Autonomous trajectory tracking control method for an agricultural robotic vehicle

Jin Yan[1], Wenguang Zhang[2,3], Yong Liu[1*], Wei Pan[1], Xiaoyu Hou[1], Zhiyu Liu[4]

(1. *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*;
2. *School of Information Engineering, Xuzhou University of Technology, Xuzhou 221018, Jiangsu, China*;
3. *National Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China*;
4. *School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*)

Abstract: To address the nonlinearities and external disturbances in unstructured and complex agricultural environments, this paper investigates an autonomous trajectory tracking control method for agricultural ground vehicles. Firstly, this paper presents the design and implementation of a lightweight, modular two-wheeled differential drive vehicle equipped with two drive wheels and two caster wheels. The vehicle comprises drive wheel modules, passive wheel modules, battery modules, a vehicle frame, a sensor system, and a control system. Secondly, a novel robust trajectory tracking method was proposed, utilizing an improved pure pursuit algorithm. Additionally, an Online Particle Swarm Optimization Continuously Tuned PID (OPSO-CTPID) controller was introduced to dynamically search for optimal control gains for the PID controller. Simulation results demonstrate the superiority of the improved pure pursuit algorithm and the OPSO-CTPID control strategy. To validate the performance, the vehicle was integrated with a seeding and fertilizing machine to realize autonomous wheat seeding in an agricultural environment. Experimental outcomes reveal that the vehicle of this study completed a seeding operation exceeding 1 km in distance. The proposed method can robustly and smoothly track the desired trajectory with an accuracy of less than 10 cm for the root mean square error (RMSE) of the curve and straight lines, given a suitable set of parameters, meeting the requirements of agricultural applications. The findings of this study hold significant reference value for subsequent research on trajectory tracking algorithms for ground-based agricultural robots.

Keywords: trajectory tracking, autonomy control, agricultural robotic vehicle, online PSO continuously tuned PID, dynamic pure pursuit algorithm

DOI: 10.25165/j.ijabe.20241701.7296

Citation: Yan J, Zhang W G, Liu Y, Pan W, Hou X Y, Liu Z Y. Autonomous trajectory tracking control method for an agricultural robotic vehicle. Int J Agric & Biol Eng, 2024; 17(1): 215–224.

1 Introduction

Agriculture, as a fundamental and public welfare industry, plays a crucial role in the national economy and people's livelihood. Over the recent decades, the emergence of smart agriculture has begun to overturn traditional agriculture, and smart agriculture is ushering in a period of development[1]. Advancements in sensor technologies[2], artificial intelligence[3], the Internet of Things[4], and 5G[5] have accelerated the transformation of traditional agriculture to intensification, precision, and intelligence. Smart agriculture encompasses a wide array of processes, including farmland management, sowing, fertilization, irrigation, and crop conservation.

Agricultural robots are widely used to replace humans in agricultural operations, and there has been a large amount of research into transforming or designing vehicles appropriate for agricultural applications. Bell[6] applied a John Deere 7800 farm tractor to realize fully autonomous row guidance. Matveev et al.[7] realized automatic path tracking by using a John Deere 4210 Compact Utility Tractor. Zhang et al.[8] utilized a Foton Lovol TG1254 tractor to conduct automatic navigation tests in the field. The above-mentioned agricultural robots are on the basis of the existing tractors. Some researchers have designed different types of agriculture robots based on specific requirements. The most representative agricultural robot-- BoniRob was initially developed for phenotyping[9], and several other application modules ("apps"), for example, precision spraying app[10] and penetrometer app, have been developed for the robot. Bakker et al.[11,12] developed an autonomous platform for robotic weeding by using a structured design approach. Bawden et al.[13] described a lightweight, modular, and energy-efficient robotic vehicle platform designed for broadacre agriculture and to promote the sustainable intensification of agriculture by allowing farmers to concentrate on critical farm management tasks. Grimstad et al.[14] presented a modular mobile agricultural robot designed explicitly for phenotyping tasks. The robot design should be consistent with the working environment and the function. This study designed a Modular Autonomous Farm Vehicle (MAFV) according to the authors' preliminary investigations and field trips into vehicle structure, agricultural environments, and other demands. Through feasibility analysis and experimental verification, the vehicle is flexible enough to adapt to agricultural terrain and conduct agricultural operations. In addition, a Qt-based remote communicating and control interface is created and available at https://github.com/Jimmy101250/Agri_remote_

Received date: 2021-12-24 **Accepted date:** 2023-06-23

Biographies: Jin Yan, PhD candidate, research interest: agricultural robot technology, Email: yanjin@njust.edu.cn; **Wenguang Zhang**, PhD, Lecture, research interest: robot control technology, Email: wgz8911@163.com; **Wei Pan**, Master, research interest: robot mapping, Email: panwei0914@njust.edu.cn; **Xiaoyu Hou**, Master, research interest: agricultural engineering, Email: 119106010625@njust.edu.cn; **Zhiyu Liu**, Master, research interest: mechanical Engineering, Email: liuzhiyu@njust.edu.cn.

***Corresponding author: Yong Liu**, PhD, Professor, research interest: agricultural robot technology. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. Tel: +86-13813972137, Email: liuy1602@njust.edu.cn.

interface.git.

Path tracking control, which involves generating control commands to follow the predefined path, is a fundamental and crucial technology for autonomous farm vehicles. Various path-tracking control methods are documented in the literature, generally categorized into three groups[15]: geometric methods (e.g., pure pursuit, Stanley controller), optimal control-based methods, and model-based methods (e.g., PID, model predictive control, sliding mode control). Compared with other path-tracking algorithms, the pure pursuit algorithm (PPA) has a simpler implementation principle and better tracking results[16], aiming to calculate the circular path to arrive at the instantaneous goal point. Wallace et al.[17] first proposed a pure pursuit strategy to estimate the steering angle and follow the path on the road. Cao et al.[18] used the pure pursuit algorithm to obtain the target steering wheel angle and vehicle speed. The preview distance and command speed were fixed in this study. Some researchers adjusted the preview distance in a way that the preview distance is directly proportional to the vehicle velocity[19,20]. Yu et al.[21] used the fuzzy rule controller to adjust the preview distance. Gámez Serna et al.[22] estimated dynamic preview distance based on the vehicle speed and lateral error. Wang et al.[23] improved the preview distance with a 2-degree polynomial function. The aforementioned improved pure pursuit algorithms[19-23] were deployed on the Ackerman model vehicles. The procedure of the algorithms is determining the preview distance via velocity, and then calculating the steering angle using Ackerman geometry.

Besides pure pursuit algorithms, PID control-based algorithms are widely used in autonomous driving. Bakker et al.[11,24] applied the PID control method to be deployed on an automatic weeding robotic platform. To improve the performance of PID controller, intelligent algorithms such as Fuzzy control, Particle Swarm Optimization (PSO), or neural networks (NNs) were utilized. Han et al.[25] established a neural network PID controller for lateral path tracking. As an intelligent evolutionary method, the PSO algorithm has been combined with the traditional PID method to achieve a satisfactory control performance. Al-Mayyahi et al.[26] utilized a fractional order PID controller for achieving autonomous path tracking and a PSO algorithm was used to optimize the controllers' parameters. This method can easily be used in engineering applications, but there is the disadvantage that the algorithms and parameters are application-dependent. Poultangari et al.[27] proposed a radial basis function (RBF) neural network-based PI controller for collective pitch control. PSO algorithm is used to provide an optimal dataset to train the RBF neural network. Kashyap et al.[28] introduced a PSO-tuned PID controller, tuned the parameters of the conventional PID controller, and provided an optimum turning angle.

This study proposed a robust trajectory tracking control method combining an improved dynamic pure pursuit algorithm and an Online PSO Continuously Tuned PID (OPSO-CTPID) controller. The main contributions of this work are three-fold: 1) A lightweight, modular farm vehicle was designed and set up. In addition, a Qt-based remote communicating and control interface was created; 2) To achieve robust trajectory tracking, the pure pursuit algorithm was modified according to the kinematic model of the two-wheeled differential steering vehicle and a dynamic preview distance adjustment method was proposed; 3) A novel Online PSO Continuously Tuned PID (OPSO-CTPID) controller was introduced that effectively addresses the nonlinearities and external disturbances present in agricultural environments. Compared with the results of References [29-31], the proposed control scheme does not rely on the actual output of the system to calculate the fitness function, or carefully chosen initial values for the PID parameters. What's more, it is cascade-connected to a modified pure pursuit algorithm and kinematic model to realize precise control.

2 Materials and methods

2.1 Vehicle platform

The robotic vehicle was developed as a Modular Autonomous Farm Vehicle (MAFV). Specifically, a lightweight, modular two-wheeled differential drive vehicle was designed and established with two drive modules connected directly to the front of the vehicle frame and two caster wheels at the rear. The vehicle comprises drive modules, passive wheel modules, battery modules, a vehicle frame, a sensor system, and a control system, which exhibits high flexibility and extendibility. The vehicle's modularity enables it to be remodeled as a four-wheel-diving (4WD) vehicle or a four-wheel-steering four-wheel-drive (4WS/4WD) vehicle by refitting passive wheel modules or adding steering modules, thereby enhancing its flexibility. Furthermore, the vehicle's extendibility allows it to accommodate a sowing mechanism, fertilizing mechanism, or spraying mechanism for various applications, such as seeding, fertilizing, or spraying. The software platform is based on ROS and we created a Qt-based remote communication and control interface to control the vehicle remotely and obtain the vehicle position information in real-time.

2.1.1 Vehicle design

As shown in Figures 1 and 2, the design specifications of the MAFV are a vehicle mass of about 200 kg, a rated speed of no less than 5 km/h, a passable height of no less than 60 cm, a continuous working time of no less than 8 h, the operating gradient of no less than 15°, and the emergency brake equipped.

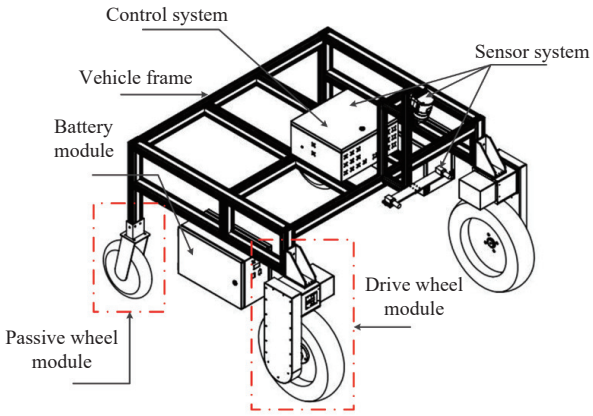


Figure 1 Design schematic diagram of the Modular Autonomous Farm Vehicle (MAFV) in this study

2.1.2 Drive system

Figure 3 presents the exploded view of the drive wheel module. After rigorous force calculations, a chain transmission mechanism was selected as the medium for transmitting motion between the motor and the wheel. The 750 W motor is directly connected to a reducer with a transmission ratio of 60:1. The reducer is secured to the upper end of the seat plate using fastening elements, and its output shaft is rigidly connected to a sprocket. A bearing sleeve is affixed to the lower end of the seat plate, while the wheel and another sprocket are attached to the two ends of the wheel shaft. The bearing sleeve and wheel shaft achieve relative rotation through the bearings. Motion is transmitted between the two sprockets via a chain.

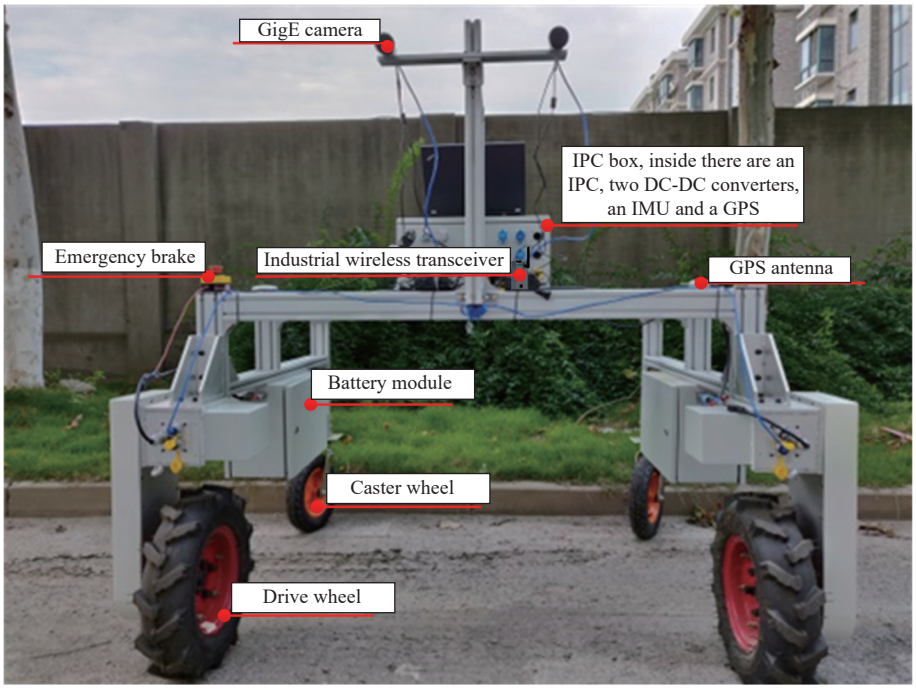


Figure 2 Actual mechanical structure display of MAFV

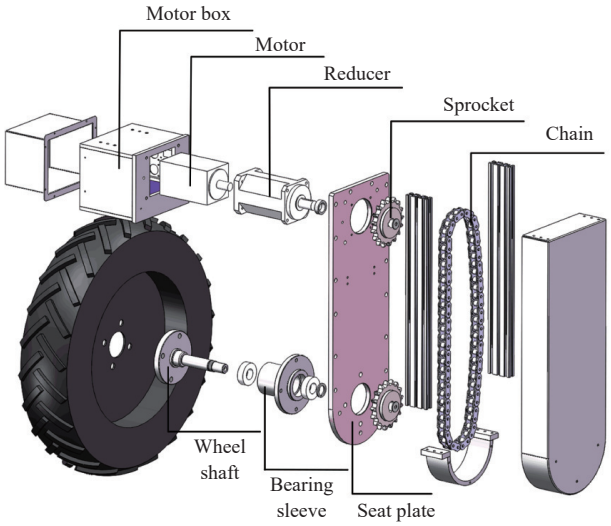


Figure 3 Exploded view of the drive wheel module

2.1.3 Power analysis

Some significant detailed specifications of the wheel, battery, and motor are listed in Table 1 so that we can analyze the feasibility of the vehicle.

Table 1 Detailed specifications of the MAFV

Detailed Specification	Measurement
Drive wheel diameter/mm	580
Battery's capacity/Ah	80
Power of electric generator/kW	0.75
Continuous drive torque/N·m	2.39
Peak drive torque/N·m	6
Reduction ratio	60
Rated voltage/V	48
Rated speed/r·min^{-1}	3000
Rated current/A	20

1) Speed

With the parameters of the rated speed of the motor, reduction ratio of the gear reducer, and drive wheel diameter, the rated speed of the vehicle is 5.46 km/h (1.52 m/s), which satisfies a majority of agricultural demands.

2) Force analysis

The force required to drive a vehicle is given by

$$F = F_{\text{rolling}} + F_{\text{gradient}} + F_{\text{acceleration}} = C_r mg\cos\theta_{\text{gradient}} + mg\sin\theta_{\text{gradient}} + ma \quad (1)$$

where, F is the total required force, N; F_{rolling} is the force of rolling friction, N; F_{gradient} denotes the force needed to overcome gravity, N; $F_{\text{acceleration}}$ is the force required to provide acceleration, N. m represents the total mass of the vehicle, kg; g represents the acceleration of gravity, N/kg; a is the acceleration, m/s^2; θ_{gradient} is the incline angle, (°); C_r is the coefficient of rolling friction, on the surface of loose soil and wet soil/mud, C_r is 0.1 and 0.2, respectively.

Two important forces need to be calculated, one is the average force requirement under normal conditions which is used to calculate average power consumption and continuous working time on a charge. Another is peak force requirement under the worst conditions, which is used to analyze the adaptive capacity of the vehicle under harsh agricultural environments.

Under normal conditions, given the mass of the vehicle is 200 kg and with the payload of 100 kg, to make the vehicle work on the soil at a constant speed of 5 km/h, the force demand is 300 N, and the power requirement is 417 W.

When the vehicle is required to accelerate (about 0.6 m/s^2) on wet soil up a gradient of 15° under the supposed worst condition, the peak force requirement is 1509 N. With the parameters of drive wheel diameter, continuous drive torque (each motor), peak drive torque (each motor), and reduction ratio in Table 1, the average force and the maximum force the vehicle can provide are 988 N and 2482 N, respectively.

3) Batteries

One battery module holds an 80 Ah, 48 V lithium battery pack and there are two battery modules installed on the left and right sides of the vehicle frame. Inside the Industrial Personal Computer

(IPC) box, there are two DC-DC converters, 48 V-24 V and 48 V-12 V, respectively. The inputs of the DC-DC converters are connected to the 48V battery, and the outputs are the power supply of the IPC and other sensors. Assume that the IPC and other sensors consume 250 W in total and overall efficiency of the battery is 75%, and the continuous working time of the vehicle is over 8 h.

2.1.4 Control and computation system

The control and computation system consists of an IPC and a PC. On the vehicle there is an IPC, whose model is ADLINK MVP-6001, with an Intel Core i7-6700TE CPU and 8GB RAM, installing Ubuntu Linux 16.04, mainly running path tracking node, processing sensor data and sending control commands to motor drivers via CAN communication. PC was used as a remote controller, installing Windows 10, mainly running remote operations, path planning algorithm, and logging in the IPC remotely.

2.1.5 Sensors

A multi-modal sensor suite provides information for autonomy and perception. In Figure 2, the vehicle of this study was equipped with two industrial GigE cameras, dual antenna direction finding RTK-GPS, IMU, an industrial wireless transceiver, and a wireless joystick.

2.2 Software development

2.2.1 Software platform

To make the vehicle's code modular and simplify the task, a Robot Operating System (ROS)[32] was chosen as the software framework. As mentioned above, the IPC runs Ubuntu16.04, and the ROS version is Kinetic. IPC runs ROS nodes for trajectory tracking, localization, communicating with motor drivers over the CAN ethernet, and communicating with another PC.

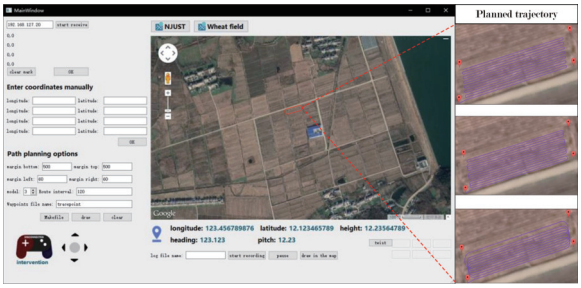
2.2.2 Remote operation

A Qt-based remote communication and control interface to control the vehicle remotely and obtain the vehicle position information in real-time was created as shown in Figure 4. The following functions are integrated into the interface according to actual needs map display. Offline Google tile maps of the farm are integrated into the remote operation interface.

Remote sharing. PC connects with a wireless joystick, and there are two modes, normally the vehicle will track the path autonomously; once an emergency occurs, the joystick takes over control rights and the operator can control the vehicle out of trouble, for instance, serious slipping or sudden obstacles.

Path planning. Assign coordinates in the map, and define some necessary parameters such as linewidth, and margin length, the program will generate a series of waypoints and display the planned path in the interface.

Remote communication. The remote side is able to communicate with the master (ROS) and obtain the node information. Thus, the vehicle's position and state information can be grasped and displayed in the interface. The real-time position of the vehicle will be shown on the map.



Note: The coordinate and orientation here are initial values, not real values.

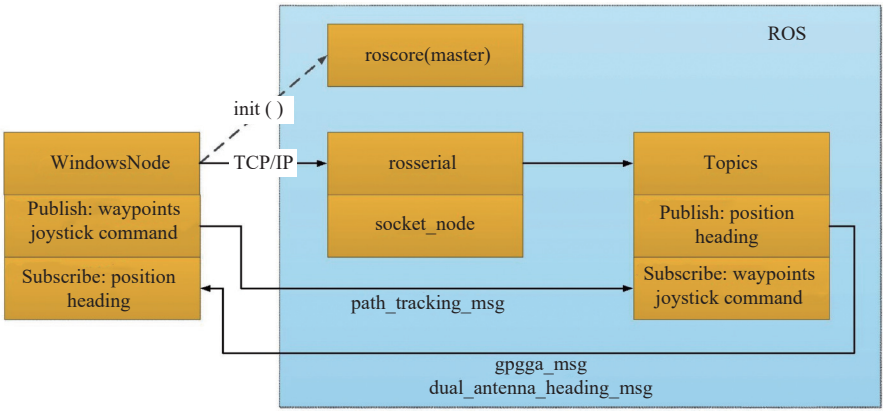
Figure 4 Remote communication and control interface of the system

2.2.3 Communication mode

The communication network was provided by an industrial wireless transceiver, whose model is MOXA AWK-1131A. With a special antenna, MOXA can establish a local area network (LAN) within 200 m. One of the advantages of this work is that the vehicle can be supervised and controlled remotely. The open-source code rosserial_windows (https://github.com/ros-drivers/rosserial) was applied to realize Windows-ROS communication and interactive operation. The communication mode is shown in Figure 5. ROS publishes topics of position (gpgga) and heading (dual_antenna_heading) messages to Windows node and subscribes waypoints and joystick commands from Windows node.

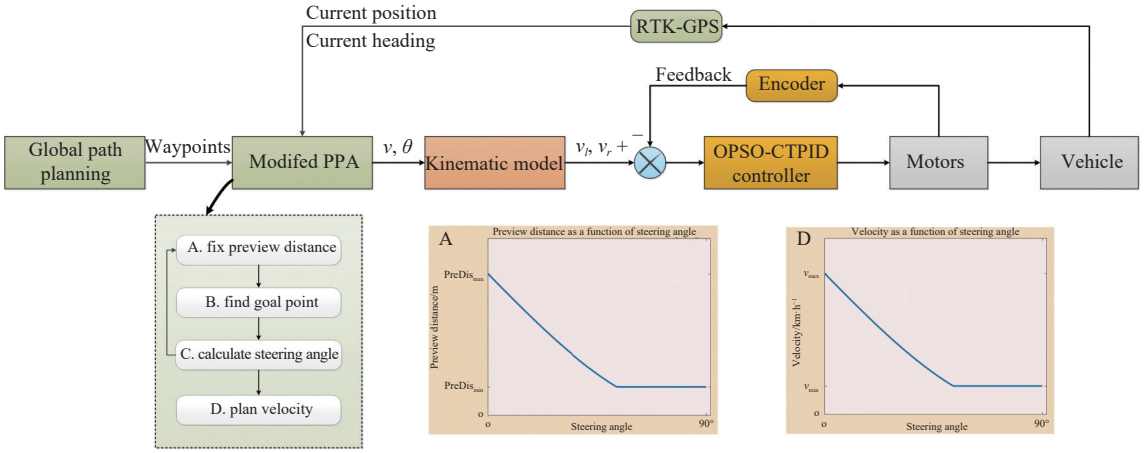
2.3 Methods

In this study, a novel trajectory tracking control method was proposed based on the farm vehicle. As is demonstrated in Figure 6, through the global path planning method, waypoints are generated. According to the current position and current heading obtained from RTK-GPS, the vehicle applies the improved pure pursuit algorithm to fix preview distance, find the current goal point, and determine steering angle and velocity. Based on the vehicle kinematic model, left and right wheel velocities are planned and then sent to the proposed OPSO-CTPID controller. The OPSO-CTPID controller combines the left and right command speed and feedback from the motors to generate control signals to the motors so that the vehicle can track the trajectory as planned. This section describes the vehicle kinematic model, improved dynamic pure pursuit algorithm, and OPSO-CTPID controller.



Note: ROS, Robot Operating System.

Figure 5 Communication mode of the whole system

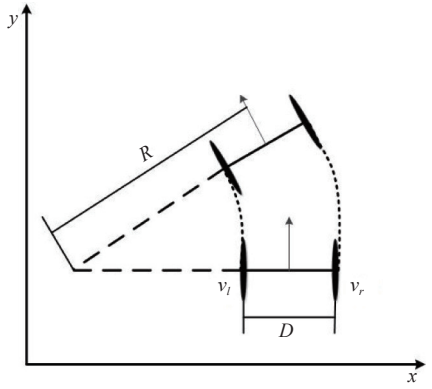


Note: RTK-GPS, Real-Time Kinematic Global Positioning System. v is the vehicle speed, m/s; θ is the steering angle; v_l is the planned left wheel velocity and v_r is the planned right wheel velocity, m/s.

Figure 6 Block diagram of vehicle trajectory tracking control system

2.3.1 Vehicle kinematic model

The vehicle is designed with two drive wheels and two passive wheels, where the front-drive wheels provide power and the passive rear wheels serve as a support and do not promote its movement. Thus, the vehicle model can be simplified as a two-wheeled differential steering model shown in Figure 7.



Note: R is the instantaneous radius of vehicle trajectory, m; D is the distance between the center of two drive wheels, m.

Figure 7 Kinematic model of the vehicle

Let v and ω be the linear velocity and angular velocity of the vehicle, respectively. D is the distance between the center of two drive wheels.

Obviously,

$$\frac{v_r}{\omega}-\frac{v_l}{\omega}=D \tag{2}$$

The kinematic model can be represented as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix}=\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{D} & -\frac{1}{D} \end{bmatrix}\begin{bmatrix} v_r \\ v_l \end{bmatrix} \tag{3}$$

Thus, the instantaneous radius of vehicle trajectory R is derived from Equation (3) as:

$$R=\frac{v}{\omega}=\frac{D(v_r+v_l)}{2(v_r-v_l)} \tag{4}$$

2.3.2 Improved pure pursuit algorithm

The pure pursuit algorithm (PPA) is extensively employed in robot trajectory tracking applications. However, PPA is predominantly applied to the Ackerman vehicle, where the algorithm controls the steering wheel deflection angle by pursuing a preview point to closely follow the planned trajectory. In our research, the pure pursuit algorithm was adapted to accommodate the kinematic model of the two-wheeled differential steering vehicle. Unlike the Ackerman-model-based pure pursuit algorithm, the modified pure pursuit algorithm directly obtains left and right demand speed based on the differential steering model. In addition, other than traditional PPA, parameters (preview distance and command speed) of the improved pure pursuit algorithm in this study change dynamically to fit the trajectory.

The performance of the controller mainly depends on the preview distance. Generally, a larger preview distance results in smoother control effects, whereas a shorter preview distance leads to more accurate control effects but may also introduce some oscillation[21]. Humans adjust their preview distance when driving a vehicle at different speeds, with shorter preview distances at lower speeds and longer preview distances at higher speeds[19]. Another experience we can refer to is that when driving in a straight line, the driving speed is high, and when the turning angle is large, the speed is relatively low. According to the kinematic model and these experiences, the pure pursuit algorithm was modified as in the following context. Table 2 lists the modified pure pursuit algorithm's parameters and descriptions.

In our study, the RTK-GPS was applied to obtain precise longitude and latitude. Firstly, latitude and longitude need to be converted to x-y coordinates in the geodetic coordinate system via Gauss projection.

1) Find the goal point. Using the path planning algorithm, a series of waypoints were obtained, and then, the distance between the vehicle body and each waypoint in order was calculated. If the distance between the vehicle body and the current waypoint is greater than the preview distance PreDis and the distance between the vehicle body and the previous waypoint is less than the preview distance PreDis, the waypoint (X_G, Y_G) was locked as the current goal point. As mentioned above, preview distance and speed are changed with the turning angle θ.

The online preview distance PreDis and vehicle demand speed v were calculated based on the following qualifications:

(1) In our study, the goal point is always ahead of the vehicle, so we have $-\frac{\pi}{2}\leqslant\theta\leqslant\frac{\pi}{2}$.

(2) PreDis>0 and v>0.

(3) While θ=0, PreDis=PreDis$_{max}$ and v=v_{max}.

(4) PreDis monotonically decreases with $|\theta|$.

(5) PreDis monotonically increases with v. If take $|\theta|$ as the indicator, v decreases with the increase of $|\theta|$.

(6) $\text{PreDis}_{\min}$ and $v_{\min}$ should be defined to avoid moving too slowly or even 0 speed.

According to the above qualifications, PreDis and v are calculated as follows:

$$\text{PreDis} = \text{Max}\{\text{PreDis}_{\max} * f, \text{PreDis}_{\min}\} \tag{5}$$

$$v = \max\{v_{\max} \times f, v_{\min}\} \tag{6}$$

where, f is an adaptor, which is defined as $f = 1 - \sin|\theta|$. Thus, the above constraints can be satisfied at the same time.

Table 2 Parameters and descriptions of the modified pure pursuit algorithm

Parameter/unit	Description
x_v/m·s^{-1}	Vehicle X coordinate in the global coordinate system
y_v/m·s^{-1}	Vehicle Y coordinate in the global coordinate system
x_G/m	Goal point X coordinate in the global coordinate system
y_G/m	Goal point Y coordinate in the global coordinate system
x/m	Goal point X coordinate in the vehicle coordinate system
y/m	Goal point Y coordinate in the vehicle coordinate system
θ_{heading}/rad	Heading angle between the orientation of the vehicle body and the X-axis in the global coordinate system
$\theta_{V\text{-}G}$/rad	Intersection angle between the position of the vehicle body point to the goal point and the X-axis in the global coordinate system
θ/rad	Angle of the goal point relative to the vehicle body in the vehicle coordinate system
v/m·s^{-1}	Average demand speed of left wheel speed and right wheel speed
$v_{l,d}$/ m·s^{-1}	Demand speed of the left wheel
$v_{r,d}$/ m·s^{-1}	Demand speed of the right wheel
R/m	Radius of curvature of the planned arc path
D/m	Distance between the center of two drive wheels
$v_{\max}$/m·s^{-1}	Set maximum speed, an important parameter of modified PPA
$v_{\min}$/m·s^{-1}	Set minimum speed
$\text{PreDis}_{\max}$/m	Set maximum preview distance, an important parameter of modified PPA
$\text{PreDis}_{\min}$/m	Set minimum preview distance
PreDis/m	Actual preview distance

2) Calculate the left and right wheel speed. Combine Figure 8. It is obvious to get the following equations:

$$(x, y) = (x_G, y_G) - (x_V, y_V) \tag{7}$$

$$\text{dis} = \sqrt{x^2 + y^2} \tag{8}$$

$$\theta_{V-G} = \arctan\frac{y_G - y_V}{x_G - x_V} \tag{9}$$

$$\theta = \theta_{\text{heading}} - \theta_{V-G} \tag{10}$$

$$R = \frac{\text{dis}}{2\sin\theta} \tag{11}$$

where, θ_{heading} can be acquired by sensors such as IMU or GPS. Here, the dual-antenna direction-finding GPS was used to get θ_{heading} value.

From Figure 8, it can be seen that the left wheel and right wheel move around the same instantaneous circle center and have the same angular speed ω. The ratio of left and right wheel speed equals the ratio of left wheel circle radius and right wheel circle radius. So, the following can be obtained:

$$\frac{v_l}{v_r} = \frac{R + D/2}{R - D/2} \tag{12}$$

Referring to the vehicle kinematic model, combining Equations (3), (11), and (12), left wheel speed and right wheel speed can be derived.

$$v_{l,d} = \frac{\text{dis} + D\cdot\sin\theta}{\text{dis}}v \tag{13}$$

$$v_{r,d} = \frac{\text{dis} - D\cdot\sin\theta}{\text{dis}}v \tag{14}$$

The variables are updated in each control period. The control period of the pure pursuit module is 200 ms because the frequency of the GPS board updating the position and orientation information is 5 Hz.

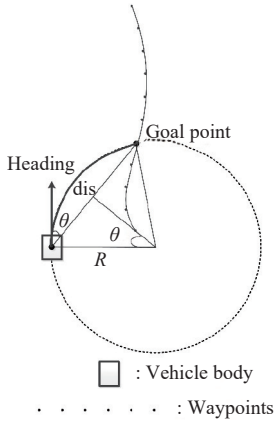


Figure 8 Modified Pure Pursuit Algorithm

2.3.3 Online Particle Swarm Optimization Continuously Tuned PID (OPSO-CTPID) controller

Proportional-integral-derivative (PID) controller is widely utilized in motor systems control. The performance of the PID controller relies on the rationality of the selected control gains. Traditionally, these gains are adjusted by artificial experiences, which may suffer from two significant drawbacks. Firstly, this tuning procedure can be challenging to execute. Secondly, the selected gains may cause overshoot and hysteresis problems, especially when the controlled system has high nonlinearity and time-varying parameters.

To address the aforementioned challenges and achieve precise wheel speed tracking control, we proposed an online optimization algorithm based on PSO. This approach dynamically searches for the optimal control gains for the PID controller, enabling the system to effectively adapt to complex agricultural environments. The block diagram of the proposed control scheme is depicted in Figure 9.

In Figure 9, the right motor-wheel dynamics can be modeled as:

$$v_r(k+1) = -\frac{d}{D^2}v_r^2(k) + \frac{1}{rm}\tau_r(k) \tag{15}$$

where, k denotes the time step; t_k satisfies t_k=$k\Delta T$ with ΔT denoting the sample time; d is the distance between the center of the mass of the vehicle and the wheel axis; r is the radius of the wheel; m is the mass of the vehicle; τ_r is the torque generated by the motor. Similarly, the dynamics of the left motor wheel can be easily deduced.

To improve the drive stability of the vehicle, $v_{r,f}$ is obtained by letting the demand speed signal of the right wheel, i.e., $v_{r,d}$ go through a proper low pass filter. Therefore, by defining the right wheel speed tracking error as $e_r(k) = v_{r,f}(k) - v_r(k)$ using the proposed OPSO-CTPID concept, the drive control signal $\tau_r(k)$ can be designed as,

$$\tau_r(k) = K_p(k)e_r(k) + K_i(k)\sum_{j=0}^{k} e_r(j) + K_d[e_r(k) - e_r(k-1)] \quad (16)$$

The proposed online PSO algorithm is composed of 6 steps; before which the position needs to be initialized for the jth particle, X_j^1:

$$X_j^1(k) = \begin{bmatrix} K_p(k) & K_i(k) & K_d(k) \end{bmatrix}^{\mathrm{T}}, \ j \in [1,n] \quad (17)$$

where, n is the number the particles.

Step 1: Predict the future speeds.

$$v_{j,r,p}^g(k+i) = -\frac{d}{D^2}[v_{j,r,p}^g(k+i-1)]^2 + \frac{1}{rm}\left\{K_p(k)e_{j,r}^g(k+i-1) + K_i(k)\sum_{j=0}^{k+i-1} e_{j,r,f}(j) + K_d\left[e_{j,r}^g(k+i-1) - e_{j,r}^g(k+i-2)\right]\right\}, \ i \in \left[1, T_p\right] \quad (18)$$

where, $g \in [1, g_{\max}]$ denotes the iteration number, and T_p denotes the prediction time length. The predicted wheel speed tracking error signal $e_{j,r}^g$ satisfies $e_{j,r}^g = v_{r,f} - v_{j,r,p}^g$. It should be mentioned that the future reference speed $v_r(k+i)$ is calculated based on the predicted wheel speed (15) and Equations (16) and (18); $v_{j,r}=v_r(k)$, $e_{j,r}^g(k-1) = v_{r,f}(k-1) - v_r(k-1)$.

Step 2: Calculate the fitness function.

$$J_{j,\mathrm{ITAE}}^g = \sum_{i=k+1}^{k+T_p} t_i |v_{r,f}(i) - v_{j,r,p}^g(i)|\Delta T \quad (19)$$

where, $v_{j,r,p}^g(i)$ is the predicted speed obtained in Step 1, and the subscript "ITAE" is an abbreviation for integral time absolute error.

Step 3: Update the local best experience $P_{j,\mathrm{best}}^g(k)$ and the global best experience $G_{\mathrm{best}}^g(k)$.

Step 4: Update the velocity and the position of the jth particle.

$$V_j^{g+1}(k) = \omega V_j^g + c_1 r_1 \left(P_{j,\mathrm{best}}^g(k) - X_j^g(k)\right) + c_2 r_2 \left(G_{\mathrm{best}}^g(k) - X_j^g(k)\right) \quad (20)$$

$$X_j^{g+1}(k) = X_j^g(k) + V_j^{g+1}(k) \quad (21)$$

where, ω is the inertial weight; c_1 and c_2 are the cognitive coefficient and social coefficient, respectively, and r_1, $r_2 \in [0,1]$ are random values.

Step 5: Stop condition. Once the maximum iteration number is reached, i.e., $l=l_{\max}$, the iteration will be stopped. Otherwise, repeat Steps 1-4.

Step 6: Calculate the gains of the PID controller.

$$X(k) = \begin{bmatrix} K_p(k) & K_i(k) & K_d(k) \end{bmatrix}^{\mathrm{T}} = \frac{1}{h}\sum_{j\in\hbar} X_j^{g_{\max}}(k) \quad (22)$$

where, $\hbar$ denotes the set of top particles, and $h = |\hbar| < n$.

Remark 1. The proposed online PSO continuously tuned PID algorithm features that can dynamically calibrate the control gains, which is different from the traditional methods[27,28]. This online gain adjustment mechanism ensures the rationality of the PID controller in the actual complex application environment.

Remark 2. In step 2 of the proposed algorithm, we construct a tracking precision-oriented fitness function by making full use of the prediction information of the controlled system state, which is quite different from the widely used fitness functions, such as $J = \int_0^\infty [w_1|e(t)|]\,\mathrm{d}t + w_3 t_s$[29], and $\mathrm{ISE} = \int_0^t e(t)^2\mathrm{d}t$[31]. The common feature of the methods in References [29] and [31] lies in that they rely on the actual output information of the controlled system, which means that the PID parameters should be kept unchanged within a certain period to enable the computation of their fitness functions. Thus, they cannot achieve dynamic adjustment of the PID parameters. What's more, compared with the bi-objective function used in [30], which is also based on the system's actual output information, our algorithm does not require some carefully chosen initial value for the PID parameters and enables parameters to be adjusted over a larger range of values.

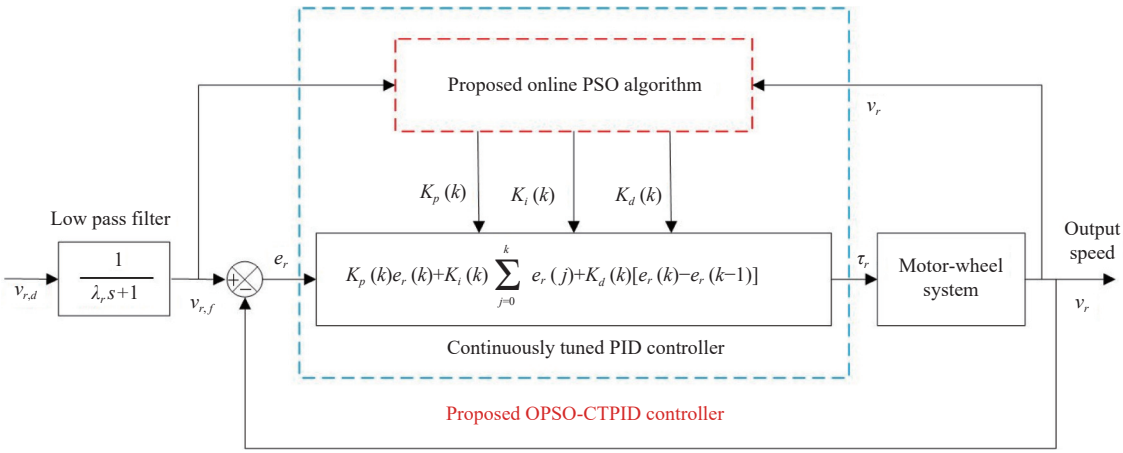


Figure 9 Block diagram of the motor-wheel control system with OPSO-CTPID controller

3 Results and discussion

3.1 Simulation results

3.1.1 Selection of adaptor f for the PPA

To choose a better adaption strategy for our proposed dynamic PPA method, several different strategies were explored to generate the adaptor f. As mentioned in Section 2.3.2, the following adaptor functions were tested that satisfy the qualifications, e.g., linear decay, cosine decay, and sine decay. To verify the superiority of dynamic PPA, the equal weight function was tested. The metric of mean absolute error (MAE) was used to evaluate the performance of these adaptors. As shown in Table 3, the decay strategies (i.e., linear decay, cosine decay, and our sine decay) can yield better results than the equal strategy (used for non-dynamic PPA). Among these decay strategies, the best way to generate the adaptor f is the proposed sine decay function.

Table 3 Studies of different adaptor strategies of the proposed dynamic pure pursuit algorithm

Adaptors	f	MAE/cm
Equal weight	1	2.89
Linear decay	$1-\frac{2\|\theta\|}{\pi}$	1.53
Cosine decay	$\cos\theta$	1.85
Sine decay (Proposed in this study)	$1-\sin\theta$	1.09

3.1.2 Simulation results comparison of four different path tracking control methods

To prove the superiority of our proposed method, the trajectory tracking methods of the Stanley algorithm with PID controller[33], pure pursuit algorithm with PID controller[14], sliding mode control (SMC)[34], and the proposed dynamic pure pursuit algorithm with OPSO-CTPID controller were simulated.

Given a desired path with the initial coordinate of (0,0), the initial state of the differential steering robot was x=−2, y=−3, yaw=0, v=0. The demand speed of the robot was set to 1.5 m/s and the control period was 0.01 s. For the pure pursuit algorithm, the preview distance is set to 3 m. The maximum preview distance in the dynamic pure pursuit algorithm is set to 3 m. Figure 10 shows the simulation results and Table 4 lists the numerical comparison.

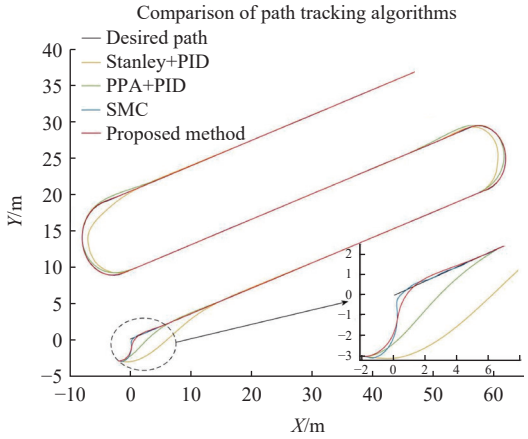


Figure 10 Simulation results of the path tracking control methods

Table 4 Numerical comparison of the path tracking control methods

Methods	MAE/cm
Stanley+PID	14.50
PPA+PID	6.32
SMC	1.61
dynamic PPA+OPSO-CTPID	1.09

From Figure 10, it is obvious that the Stanley algorithm with PID controller converges slower than other methods. Compared with the pure pursuit algorithm, the tracking error of the dynamic pure pursuit algorithm with OPSO-CTPID controller is less, especially where the curvature radius is smaller. From Table 4, it can be seen that the MAE values of the Stanley algorithm with PID controller and pure pursuit algorithm with PID controller are prominently higher than the value of our dynamic pure pursuit algorithm with OPSO-CTPID controller. Most worthy of mention is the SMC method. The SMC method achieves a fairly close MAE value with our proposed method. However, the tracking result of our proposed method is smoother than the result of the SMC method, while the SMC tracking curve is vibrative.

3.1.3 Comparison of the OPSO-CTPID controller with the traditional PID controller

To verify the effectiveness of the proposed OPSO-CTPID controller, the OPSO-CTPID was compared with the traditional PID method. Numerical simulations are carried out in MATLAB & Simulink environment. In order to obtain more realistic simulation results, we collected speed commands during real experiments. Figure 11 depicts the simulation result of wheel speed tracking error comparison between the traditional PID controller and our proposed OPSO-CTPID controller, from where it can be seen that the speed tracking error of the OPSO-CTPID controller converges very fast, and the steady-state error is satisfactory. Therefore, one can conclude that the proposed gain-tuning algorithm is essential to guarantee precise speed-tracking control. To further explore the proposed OPSO-CTPID controller and traditional PID controller, we calculate the integral absolute error (IAE) of the two controller. Left and right IAE of traditional PID controller are 10.49 and 11.20, respectively, while IAE of the proposed OPSO-CTPID controller are 2.66 and 2.89, It can be seen that the proposed method achieves a smaller IAE than the traditional PID method.

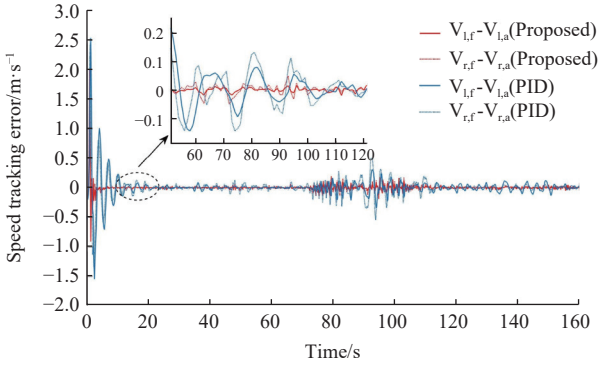


Figure 11 Simulation result of wheel speed tracking error comparison between traditional PID controller and our proposed OPSO-CTPID controller

3.2 Experiment setup

3.2.1 Experimental field

Our study location was beside Sanyou Reservoir (Liuhe District, Nanjing, China), and our research was carried out during the winter wheat seeding season. The experimental farmlands were managed by Jiangsu Aijin Group (Jiangsu Aijin Agri-chemical Co., Ltd.), and the coordinate is 32.384 773 N (latitude), 118.994 526 E (longitude). The whole field was plotted out, and the size of each grid is 80 m×25 m.

3.2.2 Seeding mechanism

The seeding and fertilizing machine is connected to our vehicle via an aluminum profile and a traction frame with a pin. The seeding and fertilizing speed is along with the speed of the vehicle due to the synchronous chain. Figure 12 is the picture that our vehicle carrying out a seeding operation in the experimental field.

Figure 12 Vehicle carrying out seeding operation

3.3 Experimental results

3.3.1 Selection of parameters for the PPA

According to the theory of our modified pure pursuit algorithm, maximum preview distance and maximum speed are two significant parameters. The simulation environment made it hard to simulate the complex real agricultural environment, so three groups of experiments were done to verify the influence of the path-tracking effect of the two parameters of maximum preview distance and maximum speed. The desired path is given as Figure 10 and the curve turning radius is around 5 m. The first set of parameters is 6 m and 5 km/h, the second set of parameters is 4 m, and 5 km/h, and the third set of parameters is 4 m and 4 km/h. In this research, the minimum preview distance and minimum speed are set as 2 m and 1.5 km/h.

The MAE and root mean square error (RMSE) of straight-line, curve-line, and overall path tracking results under three sets of parameters are summarized in Table 5. From Table 5, the first set of parameters performed worst, and the second and third sets of parameters had a similar performance while the result of the second set of parameters was a little superior to the third. It is obvious that preview distance has a clear influence on path tracking influence, and no sufficient proof shows that set speed has a great impact on path tracking results. Finally, the second set of parameters is chosen.

Table 5 MAE and RMSE of path tracking error under 3 sets of parameters

Error	MAE/cm			RMSE/cm		
	1	2	3	1	2	3
straight-line	16.30	**7.07**	8.56	13.29	**5.69**	6.77
curve-line	23.94	**10.70**	16.34	16.49	**9.59**	11.23
overall	17.65	**7.67**	10.30	14.26	**6.64**	8.62

3.3.2 Experimental results and discussion of path tracking control method

The results of the vehicle test are shown in Figures 13 and 14, where it can be seen the MAFV behaved as expected. Figures 14a and 14b demonstrate the right and left wheel speed responses respectively, from which one can observe that the wheel speed response curves under the proposed controller can track the command signal quickly and maintain good tracking accuracy in the whole control process. Figure 14c records the yaw value histories of the vehicle body and Figure 14d presents the absolute offset path tracking error during the experiment. Combining Figure 13 and Figure 14, it can be seen that the desired path is composed of three sets of straight lines and two sets of curved lines. In around 60-80 s and 140-160 s, the vehicle is turning around. Command and actual wheel speed decreased during these periods. Offset errors are relatively larger during the turning period. Maximum offset errors of straight and curved lines are 39.22 cm and 24.73 cm, respectively. Path tracking RMSE is 6.64 cm, which ensures the vehicle performs well while proceeding with agricultural operations. Our proposed method's path tracking result is superior to the SMC methods[7,34], based on which RMSE of straight-line and curve-line in Reference [34] is 7.9 cm and 13.0 cm, and the RSME in Reference [7] is 31.99 cm. One of the advantages of the pure pursuit algorithm is that the tracking curve is smooth due to the forward-looking tracking mechanism.

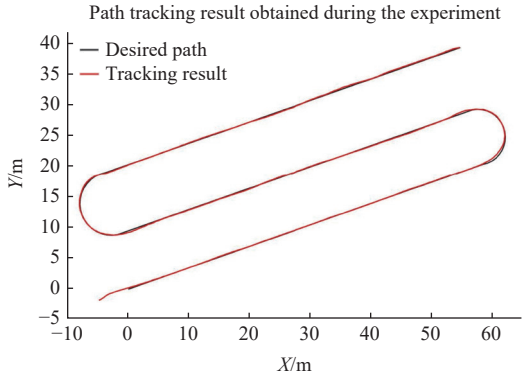


Figure 13 Path tracking result obtained during the experiment

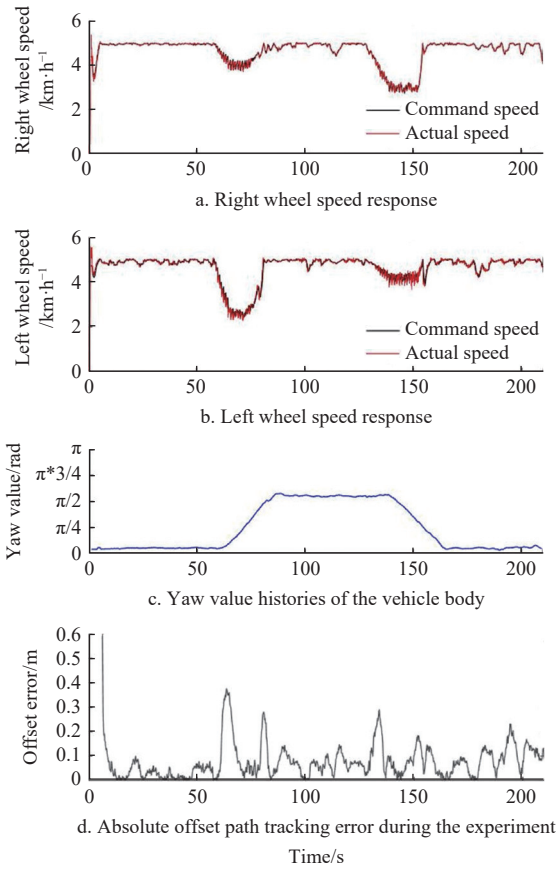


Figure 14 Details of the path tracking result obtained during the experiment.

4 Conclusions

This paper presents a feasible design for an agricultural robotic vehicle that conducts autonomous seeding operations in farmland. The vehicle is a lightweight, modular two-wheeled differential drive vehicle that can accommodate various agricultural applications. Promising autonomous seeding experiment results demonstrate the effectiveness of the hardware design, software platform, and path-tracking control method, which incorporate the proposed improved pure pursuit algorithm and OPSO-CTPID controller. Experimental results indicate that the proposed path-tracking control method exhibits high precision and robustness, making it well-suited for agricultural unstructured, complex, and uneven fields.

Of particular significance, our proposed method enables online adjustment of PID parameters, providing the agricultural vehicle with strong adaptability. This feature further enhances the practicality of our approach.

Acknowledgements

This work was financially supported in part by Jiangsu Provincial Key Research and Development Program (Grant No.

BE2017301), Jiangsu Provincial Key Research and Development Program (Grant No. BE2022363), Project of Jiangsu Modern Agricultural Machinery Equipment & Technology Demonstration and Promotion (Grant No. NJ2022-03), National Natural Science Fund of China (Grant No. 61473155), and Six Talent Peaks Project in Jiangsu Province of China (Grant No. GDZB-039).

[References]

[1] Zhai Z Y, Martínez J F, Beltran V, Martínez N L. Decision support systems for agriculture 4.0: Survey and challenges. Computers and Electronics in Agriculture, 2020; 170: 105256.

[2] Ferrández-Pastor F J, Ferrández-Pastor J M, Nieto-Hidalgo M, Mora-Pascual J, Ferrández-Pastor J. Developing ubiquitous sensor network platform using Internet of Things: Application in precision agriculture. Sensors, 2016; 16(7): 1141.

[3] Bannerjee G, Sarkar U, Das S, Ghosh I. Artificial intelligence in agriculture: A literature survey. International Journal of Scientific Research in Computer Science Applications and Management Studies, 2018; 7(3): 1–6.

[4] Bhatnagar V, Singh G, Kumar G, Gupta R. Internet of things in smart agriculture: Applications and open challenges. International Journal of Students' Research in Technology & Management, 2020; 8(1): 11–17.

[5] Gu H L, Wang H W. Innovative design of modern agricultural industry chain under 5G era. Value Engineering, 2019; 38(16): 69–71.

[6] Bell T. Automatic tractor guidance using carrier-phase differential GPS. Computers and Electronics in Agriculture, 2000; 25(1-2): 53–66.

[7] Matveev A S, Hoy M, Katupitiya J, Savkin A. Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation. Robotics and Autonomous Systems, 2013; 61(9): 973–987.

[8] Zhang S, Liu J Y, Du Y F, Zhu Z X, Mao E R, Song Z H. Method on automatic navigation control of tractor based on speed adaptation. Transactions of the CSAE, 2017; 33(23): 48–55. (in Chinese)

[9] Ruckelshausen A, Biber P, Dorna M, Gremmes H, Klose R Linz A, et al. BoniRob: An autonomous field robot platform for individual plant phenotyping. Precision Agriculture '09, 2009; pp.841-847.

[10] Scholz C, Kohlbrecher M, Ruckelshausen A, Kinski D, Mentrup D. Camera-based selective weed control application module ("Precision Spraying App") for the autonomous field robot platform BoniRob. Proceedings InInternational Conference of Agricultural Engineering, Zurich, 2014; Paper No. C0598. Available:

[11] Bakker T, van Asselt, Bontsema J, Müller J, van Straten G. Autonomous navigation using a robot platform in a sugar beet field. Biosystems Engineering, 2011; 109(4): 357–368.

[12] Bakker T, van K A, Bontsema J, Müller J, van G S. Systematic design of an autonomous platform for robotic weeding. Journal of Terramechanics, 2010; 47(2): 63–73.

[13] Bawden O, Ball D, Kulk J, Perez T, Russell R. A lightweight, modular robotic vehicle for the sustainable intensification of agriculture. In: Proceedings of the 16th Australasian Conference on Robotics and Automation Australian Robotics and Automation Association (ARAA), 2014; pp.1-9.

[14] Grimstad L, Skattum K, Solberg E, Loureiro G, From P J. Thorvald II configuration for wheat phenotyping. In: Proceedings of the IROS Workshop on Agri - Food Robotics: Learning from Industry. 2017; 4. Available: https://agrifoodroboticsworkshop.files.wordpress.com/2017/09/agrob_2017_paper_7.pdf. Accessed on [2021-06-05].

[15] Hossain T, Habibullah H, Islam R. Steering and speed control system design for autonomous vehicles by developing an optimal hybrid controller to track reference trajectory. Machines, 2022; 10(6): 420.

[16] Wang R, Li Y, Fan J H, Wang T, Chen X T. A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller. IEEE Access, 2020; 8: 166525–166540.

[17] Wallace R S, Stentz A, Thorpe C E, Moravec H, Whittaker W, Kanade T, et al. First results in robot road-following. In: IJCAI, 1985; pp.1089–1095.

[18] Cao S X, Jin Y, Trautmann T, Liu K. Design and experiments of autonomous path tracking based on dead reckoning. Applied Sciences, 2022; 13(1): 317.

[19] Ohta H, Akai N, Takeuchi E, Kato S, Edahiro M. Pure pursuit revisited: field testing of autonomous vehicles in urban areas. In: 2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), Nagoya: IEEE, 2016; pp.7–12. doi: 10.1109/CPSNA.2016.10.

[20] Park M W, Lee S W, Han W Y. Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In: 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Korea: IEEE, 2014; pp.1443–1447. doi: 10.1109/OCCAS.2014.6987787.

[21] Yu L L, Yan X X, Kuang Z X, Chen B F, Zhao Y Q. Driverless bus path tracking based on fuzzy pure pursuit control with a front axle reference. Applied Sciences, 2020; 10(1): 230.

[22] Gámez Serna C, Lombard A, Ruichek Y, Abbas-Turki A. GPS-based curve estimation for an adaptive pure pursuit algorithm. Advances in Computational Intelligence: 15th Mexican International Conference on Artificial Intelligence, MICAI 2016, Cancún: Springer International Publishing, 2017; pp.497-511. doi: 10.1007/978-3-319-62434-1_40.

[23] Wang W J, Hsu T M, Wu T S. The improved pure pursuit algorithm for autonomous driving advanced system. In: 2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA). Hiroshima: IEEE, 2017; pp.33-38. doi: 10.1109/IWCIA.2017.8203557.

[24] Bakker T, van Asselt K, Bontsema J, Müller J, van Straten G. A path following algorithm for mobile robots. Autonomous Robots, 2010; 29: 85–97.

[25] Han G N, Fu W P, Wang W, Wu Z S. The lateral tracking control for the intelligent vehicle based on adaptive PID neural network. Sensors, 2017; 17(6): 1244.

[26] Al-Mayyahi A, Wang W, Birch P. Path tracking of autonomous ground vehicle based on fractional order PID controller optimized by PSO. In: 2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any: IEEE, 2015; pp.109-114.

[27] Poultangari I, Shahnazi R, Sheikhan M. RBF neural network based PI pitch controller for a class of 5-MW wind turbines using particle swarm optimization algorithm. ISA Transactions, 2012; 51(5): 641–648.

[28] Kashyap A K, Parhi D R. Particle swarm optimization aided PID gait controller design for a humanoid robot. ISA Transactions, 2021; 114: 306–330.

[29] Wang Y B, Peng X, Wei B Z. A new particle swarm optimization based auto-tuning of PID controller. In: 2008 International Conference on Machine Learning and Cybernetics, Kunming: IEEE, 2008; pp.1818-1823. doi: 10.1109/ICMLC.2008.4620701.

[30] Salamat B, Tonello A M. Adaptive nonlinear PID control for a quadrotor UAV using particle swarm optimization. In: 2019 IEEE Aerospace Conference, Gig Sky: IEEE, 2019; pp.1-12. doi: 10.1109/AERO.2019.8741829.

[31] Rajinikanth V, Latha K. Tuning and retuning of PID controller for unstable systems using evolutionary algorithm. International Scholarly Research Notices, 2012; 2012: 693545.

[32] Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, et al. ROS: An open-source Robot Operating System. ICRA Workshop on Open Source Software, 2009; 3: 5.

[33] Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, et al. Stanley: The robot that won the DARPA grand challenge. Journal of Field Robotics, 2006; 23(9): 661–692.

[34] Tu X Y, Gai J Y, Tang L. Robust navigation control of a 4WD/4WS agricultural robotic vehicle. Computers and Electronics in Agriculture, 2019; 164: 104892.