, \mathbf{u}, t)}{\partial x} & \frac{\partial f_3(\mathbf{X}, \mathbf{u}, t)}{\partial y} & \frac{\partial f_3(\mathbf{X}, \mathbf{u}, t)}{\partial \gamma} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & -s\gamma_r c\beta v_{xr}^B + (-s\gamma_r s\alpha s\beta - c\gamma_r c\alpha)v_y^B + (-s\gamma_r s\beta c\alpha + c\gamma_r s\alpha)v_z^B \\ 0 & 0 & c\gamma_r c\beta v_{xr}^B + (c\gamma_r s\alpha s\beta - s\gamma_r c\alpha)v_y^B + (c\gamma_r s\beta c\alpha + s\gamma_r s\alpha)v_z^B \\ 0 & 0 & 0 \end{pmatrix}, \\ \frac{\partial f(\mathbf{X}, \mathbf{u}, t)}{\partial \mathbf{u}} \Big|_{(\mathbf{X}_r, \mathbf{u}_r)} &= \begin{pmatrix} \frac{\partial f_1(\mathbf{X}, \mathbf{u}, t)}{\partial v_x^B} & \frac{\partial f_1(\mathbf{X}, \mathbf{u}, t)}{\partial \omega_z^B} \\ \frac{\partial f_2(\mathbf{X}, \mathbf{u}, t)}{\partial v_x^B} & \frac{\partial f_2(\mathbf{X}, \mathbf{u}, t)}{\partial \omega_z^B} \\ \frac{\partial f_3(\mathbf{X}, \mathbf{u}, t)}{\partial v_x^B} & \frac{\partial f_3(\mathbf{X}, \mathbf{u}, t)}{\partial \omega_z^B} \end{pmatrix} = \begin{pmatrix} c\beta c\gamma_r & 0 \\ c\beta s\gamma_r & 0 \\ 0 & c\alpha / c\beta \end{pmatrix}. \end{aligned}$$

Accordingly, the state space equation can be denoted by

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}(t)\hat{\mathbf{X}} + \mathbf{B}(t)\hat{\mathbf{u}} \quad (12)$$

where $\hat{\mathbf{X}} = (x - x_r, y - y_r, \gamma - \gamma_r)^T$, $\dot{\hat{\mathbf{X}}} = (\dot{x} - \dot{x}_r, \dot{y} - \dot{y}_r, \dot{\gamma} - \dot{\gamma}_r)^T$, $\hat{\mathbf{u}} = (\hat{u}_v, \hat{u}_\omega) = (v - v_{xr}^B, \omega - \omega_{zr}^B)^T$, $\mathbf{A}(t) = \frac{\partial f(\mathbf{X}, \mathbf{u}, t)}{\partial \mathbf{X}} \Big|_{(\mathbf{X}_r, \mathbf{u}_r)}$, and $\mathbf{B}(t) = \frac{\partial f(\mathbf{X}, \mathbf{u}, t)}{\partial \mathbf{u}} \Big|_{(\mathbf{X}_r, \mathbf{u}_r)}$. $\hat{\mathbf{X}}$ is the error with respect to the reference trajectory, and $\hat{\mathbf{u}}$ is its associated perturbation control input.

Using forward differences, the approximation of $\dot{\hat{\mathbf{X}}}$ can be obtained as the following discrete-time form:

$$\hat{\mathbf{X}}(k+1) = \mathbf{G}_k \hat{\mathbf{X}}(k) + \mathbf{H}_k \hat{\mathbf{u}}(k) \quad (13)$$

where $\mathbf{G}_k = \mathbf{T}\mathbf{A}_k + \mathbf{I}$, and $\mathbf{H}_k = \mathbf{T}\mathbf{B}_k$. T and k are the sampling period and the sampling time. \mathbf{I} is the identity matrix.

3.2. Trajectory Tracking Model Based on MPC

3.2.1. Objective Function

A controller was designed for the wheeled-legged robot to track the desired trajectory precisely and stably. By changing the current and future inputs of the control system, the optimization problem is the minimization of a predicted performance cost, which is a quadratic function of the states and control inputs as follows:

$$J(t) = \sum_{i=1}^{N_p} \hat{\mathbf{X}}^T(t+i|t) \mathbf{Q} \hat{\mathbf{X}}(t+i|t) + \sum_{j=1}^{N_c-1} \Delta \hat{\mathbf{u}}^T(t+i|t) \mathbf{R} \Delta \hat{\mathbf{u}}(t+i|t) + \rho \varepsilon^2, \quad (14)$$

where N_p and N_c are the prediction and control horizons, respectively. Here, Q and R are the weighting matrices; ρ is the weight coefficient, and ε is the relaxation factor.

Let

$$\tilde{\zeta}(k|t) = \begin{pmatrix} \hat{X}(k|t) \\ \hat{u}(k-1|t) \end{pmatrix}, \tag{15}$$

we obtain

$$\zeta(k+1|t) = \hat{A}_k \zeta(k|t) + \hat{B}_k \Delta u(k|t), \tag{16}$$

$$\eta(k|t) = \hat{C}_k \zeta(k|t), \tag{17}$$

where $\hat{A}_k = \begin{pmatrix} \mathbf{G}_k & \mathbf{H}_k \\ \mathbf{0} & I \end{pmatrix}$, and $\hat{B}_k = \begin{pmatrix} \mathbf{H}_k \\ I \end{pmatrix}$.

Furthermore, Equations (16) and (17) can be rewritten as the following matrix form:

$$\mathbf{Y}(t) = \mathbf{\Omega} \zeta(t) + \mathbf{\Phi} \Delta \mathbf{U} \tag{18}$$

where

$$\mathbf{Y}(t) = \begin{pmatrix} \eta(k+1|t) \\ \eta(k+2|t) \\ \dots \\ \eta(k+N_c|t) \\ \dots \\ \eta(k+N_p|t) \end{pmatrix}, \mathbf{\Omega} = \begin{pmatrix} \hat{C}_k \hat{A}_k \\ \hat{C}_k \hat{A}_k^2 \\ \dots \\ \hat{C}_k \hat{A}_k^{N_c} \\ \dots \\ \hat{C}_k \hat{A}_k^{N_p} \end{pmatrix}, \Delta \mathbf{U} = \begin{pmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \dots \\ \Delta u(t+N_c|t) \end{pmatrix}, \text{ and}$$

$$\mathbf{\Phi} = \begin{pmatrix} \hat{C}_k \hat{B}_k & 0 & 0 & 0 \\ \hat{C}_k \hat{A}_k \hat{B}_k & \hat{C}_k \hat{B}_k & 0 & 0 \\ \dots & \dots & \ddots & \dots \\ \hat{C}_k \hat{A}_k^{N_c-1} \hat{B}_k & \hat{C}_k \hat{A}_k^{N_c-2} \hat{B}_k & \dots & \hat{C}_k \hat{B}_k \\ \hat{C}_k \hat{A}_k^{N_c} \hat{B}_k & \hat{C}_k \hat{A}_k^{N_c-1} \hat{B}_k & \dots & \hat{C}_k \hat{A}_k \hat{B}_k \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_k \hat{A}_k^{N_p-1} \hat{B}_k & \hat{C}_k \hat{A}_k^{N_p-2} \hat{B}_k & \dots & \hat{C}_k \hat{A}_k^{N_p-N_c-1} \hat{B}_k \end{pmatrix}$$

Equations (14) and (18) yield

$$\begin{aligned} J(t) &= \Delta \mathbf{U}^T(t) \mathbf{R} \Delta \mathbf{U}(t) + \mathbf{Y}^T(t) \mathbf{Q} \mathbf{Y}(t) + \rho \varepsilon^2 \\ &= \Delta \mathbf{U}^T(t) \mathbf{R} \Delta \mathbf{U}(t) + (\mathbf{\Phi} \Delta \mathbf{U}(t))^T \mathbf{Q} (\mathbf{\Phi} \Delta \mathbf{U}(t)) \\ &\quad + 2(\mathbf{\Omega} \zeta(t))^T \mathbf{Q} (\mathbf{\Phi} \Delta \mathbf{U}(t)) + (\mathbf{\Omega} \zeta(t))^T \mathbf{Q} (\mathbf{\Omega} \zeta(t)) + \rho \varepsilon^2. \end{aligned} \tag{19}$$

Here, $\mathbf{\Omega} \zeta(t)$ is not affected by the inputs and can thus be discarded. Therefore, the objective function is rewritten as a standard quadratic form:

$$J(t) = (\Delta \mathbf{U}^T(t) \quad \varepsilon) \mathbf{H}(t) (\Delta \mathbf{U}^T(t) \quad \varepsilon)^T + F(t) (\Delta \mathbf{U}^T(t) \quad \varepsilon)^T, \tag{20}$$

where

$$\mathbf{H}(t) = \begin{pmatrix} \mathbf{\Phi}^T \mathbf{Q} \mathbf{\Phi} + \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \rho \end{pmatrix}, F(t) = \begin{pmatrix} 2(\mathbf{\Omega} \zeta(t))^T \mathbf{Q} \mathbf{\Phi} & \mathbf{0} \end{pmatrix}.$$

3.2.2. Constraints

There are some constraints when the wheeled-legged robot carries out trajectory tracking tasks. The amplitude of the control input \mathbf{u} and control input increment $\Delta \mathbf{u}$ satisfy

$$\mathbf{u}_{min}(t+k|t) \leq \mathbf{u}(t+k|t) \leq \mathbf{u}_{max}(t+k|t), k = 0, 1 \dots N_c - 1, \tag{21}$$

$$\Delta \mathbf{u}_{min}(t+k|t) \leq \Delta \mathbf{u}(t+k|t) \leq \Delta \mathbf{u}_{max}(t+k|t), k = 0, 1 \dots N_c - 1, \tag{22}$$

where u_{min} and Δu_{min} are the predefined lower bounds, and u_{max} and Δu_{max} are the predefined upper bounds. Furthermore, the variable to be solved in the objective function are the control increment in the control horizon. Therefore, the constraints need be converted into the product form of the control increment and the transformation matrix.

The following relationship exists:

$$u(t+k|t) = u(t+k-1|t) + \Delta u(t+k|t). \tag{23}$$

Furthermore, Equation (23) can be reformulated as a matrix form:

$$\mathbf{U}(t) = \mathbf{E}\Delta\mathbf{U}(t) + \mathbf{U}(t-1) \tag{24}$$

where

$$\mathbf{U}(t) = \begin{pmatrix} u(t|t) \\ u(t+1|t) \\ \vdots \\ u(t+N_c-1|t) \end{pmatrix}, \mathbf{E} = \begin{pmatrix} I & 0 & 0 & 0 \\ I & I & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & I & I \end{pmatrix},$$

$$\Delta\mathbf{U}(t) = \begin{pmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \vdots \\ \Delta u(t+N_c-1|t) \end{pmatrix}, \mathbf{U}(t-1) = \begin{pmatrix} u(t-1) \\ u(t-1) \\ \vdots \\ u(t-1) \end{pmatrix}.$$

Moreover, from Equations (19) and (23), we obtain

$$\mathbf{U}_{min}(t) \leq \mathbf{E}\Delta\mathbf{U}(t) + \mathbf{U}(t-1) \leq \mathbf{U}_{max}(t), \tag{25}$$

where

$$\mathbf{U}_{min}(t) = \begin{pmatrix} u_{min}(t|t) \\ u_{min}(t+1|t) \\ \vdots \\ u_{min}(t+N_c-1|t) \end{pmatrix}, \mathbf{U}_{max}(t) = \begin{pmatrix} u_{max}(t|t) \\ u_{max}(t+1|t) \\ \vdots \\ u_{max}(t+N_c-1|t) \end{pmatrix}.$$

For the control increment, we have

$$\Delta\mathbf{U}_{min}(t) \leq \Delta\mathbf{U}(t) \leq \Delta\mathbf{U}_{max}(t), \tag{26}$$

where

$$\Delta\mathbf{U}_{min}(t) = \begin{pmatrix} \Delta u_{min}(t|t) \\ \Delta u_{min}(t+1|t) \\ \vdots \\ \Delta u_{min}(t+N_c-1|t) \end{pmatrix}, \Delta\mathbf{U}_{max}(t) = \begin{pmatrix} \Delta u_{max}(t|t) \\ \Delta u_{max}(t+1|t) \\ \vdots \\ \Delta u_{max}(t+N_c-1|t) \end{pmatrix}.$$

Accordingly, Equations (20), (25) and (26) yield the following quadratic programming problem

$$J(t) = (\Delta\mathbf{U}^T(t) \quad \varepsilon)\mathbf{H}(t)(\Delta\mathbf{U}^T(t) \quad \varepsilon)^T + \mathbf{F}(t)(\Delta\mathbf{U}^T(t) \quad \varepsilon)^T, \tag{27}$$

$$s.t. \mathbf{U}_{min}(t) \leq \mathbf{E}\Delta\mathbf{U}(t) + \mathbf{U}(t-1) \leq \mathbf{U}_{max}(t),$$

$$\Delta\mathbf{U}_{min}(t) \leq \Delta\mathbf{U}(t) \leq \Delta\mathbf{U}_{max}(t).$$

Solving Equation (27) in each control cycle leads to a series of control increments in the control time domain:

$$\Delta\mathbf{U}^*(t) = (\Delta u^*(t|t) \quad \Delta u^*(t+1|t) \quad \dots \quad \Delta u^*(t+N_c-1|t)), \tag{28}$$

Furthermore, the first element in the sequence was adopted for the actual control increment

$$u^*(t|t) = u(t - 1|t) + \Delta u^*(t|t). \tag{29}$$

Finally, by repeating the above process in each control cycle, the desired trajectory is tracked.

3.3. Steering Strategy

Using the aforementioned MPC method, we can obtain the optimal control inputs, $u^*(t|t) = (v_x^B, \omega_z^B)$. Furthermore, the speed and the steering angle of each wheel need to be derived. In the present study, the steering strategy in which all of the wheels make the uniform circular motion was adopted, as seen in Figure 6.

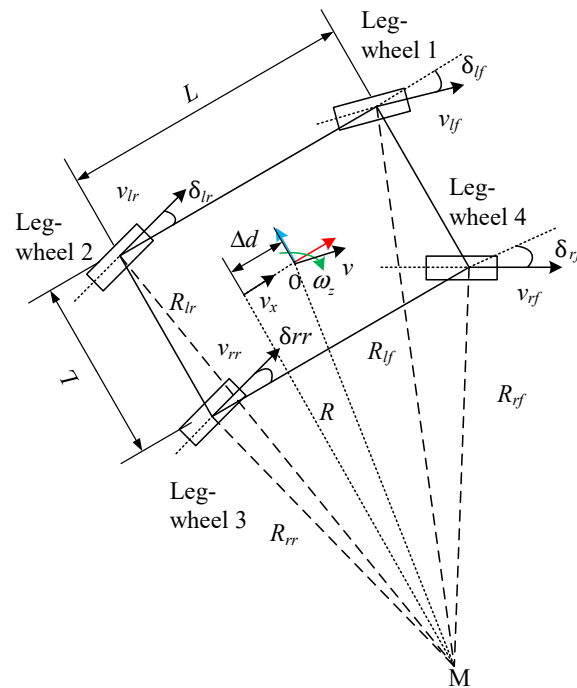


Figure 6. The steering strategy.

The point M is the steering center and the steering radius of the robot R is

$$R = \frac{v_x^B}{\omega_z^B}, \tag{30}$$

where $\omega_z^B \neq 0$. Furthermore, the steering radii of the four wheels are written as

$$\begin{cases} R_{rf} = \sqrt{\left(\frac{L}{2} - \Delta d\right)^2 + \left(R - \frac{L}{2}\right)^2} \\ R_{lf} = \sqrt{\left(\frac{L}{2} - \Delta d\right)^2 + \left(R + \frac{L}{2}\right)^2} \\ R_{rr} = \sqrt{\left(\frac{L}{2} + \Delta d\right)^2 + \left(R - \frac{L}{2}\right)^2} \\ R_{lr} = \sqrt{\left(\frac{L}{2} + \Delta d\right)^2 + \left(R + \frac{L}{2}\right)^2} \end{cases} \tag{31}$$

According to Ackermann's principle, the wheel speeds and the steering angles are obtained as follows:

$$\begin{cases} \omega_{rf} = \frac{\omega_z^B R_{rf}}{R_w} \\ \omega_{lf} = \frac{\omega_z^B R_{lf}}{R_w} \\ \omega_{rr} = \frac{\omega_z^B R_{rr}}{R_w} \\ \omega_{lr} = \frac{\omega_z^B R_{lr}}{R_w} \end{cases} \tag{32}$$

$$\begin{cases} \delta_{rf} = \text{sign}(k) \tan^{-1} \frac{\frac{l}{2} - \Delta d}{R - L/2} \\ \delta_{lf} = \text{sign}(k) \tan^{-1} \frac{\frac{l}{2} - \Delta d}{R + L/2} \\ \delta_{rr} = -\text{sign}(k) \tan^{-1} \frac{\frac{l}{2} + \Delta d}{R - L/2} \\ \delta_{lr} = -\text{sign}(k) \tan^{-1} \frac{\frac{l}{2} + \Delta d}{R + L/2} \end{cases} \quad (33)$$

where ω_{rf} , ω_{lf} , ω_{rr} , and ω_{lr} are the wheel speeds, and δ_{rf} , δ_{lf} , δ_{rr} , and δ_{lr} are the steering angles. Here, $\text{sign}(k)$ is a signum function, and $\text{sign}(k) = -1$ when the wheel rotates clockwise; $\text{sign}(k) = 1$ when the wheel rotates anticlockwise. When $\omega_z^B = 0$ in Equation (30), the four steering angles are all zero, i.e., $\delta_{rf} = \delta_{lf} = \delta_{rr} = \delta_{lr} = 0$.

3.4. Wheel Speed Allocation (WSA)

The WSA module calculates the suitable speed for each wheel according to the motion characteristics of the whole robot on rough terrain.

First, according to the kinematic equations, the linear velocity of the wheel center and the angular velocity of the lower leg (i.e., Frame 4) in Figure 4 are written as

$$\begin{pmatrix} v_{iw} \\ \omega_{iw} \end{pmatrix} = \begin{pmatrix} {}^4_B R_i & {}^4_B R_i S_{iw} \\ 0 & {}^4_B R_i \end{pmatrix} \begin{pmatrix} v_{i0}^B \\ \omega_{i0}^B \end{pmatrix} + {}^4_0 R_i J_i \begin{pmatrix} \dot{\theta}_{i1} \\ \dot{\theta}_{i2} \\ \dot{\theta}_{i3} \\ \dot{\theta}_{i4} \end{pmatrix}, \quad (34)$$

where S_{iw} is the position vector of the wheel center with respect to the body frame; ${}^4_B R_i$ is the rotation transformation matrix from the body frame to frame 4; ${}^4_0 R_i$ is the rotation transformation matrix from the leg frame to frame 4; J_i is the Jacobian matrix with respect to the leg frame; $i = 1-4$ denotes the leg number; and v_{i0}^B and ω_{i0}^B are the linear and angular velocities of the leg frame with respect to the body frame, which are given by

$$\begin{pmatrix} v_{i0}^B \\ \omega_{i0}^B \end{pmatrix} = \begin{pmatrix} \omega^B \times (O_{0i} - M) \\ \omega^B \end{pmatrix}, \quad (35)$$

where ω^B is the angular velocity of the body.

Second, the ideal (no-slip) linear velocity of the wheel center comes from the driving motor and the rotation of the lower leg, which can be denoted by

$$\|v_{iw}\| = (\omega_{iwz} + \omega_{id})R_w, \quad (36)$$

where ω_{iwz} is the projection of ω_{iw} on the direction of the wheel axis.

Accordingly, the rotational speeds of the wheel motors are obtained by

$$\omega_{id} = \frac{\|v_{iw}\|}{R_w} - \omega_{iwz}. \quad (37)$$

4. Results and Discussion

4.1. Simulations

The numerical program of the control strategy was first developed using MATLAB software. Then, the joint simulation model was established by SIMULINK and UG Motion software. UG motion software provides the joint angles as well as the pitch and roll angles for each control block. In the meantime, the control blocks calculate the joint angles and wheel speeds and provide them to the virtual prototype. There are some system and control parameters that can be grouped into three categories, namely the input parameters, the output parameters, and the control parameters, as seen in Table 2. The input parameters include the points on the reference path, the desired yaw angle of the robot body, and the desired linear and angular velocities of the robot body. The output parameters are the

linear and angular velocities of the robot body from the MPC and the steering angles and the wheel speeds. In addition, there are control parameters, including the prediction and control horizons, the weight coefficient and the relaxation factor, and the PID parameters to control the linear and angular velocities of the robot body.

Table 2. The system and control parameters.

Categories	Terminology	Definition
Input parameters	x_{cr}^w, y_{cr}^w	Points on reference path
	γ_r^w	Desired yaw angle of body
	v_{xr}^B, ω_{zr}^B	Desired linear and angular velocities of body
Control parameters	N_p, N_c	Prediction and control horizons
	ρ, ε	Weight coefficient and the relaxation factor
	k_{p1}, k_{i1}, k_{d1}	PID parameters for control of the linear velocity of body
	k_{p2}, k_{i2}, k_{d2}	PID parameters for control of the angular velocity of body
	$u^*(t t) = (v_x^B, \omega_z^B)$	The linear and angular velocities of the body
Output parameters	$\delta_{rf}, \delta_{lf}, \delta_{lr}$	Steering angles
	ω_{id}	Rotational speeds of the wheel motors

First, the WSA module was verified. For the simulation, the terrain included two trapezoid and two arc obstacles. The posture as well as the linear and angular velocities of the body, the joint angles and the angular velocities of the joints, and the driving speeds of the wheels were all measured using the virtual model in UG Motion software. Note that the terrain with the obstacles led to the changes of rover attitude angles (α, β). Here, the terrain-adaptive algorithm in [27] was adopted to control the rover attitude. With this algorithm, the robot attitude was almost kept unchanged in irregular terrains. Then, the practical linear velocity of the wheel centers and the practical angular velocity of the wheels were calculated according to Equations (36) and (37). Therefore, the slip percent could be obtained from

$$\mu = \frac{((\omega_{id} + \omega_{iwz})R_w - v_i)}{(\omega_{id} + \omega_{iwz})R_w} \times 100\%, \quad (38)$$

where R_w is the radius of the wheel; ω_{id} is the practical angular velocity of the wheel, which can be measured by the wheels' encoders; and v_i is the practical linear velocity of the wheel center. Figure 7 shows the comparison of the slippage percentages with and without the WSA module. Without the WSA module, the slippage reached up to 0.25, while with the WSA, the maximum of the slippage was less than 0.13. It was found that wheel slip was obviously decreased by the WSA component.

Second, the trajectory tracking based on MPC was verified. An arc trajectory with a radius of 30 m in the plane was selected for the validation simulation. In the simulations, there were three speeds, i.e., 0.1 m/s, 0.2 m/s, and 0.4 m/s. The control parameters for the simulations of the linear trajectory were: $N_p = 6$; $N_c = 3$; $\rho = 10$; $\varepsilon = 0$ for the lower limit and $\varepsilon = 10$ for the upper limit. The PID parameters for the linear velocity were set as: $k_{p1} = 2$, $k_{i1} = 1$, $k_{d1} = 0$. Since there was only a linear velocity in the body frame, the PID module for the control of angular velocities did not work. Figures 8–10 show the trajectory tracking results for three speeds. It was found that the robot could track the corresponding target values within a short period of time under the three different speeds. There was a large increase in trajectory errors at the beginning of the tracking process. The reason for this is that the initial direction of the target speed was the same as the x axis in the global coordinate system. There was an obvious delay before the actual speed reached the target value, and the speed error was relatively larger at the beginning. Furthermore, it was found that there were obvious overshoots in the velocity responses from the MPC method in the beginning. These overshoots facilitated trajectory tracking, and thus, the forward velocity of the robot could approximate the desired value quickly. In the meantime, the overshoot

increased as the desired speed increases. The overshoot at 0.4 m/s was the largest one among the three forward speeds. It should also be noted that the final velocity response errors increased as the target speed increased. However, as a whole, the trajectory errors for the three speeds were all relatively small, validating the MPC module and the whole control strategy.

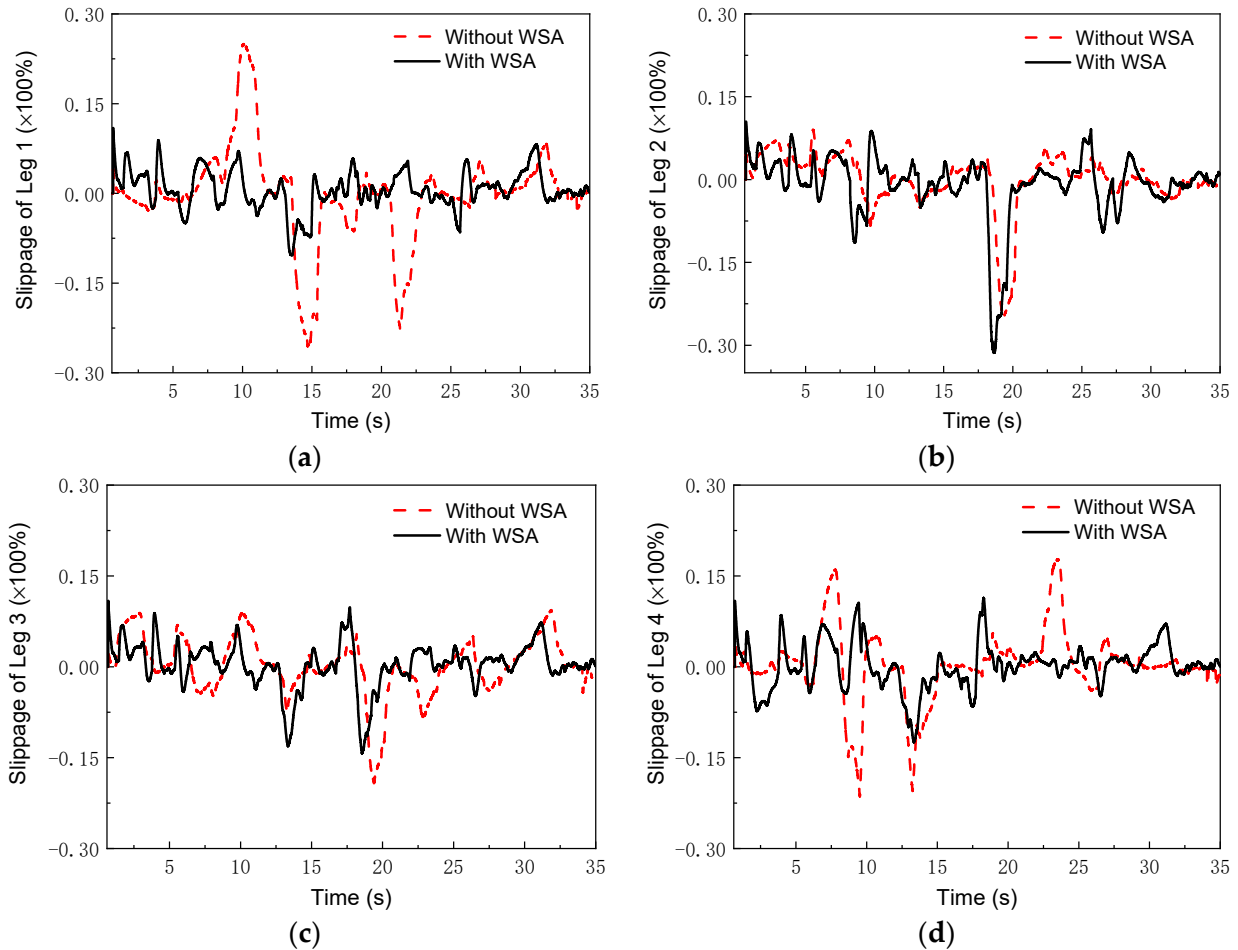


Figure 7. Slippage in simulations: (a) leg 1; (b) leg 2; (c) leg 3; (d) leg 4.

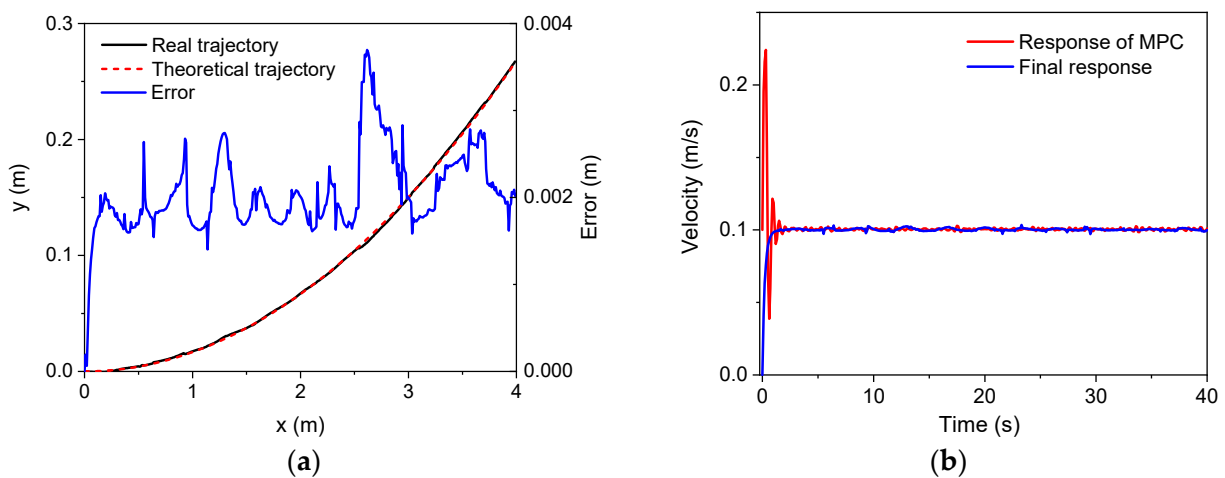


Figure 8. The simulation results ($v = 0.1$ m/s): (a) trajectory; (b) velocity.

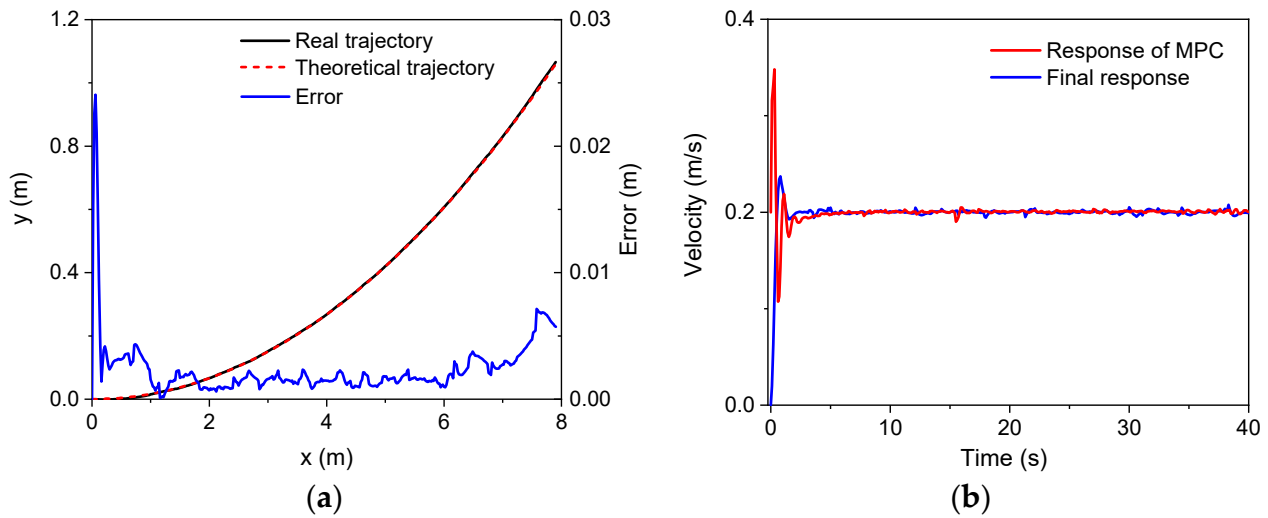


Figure 9. The simulation results ($v = 0.2$ m/s): (a) trajectory; (b) velocity.

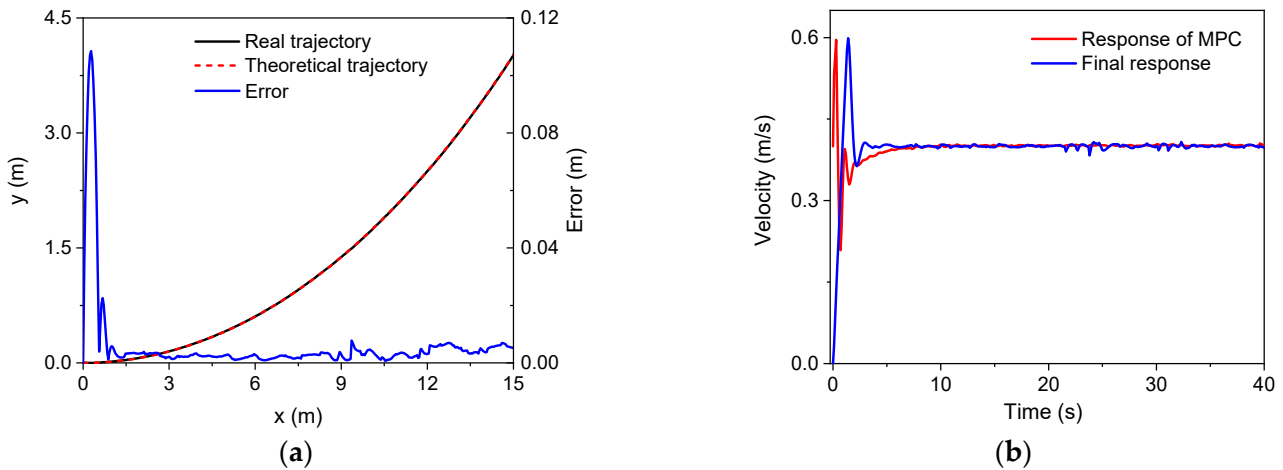


Figure 10. The simulation results ($v = 0.4$ m/s): (a) trajectory; (b) velocity.

To further validate the control strategy, a more complicated trajectory, i.e., an S-type trajectory was selected for the tracking simulations. The S-type trajectory consisted of two semicircles with a radius of $R = 20$ m, which can be described as follows:

$$x_{cr}^w = \begin{cases} R \sin(\omega_z t), & 0 \leq t < T_s \\ R \sin[-\omega_z(t - T_s)], & T_s \leq t \leq 2T_s \end{cases} \quad (39)$$

$$y_{cr}^w = \begin{cases} R - R \cos(\omega_z t), & 0 \leq t < T_s \\ 2R - R \cos(\omega_z T_s) - R \cos[-\omega_z(t - T_s)], & T_s \leq t \leq 2T_s \end{cases} \quad (40)$$

$$\gamma_{cr}^w = \begin{cases} \omega_z t, & 0 \leq t < T_s \\ \omega_z T_s - \omega_z(t - T_s), & T_s \leq t \leq 2T_s \end{cases} \quad (41)$$

where $v_x = 0.4$ m/s; $\omega_z = v_x/R$; $T_s = \pi/\omega_z$. During the simulation, the control parameters were set as: $N_p = 6$; $N_c = 3$; $\rho = 10$; $\varepsilon = 0$ for the lower limit and $\varepsilon = 10$ for the upper limit. The PID parameters for the linear velocity were set as: $k_{p1} = 0.7$, $k_{i1} = 0$, $k_{d1} = 5$. Figure 11 gives the comparisons of the theoretical and real trajectories and velocities. It was found that the robot could track well the reference trajectory and the reference velocity when running along the S-type trajectory. Furthermore, the errors in the x and y coordinates were very small, the relative error of which were less than 2% and 1.75%, respectively, as seen in Figure 12.

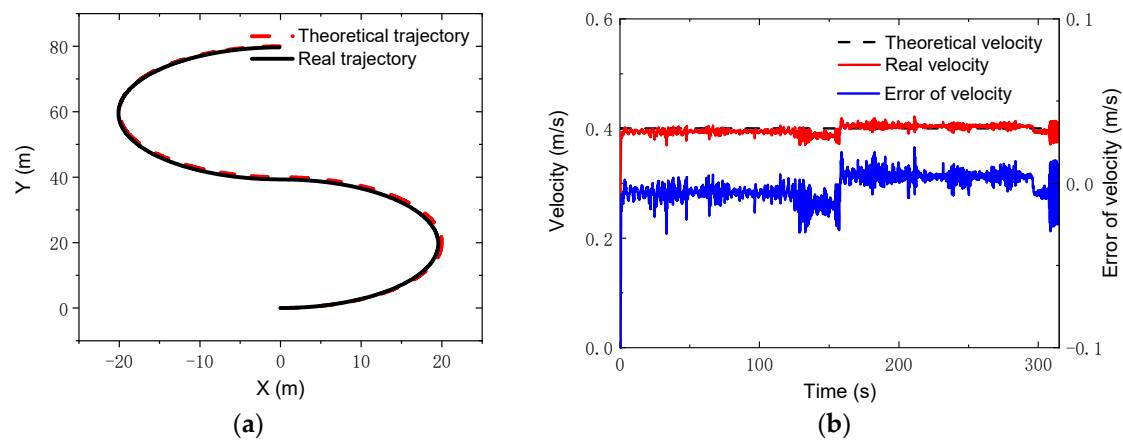


Figure 11. S-type trajectory and velocity: (a) trajectories; (b) velocities and errors.

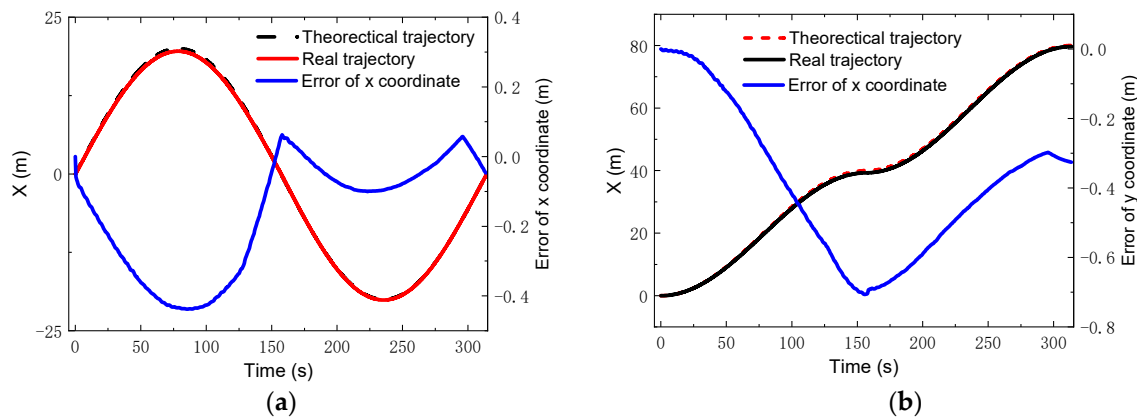


Figure 12. Coordinates and errors for S-type trajectory: (a) x coordinate; (b) y coordinate.

In addition, tracking simulations for high speeds were also carried out. Two speeds, i.e., $v_x = 2$ m/s and 4 m/s, were chosen, which were ten times as large as the speeds in the previous simulations. A circle trajectory with a radius of $R = 35$ m was selected for the simulations, which can be described by

$$\begin{cases} x_{cr}^w = R \sin(\omega_z t) \\ y_{cr}^w = R - R \cos(\omega_z t) \\ \gamma_r^w = \omega_z t \end{cases}, \quad (42)$$

where $\omega_z = v_x/R$. The control parameters for the simulations of the linear trajectory were set as: $N_p = 6$; $N_c = 3$; $\rho = 10$; for the lower limit and $\varepsilon = 10$ for the upper limit. The PID parameters for the linear velocity were set as: $k_{p1} = 2$, $k_{i1} = 1$, $k_{d1} = 0$. Figure 13 gives the changes of the real velocities. It was found that the robot could still track the reference velocity after a relatively short time. Moreover, with the control, the robot could track the reference trajectories of both the x coordinate and the y coordinate, depicted in Figures 14 and 15. Compared to the lower speed, the position errors increased. However, the relative errors of the position points were small. The maxima of the relative position errors at 2 m/s and 4 m/s were less than 3% and 8.5%, respectively.

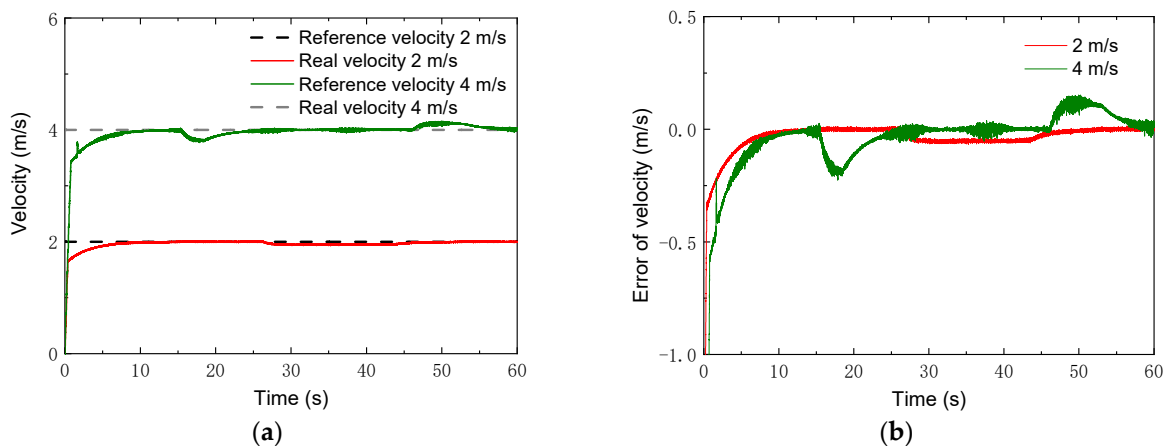


Figure 13. Velocity results with higher speeds: (a) velocity; (b) error of velocity.

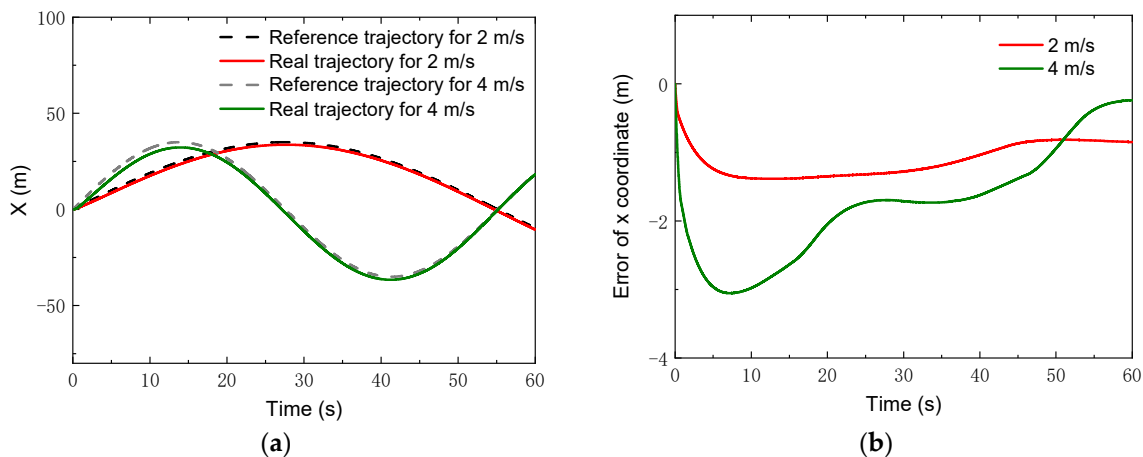


Figure 14. x Coordinates with higher speeds: (a) x coordinate; (b) error of x coordinate.

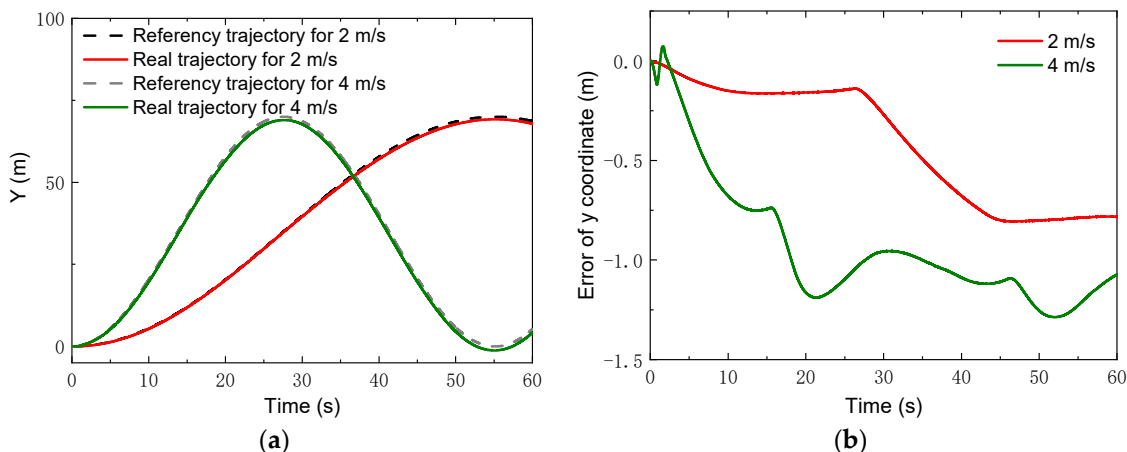


Figure 15. y Coordinates with higher speeds: (a) y coordinate; (b) error of y coordinate.

4.2. Experiments

To further verify the control strategy, an experimental setup based on the NOKOV vision motion capture system was established that consisted of six cameras, as shown in Figure 16. An L-type tool was used for the benchmark calibration of the vision system. After there were enough cameras placed around the robot, the vision capture system was calibrated. The L-type tool was mounted on the body, as seen in Figure 17. There are four markers on the L-type tool. The cameras recognize the markers, and thus, the vision

frame (i.e., the global coordinate system) can be established. According to the geometrical relationship between the mounting location of the L-type tool and the robot's body frame, the initial transformation matrix between the vision frame and the body frame could be obtained. Several markers were bonded to the body of the rover, and thus, the body coordinate system could be established in the world coordinate system. Therefore, the real motion trajectory of the rover could be measured in real time and sent to the rover control system. To clarify the effectiveness of the WSA module, a terrain with a flat surface and two trapezoids was employed in the experiments. Since the terrain included obstacles, the terrain-adaptive algorithm [27] was adopted, similar to the simulations. During the experiments, the control parameters for the MPC module were set as: $N_p = 6$; $N_c = 3$; $\rho = 10$; $\varepsilon = 0$ for the lower limit and $\varepsilon = 10$ for the upper limit. The PID parameters for the control of linear velocities were set as: $k_{p1} = 8$, $k_{i1} = 0$, $k_{d1} = 0.2$. Figure 18 shows the slippage in the experiments. Note that the slippages of Leg 1 and 4 (Leg 2 and 3) are almost the same because they suffer the same terrain condition. It was found that the average slippage of all of the legs demonstrated an obvious decrease after WSA control, up to 20%. Figure 19 shows the experimental results of trajectory tracking. As it can be seen, the rover could strictly track the reference trajectory. The deflection with respect to the reference trajectory was less than 2% F.S., which is a relatively small error. Accordingly, the control strategy was validated by the experiments.

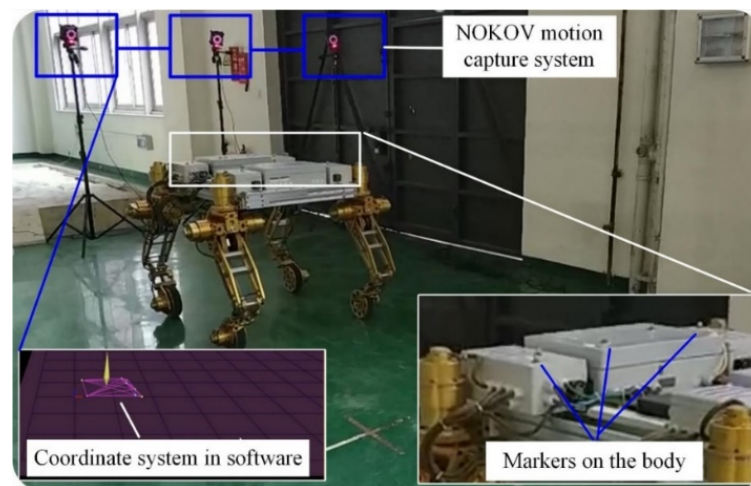


Figure 16. The experimental setup.

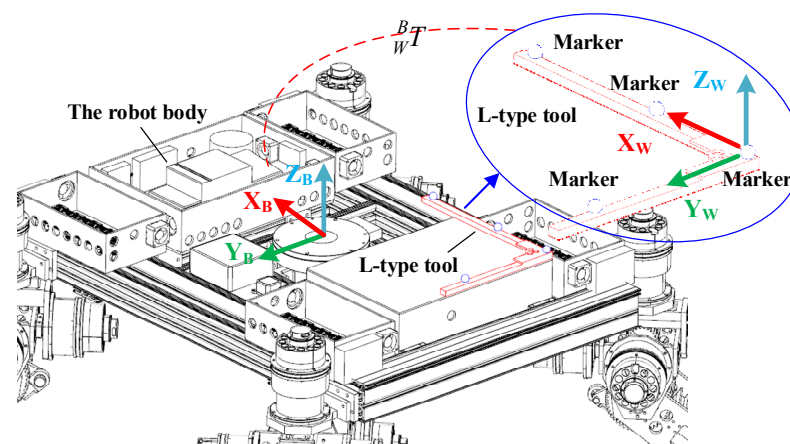


Figure 17. The calibration of the vision frame.

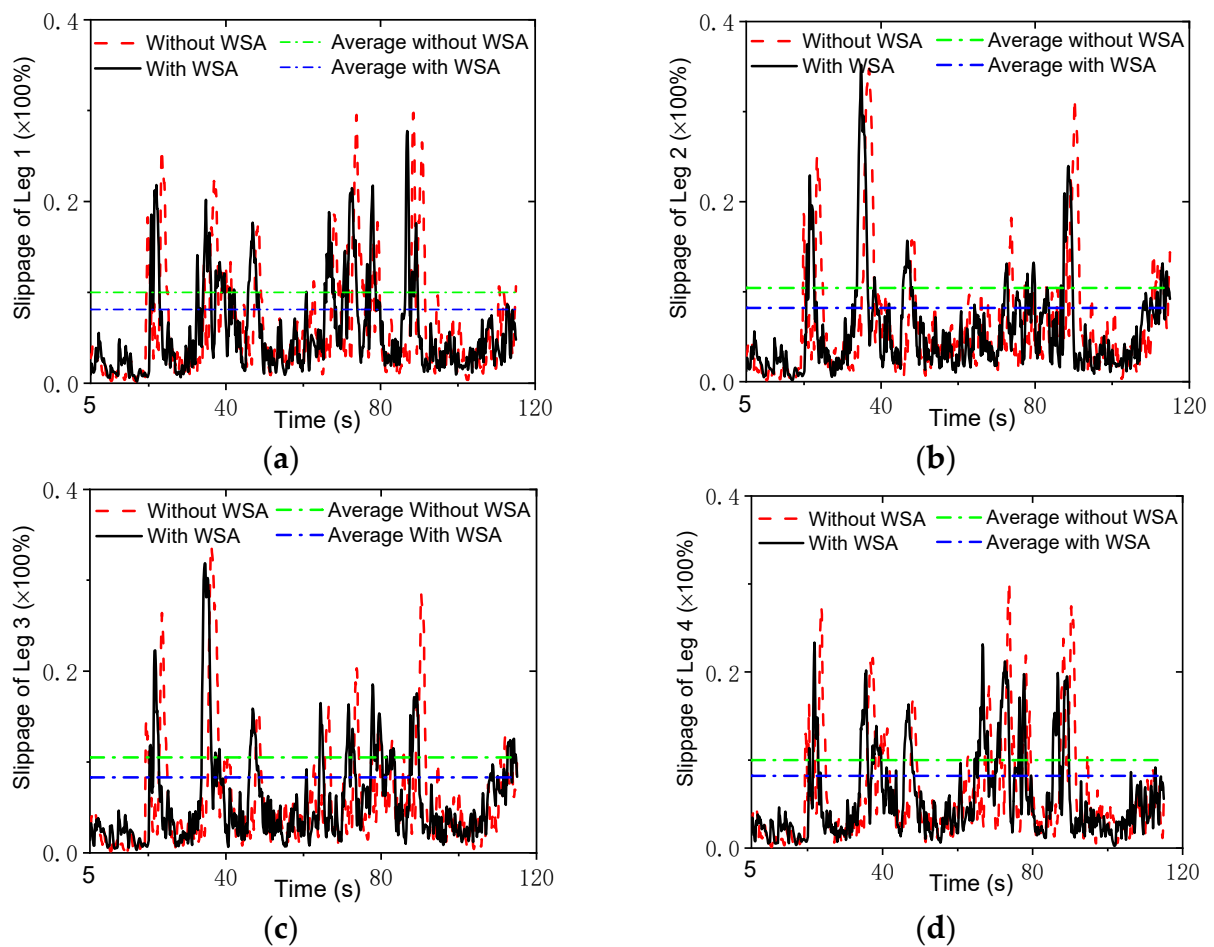


Figure 18. Slippage in the experiments: (a) Leg 1; (b) Leg 2; (c) Leg 3; (d) Leg 4.

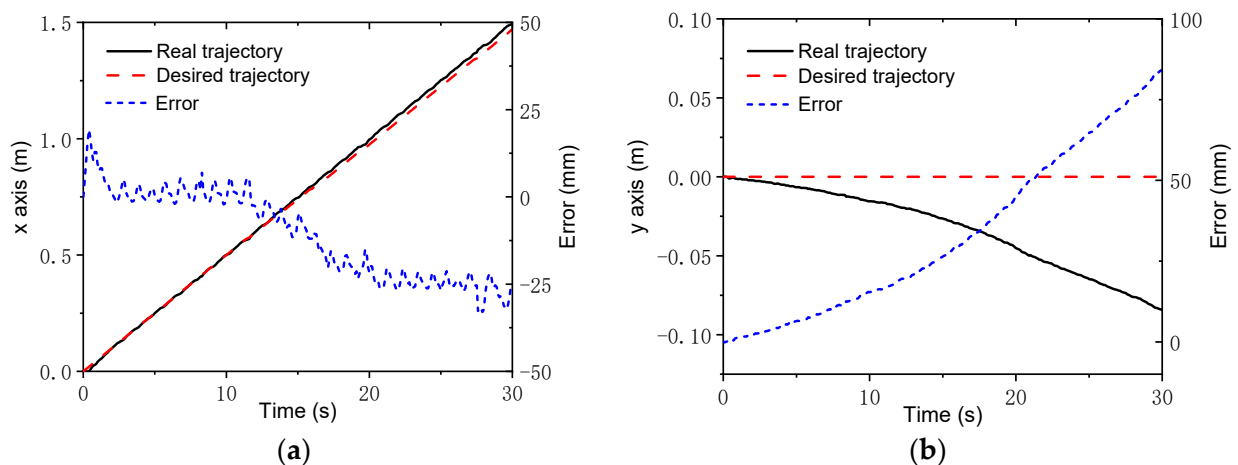


Figure 19. Experimental trajectory tracking results: (a) x axis; (b) y axis.

5. Conclusions

In this paper, a novel wheeled–legged planetary rover with four legs was proposed, and each leg had four DOFs with an actuated wheel. The articulated legs utilized a serial–parallel hybrid configuration, and it had the merits of both serial and parallel mechanisms. Moreover, the legs had a rigid–flexible coupling structure that could conform to unstructured terrain using both active and passive compliance. The kinematics equations of the rover were derived. Then, a control scheme including trajectory tracking, the steering

strategy, and the WSA module was proposed. A trajectory tracking model based on MPC that could handle the line and arc trajectory with quite a good accuracy was established. In addition, the WSA was introduced into the control strategy to decrease the slippage. After that, to validate the control method, three groups of cosimulations, i.e., tracking an arc trajectory, tracking a S-type trajectory, and trajectory tracking with high speeds were carried out. Finally, trajectory tracking experiments were conducted through a vision motion capture system. It was found that the average slippage of all of the legs decreased obviously after WSA control, with a slippage up to 20% in our experiments. Moreover, the rover could strictly track the reference trajectory. With respect to the reference trajectory, the deflection was found to be less than 2% F.S., which is a relatively small error. Accordingly, the proposed control strategy was thoroughly verified by the simulations and experiments.

Author Contributions: Conceptualization, J.H. and Y.S.; methodology, J.H. and Y.S.; software, Y.S. and L.Y.; investigation, Y.S. and L.Y.; writing—original draft, J.H.; writing—review and editing, J.H.; supervision, F.G.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant no. 52175022, the State Key Laboratory of Mechanical System and Vibration, grant no. MSVZD202106, and Shanghai R&D public service platform project, grant no. 19DZ2291400.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available upon reasonable request by the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rankin, A.; Maimone, M.; Biesiadecki, J.; Patel, N.; Levine, D.; Toupet, O. Driving curiosity: Mars rover mobility trends during the first seven years. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2020.
2. Dodge, R.; Parsons, D.; Abid, M.; Chrystal, K.; Kartolov, B. Dynamics associated with the Corer on M2020 Perseverance Rover. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 6–13 March 2021.
3. Zheng, J.; Gao, H.; Yuan, B.; Liu, Z.; Yu, H.; Ding, L.; Deng, Z. Design and terramechanics analysis of a Mars rover utilizing active suspension. *Mech. Mach. Theory* **2018**, *128*, 125–149. [\[CrossRef\]](#)
4. Michaud, F.; Letourneau, D.; Arsenault, M.; Bergeron, Y.; Cadrin, R.; Gagnon, F.; Legault, M.A.; Millette, M.; Paré, J.F.; Tremblay, M.C.; et al. Multi-modal locomotion robotic platform using leg-track-wheel articulations. *Auton. Robots* **2005**, *18*, 137–156. [\[CrossRef\]](#)
5. Hauser, K.; Bretl, T.; Latombe, J.C.; Harada, K.; Wilcox, B. Motion Planning for Legged Robots on Varied Terrain. *Int. J. Robot. Res.* **2008**, *27*, 1325–1349. [\[CrossRef\]](#)
6. Grand, C.; Benamar, F.; Plumet, F.; Bidaud, P. Stability and traction optimization of a reconfigurable wheel-legged robot. *Int. J. Robot. Res.* **2004**, *23*, 1041–1058. [\[CrossRef\]](#)
7. Smith, J.A.; Poulakakis, I.; Trentini, M.; Sharf, I. Bounding with active wheels and liftoff angle velocity adjustment. *Int. J. Robot. Res.* **2010**, *29*, 414–427. [\[CrossRef\]](#)
8. Xu, K.; Wang, S.; Yue, B.; Wang, J.; Peng, H.; Liu, D.; Chen, Z.; Shi, M. Adaptive impedance control with variable target stiffness for wheel-legged robot on complex unknown terrain. *Mechatronics* **2020**, *69*, 102388. [\[CrossRef\]](#)
9. He, J.; Gao, F. Type Synthesis for bionic quadruped walking robots. *J. Biol. Eng.* **2015**, *12*, 527–538. [\[CrossRef\]](#)
10. He, J.; Gao, F. Mechanism, actuation, perception, and control of highly dynamic multi-legged robots: A Review. *Chin. J. Mech. Eng.* **2020**, *33*, 79. [\[CrossRef\]](#)
11. Lewinger, W.A.; Harley, C.M.; Ritzmann, R.E.; Branicky, M.S.; Quinn, R.D. Insect-like antennal sensing for climbing and tunneling behavior in a biologically-inspired mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005.
12. Daltorio, K.A.; Wei, T.E.; Gorb, S.N.; Ritzmann, R.E.; Quinn, R.D. Passive foot design and contact area analysis for climbing mini-whegs. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007.
13. Chen, S.C.; Huang, K.J.; Chen, W.H.; Shen, S.Y.; Li, C.H.; Lin, P.C. Quattroped: A leg-wheel transformable robot. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 730–742. [\[CrossRef\]](#)
14. Kim, Y.S.; Jung, G.P.; Kim, H.; Cho, K.J.; Chu, C.N. Wheel transformer: A wheel-leg hybrid robot with passive transformable wheels. *IEEE Trans. Robot.* **2014**, *30*, 1487–1498. [\[CrossRef\]](#)

15. Chen, W.H.; Lin, H.S.; Lin, Y.M.; Lin, P.C. TurboQuad: A novel leg-wheel transformable robot with smooth and fast behavioral transitions. *IEEE Trans. Robot.* **2017**, *33*, 1025–1040. [[CrossRef](#)]
16. Kim, Y.; Lee, Y.; Lee, S.; Kim, J.; Kim, H.S.; Seo, T. STEP: A new mobile platform with 2-DOF transformable wheels for service robots. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1859–1868. [[CrossRef](#)]
17. Sun, T.; Xiang, X.; Su, W.; Wu, H.; Song, Y. A transformable wheel-legged mobile robot: Design, analysis and experiment. *Robot. Auton. Syst.* **2017**, *98*, 30–41. [[CrossRef](#)]
18. Grotzinger, J.P.; Crisp, J.; Vasavada, A.R.; Anderson, R.C.; Baker, C.J.; Barry, R.; Blake, D.F.; Conrad, P.; Edgett, K.S.; Ferdowski, B.; et al. Mars science laboratory mission and science investigation. *Space Sci. Rev.* **2012**, *170*, 5–56. [[CrossRef](#)]
19. Cordes, F.; Kirchner, F.; Babu, A. Design and field testing of a rover with an actively articulated suspension system in a Mars analog terrain. *J. Field Robot.* **2018**, *35*, 1149–1181. [[CrossRef](#)]
20. Lamon, P. 3D-position tracking and control for all-terrain robots. *Adv. Robot.* **2008**, *43*, 33–51.
21. Chwa, D. Robust distance-based tracking control of wheeled mobile robots using vision sensors in the presence of kinematic disturbances. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6172–6183. [[CrossRef](#)]
22. Liang, Z.; Chen, J.; Wang, Y. Equivalent acceleration imitation for single wheel of manned lunar rover by varying torque on earth. *IEEE/ASME Trans. Mechatron.* **2019**, *25*, 282–293. [[CrossRef](#)]
23. Krid, M.; Amar, F.B. A dynamic based path tracking controller for a fast rover with independent steering and drive. In Proceedings of the CLAWAR 2011, Paris, France, 6–8 September 2011.
24. Yang, H.; Zhao, H.; Xia, Y.; Zhang, J. Nonlinear MPC with time-varying terminal cost for tracking unreachable periodic references. *Automatica* **2021**, *123*, 109337. [[CrossRef](#)]
25. Liu, X.; Wang, W.; Li, X.; Liu, F.; He, Z.; Yao, Y.; Ruan, H.; Zhang, T. MPC-based high-speed trajectory tracking for 4WIS robot. *ISA Trans.* **2022**, *123*, 413–424. [[CrossRef](#)]
26. Ding, T.; Zhang, Y.; Ma, G.; Cao, Z.; Zhao, X.; Tao, B. Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint. *Mechatronics* **2022**, *84*, 102779. [[CrossRef](#)]
27. Sun, Y.L.; He, J.; Xing, Y. Multi-target coordinated control of wheel-legged Mars rover. *Acta Aeronautica Astronaut. Sin.* **2021**, *42*, 524246.
28. Li, W.; Ding, L.; Gao, H.; Tavakoli, M. Haptic tele-driving of wheeled mobile robots under nonideal wheel rolling, kinematic control and communication time delay. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 336–347. [[CrossRef](#)]
29. Chen, C.; Shu, M.; Wang, Y.; Ding, L.; Gao, H.; Liu, H.; Zhou, S. Simultaneous control of trajectory tracking and coordinated allocation of rocker-bogie planetary rovers. *Mech. Syst. Signal Proc.* **2021**, *151*, 107312. [[CrossRef](#)]