2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Path Tracking Control for A Transformable Wheel-Legged Robot Using Model Predictive Control

Chongping Sun, Na Zhao†, Kaijie Zhao, Yudong Luo, and Yantao Shen

Abstract— **Transformable wheel-legged robots can adjust their configuration according to terrain conditions, enabling effective operation in harsh environments. While existing controllers based on preset commands have successfully demonstrated the feasibility of reconfigurable mechanisms, they still struggle to handle complex autonomous operations. To address this, we develop a comprehensive motion model for such robots, encompassing chassis kinematics, chassis-wheel kinematics, and stability models, along with a hierarchical path tracking method. The upper controller uses model predictive control with an error state-space model to optimize real-time tracking error under input constraints and generate desired commands. The lower controller utilizes feedforward control to convert desired inputs into actual ones, while accommodating physical constraints and geometric coupling associated with variable-radius wheels. Comparative analyses confirm the effectiveness of the proposed approach and demonstrate the robot's performance under different wheel modes.**

I. Introduction

Wheeled robots have extensive applications across industry, agriculture, logistics, and healthcare, as well as in hazardous environments such as defense operations and outdoor expeditions due to their simple actuation mechanism, good terrain adaptability, and high energy efficiency [1]. Traditional wheeled robots, constrained by fixed wheel sizes, can only surmount obstacles no bigger than their wheel diameter. Yet enlarging the wheels leads to increased robot size and weight, restricting access to confined spaces [2]. To improve their obstacle-crossing capabilities and navigation in narrow spaces, extensive research has centered on wheel mechanism design [3]. For instance, rimless wheels and multi-spoke rotating-leg wheels are engineered to climb obstacles taller than their radius [4], though this comes at the cost of reduced stability and maneuverability. A transformable mechanism that can switch between wheel mode and leg mode depending on the terrain has been proposed in [5], [6]. Consequently, wheel-legged robots equipped with this mechanism can maneuver more flexibly and maintain more stable motion compared to non-morphing wheel-legged robots. Since then, morphings have garnered significant attention due to their notable advantages in obstacle negotiation, maneuverability, and stability [7], [8].

This work was supported by the National Natural Science Foundation of China under Grant 52305009 and the Fundamental Research Funds for the Central Universities under Grant 3132025271.

Chongping Sun, Na Zhao†, Kaijie Zhao and Yudong Luo are with the Department of Computer Science and Technology, Dalian Maritime University, Dalian, Liaoning, 116026, China. Yantao Shen is with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557, USA.

†Corresponding author. Email: *zna@dlmu.edu.cn*.

In the field of motion control for transformable wheel-legged robots, some studies have focused on the modeling and control of individual wheel-leg modules. For example, Ryu et al. [7] proposed a centrifugal-force-driven three-spoke passive wheel mechanism and simplified the entire robot to a constrained single-wheel system, which operated entirely under open-loop velocity control. Zheng et al. [2] designed a gear-leg transformation mechanism that relies on external sensors and proportional feedback laws to generate instructions to achieve the transformation of a single wheel. Other studies have attempted to establish kinematic models at the whole-body level and proposed various control frameworks. Chen et al. [8] employed a closed-loop control approach based on whole-body kinematics for the wheeled mode, whereas the legged mode adopts an open-loop joint-space trajectory tracking strategy, which cannot map body-level motion objectives to leg actions. Moger and Varol [9] established dynamic equations for the wheel diameter and designed an open-loop feedforward control method based on these equations. Although these studies demonstrate the feasibility of the transformable mechanisms in reconfigurable robots, they remain subject to notable limitations. Existing research predominantly focuses on the modeling of individual wheels, whereas complete models that map whole-body motion objectives into coordinated wheel-leg control remain scarce. Besides, most approaches rely on open-loop control or passive triggering mechanisms, which makes it difficult to achieve accurate tracking under complex trajectories.

Path tracking, aiming to guide robots to follow a predefined path or trajectory with high precision, is essential in numerous application scenarios. For conventional rigid wheeled platforms, researchers have proposed various path-tracking control algorithms. For example, Xiong et al. [10] proposed a cascaded NTSM-PID controller that combines nonsingular terminal sliding mode control at the upper level with model-based feedforward and robust feedback, achieving finite-time convergence and disturbance rejection for skid-steering robots. Chen et al. [11] designed a hierarchical control architecture that integrates guidance vector field-based online optimization with model feedforward and sliding mode control to achieve robust path tracking for differential-drive mobile robots. Sun et al. [12] proposed a MIMO nonsingular terminal sliding mode control strategy for a Mecanum omnidirectional platform, achieving high-precision pose tracking. Hassan and Saleem [13] integrated neural network adaptation with MRAC to enhance system adaptability to uncertainties and disturbances. Prado et al. [14] designed a centralized robust MPC that addresses tire

slip through feedforward compensation and online constraint tightening. These control methods can achieve path tracking for most conventional wheeled mobile robots, but they are not directly applicable to morphing wheeled mobile robots. This is because the variable wheel radius and increased degrees of freedom significantly amplify the required control effort. Consequently, low-level PID struggle to maintain coordination, and the computational burden of online training for neural networks or MRAC increases accordingly. Additionally, changes in structural morphology introduce additional control inputs and complex nonlinear coupling. Under such circumstances, Jacobian-based mappings, guidance vector fields, or fixed-structure MPC may fail or experience performance degradation, potentially resulting in infeasible optimization problems or abrupt changes in control inputs.

To this end, we propose a control framework composed of a three-layer model, as well as a two-level controller consisting of a high-level pose controller and a low-level radius controller. Specifically, as the robot has both the chassis structure of the traditional wheeled mobile robot and wheels with a transformable wheel-leg mechanism, this work first presents a three-layer model to capture different aspects of its kinematic behavior: chassis kinematic model, chassis-wheel coupling model, and stability model. These three layers respectively characterize the overall chassis motion, the nonlinear coupling between the chassis and the variable-radius wheels, and the stability characteristics, thereby enabling a continuous mapping from the global motion objectives of the robot to coordinated wheel radius adjustments. A two-level controller is then constructed. The high-level pose controller adopts the MPC, focusing on tracking the desired path and generating the desired control law for the lower-level controller. The lower-level radius controller employs a feedforward strategy to handle multiple coupled actual control inputs introduced by the variable-diameter wheels to track the desired control law. Furthermore, a stability control strategy is introduced to prevent the robot from tipping over during morphing. The main contributions of this paper are as follows. *i*) Unlike the current work of modeling a single wheel-leg module, a complete whole-body kinematic model is established for the transformable wheel-legged robot, which can systematically capture the relationship between chassis motion and wheel radius variation, while incorporating posture stability constraints to enhance the model's completeness. *ii*) A hierarchical control framework is proposed for the first time in the path-tracking control of the transformable wheel-legged robot to balance path tracking precision and adaptability to terrain disturbances.

This paper is organized as follows. Sect.II details the three-layer model, including chassis kinematics, chassis-wheel kinematics, and stability model. Sect.III presents the design of the two-level controller, with the emphasis on constructing the position and orientation controller as well as the radius controller. Sect.IV demonstrates the effectiveness of the proposed method through comparative experiments based on the customized robot. Finally, Sect.V summarizes the whole paper and suggests future research direction.

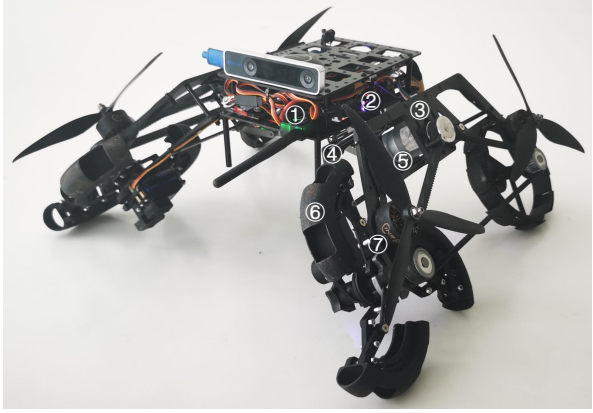


Fig. 1: The wheel-legged robot equipped with variable-radius wheels and foldable arms. The main components are labeled with serial numbers as follows: ① ROS controller, ② flight controller (PX4), ③ foldable arm, ④ high-voltage servo motor, ⑤ reduction motor, ⑥ variable-radius wheel, ⑦ low-voltage servo motor. (Photographed at Dalian Maritime University.)

II. Kinematic Modeling and Stability Analysis

In this section, we first provide a brief introduction to the robot platform and then derive the chassis kinematic model, the chassis-wheel kinematic model, and a detailed stability analysis of the robot, all of which support the subsequent design of the controller.

A. System Description

As shown in Fig. 1, we adopt an enhanced iteration of the robot platform described in [15] in this study, which features two key structural highlights: folding arms and variable-diameter wheels, thus enabling seamless transitions between flight and ground modes and enhancing adaptability to complex terrain. The foldable arms are symmetrically arranged and can adjust their folding angles to switch modes or avoid obstacles. The variable-diameter wheel features a single-degree-of-freedom structure, characterized by high stiffness and a large morphing range, which enables the robot to transition seamlessly from continuous wheeled mode to discrete legged mode, thereby overcoming obstacles. The control system is designed to support its morphing capabilities. The robot's aerial motion is controlled by a PX4 flight controller, with the ground motion managed by a ROS controller. The PX4 controller also centrally manages the servo motors responsible for folding the arm and adjusting the wheel size. This integrated hardware and control architecture provides a solid foundation for the subsequent real-world experiments on motion control and path tracking.

B. Kinematic Modeling

The kinematics consists of two parts: the chassis model, mapping from the desired position and orientation to the desired velocity and angular velocity, and the chassis-wheel model, describing the transformation from the desired velocity and angular velocity to the radius and wheel speed. Define $O_gX_gY_g$ as the global coordinate system (GCS) and

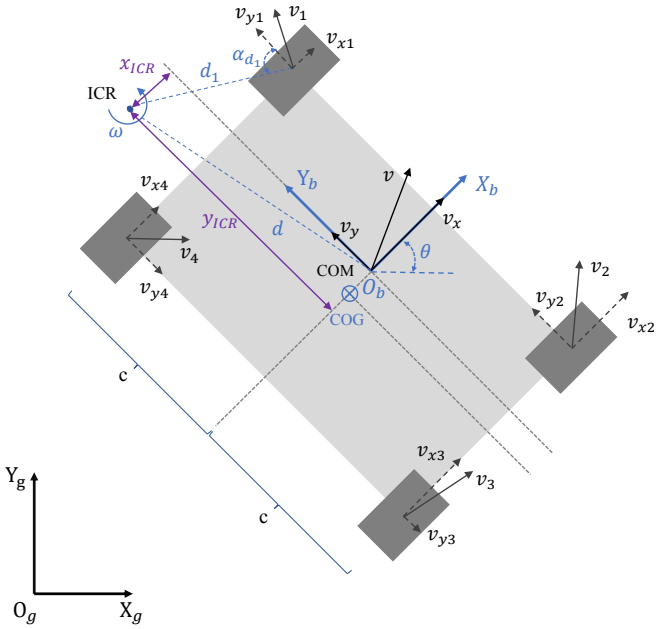


Fig. 2: Schematic diagram of the robot coordinate systems.

$O_bX_bY_b$ as the body coordinate system (BCS), and the origin of the BCS is located at the center of mass (COM) of the robot chassis, as shown in Fig.2. The distance from the wheel to the center of geometry (COG) along the X_b axis is defined as c. ICR indicates the robot's instantaneous center of rotation, with the distance from COM to ICR denoted by d, and (x_{ICR}, y_{ICR}) defines the coordinates in the BCS. The kinematics for wheeled robots is empirically written as

$$\dot{q} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{1}$$

where X and Y indicate the position of the robot in GCS, θ is the orientation relative to GCS, ω is the angular velocity, and v denotes the overall speed with v_x and v_y representing its components along the X_b and Y_b, respectively.

The chassis is modeled as a rigid body where every point has the same angular velocity as the COM when rotating. Accordingly, the angular velocity of the four wheels around the ICR is ω. Thus, the relationship between the wheel velocities and the chassis angular velocity is expressed as

$$\begin{aligned} w &= \frac{v_i}{d_i} = \frac{v_i \sin\alpha_{d_i}}{d_i \sin\alpha_{d_i}} = \frac{v_{yi}}{d_{xi}} \\ w &= \frac{v_i}{d_i} = \frac{v_i \cos\alpha_{d_i}}{d_i \cos\alpha_{d_i}} = \frac{v_{xi}}{d_{yi}} \end{aligned} \tag{2}$$

where v_i denotes the velocity of the i-th wheel, d_i is the distance from the COM of the wheel to ICR, d_{xi} and d_{yi} denote the components of d_i along the X_b and Y_b axes, respectively, and α_{d_i} is the angle between d_i and Y_b, with $i \in \{1, 2, 3, 4\}$.

From the first equation in (2), the constraint on v_y can be obtained as $v_y + \omega x_{ICR} = 0$. Substituting it into (1) yields

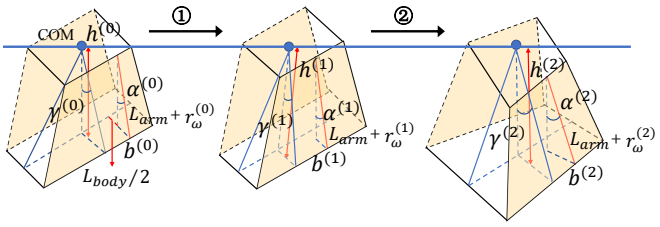


Fig. 3: Simplified 3D model for stability analysis. Action ① indicates increasing wheel radius, and action ② indicates increasing track width.

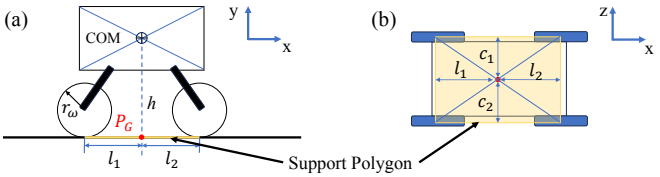


Fig. 4: Support polygon in different views: (a) robot side view and (b) top view.

the final kinematic model of the robot chassis.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & x_{ICR}\sin\theta \\ \sin\theta & -x_{ICR}\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \tag{3}$$

Similarly, the second equation in (2) can be rewritten as $v_{xi} = \omega d_{yi}$ with $d_{y1} = d_{y4} = d_y - c$ and $d_{y2} = d_{y3} = d_y + c$. Given the rigid linkage of the wheels to the chassis, and that velocity of the wheels on left and right sides is determined by the same drive system, we can obtain $v_L = v_{x1} = v_{x4}$ and $v_R = v_{x2} = v_{x3}$, where v_L and v_R are the velocity of the robot's chassis on the left and right sides, respectively. Therefore, we have $v_L = v_x - \omega c$ and $v_R = v_x + \omega c$. The kinematics of the wheel can be described as $v_L = \omega_L r_L$ and $v_R = \omega_R r_R$. By combining these equations, we can obtain the kinematic model for the chassis-wheel as

$$\begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} \omega_L/2 & \omega_R/2 \\ -\omega_L/(2c) & \omega_R/(2c) \end{bmatrix} \begin{bmatrix} r_L \\ r_R \end{bmatrix} \tag{4}$$

where r_L and r_R denote the radius of the left and right wheels, respectively, and ω_L and ω_R are their rotational angular velocities.

C. Stability Analysis

The change in wheel radius during motion may alter the robot's posture and increase the risk of rolling or tipping over. Therefore, a stability analysis of the morphing process is required to ensure safety. Fig. 3 illustrates the proposed stability model, which is utilized to assess the robot's stability under various wheel diameters and folding angles, thereby deriving the relationship between wheel diameter and folding angle subject to stability constraints. The stability model evaluates the robot's stability using the stability margin angle, defined as the minimum angle between the perpendicular to the COM and the edge of the support polygon, as detailed in Fig. 4. The corresponding tangent value is a quantitative indicator of the robot's stability [16].

To analyze the robot's stability during the morphing process, we simplify the motion involving wheel-diameter

adjustment into three states and two actions, as shown in Fig.3. In the diagram, the vertical distance from the COM to the ground is denoted by h, the stability margin angle is denoted by γ, and α is unfolding angle representing the angle between the foldable arm and the vertical plane. The minimum distance from the projected COM to the support polygon is denoted as $b = \min\{c_1, c_2, l_1, l_2\}$, where c_1, c_2, l_1, and l_2 are the respective distances to each of its edges. The values of these parameters under three distinct system states, denoted as S_0, S_1, and S_2, are given as follows: $(h^{(0)}, h^{(1)}, h^{(2)})$, $(\gamma^{(0)}, \gamma^{(1)}, \gamma^{(2)})$, $(\alpha^{(0)}, \alpha^{(1)}, \alpha^{(2)})$, and $(b^{(0)}, b^{(1)}, b^{(2)})$. Specifically, S_0 signifies the initial state before the wheel radius adjustment begins, S_1 is the intermediate state following action ①, and S_2 marks the final state upon completion of action ②. Here are some assumptions to construct the stability model: *i*) The robot's mass is uniformly distributed. *ii*) Throughout the arm's adjustment process, the projected COM is always closer to the lateral edge of the support polygon than to the longitudinal edge, as the robot's body length is always greater than its width. The stability margin angle in state S_i is given by

$$\tan\gamma^{(i)} = \frac{b^{(i)}}{h^{(i)}} \tag{5}$$

where $b^{(i)} = L_{body}/2 + (L_{arm} + r_\omega^{(i)})\sin\alpha^{(i)}$, $h^{(i)} = (L_{arm} + r_\omega^{(i)})\cos\alpha^{(i)}$. Here, L_{body}, L_{arm}, and $r_\omega^{(i)}$ denote the body width, foldable arm length, and wheel radius in state S_i, respectively, for $i \in \{0, 1, 2\}$.

To maintain the robot's stability after morphing, the foldable arm is adjusted to restore its initial configuration, ensuring that $\gamma^{(2)} = \gamma^{(0)}$. Substituting (5) into this equation and rearranging these terms, we can derive the stability model to expresses the relationship between γ and α as

$$\tan\alpha^{(2)} = \tan\gamma^{(0)} - \frac{L_{body}/2}{(L_{arm} + r_\omega^{(2)})\cos\alpha^{(2)}} \tag{6}$$

which is introduced into the kinematic model (4) as posture stability constraints, providing a quantitative evolution metric for stability analysis in the subsequent implementation.

III. Controller Design

A. Overall Control Architecture

The transformable robot faces the following control challenges in the path-tracking task: the multi-degree-of-freedom characteristics resulting from variable-diameter wheels and folding arms, as well as the strong coupling between actual control inputs. To address these issues, a hierarchical control architecture comprising a high-level MPC and a low-level actuator coordination controller (radius controller) has been proposed, as illustrated in Fig. 5. In detail, the high-level controller is designed for path tracking, generating the required virtual linear and angular velocities (the virtual control law $\boldsymbol{u}^{des} = \left[v_x^{des}, \omega^{des}\right]^T$) such that the robot can follow the desired path $\boldsymbol{x}_r$ with $\boldsymbol{u} = \boldsymbol{u}^{des}$. The low-level controller is designed to compute the actual control inputs. Based on the desired virtual command $\boldsymbol{u}^{des}$, it calculates the wheel radii,

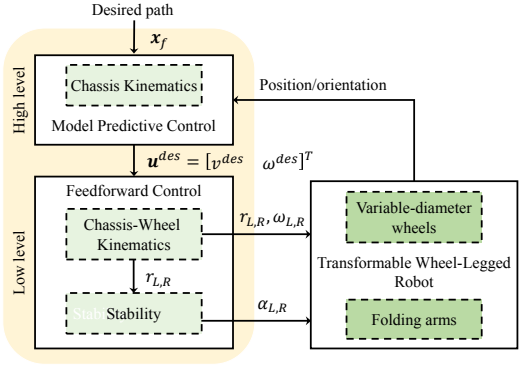


Fig. 5: Schematic diagram of the two-level control structure and system model.

wheel angular velocities, and folding arm angles, ensuring that the actual linear and angular velocities track $\boldsymbol{u}^{des}$ while maintaining posture stability during motion.

B. High-Level Controller

Firstly, the robot's chassis kinematic model (3) can be represented through $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$, where $\boldsymbol{x} = \left[X, Y, \theta\right]^T$ denotes the state vector, $(X, Y) \in \mathbb{R}^2$, $\theta \in (-\pi, \pi)$, and $\boldsymbol{u} = \left[v_x, \omega\right]^T$ denotes control vector, $v_x \in \mathbb{R}$, $\omega \in \mathbb{R}$, and $f(\cdot)$ describes the evolution of the system state with the control inputs. Further, the state-space equation can be linearized near the reference point and discretized to obtain the state-space equations applicable to discrete time.

$$\boldsymbol{x}_e(k+1) = A_d(k)\boldsymbol{x}_e(k) + B_d(k)\boldsymbol{u}_e(k) \tag{7}$$

where $\boldsymbol{x}_e(k)$ and $\boldsymbol{x}_e(k+1)$ denote the state errors at time k and $k+1$, defined as $\boldsymbol{x}_e = \boldsymbol{x} - \boldsymbol{x}_r$, and $\boldsymbol{u}_e(k)$ is the control input error, defined as $\boldsymbol{u}_e = \boldsymbol{u} - \boldsymbol{u}_r$. The reference state and control vectors are $\boldsymbol{x}_r = \left[X_r, Y_r, \theta_r\right]^T$ and $\boldsymbol{u}_r = \left[v_{xr}, \omega_r\right]^T$, respectively. The matrices $A_d(k)$ and $B_d(k)$ denote the discretized state and input matrices at step k.

The cost function is formulated as follows

$$\begin{aligned} J(k) = \sum_{j=1}^{N} \{ & \boldsymbol{x}_e^T(k+j|k) q \boldsymbol{x}_e(k+j|k) \\ & + \boldsymbol{u}_e^T(k+j-1|k) r \boldsymbol{u}_e(k+j-1|k)\} \end{aligned} \tag{8}$$

where q and r are the weight matrices, satisfying $q, r \geq 0$, N is the length of the prediction horizon, and T is the time step size. The conditional subscript $(k+j|k)$ denotes the prediction of the state at time step $k+j$ from time step k.

We use (7) to recursively predict the system states over the next N steps, computing the state sequence within the prediction horizon using the following recursive.

$$\xi_{\boldsymbol{x}}(k) = \widetilde{A}(k)\boldsymbol{x}_e(k|k) + \widetilde{B}(k)\xi_{\boldsymbol{u}}(k) \tag{9}$$

where $\xi_{\boldsymbol{u}}(k)$ is the stacked vector of input errors over the prediction horizon, and $\widetilde{A}(k)$ and $\widetilde{B}(k)$ are the corresponding block matrices formed by $A_d(k)$ and $B_d(k)$.

Substitute (9) into (8), we can obtain

$$\begin{aligned} J(k) &= \xi_{\boldsymbol{x}}^T(k) Q \xi_{\boldsymbol{x}}(k) + \xi_{\boldsymbol{u}}^T(k) R \xi_{\boldsymbol{u}}(k) \\ &= \frac{1}{2}\xi_{\boldsymbol{u}}^T(k) H \xi_{\boldsymbol{u}}(k) + F^T \xi_{\boldsymbol{u}}(k) + P \end{aligned}$$

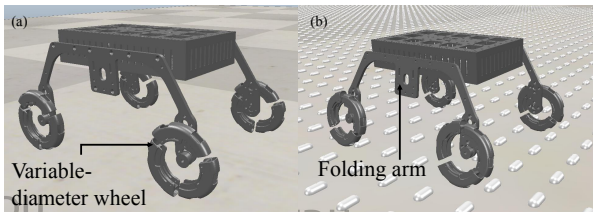


Fig. 6: The robot on various terrains implemented in CoppeliaSim: (a) flat terrain and (b) uneven terrain. (Simulated at Dalian Maritime University on an HP Z2 Mini G9 workstation.)

where $H = 2(\widetilde{B}^T Q \widetilde{B} + R)$, $F = 2\widetilde{B}^T Q^T \widetilde{A}\xi_{\boldsymbol{x}}(k)$, $P = \xi_{\boldsymbol{x}}^T(k)\widetilde{A}^T(k)Q\widetilde{A}(k)\xi_{\boldsymbol{x}}(k)$, Q and R are diagonal matrices with q and r repeated N times, respectively.

The constrained optimization problem at each step can be expressed as a standard quadratic programming problem, $\xi_{\boldsymbol{u}}^* = \arg\min_{\boldsymbol{u}_e} J(k)$, subject to $\boldsymbol{u}_{min} \le \boldsymbol{u}_e(k+j|k) \le \boldsymbol{u}_{max}$, $j \in [0, N-1]$, and $\xi_{\boldsymbol{u}}^*$ denotes the optimal control sequence. After solving the optimization problem, only the control input at the current time step is retained as the virtual control law, which is tracked in real time by the low-level controller, thereby forming a complete hierarchical control loop.

C. Low-Level Controller

The low-level controller needs to convert the virtual control law $\boldsymbol{u}^{des} = \left[v_x^{des}, \omega^{des}\right]^T$ from the high-level controller into actual commands for the actuators, $\boldsymbol{u}^{act} = \left[r_L, r_R, \omega_L, \omega_R\right]^T$. To achieve the desired motion while maintaining posture stability during wheel radius adjustments, the controller adopts a feedforward control based on the robot's kinematic model (4) and the stability model (6).

To achieve stable motion and prevent overturning, the robot needs to adjust the unfolding angles of its two foldable arms, α_L and α_R, based on the current wheel radius. According to (6), α_L and α_R must satisfy $\alpha_i = \arctan\{\tan\gamma - L_{body}/(2L_{arm} + 2r_i)\}$, $i \in \{L, R\}$, where γ denotes the desired stability margin angle. After adjusting α_L and α_R, the effective lateral distances from the COM to the left and right wheels, c_L and c_R are updated as $c_i = L_{body}/2 + (L_{arm} + r_i)\sin\alpha_i$, $i \in \{L, R\}$.

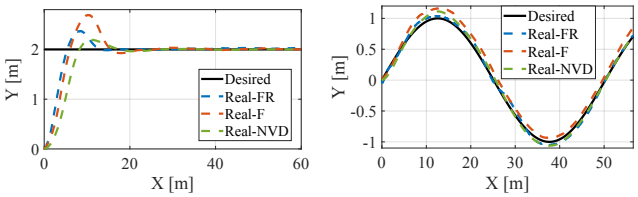
Define the intermediate calculated values at the current time t as $\tilde{r}_{L_t}$ and $\tilde{r}_{R_t}$, with

$$\tilde{r}_{L_t} = \frac{v_{x_t}^{des} - c_{L_{t-1}}\omega_t^{des}}{\omega_{L_{t-1}}}, \quad \tilde{r}_{R_t} = \frac{v_{x_t}^{des} + c_{R_{t-1}}\omega_t^{des}}{\omega_{R_{t-1}}}$$

which will then be determined whether they are within the allowable range $[r_{min}, r_{max}]$, so that the following different strategies can be used to calculate the control input $\boldsymbol{u}_t^{act} = \left[r_{L_t}, r_{R_t}, \omega_{L_t}, \omega_{R_t}\right]^T$ and update the relevant parameters.

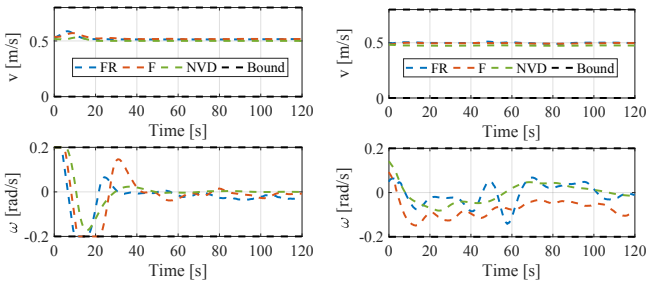
If $\tilde{r}_{L_t} \in [r_{min}, r_{max}]$ and $\tilde{r}_{R_t} \in [r_{min}, r_{max}]$, we have

$$\omega_{L_t} = \omega_{L_{t-1}}, \quad \omega_{R_t} = \omega_{R_{t-1}}, \quad r_{i_t} = \tilde{r}_{i_t}$$
$$\alpha_{i_t} = \arctan\left(\tan\gamma - \frac{L_{body}/2}{L_{arm} + r_{i_t}}\right)$$
$$c_{i_t} = \frac{L_{body}}{2} + (L_{arm} + r_{i_t})\sin\alpha_{i_t}$$



(a) (b)

Fig. 7: Comparison of desired and actual paths under three modes on flat terrain: (a) straight-line path tracking and (b) curved path tracking.



(a) (b)

Fig. 8: Control inputs during path tracking on flat terrain: (a) straight-line path and (b) curved path.

Otherwise, we have

$$r_{i_t} = r_{i_{t-1}}, \quad \alpha_{i_t} = \alpha_{i_{t-1}}, \quad c_{i_t} = c_{i_{t-1}}$$
$$\omega_{L_t} = \frac{v_{x_t}^{des} - c_{L_t}\omega_t^{des}}{r_{L_t}}, \quad \omega_{R_t} = \frac{v_{x_t}^{des} + c_{R_t}\omega_t^{des}}{r_{R_t}}$$

where $i \in \{L, R\}$.

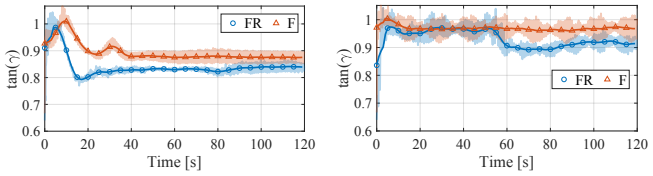
IV. Experimental Results

The robot equipped with four morphing wheels supports three configuration modes: FR, F, and NVD. In FR mode, both the front and rear wheels of the robot can be changed. In F mode, only the front wheels can be changed, while the rear wheels are fixed, thus requiring fewer actuators than FR mode. In NVD mode, four wheels are fixed. We conducted a series of comparative experiments using CoppeliaSim to evaluate the robot's tracking performance across different terrain conditions and to compare the three configuration modes in path-tracking tasks, as shown in Fig. 6.

A. Path Tracking Performance on Flat Pavements

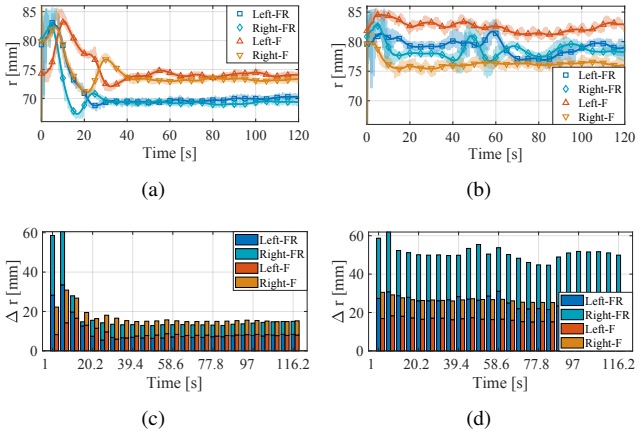
We designed two different paths, a straight path and a curved path, and compared them in terms of path tracking performance and radius variation.

Path tracking performance: To describe the tracking performance, we define the tracking error as $e_j^i(t) = j^i(t) - j_r^i(t)$, where $i \in \{FR, F, NVD\}$ and $j \in \{X, Y, \theta\}$ denote the configuration mode and the state variable, respectively, and j_r denotes the reference state. The experimental results shown in Fig. 7 demonstrate that the proposed controller achieves accurate and stable path tracking in all three configurations. For both straight and curved paths, e_X converges to $[-0.09, -0.05]$ m, e_Y to $[-0.08, 0.13]$ m, and e_θ to



(a) (b)

Fig. 9: The stability margin angle ($\tan\gamma$) during path tracking on flat terrain: (a) straight-line path and (b) curved path.

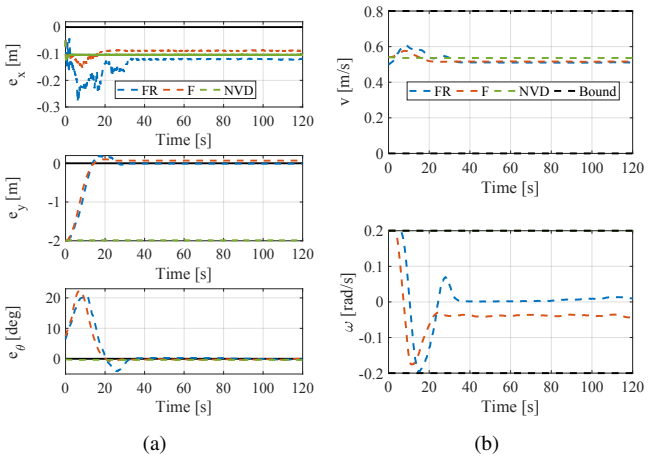


(a) (b) (c) (d)

Fig. 10: Wheel radius variations and increments during path tracking on flat terrain: (a) wheel radius for straight-line path, (b) radius increment for straight-line path, (c) wheel radius for curved path, and (d) radius increment for curved path.

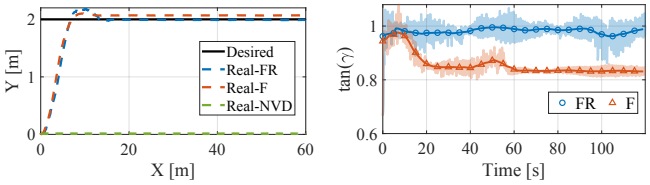
$[-2.3°, 3.4°]$, which verifies the effectiveness and robustness of the controller in different task scenarios. Further comparison of the tracking performance across the three modes reveals a convergence trend of $e_j^{NVD} \leq e_j^{FR} \leq e_j^{F}$. Among them, the NVD mode exhibits the smallest tracking error, as the controller does not need to handle multi-dimensional input allocation and dynamic constraints. The tracking error in the FR mode is close to that in the NVD mode, indicating that the proposed complete kinematic model can accurately capture structural variations during dynamic wheel-radius adjustment, and that the controller can effectively handle these changes during path tracking. The F mode exhibits slightly larger errors, primarily due to reduced input allocation flexibility caused by constraints on certain wheel radii. Nevertheless, the errors remain convergent, indicating that the controller retains robustness under partially uncontrollable conditions. As shown in Fig. 8, the control inputs v_x and ω always maintain within the limits in all modes. In addition, according to Fig. 9, it can be found that in both FR and F modes, $\tan(\gamma)$ tends to stabilize after 30 seconds, and the robot does not experience any tipping or overturning, thus ensuring the feasibility and safety of the proposed controller in terms of actuator constraints and posture stability.

Radius variation: As the NVD mode does not support wheel radius adjustment, the following discussion focuses solely on FR and F modes. To analyze the wheel behavior under two modes, we introduce the variables of $i \in \{FR, F\}$ and $j \in \{L, R\}$ to denote configuration mode and left/right



(a) (b)

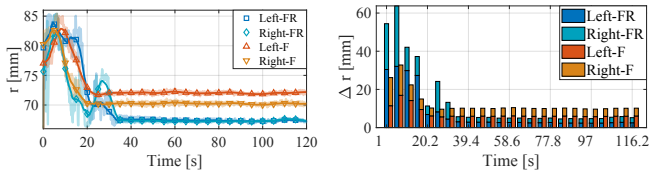
Fig. 11: Straight-line path tracking on uneven terrain: (a) state errors and (b) control inputs.



(a) (b)

Fig. 12: Path tracking comparison and stability analysis: (a) desired path vs. actual paths under three modes and (b) variation of $\tan(\gamma)$ during tracking in FR and F modes.

wheel index, respectively. The radius of the j wheel in mode i at time t is denoted by $r_j^i(t)$, and the radius difference is defined as $r_{LR}^i(t) = |r_L^i(t) - r_R^i(t)|$ with the increment given by $\Delta r_j^i(t) = |r_j^i(t) - r_j^i(t-1)|$. Since the front and rear wheels exhibit identical morphing behavior in the FR mode, we only present the radius variation of each morphing wheel on the left and right sides for comparison with the corresponding wheels in the F mode. As shown in Fig. 10(a), in straight-line tracking task, $r_j^F(t) \geq r_j^{FR}(t)$. This is because the FR mode distributes the wheel-radius adjustment tasks to the four wheels, whereas the F mode can only rely on the front wheels, thus requiring a larger radius adjustment for single wheel. Similarly, when tracking a curved path, steering depends on the radius difference between the left and right wheels, and the F mode still requires a larger adjustment, as reflected by $r_{LR}^F(t) \geq r_{LR}^{FR}(t)$ in Fig. 10(b). We further collected the cumulative changes of all morphing wheels in wheel radius to evaluate energy consumption, and the results are shown in Fig. 10(c) and Fig. 10(d). In general, the energy consumption caused by the variable radius in FR mode is higher than that in F mode, satisfying $\Delta r_L^{FR}(t) + \Delta r_R^{FR}(t) \geq \Delta r_L^{F}(t) + \Delta r_R^{F}(t)$, which is significantly higher in curve tracking and slightly higher in straight-line tracking. This is because, although the radius variation of each morphing wheel on the left and right sides in the FR mode is smaller than in the F mode, both the front and rear wheels will participate in adjusting the radius, resulting in a larger overall cumulative variation.



(a) (b)

Fig. 13: Wheel radius variations and increments during path tracking on uneven terrain: (a) wheel radius and (b) radius increment.

B. Path Tracking Performance on Rough Pavements

To test the robustness of the controller, we designed a scenario that consists of a standard road surface with gravel spread evenly to simulate a real-world uneven road. The gravel size was set to 0.048 m×0.024 m×0.012 m with the gravel spacing of 0.1 m.

Path tracking performance: During testing, it was found that the robot could converge in both F and FR modes, with convergence errors of $e_X \in [-0.12, -0.08]$ m, $e_Y \in [-0.01, 0.07]$ m, and $e_\theta \in [-0.09°, 0.06°]$, as shown in Fig. 11(a) and Fig. 12(a). However, it failed to converge in NVD mode. This demonstrates that the controller exhibits robust and adaptive path tracking under terrain disturbances in both FR and F modes. The failure to converge in NVD mode stems from the lack of wheel radius adjustment, which prevents the controller from fully compensating for terrain disturbances, leading to wheel slippage and a complete loss of tracking capability. This further highlights the importance of dynamic wheel radius adjustment for path tracking performance. Furthermore, the stability curves in Fig. 12(b) and Fig. 11(b) demonstrate that the controller design is able to maintain pose stability under more challenging conditions while satisfying the control constraints.

Radius variation: The radius variation during this process are plotted in Fig. 13(a) and Fig. 13(b). Similar to the results in flat pavements, the radius change and energy consumption caused by the variable radius of uneven roads also follow the same rules: $r_j^F(t) \geq r_j^{FR}(t)$, $r_{LR}^F(t) \geq r_{LR}^{FR}(t)$, and $\Delta r_L^{FR}(t) + \Delta r_R^{FR}(t) \geq \Delta r_L^F(t) + \Delta r_R^F(t)$.

The experimental results above validate the effectiveness of the proposed controller in all three modes and also characterize the balance between terrain adaptability and actuation complexity for each mode. Specifically, the NVD mode performs best on flat surfaces, but is prone to slipping on rough pavement. The FR mode demonstrates the best adaptability to terrain disturbances, though it requires higher actuation complexity. The F mode, while slightly less adaptable than the FR mode, achieves robustness to terrain disturbances with fewer actuators and demonstrates better cost-effectiveness.

V. Conclusion

The robot in this study is a novel robot equipped with variable-diameter wheels and foldable arms, exhibiting strong terrain adaptability. To address these characteristics, we proposed a three-layer model and a two-level controller framework that effectively balances path-tracking accuracy and adaptability to terrain disturbances. Different scenarios were designed to verify the path-tracking performance and robustness of the controller under three mode configurations (FR, F, and NVD). Additionally, each configuration has its own advantages under different environmental conditions. In particular, FR and F exhibit excellent steering performance even in complex terrain conditions, demonstrating superior terrain adaptability. Future research will involve physical experiments to test the controller's performance.

References

[1] J. Liao, Z. Chen, and B. Yao, "Model-based coordinated control of four-wheel independently driven skid steer mobile robot with wheel–ground interaction and wheel dynamics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1742–1752, 2019.

[2] C. Zheng, S. Sane, K. Lee, V. Kalyanram, and K. Lee, "α-waltr: Adaptive wheel-and-leg transformable robot for versatile multiterrain locomotion," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 941–958, 2023.

[3] İrem Mertyüz, A. K. Tanyıldızı, B. Taşar, A. B. Tatar, and O. Yakut, "Fuhar: A transformable wheel-legged hybrid mobile robot," *Robotics and Autonomous Systems*, vol. 133, p. 103627, 2020.

[4] J. B. Jeans and D. Hong, "Impass: Intelligent mobility platform with active spoke system," in *2009 IEEE International Conference on Robotics and Automation*, pp. 1605–1606, 2009.

[5] 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. 1487–1498, 2014.

[6] T. Sun, X. Xiang, W. Su, W. Hang, and Y. Song, "A transformable wheel-legged mobile robot: Design, analysis and experiment," *Robotics and Autonomous Systems*, vol. 98, pp. 30–41, 2017.

[7] S. Ryu, Y. Lee, and T. Seo, "Shape-morphing wheel design and analysis for step climbing in high speed locomotion," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1977–1982, 2020.

[8] S.-C. Chen, K.-J. Huang, W.-H. Chen, S.-Y. Shen, C.-H. Li, and P.-C. Lin, "Quattroped: A leg–wheel transformable robot," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 730–742, 2014.

[9] G. Moger and H. A. Varol, "Design and implementation of a mobile robot with variable-diameter wheels," *IEEE/ASME Transactions on Mechatronics*, pp. 1–10, 2024.

[10] R. Xiong, L. Li, C. Zhang, K. Ma, X. Yi, and H. Zeng, "Path tracking of a four-wheel independently driven skid steer robotic vehicle through a cascaded ntsm-pid control method," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[11] J. Chen, C. Wu, G. Yu, D. Narang, and Y. Wang, "Path following of wheeled mobile robots using online-optimization-based guidance vector field," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 1737–1744, 2021.

[12] Z. Sun, H. Xie, J. Zheng, Z. Man, and D. He, "Path-following control of mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode," *Mechanical Systems and Signal Processing*, vol. 147, p. 107128, 2021.

[13] N. Hassan and A. Saleem, "Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 10, pp. 13582–13597, 2022.

[14] Álvaro Javier Prado, M. Torres-Torriti, J. Yuz, and F. Auat Cheein, "Tube-based nonlinear model predictive control for autonomous skid-steer mobile robots with tire–terrain interactions," *Control Engineering Practice*, vol. 101, p. 104451, 2020.

[15] Z. He, N. Zhao, Y. Luo, S. Long, X. Luo, and H. Deng, "A multi-modal hybrid robot with enhanced traversal performance*," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6193–6198, 2024.

[16] L. Sun, W. Wang, and Z. Du, *Modeling and Control of Field Robots*. Harbin Institute of Technology Press, 2022.