Slope path tracking control of agricultural wheel-legged robot based on virtual sensing radar and two-level deep neural network

Yongchun Zhao[1,2], Qing Zhang[1,2*], Yong You[1,2]

(1. *College of Engineering, China Agricultural University, Haidian 100083, Beijing, China*;
2. *Research Center of Intelligent Grassland Industry and Intelligent Grassland Equipment, China Agricultural University, Haidian 100083, Beijing, China*)

Abstract: The continuous development of smart agriculture puts forward the requirement of high accuracy slope path tracking for the agricultural wheel-legged robot. Compared to flat terrain, path tracking control on sloped terrain faces the obstacle of motion instability of the wheel-legged robot induced by the slope gravitational force component, which causes instantaneous steering center to offset. To address this problem, this study proposed a slope path tracking control algorithm by combining the methods of virtual sensing radar and two-level neural network. Firstly, the kinematic and dynamic models of the wheel-legged robot are deduced, from which the crucial factors affecting control accuracy of slope path tracking are recognized. Secondly, this study constructs the slope path tracking control algorithm, in which the virtual sensing radar is utilized to realize route perception, and the two-level neural network is employed to provide drive motors' speeds to adapt to path tracking on different slopes. Furthermore, the corresponding compensation methods of the identified impacting factors are embedded in the proposed algorithm, including the lateral tracking deviation factor, heading angle deviation factor, slope change factor, and slip rate factor. Finally, the co-simulation model of slope path tracking control is constructed, including the multi-body dynamic model of the wheel-legged robot in RecurDyn and the proposed slope path tracking algorithm complied by Python. Subsequently, the simulation tests of the wheel-legged robot are carried out under various slope angles and velocities. The results reveal that the proposed algorithm's effectiveness and accuracy are superior, with tracking errors reduced by more than 47.2% compared to an optimized pure pursuit algorithm.

Keywords: deep neural network, virtual sensing radar, slope path tracking, wheel-legged robot
DOI: 10.25165/j.ijabe.20251803.8737

Citation: Zhao Y C, Zhang Q, You Y. Slope path tracking control of agricultural wheel-legged robot based on virtual sensing radar and two-level deep neural network. Int J Agric & Biol Eng, 2025; 18(3): 223–235.

1 Introduction

With the continuous development of smart agriculture, the scope of application for agricultural robots in various working environments is gradually expanding. Path tracking control is a key factor of automatic navigation technology for agricultural robots under flat terrains. Nevertheless, there is currently a scarcity of automatic navigation systems suitable for hilly and mountainous terrains. The varying gradients of sloped terrains will bring new challenges in path tracking control since the navigation equipment relies on obtaining position information from an earth plane coordinate system. Under these circumstances, direct utilization of position information without proper conversion will lead to unexpected tracking errors. Moreover, the unstructured slopes will have significant impact on the unmanned agricultural robot in two main aspects. On the one hand, the steep road slope will disrupt the motion stability of the agricultural robot, resulting in reduced perception with the target path. On the other hand, the unstable lateral tire slippage and sinking of agricultural robots resulting from various road surface hardnesses and gravity components along the slope will also lead to unexpected lateral error and heading error in the path tracking implementation. The commonly used control methods typically focus solely on the planar motion of the robot, neglecting or directly disregarding the influence of gravity. Such oversights can result in a significant decline in the robot's tracking progress when navigating sloped paths. Therefore, it is in urgent demand to verify the crucial disturbance factors affecting the tracking accuracy of agricultural robots under sloped terrains and improve the adaptation of the control algorithm.

With regards to slope path tracking control, there is little related research. Auria et al.[1] performed real agricultural robot tests in sloped terrain conditions to investigate the potential control issues in the path tracking process. The fundamental control elements for slope path tracking control are discussed in their study. Dogan et al.[2] proposed a robust adaptive model and position control method with adaptive state variables, consisting of the robot' s dynamic model, the slope angle, and other related parameters to establish the longitudinal force and lateral force models of the robot. Additionally, the relationship between slip and friction was established by using a magic formula wheel model, as well as the relationship between body speed and wheel speed by using a slip system. Jeong[3] presented a path tracking control algorithm for slope autonomous navigation robots with four-wheel steering, in which the direct yaw moment control was employed to achieve accurate path tracking performance considering the various influencing factors such as control forces, actuator limits, ride comfort, and body slip when determining steering inputs for front and rear wheels, respectively. Qi et al.[4] proposed a model

Received date: 2023-12-13 **Accepted date:** 2025-03-27

Biographies: Yongchun Zhao, PhD, research interest: intelligent agricultural machinery, Email: 1727667355@qq.com; **Yong You**, Professor, research interest: intelligent agricultural machinery, Email: youyong@cau.edu.cn.

***Corresponding author: Qing Zhang**, Professor, research interest: agricultural machinery automation. College of Engineering, China Agricultural University, Haidian 100083, Beijing, China. Tel: +86-010-62736076, Email: zhangqingbit@163.com.

predictive control-based path tracking method for a car-like mobile robot on slopes, considering the effects of gravity and internal load changes. The method enables accurate path tracking on slopes by addressing the complex relationship between the wheel and road surface due to the influence of gravity, as well as accounting for slope variations. Hu et al.[5] proposed a cascaded navigation control method for straight path tracking. The navigation control task was decomposed into two cascaded control tasks, namely, the path tracking control task and the steering control task. Then the controller is obtained by establishing a kinematic model, and its feasibility is verified by experiments.

Yue et al.[6] highlighted that during slope path tracking, the interaction between the robot's wheel and the road surface becomes increasingly intricate due to gravitational effects, thereby rendering tracking control more challenging amidst varying slopes. Therefore, they proposed a model predictive control method combined with a fuzzy system that effectively incorporates both the dynamic characteristics of the body and wheels on sloped roads. By enhancing the two-dimensional kinematics and dynamics model of the robot, their approach enables it to adapt to varying slopes and road conditions while accommodating curvature mutations. However, the approach does not fully consider the body slip phenomenon in hilly and mountainous areas, resulting in poor portability of the control algorithm.

Given the intricate and dynamic control environment, neural networks have been increasingly employed by scholars to optimize the control process, making machine learning with neural networks the primary prominent research area in agricultural machinery identification and control. Yang et al.[7] proposed an asymmetric dual-priority polling control system model and utilized a neural network algorithm for performance prediction and analysis. The mathematical model of the continuous-time state system was established using the embedded Markov chain theory and probability generating function. Furthermore, accurate analysis and verification of system characteristics such as average queue length and average cycle were conducted through simulation experiments. The results demonstrate that the model not only distinguishes multi-service tasks effectively but also ensures minimal system delay, thereby highlighting the advantages of hierarchical neural networks. Zhou et al.[8] introduced a multi-level attention network combined with policy reinforcement learning for image captioning to address limitations observed in existing methods that solely focus on global representation at the image level or specific concepts alone. This model comprises a multi-level attention network module aiming to capture both global and local object details, along with a region attention network module targeting local features within regions of interest. The learning task is graded in this model, simplifying each learning module's content while facilitating communication between both modules. Experimental results indicate superior learning capabilities compared to other existing image captioning methods. Li et al.[9] proposed a prediction model, namely the two-level resolution deep neural network, which addresses the limitation of traditional GCN's small receptive field that hampers referencing traffic characteristics from remote sensors and consequently leads to inaccurate long-term predictions. This model comprises two resolution blocks: low-resolution blocks for macroscopic scale traffic prediction, such as regional traffic changes; and high-resolution blocks that utilize GCN to extract spatial correlation and reference regional variations generated by the low-resolution block for microscopic scale flow prediction. Experimental results demonstrate that hierarchical processing in neural networks refines perceptual content, thereby yielding more accurate prediction outcomes. Based on these studies, hierarchical neural networks exhibit promising development potential across various fields due to their efficient and accurate information processing capabilities along with their unique ability to decompose complex tasks and fuse network correlations. Bai et al.[10] proposed a neural network control method with NMPC as the learning sample. The design process of this control method includes establishing the NMPC controller based on the time-varying local model, generating learning samples based on this NMPC controller, and training to obtain the neural network controller. But this method considers fewer control factors, resulting in lower control accuracy.

In order to improve the tracking accuracy of agricultural robot under sloped terrains, the present study proposes a slope path tracking control algorithm based on virtual sensing radar and two-level deep neural network. The algorithm aims to address the control instability caused by slopes in hilly and mountainous terrains, as well as active steering control. The research content is mainly divided into the following aspects: 1) The kinematic and dynamic models of the agricultural robot under sloped terrain were established, and the critical factors affecting motion stability were identified. 2) The slope path tracking control algorithm was constructed based on virtual sensing radar and two-level deep neural network. 3) The co-simulation model was accomplished to verify the proposed algorithm under various tracking conditions.

2 Materials and methods

In this part, the necessary control parameters are obtained by building the kinematic and dynamic models of the wheel-legged robot in the hills and mountains, and the final control value is obtained by sorting out the control logic and building the deep neural network framework, combining the virtual sensing radar map and various control influence factors.

2.1 Kinematic and dynamic models of the wheel-legged robot on the slope

2.1.1 Kinematic model of the wheel-legged robot on the slope

The kinematic model of the wheel-legged robot is established based on the following assumptions:

a) The wheel-legged robot is regarded as a rigid body at all times;

b) The physical properties of the four wheels of the wheel-legged robot are the same;

c) The mass center of the whole robot coincides with its geometric center.

The kinematic model of the wheel-legged robot, as shown in Figure 1, is built in the spatial coordinate system.

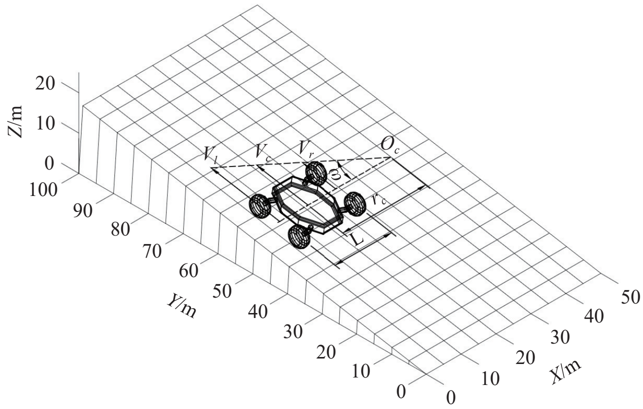


Figure 1 Kinematic model of the agricultural wheel-legged robot on the slope

Through the geometric relationship, the angular velocity of the wheel-legged robot at a certain time can be derived as follows:

$$\omega = \frac{V_c}{r_c} = \frac{V_r}{r_c + \dfrac{L}{2}} = \frac{V_l}{r_c - \dfrac{L}{2}} \tag{1}$$

where, ω, V_c, V_r, V_l, r_c, and L are the angular velocity, rad/s; the centroid velocity, m/s; the converted speed of the right wheel, m/s; the converted speed of the left wheel, m/s; the steering radius, m; and the width of the wheel-legged robot, m, respectively.

When $v_r = v_l \neq 0$, the wheel-legged robot will go straight at a uniform speed; when $v_r > v_l$, $\omega > 0$, the wheel-legged robot will turn right; when $v_r < v_l$, $\omega < 0$, the wheel-legged robot will turn left; when $v_r = v_l = 0$, the wheel-legged robot will be parking. Among them, the speed of turning right or left depends on the speed differential of the drive motors on both sides.

The forward kinematic model can be obtained based on the velocities of the left and right wheels:

$$\begin{bmatrix} V_c \\ \omega \end{bmatrix} = \begin{bmatrix} \dfrac{V_r + V_l}{2} \\ \dfrac{V_r - V_l}{L} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{2} & \dfrac{1}{2} \\ \dfrac{1}{L} & -\dfrac{1}{L} \end{bmatrix} \begin{bmatrix} V_r \\ V_l \end{bmatrix} \tag{2}$$

Based on the velocity decomposition of the geometric center point, the inverse kinematic model can be obtained:

$$\begin{bmatrix} V_r \\ V_l \end{bmatrix} = \begin{bmatrix} V_c + \dfrac{L}{2}\omega \\ V_c - \dfrac{L}{2}\omega \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{L}{2} \\ 1 & -\dfrac{L}{2} \end{bmatrix} \begin{bmatrix} V_c \\ \omega \end{bmatrix} \tag{3}$$

In which $[V_c \quad \omega]^{\mathrm{T}}$ can be set according to the inverse kinematic model to obtain $[V_l \quad V_r]$, to precisely control the rotation of the driving wheel; the above formula is not affected by gravity.

However, the position and heading information obtained by the navigation equipment are based on the coordinate system values of the earth plane, while the value used in the actual tracking process is its projection on the slopes. Therefore, coordinate conversion is necessary in slope path tracking, as shown in Figure 2.

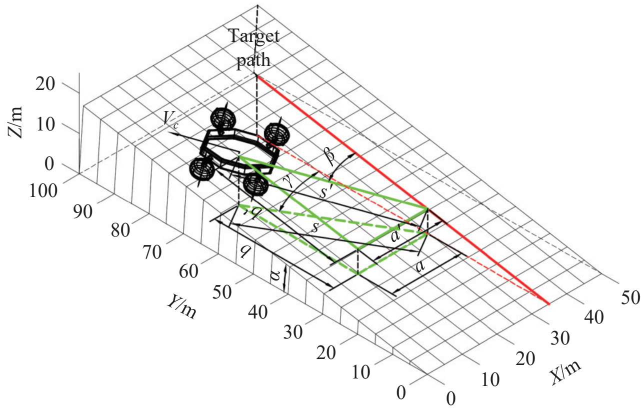


Figure 2 Navigation and positioning data conversion of the wheel-legged robot on the slope

Figure 2 illustrates how the wheel-legged robot's relative coordinates (a, b) on the global plane are derived. The navigation system provides the robot's absolute position (x_c, y_c) and heading angle γ, which are then combined with the reference path coordinates $[x_r, y_r]^{\mathrm{T}}$ to compute (a, b).

The slope angle is set as α. Then the projection of coordinate points on the reference slope path can be expressed as $\left[x_r, \dfrac{y_r}{\cos\alpha}\right]^{\mathrm{T}}$. The other parameter settings are shown in Figure 2, where the solid line is on the slope road surface and the dotted line is on the earth plane. Then:

$$b' = \frac{b}{\cos\alpha} \tag{4}$$

$$s' = \sqrt{a^2 + \frac{b^2}{\cos^2\alpha}} \tag{5}$$

$$\sin\beta = \frac{a}{s'} \tag{6}$$

$$\beta = \arcsin\frac{a}{\sqrt{a^2 + \dfrac{b^2}{\cos^2\alpha}}} \tag{7}$$

where, a, b, b', α, β are the horizontal coordinate value of the wheel-legged robot earth plane absolute coordinate system, m; the vertical coordinate value of the wheel-legged robot earth plane absolute coordinate system, m; the vertical coordinate value of the wheel-legged robot slope projection coordinate system, m; the slope angle and the heading angle in the wheel-legged robot slope projection coordinate system, respectively.

It can be deduced that the coordinates of the wheel-legged robot on the sloped road at any time can be expressed as $\left(a, \dfrac{b}{\cos\alpha}\right)$, and the heading angle can be expressed as arcsin $\dfrac{a}{\sqrt{a^2 + \dfrac{b^2}{\cos^2\alpha}}}$, which can be used directly by the navigation system.

2.1.2 Dynamic model of the wheel-legged robot on the slope

Figure 3 shows the dynamic relationship between the forces on each wheel and the robot body.

According to the geometric relationship in Figure 3, the dynamic model of the wheel-legged robot can be deduced:

$$m(\dot{u} - \omega v) = F_{x1} + F_{x2} + F_{x3} + F_{x4} - mg\sin\alpha\cos\theta \tag{8}$$

$$m(\dot{v} - \omega u) = F_{y1} + F_{y2} + F_{y3} + F_{y4} - mg\sin\alpha\sin\theta \tag{9}$$

$$J\dot{\omega} = \frac{L}{2}(-F_{y1} + F_{y2} - F_{y3} + F_{y4}) + L_f(F_{y1} + F_{y2}) - L_r(F_{y3} + F_{y4}) \tag{10}$$

where, m is the mass, kg; J is the moment of inertia, kgm^2; α is the slope of the simulated road surface, °; F_{xi} is the longitudinal static ground friction force applied to each wheel, N; F_{yi} is the transverse static ground friction force applied to each wheel, N; L_f and L_r are the front and rear wheelbases, respectively, m; and L is the width, m.

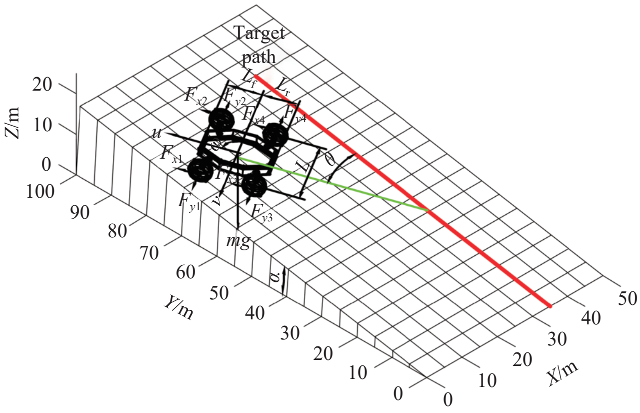


Figure 3 Dynamic model of wheel-legged robot in hilly mountainous areas

It is difficult to control the lateral force of the wheel-legged robot tire in the tracking control process. When wheel slip occurs, the control consideration can be taken into account after conversion, and the remaining robot body tracking parameters can be transformed into the following formula:

$$\dot{u}=\frac{F_x}{m}-g\sin\alpha\cos\theta+\omega v \tag{11}$$

$$\dot{v}=\frac{F_y}{m}-g\sin\alpha\sin\theta-\omega u \tag{12}$$

$$\dot{\omega}=\frac{M}{J} \tag{13}$$

According to the formula, the course deviation, slopes, and related wheel slip problems should be fully considered in the determination of wheel-legged robot control quantity.

2.2 The construction of slope path tracking control algorithm

2.2.1 The functioning principles of virtual sensing radar

Liu et al.[11] proposed a virtual sensing radar model for path tracking control of tractors in orchards. Instead of using lateral deviation and heading deviation as the perception input of the path navigation algorithm, the scanned virtual sensing radar map information was used as the perception input. The virtual sensing radar map can not only establish the relative position relationship between the hilly and mountainous wheel-legged robot and the target path, but also predict the direction of the target path by pre-perception of the failed path segment to a certain extent, to reduce the possible overshoot and collision, and provide convenience for further accurate control of the hilly and mountainous wheel-legged robot tracking operation. At the same time, the virtual sensing radar avoids the problem of the real radar sensor, which is easily disturbed by the environment. It is not sensitive to the real working environment and has high stability.

The use of virtual sensing radar needs to obtain the target path segment first, and the target path segment should include the coordinates of the starting and ending positions and other necessary parameters. However, the target path lines connected by the target waypoints directly obtained by the navigation equipment do not have the property of road width, so the target path needs to be widened manually. In the algorithm, the discrete target path points are connected to the target path line segment, and because the robot's width is 1 m, the target path line segment is translated to the left and right sides by 0.5 m equidistant to obtain the virtual path boundary. The virtual target path segment with a total width of 1 m is obtained by combining the two virtual path boundaries for virtual sensing radar detection.

The operation principle of the virtual sensing radar is to imitate the work of the real radar. The wheel-legged robot itself is the virtual radar beam launching center, and the detection beam is uniformly transmitted to the fixed range of the fuselage. When the detection beam contacts the boundary of the virtual path or reaches the limit of the transmission distance, it stops and records the length. All the detection beams emitted uniformly are used as the virtual sensing radar map of the wheel-legged robot at a certain time, and the virtual sensing radar map generated at a certain time in the operation of the hilly mountainous wheel-legged robot with respect to the virtual path segment is shown in the shaded part of Figure 4.

In Figure 4, $S_P(x_s,y_s)$ is the starting point of the target path segment currently, and $E_P(x_e,y_e)$ is the end point of the target path segment. The virtual sensing radar scanning center is $C(x,y)$. The virtual path width is d; the heading angle of the current wheel-legged robot is θ. The maximum detection range of the virtual sensing radar is $l_{\max}$.

The target path line equation can be expressed as follows:

$$A_0x+B_0y+C_0=0 \tag{14}$$

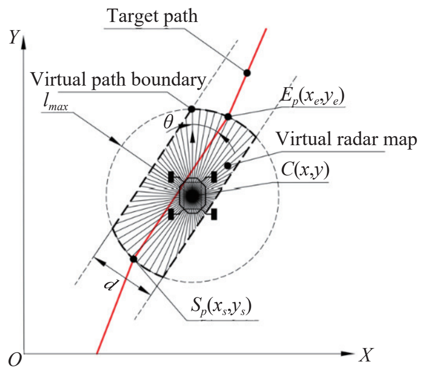


Figure 4 Schematic of a virtual sensing radar map generated at a certain time

The boundary line equation of the virtual path on both sides is denoted by:

$$A_1x+B_1y+C_1=0 \tag{15}$$

$$A_2x+B_2y+C_2=0 \tag{16}$$

The following relationship is obtained by mathematical derivation:

$$A_0=A_1=A_2=y_s-y_e \tag{17}$$

$$B_0=B_1=B_2=x_e-x_s \tag{18}$$

$$C_0=x_sy_e-x_ey_s \tag{19}$$

$$C_1=C_0+\frac{d\sqrt{A_1^2+B_1^2}}{2} \tag{20}$$

$$C_2=C_0-\frac{d\sqrt{A_2^2+B_2^2}}{2} \tag{21}$$

where, A_0, B_0, C_0, A_1, B_1, C_1, A_2, B_2, C_2 are constant parameters, and d is virtual path width, m.

The virtual sensing radar map is calculated by the existing mathematical relationship. If the virtual scanning resolution is 1°, the virtual sensing radar sends 360 detection beams uniformly to the surroundings at a certain time, and the distance between the detection beam of the ith virtual sensing radar and the boundary of the virtual path on a certain side is:

$$l_j^i=-\frac{A_jx+B_jy+C_j}{\sqrt{A_j^2+B_j^2}\sin(\theta+i)} \tag{22}$$

where i, j are the detection beam sequence number transmitted by the virtual sensing radar, and the boundary label of the virtual path, respectively.

The value of j is 1 or 2, which represents that the virtual sensing radar scanning center transmits a detection beam to the left or right virtual path boundary. When the virtual sensing radar detection range is calculated through the above equation, if the square root $\sqrt{A_j^2+B_j^2}\times\sin(\theta+i)$ is 0, it specifies that the detection range at this time is the maximum value, namely for $l_{\max}$; if $l_1^i<0$ or $l_1^i>l_{\max}$, let $l_1^i=l_{\max}$; if $l_2^i<0$ or $l_2^i>l_{\max}$, let $l_2^i=l_{\max}$. Each time the virtual sensing radar map is calculated, the virtual sensing radar center will send virtual detection beams to the virtual path boundaries on both sides at the same time, and only one of the detection distances is taken as the effective virtual sensing radar detection distance at the moment. Therefore, the distance comparison between the transmitting center and the virtual path boundaries on both sides is involved: Every time, the algorithm takes the virtual sensing radar center detection to virtual path boundary distance minimum value as the effective detection range, namely $l^i=\min(l_1^i,l_2^i)$. The detection distances of all virtual sensing radars are normalized to form the final virtual sensing radar map for the subsequent use of the neural network.

In actual path tracking on the slopes, only the following eight conditions will appear in the virtual sensing radar map, as shown in Figure 5, and the other conditions can be attributed to one of the following eight virtual sensing radar maps.

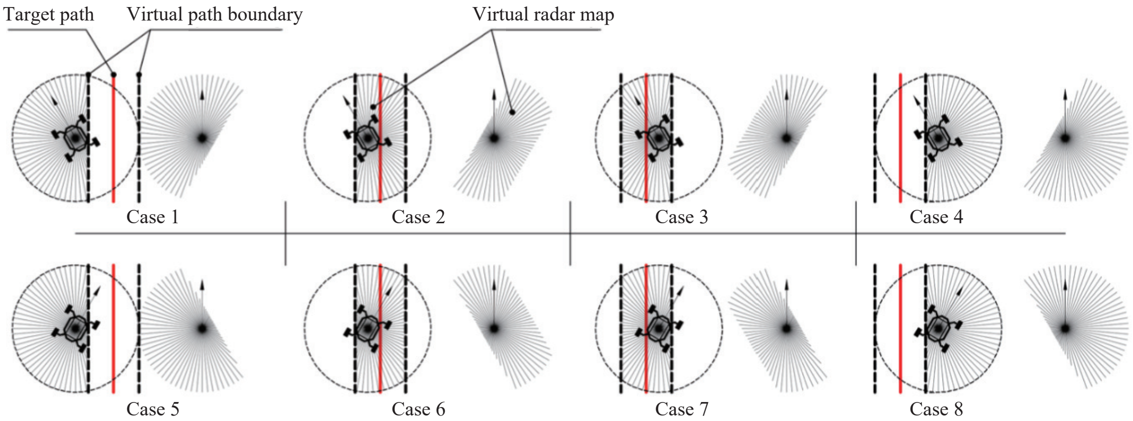


Figure 5 All the shapes that the virtual sensing radar map will appear during the actual tracking process

In the eight virtual sensing radar maps, the maximum heading deviation angle is set to 60°, the positive direction of the path is 0°, the counterclockwise deflection of the wheel-legged robot is positive, and the clockwise deflection is negative.

Case 1: The wheel-legged robot is tracked outside the left side of the virtual path boundary, and the heading angle is positive. The main control strategy corresponding to this type of virtual sensing radar map is to turn sharply to the right.

Case 2: The wheel-legged robot is tracked to the left side of the target path, inside the virtual path boundary, and the heading angle is positive. The main control strategy corresponding to this type of virtual sensing radar chart is a small turn to the right.

Case 3: The wheel-legged robot is tracked to the right of the target path, inside the boundary of the virtual path, and the heading angle is positive. The main control strategy corresponding to this type of virtual sensing radar map is to go straight.

Case 4: The wheel-legged robot is tracked outside the right side of the virtual path boundary, and the heading angle is positive. The main control strategy corresponding to this type of virtual sensing radar map is to go straight.

Case 5: The wheel-legged robot is tracked outside the left side of the virtual path boundary, and the heading angle is negative. The main control strategy corresponding to this type of virtual sensing radar map is to go straight.

Case 6: The wheel-legged robot is tracked to the left side of the target path, inside the boundary of the virtual path, and the heading angle is negative. The main control strategy corresponding to this type of virtual sensing radar map is to go straight.

Case 7: The wheel-legged robot is tracked to the right of the target path, inside the boundary of the virtual path, and the heading angle is negative. The main control strategy corresponding to this type of virtual sensing radar map is a small turn to the left.

Case 8: The wheel-legged robot is tracked outside the right side of the virtual path boundary, and the heading angle is negative. The main control strategy corresponding to this type of virtual sensing radar chart is a large turn to the left.

2.2.2 Architecture of the slope path tracking control algorithm

The essence of path tracking control for the wheel-legged robot in hilly and mountainous areas is to calculate the corresponding driving instructions according to the relative position relationship between the robot and the target path. Based on the analysis of the functions and operating characteristics of the wheel-legged robot, the basic actions can be simplified into four conditions: go straight, turn left, turn right, and stop, and the rest of the driving actions can also be composed of these four basic actions. However, due to the weak adaptive ability of a single fixed control command to the road, especially the wheel-legged robot which is affected by gravity when moving on the slope, the wheels are easy to slip and be affected by external interference, resulting in poor control accuracy or failure. Therefore, the purpose of building the first-level deep neural network in this algorithm is to describe the position relationship between the wheel-legged robot with respect to the virtual path boundary through the generated virtual sensing radar map. Then the second-level deep neural network is used to calculate multiple control influence parameters to obtain real-time and accurate control speeds of the drive motor.

Figure 6 shows the architecture of the proposed slope path tracking algorithm by fusing virtual sensing radar and the two-level deep neural network.

Firstly, the real-time wheel-legged robot position coordinates and heading angle information are obtained by RTK-GNSS positioning base station and mobile station, then converted to the sloped road. The virtual sensing radar algorithm is combined with the existing reference path and the current wheel-legged robot slope position to calculate the virtual sensing radar map in real time, and the current robot-road position relationship is preliminarily obtained through the first-level deep neural network calculation. The output results of the first-level deep neural network, the current road slope, and the wheel slip rate are included in the calculation of the second-level deep neural network. Finally, the control value of the drive motor speeds is obtained, which is sent to the motor drivers through the lower computer for execution.

2.2.3 Neural networks' structure and influencing factors

1) Structure and role of first-level deep neural network

The structure of the first-level deep neural network is shown in Figure 7. The input is the virtual sensing radar map after batch

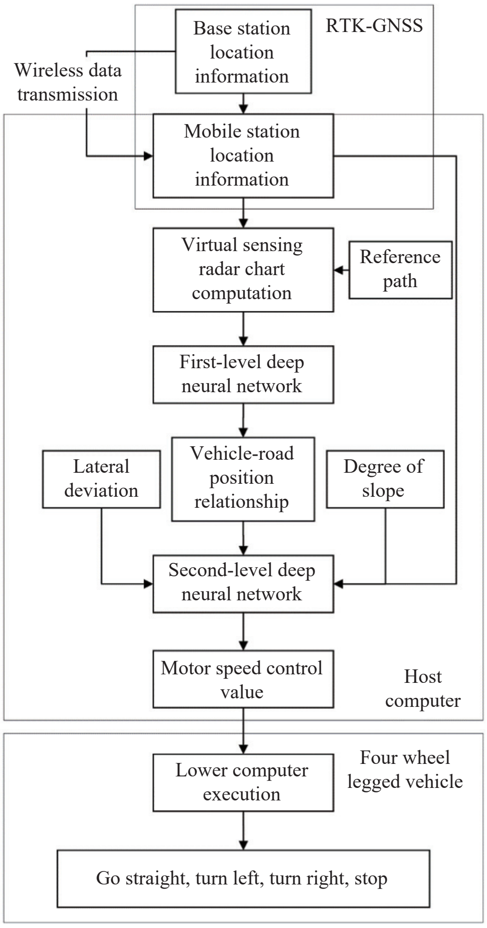

Base station location information
RTK-GNSS
Wireless data transmission
Mobile station location information
Virtual sensing radar chart computation
Reference path
First-level deep neural network
Lateral deviation
Vehicle-road position relationship
Degree of slope
Second-level deep neural network
Motor speed control value
Host computer
Lower computer execution
Four wheel legged vehicle
Go straight, turn left, turn right, stop

Figure 6 Architecture of the proposed slope path tracking algorithm fusing virtual radar and two-level deep neural network

normalization, and the output is the position relationship between the wheel-legged robot and target path segment, which is used to represent various situations on the target path in real work. The relative position relationship can be described by the different deviation. When the wheel-legged robot is on the target path left, the deviation relationship obtained by the virtual sensing radar which is between the wheel-legged robot and target path segment may occur in the following four conditions:

(1) There is a lateral deviation, the wheel-legged robot is located inside of the target path segment, and the heading deviation is greater than 0.

(2) There is a lateral deviation, the wheel-legged robot is located outside the target path segment, and the heading deviation is greater than 0.

(3) There is a lateral deviation, the wheel-legged robot is located inside of the target path segment, and the heading deviation is less than 0.

(4) There is a lateral deviation, the wheel-legged robot is located outside the target path segment, and the heading deviation is less than 0.

The relative position relationship corresponding to the above four deviation cases also exists symmetrically on the right side of the target path segment, so the first-level deep neural network sets a total of eight outputs, corresponding to all possible conditions of the virtual sensing radar.

The first-level deep neural network establishes the relative position relationship between the wheel-legged robot and the target path segment, and the mapping relationship between the virtual sensing radar map through deep learning, and converts it into the corresponding numerical label value of [0,7] as the input of the second-level deep neural network.

2) Structure and role of second-level deep neural network

The structure of the second-level deep neural network collects the label values of the relative position relationship between the wheel-legged robot and the target path generated by the first-level deep neural network, as well as the parameter values of the lateral tracking deviation impact factor, the heading deviation, the slope of the wheel-legged robot, and the slip rate influence factor. Through deep learning, the mapping relationship between the precise control speed of the driving motors on both sides of the current wheel-legged robot and the above multiple parameters is established. A schematic representation of the working role of the second-level deep neural network is shown in Figure 8.

3) Compensation methods of various influencing factors

(1) Lateral tracking deviation factor *e*

There will be lateral tracking deviation in the process of path tracking for the wheel-legged robot on the slope. When the lateral deviation is too large, it easily leads to insufficient control, and when the lateral deviation is small, it easily leads to control overshoot.

Figure 9 enumerates the control speed of the drive motor of the wheel-legged robot on both sides under different lateral tracking deviations when the heading deviation is 0. When the wheel-legged robot is located on the left side of the path, the control speed of the drive motor of the wheel-legged robot on the left side is greater than that on the right side. Similarly, when the wheel-legged robot is located on the right side of the path, the control speed of the drive motor of the wheel-legged robot on the right side is greater than that

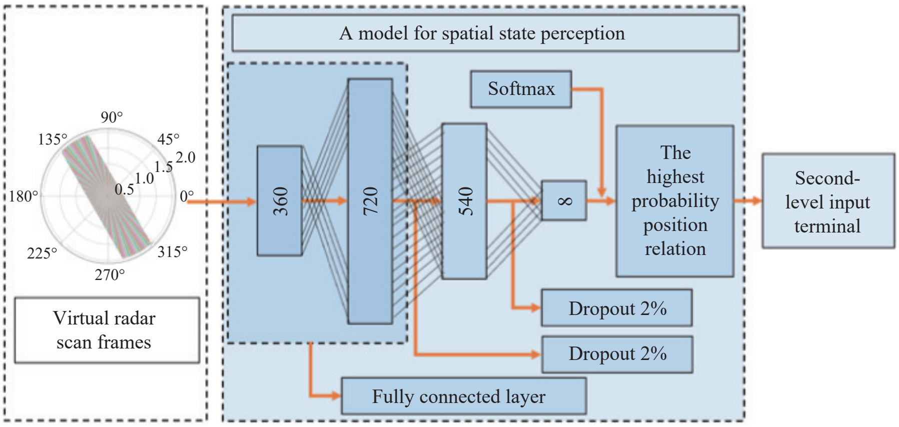

90°
45°
135°
2.0
1.5
1.0
0.5
180°
0°
225°
315°
270°
Virtual radar scan frames
A model for spatial state perception
Softmax
360
720
540
8
The highest probability position relation
Second-level input terminal
Dropout 2%
Dropout 2%
Fully connected layer

Figure 7 Structure and role of first-level deep neural network

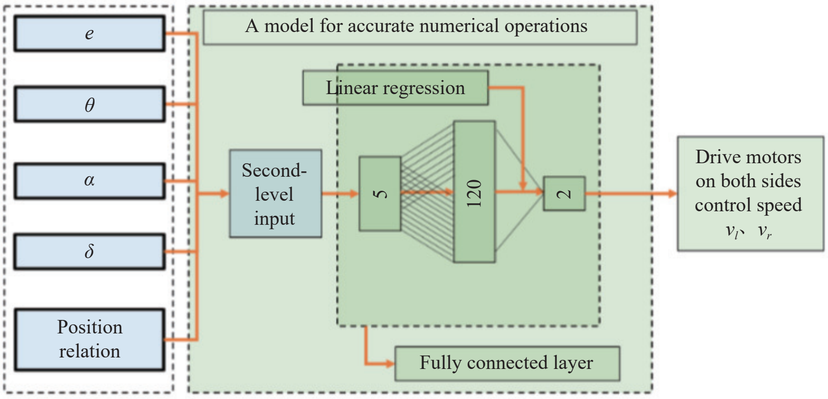

e
θ
α
δ
Position relation
A model for accurate numerical operations
Linear regression
Second-level input
5
120
2
Fully connected layer
Drive motors on both sides control speed v_l、v_r

Figure 8 Structure and role of second-level deep neural network

on the left side, in which way the tracking of the wheel-legged robot close to the target path is realized.

According to the above analysis, under the same heading deviation, the lateral tracking deviation plays an important role in the path tracking control of the wheel-legged robot in hills and mountains. If the influence of different lateral tracking deviations on the speed control of the drive motor is not considered, the control value will be single and cannot be well-adapted to the environment. Therefore, the lateral deviation distance factor e is introduced to linearly adjust the control speed, so that the wheel-legged robot can obtain the current best control speed under different lateral deviations. As shown in Figure 9, when the deviation of the heading angle is 0°, the method used in this paper is to measure the lateral deviation, and the setting method of the wheel-legged robot on the left and right of the target path is the same. The maximum value of the lateral tracking deviation factor e is taken at the point 1 m away from the target path, which is set to 1, and the corresponding wheel speed difference is 1 m/s. The minimum value of the lateral tracking deviation factor e at 0 m deviation from the target path is set to 0, and the corresponding wheel speed difference is 0 m/s at this time. Taking 0.2 m as the degree, the lateral tracking deviation influence factor e is linearly divided and set as an array of $[0.2, 0.4, 0.6, 0.8, 1]$, corresponding to the wheel speed difference of 0.2 m/s, 0.4 m/s, 0.6 m/s, 0.8 m/s, and 1.0 m/s, respectively, which is convenient for subsequent access.

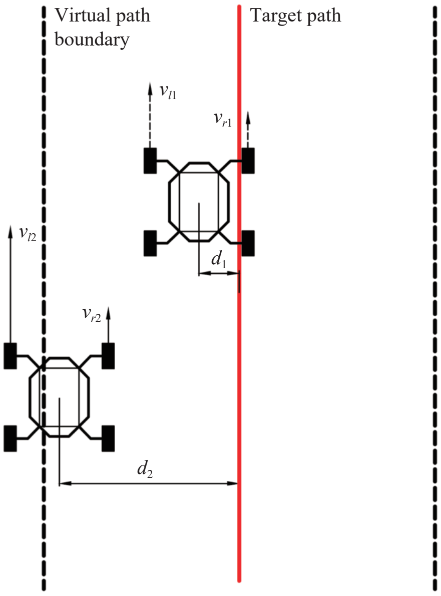


Figure 9 Control speeds under different lateral deviations and the same course deviation

(2) Heading angle deviation factor θ

Figure 10 shows the heading angle deviations of the wheel-legged robot under the same lateral tracking deviation d. When the wheel-legged robot is located on the left side of the target path, the larger the heading angle deviation value is, the larger the speed difference of the driving motor on both sides is set, so the wheel-legged robot can quickly return to the positive direction when the heading deviation is large. The smaller the heading angle deviation value is, the smaller the speed difference between the drive motors on both sides will be. At this time, the speed difference between the driving motors on both sides will gradually approach 0. Therefore, when the heading angle deviation of the wheel-legged robot is small on the slope, it can approach the target path line with the help of its current travel trend to avoid overshoot. When the wheel-legged robot is located on the right side of the target path, the speed difference of the motors on both sides of the wheel-legged robot is opposite to that when it is on the left side.

Heading angle change range is set as [–60°, 0°] or [0°, +60°], and the degree value is 1°. When the wheel-legged robot heading angle is 0°, the corresponding wheel differential speed is 0 m/s; when the heading angle of the wheel-legged robot is –60° or +60°, the corresponding wheel differential speed is 1 m/s; then the specific numerical control of the driving motor combined with the heading angle is realized.

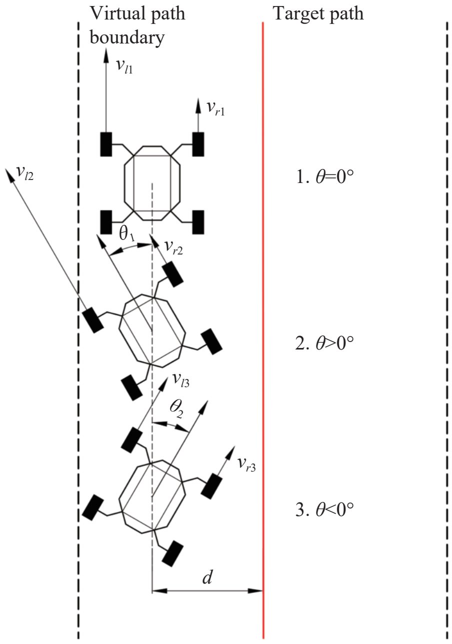


Figure 10 Heading angle deviations of wheel-legged robot under the same lateral deviation and different heading deviations

(3) Slope change factor α

The typical working environment of the wheel-legged robot in hilly and mountainous areas is set as the slope road, so it is necessary to comprehensively consider the road characteristics under the slope road and incorporate the control logic. The slope of the path will produce the component force of gravity. Combined with the previous dynamic analysis, the wheel-legged robot in hilly and mountainous areas needs to overcome the influence of gravity. The greater the slope is, the greater the influence of gravity that needs to be overcome. If the steering process is affected by gravity, it will lead to steering overshoot or understeering.

Generally, the ratio α of the vertical height and the horizontal distance of the slope is called slope. The road slope used in this paper directly takes the slope angle value, which is normalized and divided into integers $(0,30]$, and the degree is set to 1 to represent different road slopes, facilitating the learning of the neural network. When the wheel-legged robot is simulating the tracking of hilly and mountainous roads, the real-time slope is obtained through the virtual sensor installed on the robot body by the software, and the actual use is obtained by adding the inclination sensor.

(4) Slip rate factor δ

Hilly mountainous agricultural machinery is always affected by the gravity component force during tracking. When the agricultural machinery is greatly deflected, the gravity component force can easily cause the wheel to slip, which will lead to the deviation between the instantaneous center of the wheel-legged robot and the target path. However, in the actual work, it is difficult to measure the slip rate of the wheel in real time. At the same time, in order to reduce the cost and use of sensors, this paper converted the real-time slip rate of the road surface, obtained the heading angle of the agricultural wheel-legged robot at each time through a fixed time interval, combined with the content of the control command and the control time interval, and deduced through the mathematical relationship. In this paper, the agricultural wheel-legged robot adopts differential steering scheme control; that is, the wheel speed on the same side is the same at the same time, so no matter how many wheels slip at a certain time, it will lead to a deviation of the body that does not meet the expected instructions. The slope factor is calculated and derived based on the idea of the deviation degree of the whole vehicle. The various states that may occur when the wheel-legged robot is tracking are shown in Figure 11.

It is assumed that the heading angle of the wheel-legged robot is 0° before the first control command is issued; that is, it is assumed that there is no heading deviation at this time. S_l is the distance traveled by the left wheel in a control command cycle when the wheel-legged robot on the slope does not slip under ideal conditions:

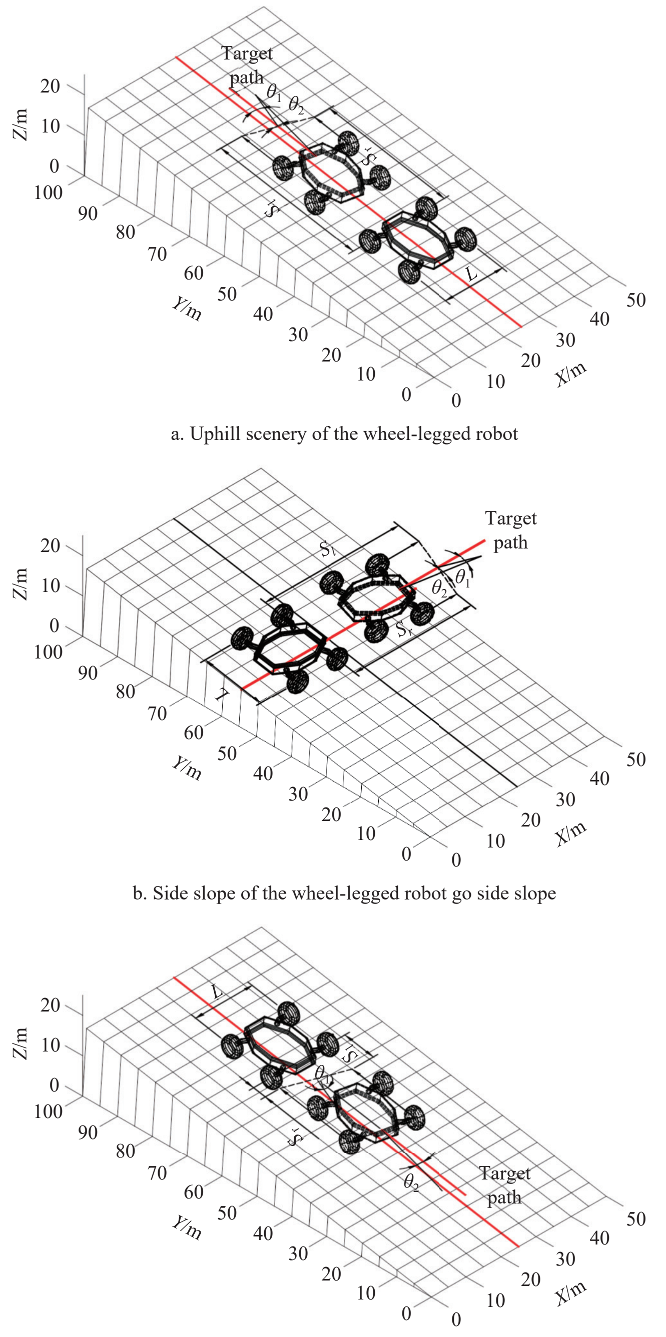
$$S_l = v_l \Delta t \tag{23}$$

where, Δt, v_l are the time interval between two wheels' speed control commands sent by the host computer, s; and the desired control speed of the left wheel at a certain time, m/s, respectively.

Similarly, the distance traveled in one instruction cycle when the right wheel does not slip can be obtained:

$$S_r = v_r \Delta t \tag{24}$$

According to the geometric relationship in Figure 11 and Equations (23) and (24), the heading angle change of the wheel-legged robot after a certain control command can be obtained as θ_1 in the ideal state:



a. Uphill scenery of the wheel-legged robot

b. Side slope of the wheel-legged robot go side slope

c. Downslope scenery of the wheel-legged robot

Figure 11 Heading angle changes of the wheel-legged robot in hilly and mountainous areas when slip occurs

$$\theta_1 = \arctan\left(\frac{S_l - S_r}{L}\right) = \arctan\left(\frac{(V_l - V_r)\Delta t}{L}\right) \tag{25}$$

However, when the slip occurs as the wheel-legged robot is running on hilly and mountainous slopes, the wheels on both sides cannot travel to the desired position. Thus the wheel-legged robot can only go to the position shown in the dotted line in Figure 11 in a control command cycle, and the current real heading angle θ_2 can be directly obtained by the navigation equipment.

Currently, the following formula is used to replace the expression of the wheel slip rate δ of the wheel-legged robot in hills and mountains:

$$\delta = \frac{\theta_1 - \theta_2}{\theta_1} = 1 - \frac{\theta_2}{\arctan\left(\dfrac{(V_l - V_r)\Delta t}{L}\right)} \tag{26}$$

Because the converted wheel slip rate needs to be used as the input data of the second-level deep neural network, and the slip rate is a decimal between $[0,1)$, which is not convenient for subsequent calculation, it is necessary to process the slip rate value. The calculated converted wheel slip rate is multiplied by 10 to a decimal whose value range is $[0,10)$. Then the integer part is taken as the valid data; that is, the converted wheel slip rate factor value range is an integer between $[0,9]$, and the degree is set to 1.

2.2.4 Construction of deep neural networks

Deep learning has been widely used in fields such as image recognition, unmanned driving, natural language processing, and so on. In this study, Python 3.9 and MXnet deep learning libraries were used to build the neural network framework, and the operating environment was macOS Monterey system.

1) First-level deep neural network construction

The number of input neurons of the first-level deep neural network is determined by the number of input virtual sensing radar detection beams. Since 1° is taken as the minimum angle of virtual sensing radar scanning per unit time, the input is set to 360 neurons for circular scanning. The two hidden layers are set to 720 and 540 neurons, respectively, through empirical formula and actual training results. The activation function is linear rectifier function (*Relu*), which is expressed as follows:

$$Relu(x) = \begin{cases} x(x>0) \\ 0(x \le 0) \end{cases} \tag{27}$$

Because the positive and negative heading angles are set, the control strategy of the wheel-legged robot on the left and right sides of the target path is symmetric about the target path, and there are eight in total. The deep learning model is set as the classification model, so the number of neurons in the output layer is set as eight. The output layer uses *SoftMax* function as the activation function, and the expression is as follows:

$$SoftMax(o)_k = \frac{e^{ok}}{\sum_{j=1}^{n} e^{ok}} \tag{28}$$

where, o_k, o_j, $Softmax(o)_k$ are the value of the k^{th} unit before input to *SoftMax*; the value of each unit before input to *SoftMax*; and the probability distribution of the k^{th} unit after processing by the *SoftMax* function, respectively.

The significance of *SoftMax* regression is to integrate the calculation results of the neural network in the output layer to obtain the probability distribution result with sum of 1. The result with the largest probability can be directly selected as the classification result, which is convenient for subsequent program processing. Figure 12 shows the structure of the first-level deep neural network.

2) First-level deep neural network training data

All neural network training data of the proposed algorithm are generated by the Python program, and the process principle is to generate all virtual sensing radar maps under a certain label as a set of training data. Values are as follows: the total heading angle deviation range $[-60°,+60°]$, each group takes $[-60°,0°]$ or $[0°,+60°]$, and the degree is 1°. The deviation range of lateral position distance is $[-1\text{ m},+1\text{ m}]$. The maximum lateral displacement span of single group training set is 0.5 m. The dividing value of lateral distance in each group is 0.005 m. Each set of training data is assigned the same label with a label value of $[0,7]$ and a scale of 1.

Through the above division method, the 2 m lateral deviation variation range of the wheel-legged robot is divided equally into four parts, and each part covers the heading angle deviation of $[-60°,0°]$ and $[0°,+60°]$. Therefore, a total of eight groups of training data are generated, and a total of 48 000 data are generated after the integration of the eight groups of training data generated by the program. Considering that various unexpected situations may occur in actual tracking, this study added 2000 training data corresponding to parking instruction labels, and finally formed a dataset of 50 000 for the training of the first-level deep neural network. This dataset is divided into training, validation, and test sets at a ratio of 8:1:1.

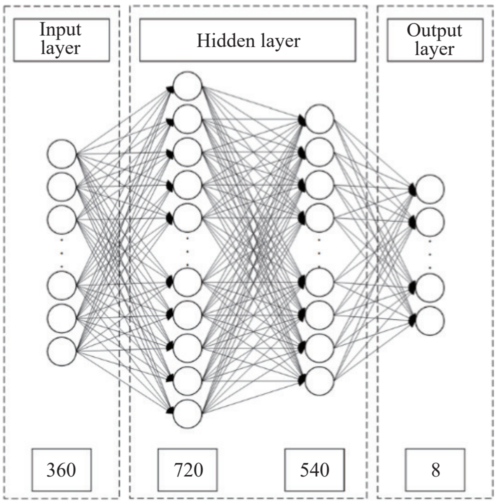


Figure 12 First-level deep neural network structure

3) Training first-level deep neural network

The weight parameters of the neural network were obtained by training the first-level deep neural network. The selected training mode was mini-batch Stochastic Gradient Descent (sgd), the training learning rate was 0.05, the batch size was 100, the number of iterations was 10, and the loss rate of the two hidden layers was 0.02. The loss function takes the cross-entropy loss function; the reason is that after the operation of the cross-entropy loss function, the output result only cares about the prediction probability of the correct class. If its value is large enough, it can ensure the correct classification result. The formula is as follows:

$$l(\Theta) = -\frac{1}{n}\sum_{i=1}^{n\hat{\sum}_{j}^{(i)}}\sum_{j=1}^{q\sum} y_j^{(i)} log \tag{29}$$

where, Θ, n, q, i, j, $y_j^{(i)}$, $\hat{y}_j^{(i)}$ are the model parameters of the first-level deep neural network; the number of samples in the training dataset; the output category of the neural network; the sample sequence number; the output category sequence number; the 0 or 1 elements in the real label probability distribution; and the predicted label probability distribution value, respectively.

4) Second-level deep neural network construction

The second-level deep neural network control system will further give specific control information by combining the relative position relationship between the wheel-legged robot and target path segments output by the first-level deep neural network. The control quantity of the robot is determined by the motor speeds on the left and right sides. Thus in this algorithm, a second-level deep neural network is employed to directly output the continuous control quantity using linear regression.

In this algorithm, the input of the second-level deep neural network includes the output of the first-level deep neural network and the heading angle of the current robot. At the same time, the slope, the converted slip rate, and the lateral deviation distance factor obtained by the simulation sensor are also used as input data. The output is the control speed of the right (left) motor, and the control wheel speed of the same side is the same at any time. A schematic diagram of the second-level deep neural network structure is shown in Figure 13.

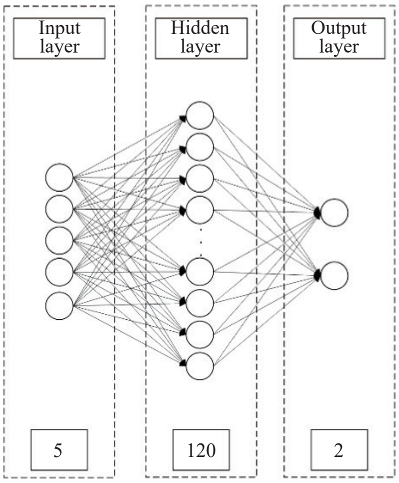


Figure 13 Second-level deep neural network structure

5) Second-level deep neural network training data

The training data of the second-level deep neural network is also directly generated by the program, and the output of the neural network is the speed control value of the motor. This includes the PWM value of the voltage of the drive motor, which is sent to the lower computer for processing and execution. The parameter settings for generating the training data are as follows: The heading angle variation range under the same lateral deviation is [–60°,0°] or [0°,+60°], and the degree is 1°. The range of slope influence factor α is (0°,30°], and the degree is 5°. Converted slip rate index range is [0,9], and degree is 1. The influence factor of lateral deviation distance is [0.2 m,1 m], and the degree is 0.2 m. The input one-level deep neural network label value is [0,7], and the degree is 1. The neural network outputs label values are set in the range [0,50] with a degree of 1. The output of the neural network is the left and right motor speed control values of [0,100].

Considering the heading angle deviation at any position, 60 sets of the determined drive motor's control speed data will be obtained. The full range of slope angle factors and slip rate factors α and δ are introduced to participate in data generation, and 3600 sets of data will be obtained. Finally, since the output label values of each first-level deep neural network correspond to a set of full-range data, a total of 144 000 arrays are used as the training data of the second-level deep neural network. It is also divided into training set, validation set, and test set at a ratio of 8:1:1.

6) Training second-level deep neural network

The network parameter training iteration of the second-level deep neural network is still using mini-batch stochastic gradient descent, and the batch size is set to 10. The learning rate is set to 0.001, and the number of iterations is set to 100. The activation function is still using linear rectifier function. Since there is only a fully connected layer and the result is relatively simple, the loss function is chosen as the square function, and its expression is:

$$l^{(i)}(\Theta) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2 \tag{30}$$

where, $l^{(i)}(\Theta)$, Θ , $\hat{y}^{(i)}$, $y^{(i)}$ are the i^{th} sample error, the model parameters of the second-level deep neural network, the i^{th} predicted value at the output of the neural network, and the standard label value corresponding to the i^{th} predicted value, respectively.

In this study, the general average method is used to evaluate the quality of the neural network training parameters; that is, the average of all sample errors in the training dataset is used to measure the quality of the model prediction, and the formula is expressed as:

$$l(\Theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2 \tag{31}$$

where, (Θ), n are the average loss value of all samples, and the number of all samples in the training set, respectively.

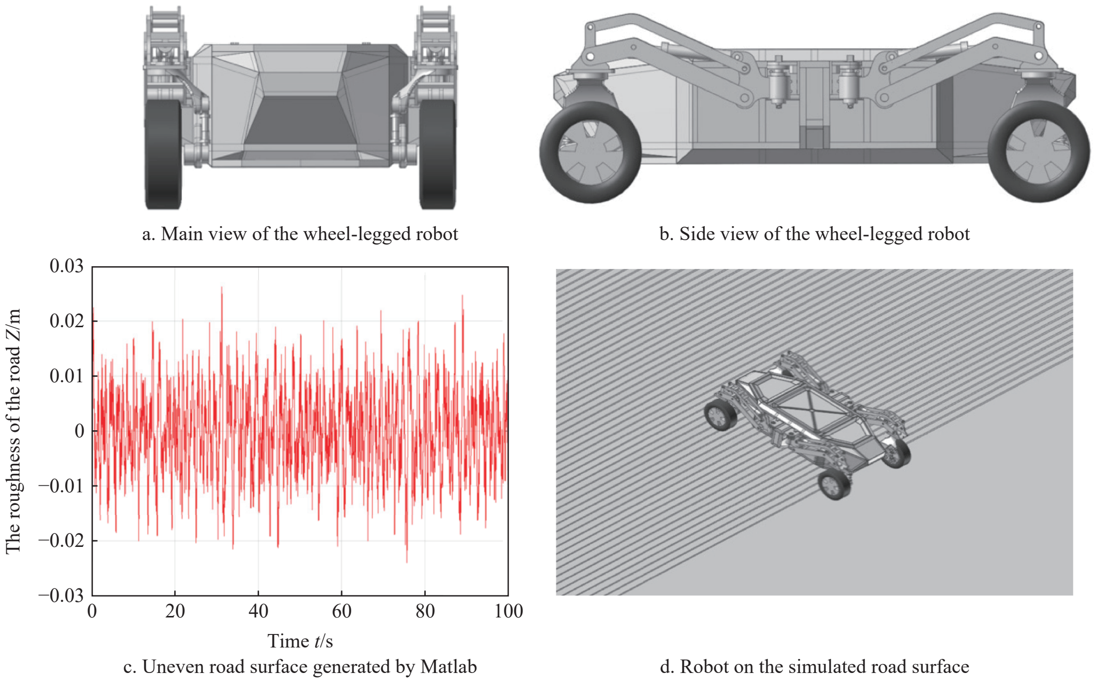
3 Results and discussion

In order to verify the effectiveness and accuracy of the proposed control algorithm, simulation tests under U-shaped paths were carried out by the Python-RecurDyn co-simulation model. The size of the target path was limited within a standard rectangle of 100 m×50 m, and the road condition was hilly and mountainous with a low adhesion coefficient, in which the wheels were easy to slip. Figure 14 shows the dynamic model of the agricultural robot in RecurDyn and the established uneven road surface generated by Matlab. The proposed slope path tracking algorithms were all run in PyCharm to complete the simulation tests in the form of co-simulation with RecurDyn, as shown in Figure 15.

Firstly, the frictional contact relationship between the wheels and the ground was established in RecurDyn software. Then the input and output of the FMU file in RecurDyn was set. The output of the FMU file used in the co-simulation is the relative coordinates, heading angle, slope, simulation time step, and other information of the current wheel-legged robot in the RecurDyn simulation environment. The input is the motor control speed value calculated through the path tracking control Python program. The generated FMU file would be imported into Python, and a virtual radar map is obtained through the proposed slope path tracking control algorithm to obtain the current robot-road relative position relationship. Combining the parameter values of the affecting factors obtained from FMU, the Python program continued to calculate the speed control value of the robot's driving motor and sent it to RecurDyn. The two softwares collaborated in the same time step to complete joint simulation testing.

In order to further verify the stability and accuracy of the proposed algorithm, the path tracking algorithm based on an optimized pure pursuit model (Yang et al.[12]) was also tested under the same simulated conditions as a comparison. Considering the influence of different slopes on the stability of the algorithm, three groups of comparative tests were carried out under the speed of 1 m/s, and the simulation road slope was set as 5°, 10°, and 15°, respectively[13-17]. Considering the influence of different robot speeds on the stability of the algorithm, three groups of comparative experiments were carried out under the road slope of 10° with the simulation robot speed of 0.5 m/s, 1 m/s, and 1.5 m/s, respectively. The benchmark target path of the test is formed by fitting a standard arc and a straight line[18-22]. The deviations of the test results were taken as absolute deviations. The path tracking results referring to different slope angles and tracking speeds are shown in Figures 16 and 17, separately. Tables 1 and 2 show the path tracking errors on slopes with different angles and speeds, separately[23-27].

It can be seen from Figure 16 that the tracking results of the wheel-legged robot controlled by the proposed algorithm are better compared with the optimized pure pursuit algorithm under different



a. Main view of the wheel-legged robot

b. Side view of the wheel-legged robot

c. Uneven road surface generated by Matlab

d. Robot on the simulated road surface

Figure 14 Multi-dynamic model of the agricultural wheel-legged robot

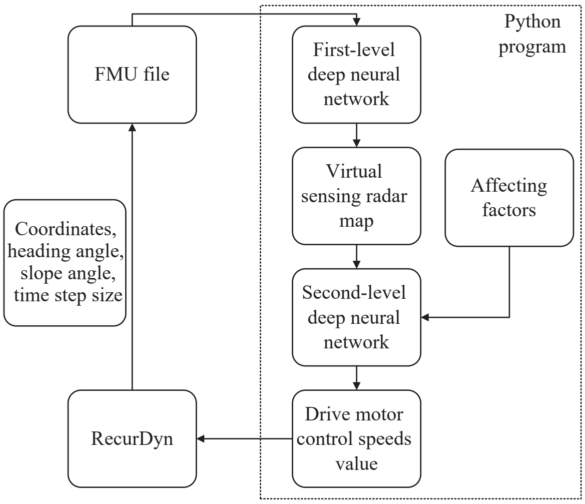

FMU file
First-level deep neural network
Python program
Virtual sensing radar map
Affecting factors
Coordinates, heading angle, slope angle, time step size
Second-level deep neural network
RecurDyn
Drive motor control speeds value

Figure 15 Principle of co-simulation model

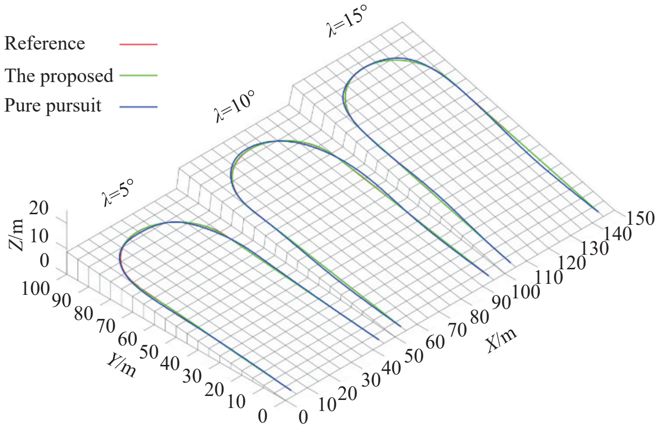


Figure 16 Simulation results under U-shaped slope paths of the proposed algorithm and the optimized pure pursuit model under slope angles of 5°, 10°, and 15°, respectively

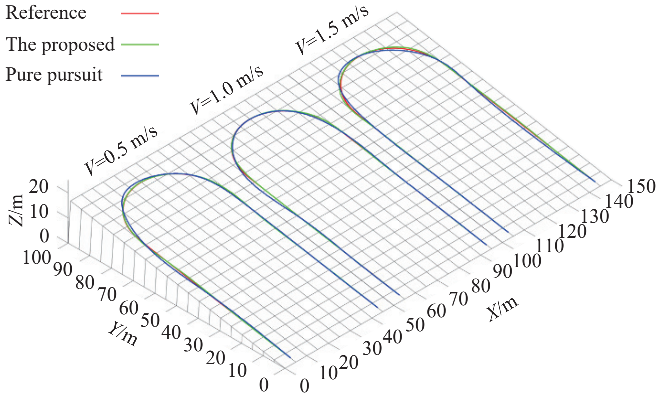


Figure 17 Simulation results under U-shaped slope paths of the proposed algorithm and the optimized pure pursuit model under robot speeds of 0.5 m/s, 1 m/s, and 1.5 m/s, respectively

Table 1 Path-following test data under different slopes

Experimental algorithm	Test slope/(°)	Maximum lateral deviation/m	Mean lateral deviation/m	Standard deviation/m
Algorithm proposed in this paper	5	0.249	0.226	0.024
	10	0.255	0.230	0.022
	15	0.367	0.235	0.023
Optimized pure pursuit algorithm	5	0.584	0.132	0.034
	10	0.673	0.326	0.035
	15	0.907	0.445	0.037

Table 2 Path-following test data at different speeds

Experimental algorithm	Test robot speed/m·s^{-1}	Maximum lateral deviation/m	Mean lateral deviation/m	Standard deviation/m
Algorithm proposed in this paper	0.5	0.246	0.227	0.023
	1.0	0.255	0.230	0.022
	1.5	0.364	0.236	0.022
Optimized pure pursuit algorithm	0.5	0.664	0.231	0.036
	1.0	0.673	0.326	0.035
	1.5	0.879	0.334	0.036

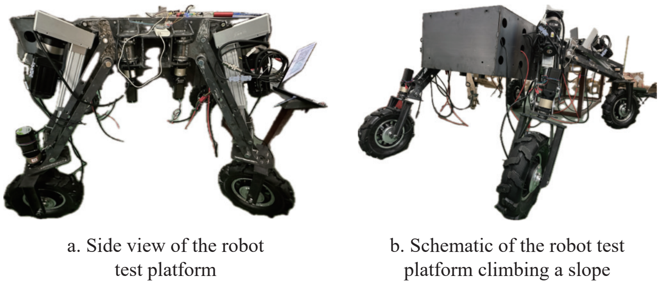
slope angles. Furthermore, under different slope angles, the maximum path tracking lateral deviation of the proposed algorithm is 0.367 m, whereas the corresponding value of the optimized pure pursuit algorithm is 0.907 m. The optimized pure pursuit algorithm has a large tracking error at path turns and a slow control response on the slope path[28-32]. As can be seen from Table 1, the agricultural robot can track the reference path accurately and stably, no matter whether the slope angle is 5°, 10°, or 15°. Even if there are some abrupt curvature changes on the slope path, the agricultural robot moves without large fluctuations and shocks. In conclusion, with the increase of the slope angle, the tracking errors change very little when using the proposed algorithm, which indicates good self-adaptability to the change of slope angles.

With respect to the tracking speed, as in the results shown in Figure 17 and Table 2, it can be seen that the proposed algorithm still maintains a relatively stable tracking error growth rate with increasing speeds. The maximum deviation of path tracking of the proposed algorithm is 0.364 m, whereas the maximum tracking deviation of the optimized pure pursuit algorithm is 0.879 m. The simulation results reveal that the proposed algorithm's effectiveness and accuracy are superior, with tracking errors reduced by more than 47.2% compared to the optimized pure pursuit algorithm. Therefore, compared with the general planar tracking control algorithm, the control system proposed in this study has better control ability for the agricultural robot's motion on the slope[33-37].

In Yang's study in which the optimized pure pursuit algorithm was proposed, the results of planar field tracking tests showed that the average value of the lateral error of the straight path is 0.012 m and the average value of the lateral error for curved paths is 0.07 m (Yang et al.[12]). However, when tracking on a slope path, the error of this optimized pure pursuit algorithm increases by two to six times, as shown in Tables 1 and 2, which can be attributed to the neglect or direct disregard of the influence of gravity in slope path tracking. On the contrary, the path tracking control algorithm proposed in this paper pays special attention to this issue and takes some consideration and compensation control. The accuracy of path tracking has been significantly improved compared to the aforementioned algorithm, although there is still room for further improvement in the absolute value of the tracking errors[38].

To further validate the effectiveness and practicality of the proposed algorithm, this study constructed a physical robot test platform and conducted relevant performance tests to adapt to the slope conditions mentioned in the algorithm. The physical platform is illustrated below.

Figure 18 depicts the constructed physical test platform. The main body of the platform is assembled from steel, with four independently controlled support legs. The robot's slope climbing is achieved through the extension and retraction of electric cylinders, while steering is controlled by a steering motor. The propulsion power is provided by four hub motors, and the entire system is powered by an onboard 48V battery. The integrated controller consists of an upper computer and a microcontroller. After completing the robot assembly, performance tests were conducted on the wheel-legged robot, and the results are presented in the Table 3 below.


a. Side view of the robot test platform

b. Schematic of the robot test platform climbing a slope

Figure 18 Constructed physical robot test platform

Table 3 Performance test results of physical robot test platform

Item		Design parameters	Actual parameters
Vehicle parameters	Full-load mass/kg	≤400	356
	Dimensions/mm	Length ≤2100, Width ≤ 1000, Height ≤700	Length 1900, Width 850, Height 585
	Drive type	Four-wheel drive	Four-wheel drive
	Endurance/h	≥5	5.2
Dynamic performance	Maximum speed/km·h^{-1}	≥30	30
	Maximum climb angle/(°)	≥30	30
Mobility	Body leveling range	Pitch angle ≥20°, Roll angle ≥10°	Pitch angle 20°, Roll angle 10°
	Obstacle clearance height/mm	≥500	550

The data in the table demonstrate that the constructed physical robot platform is capable of adapting to the slope conditions addressed by the proposed algorithm, providing a solid experimental foundation for further algorithmic improvements.

4 Conclusions and future work

In this study, the slope path tracking control problem of agricultural robot was studied. Firstly, the kinematic and dynamic models of the wheel-legged robot on the hilly terrain were established, from which the crucial factors affecting slope path tracking control accuracy were recognized. Secondly, a path tracking control algorithm based on virtual sensing radar and two-level deep neural network was proposed. The algorithm used the virtual sensing radar to detect the path boundary as the perception input of the agricultural robot, and combined the first-level deep neural network to analyze the virtual radar map to obtain the real-time robot-road position relationship, as well as to predict the road direction. Furthermore, by analyzing the identified crucial factors affecting slope path tracking control accuracy, the second-level deep neural network obtained the accurate real-time drive motor speed values. Finally, the slope path tracking control simulations of the wheel-legged robot were carried out under various angles and velocities. Compared with the traditional planar tracking control algorithm, the simulation results show that the proposed algorithm has better performance.

However, in this study, the unexpected deviation of the agricultural robot caused by slipping between wheels and road surfaces may not be well considered in the tracking process, so the stability of the algorithm under bad working conditions needs to be verified. In the future, more slope control parameters will be introduced to achieve more accurate and stable path tracking control of the agricultural wheel-legged robot in hilly and mountainous areas.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No. 2022YFD2202102) and the Key Laboratory of Modern Agricultural Intelligent Equipment in South China, Ministry of Agriculture and Rural Affairs, China.

[References]

[1] D'Auria D, Ristorto G, Raimondo G, Mazzetto F. Tracked robot over a slope path: Dynamic stability control. In: 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), Pittsburgh, PA, USA: IEEE, 2016; pp.496–499.

[2] Dogan M U, Guvenc U, Elmas C. Genetic PI based model and path

tracking control of four traction electrical robot vehicle. Electrical Engineering, 2020; 102: 2059–2073.

[3] Jeong Y. Path tracking control for four-wheel-steering autonomous vehicle based on adaptive sliding mode control with control allocation. In: 2021 21st International Conference on Control, Automation and Systems (ICCAS), South Korea: IEEE, 2021; pp.1741–1746.

[4] Qi H Y, Shangguan J Y, Fang C, Yue M. Path tracking control of car-like wheeled mobile robot on the slope based on nonlinear model predictive control. In: 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China: IEEE, 2022; pp.465–470.

[5] Hu J T, Li T C. Cascaded navigation control for agricultural vehicles tracking straight paths. Int J Agric & Biol Eng, 2014; 7(1): 36–44.

[6] Yue X, Chen J K, Li Y Q, Zou R, Sun Z H, Cao X C, et al. Path tracking control of skid-steered mobile robot on the slope based on fuzzy system and model predictive control. International Journal of Control, Automation and Systems, 2022; 20(4): 1365–1376.

[7] Yang Z J, Mao L, Yan B, Wang J, Gao W. Performance analysis and prediction of asymmetric two-level priority polling system based on BP neural network. Applied Soft Computing, 2021; 99: 106880.

[8] Zhou Z B, Zhang X M, Li Z J, Huang F R, Xu J. Multilevel attention networks and policy reinforcement learning for image caption generation. Big Data, 2022; 10(6): 481–492.

[9] Li Y W, Zang L G, Shi T, Lv T, Lin F. Design and dynamic simulation analysis of a wheel–Track composite chassis based on RecurDyn. World Electric Robot Journal, 2022; 13(1): 12.

[10] Bai G X, Meng Y, Liu L, Gu Q, Huang J X, Liang G D, et al. Path tracking for car-like robots based on neural networks with NMPC as learning samples. Electronics, 2022; 11: 4232.

[11] Liu Z J, Wang X L, Ren Z G, Mao W J, Yang F Z. Crawler tractor navigation path tracking control algorithm based on virtual sensing radar model. Transactions of the Chinese Society for Agricultural Machinery, 2021; 52(6): 376–385.

[12] Yang Y, Li Y K, Wen X, Zhang G, Ma Q L, Cheng S K, et al. An optimal goal point determination algorithm for automatic navigation of agricultural machinery: Improving the tracking accuracy of the Pure Pursuit algorithm. Computers and Electronics in Agriculture, 2022; 194: 106760.

[13] Zhang F, Zheng L M, Wang W, Wang Y F, Wang J J. Development of agricultural bionic mechanisms: Investigation of the effects of joint angle and pressure on the stability of goats moving on sloping lands. Int J Agric & Biol Eng, 2018; 11(3): 35–41.

[14] Liu Z D, Zheng W X, Wang N, Lyu Z Q, Zhang W Z. Trajectory tracking control of agricultural vehicles based on disturbance test. Int J Agric & Biol Eng, 2020; 13(2): 138–145.

[15] Yin X, Wang Y X, Chen Y L, Jin C Q, Du J. Development of autonomous navigation controller for agricultural vehicles. Int J Agric & Biol Eng, 2020; 13(4): 70–76.

[16] Bayar G, Bergerman M, Koku A B, Konukseven E I. Localization and control of an autonomous orchard vehicle. Computers and Electronics in Agriculture, 2015; 115: 118–128.

[17] Backman J, Oksanen T, Visala A. Navigation system for agricultural machines: Nonlinear Model Predictive path tracking. Computers and Electronics in Agriculture, 2012; 82: 32–43.

[18] Chaudhary N, Gupta A. Multi-body analysis for a four-bar mechanism using RecurDyn and MATLAB. In: Kumar R, Chauhan V S, Talha M, Pathak H. (eds). Machines, Mechanism and Robotics. Singapore: Springer. 2022; pp.1813–1823.

[19] Deng J, Zhou H R, Lv X, Yang L J, Shang J L, Sun Q, et al. Applying convolutional neural networks for detecting wheat stripe rust transmission centers under complex field conditions using RGB-based high spatial resolution images from UAVs. Computers and Electronics in Agriculture, 2022; 200: 107211.

[20] Ding Y, Wang L, Li Y W, Li D L. Model predictive control and its application in agriculture: A review. Computers and Electronics in Agriculture, 2018; 151: 104–117.

[21] Gratton S, Kopaničáková A, Toint P L. Multilevel objective-function-free optimization with an application to neural networks training. arXiv: 2302.07049, 2023; In press.

[22] Graf Plessen M M, Bemporad A. Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines. Biosystems Engineering, 2017; 153: 28–41.

[23] Han X, Kim H J, Jeon C W, Moon H C, Kim J H, Seo I H. Design and field testing of a polygonal paddy infield path planner for unmanned tillage operations. Computers and Electronics in Agriculture, 2021; 191: 106567.

[24] He Y Q, Zhou J, Yuan L C, Zheng P Y, Liang Z A. Local tracking path planning based on steering characteristics of Crawler-type combine harvester. Transactions of the Chinese Society for Agricultural Machinery, 2022; 53(11): 13–21.

[25] Hussain M, Qazi E U H, Aboalsamh H A, Ullah I. Emotion recognition system based on two-level ensemble of deep-convolutional neural network models. IEEE Access, 2023; 11: 16875–16895.

[26] He J, Hu L, Wang P, Liu Y X, Man Z X, Tu T P, et al. Path tracking control method and performance test based on agricultural machinery pose correction. Computers and Electronics in Agriculture, 2022; 200: 107185.

[27] Han X Z, Kim H J, Kim J Y, Yi S Y, Moon H C, Kim J H, et al. Path-tracking simulation and field tests for an auto-guidance tillage tractor for a paddy field. Computers and Electronics in Agriculture, 2015; 112: 161–171.

[28] Liu F, Meng W, Yao D Y. Bounded antisynchronization of multiple neural networks via multilevel hybrid control. IEEE Transactions on Neural Networks and Learning Systems, 2023; 34(11): 8250–8261.

[29] Li D, Kwak S, Geroliminis N. TwoResNet: Two-level resolution neural network for traffic forecasting on freeway networks. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China: IEEE, 2022; pp.3963–3969.

[30] Murillo M, Sánchez G, Deniz N, Genzelis L, Giovanini L. Improving path-tracking performance of an articulated tractor-trailer system using a non-linear kinematic model. Computers and Electronics in Agriculture, 2022; 196: 106826.

[31] Milo V, Anzalone F, Zambelli C, Perez E, Mahadevaiah M K, Ossorio O G, et al. Optimized programming algorithms for multilevel RRAM in hardware neural networks. In: 2021 IEEE International Reliability Physics Symposium (IRPS), Monterey, CA, USA: IEEE, 2021; pp.1–6. doi: 10.1109/IRPS46558.2021.9405119.

[32] Naqi M, Kang M S, Liu N, Kim T, Baek S, Bala A, et al. Multilevel artificial electronic synaptic device of direct grown robust MoS2 based memristor array for in-memory deep neural network. Npj 2D Materials and Applications, 2022; 6: 53.

[33] Sohn J H, Lee C H, Kim Y J, Kim S S. Evaluation of path tracking performance of a self-driving tracked vehicle. Transactions of the Korean Society of Mechanical Engineers, 2022; 45(12): 1167–1176.

[34] Wu Y H, Duan Y H, Wei Y G, An D, Liu J C. Application of intelligent and unmanned equipment in aquaculture: A review. Computers and Electronics in Agriculture, 2022; 199: 107201.

[35] Zyarah A M, Soures N, Hays L, Jacobs-Gedrim R B, Agarwal S, Marinella M, et al. Ziksa: On-chip learning accelerator with memristor crossbars for multilevel neural networks. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA: IEEE, 2017; pp.1–4.

[36] Zhan Z Q, Si G M, Zhi J J, Liu L. Design and analysis based on RecurDyn of the electric crawler type remote controlled hedge trimmer. Journal of Physics: Conference Series, 2020; 1633(1): 012010.

[37] Zhang L H, Zhang R R, Li L L, Ding C C, Zhang D Z, Chen L P. Research on virtual Ackerman steering model based navigation system for tracked vehicles. Computers and Electronics in Agriculture, 2022; 192: 106615.

[38] Zhang W Y, Gai J Y, Zhang Z G, Tang L, Liao Q X, Ding Y C. Double-DQN based path smoothing and tracking control method for robotic vehicle navigation. Computers and Electronics in Agriculture, 2019; 166: 104985.