

# Research on robot motion control and trajectory tracking based on agricultural seeding

L.L. Chen<sup>1</sup>

<sup>1</sup> *Mechanical and Electronic Engineering Department, Weihai Ocean Vocational College, Rongcheng 264300, Shandong, China*

## Abstract

With the development of science and technology, agricultural production has been gradually industrialized, and the use of robots instead of humans for seeding is one of the agricultural industrializations. This paper studied the seeding path planning and path tracking algorithms of the seeding robot, carried out experiments, and compared the improved proportion, integral, differential (PID) algorithm with the traditional PID control algorithm. The results demonstrated that both the improved and non-improved control algorithms played a good role in tracking on the straight path, but the improved control algorithm had a better tracking effect on the turning path; the displacement deviation and angle deviation of the tracking trajectory of the improved PID algorithm were reduced faster and more stable than the traditional PID algorithm; the tracking trajectory was shorter and the operation time of the robot was less under the improved PID algorithm than the traditional one.

**Keywords:** seeding robot, trajectory tracking, PID, motion control.

**Citation:** Chen LL. Research on robot motion control and trajectory tracking based on agricultural seeding. *Computer Optics* 2023; 47(1): 179-184. DOI: 10.18287/2412-6179-CO-1171.

## Introduction

Agricultural production is an important issue for any country, and China is a large agricultural country. Traditional agricultural production requires a lot of manpower, especially in the seeding process, which requires not only sufficient manpower but also the experience of the seeders to avoid yield reduction due to seeding errors as much as possible [1]. With the advance of industrialization, agricultural production has been gradually mechanized and intelligent, effectively making up for the shortage of agricultural labor [2], for example, seeding robots can perform seeding tasks in the field according to set instructions. Compared with humans, robots do not suffer from reduced efficiency and seeding accuracy due to fatigue [3]. Path planning and path trajectory tracking are important components of seeding robot intelligence. Path planning can provide efficient seeding paths for robots, and path trajectory tracking can guarantee stable robot operation along the planned path. Rijnen et al. [4] proposed a time-varying feedback gain and investigated the effectiveness of this control law for mechanical systems with one-sided position constraints. Yu et al. [5] used an iterative learning control algorithm to solve the high-precision trajectory tracking problem and verified the effectiveness of the algorithm. Klauer et al. [6] proposed an autonomous vehicle path tracking controller in an urban environment and validated the method on a closed test site simulating an urban scenario. Ostafew et al. [7] proposed a learning-based nonlinear model predictive control algorithm for autonomous mobile robots. The experimental results showed that the system started with a generic a priori vehicle model and then reduced vehicle- and trajectory-specific path tracking errors based on empirical learning. Oberherber et al. [8] proposed a method

to partition the path into segments in order to make the end effector path trajectory of an industrial robot as smooth as possible. The optimal smooth trajectory was determined for each individual segment, and the trajectories were combined. The experimental results showed that the method was effective in providing smooth time-optimal trajectories for arbitrarily long geometric paths. Zhao et al. [9] proposed an iterative learning identification method for identifying the kinematic parameters of the robot along the path in the local working area and accurately tracked the path of the industrial robot based on the identified kinematic parameters. The simulation experimental results showed that the method significantly improved the performance of path tracking. Chen et al. [10] derived a generalized explicit formulation for automatically generating translational crawling gaits in any direction, controlling joint positions, and estimating robot positioning in the walking environment. The experimental results demonstrated that the method was effective. Kapania et al. [11] used iterative learning control to improve the tracking performance of car auto driving and conducted experiments on an Audi TTS race car to verify the effectiveness of the control method. Zhao et al. [12] proposed a new discrete-time iterative learning control framework for the robust path tracking problem of a non-complete mobile robot and verified the effectiveness of the scheme through simulations. Wang et al. [13] proposed a new vision feedback-based trajectory tracking controller for non-complete mobile robots and conducted experiments on the mobile robot to verify the effectiveness and robustness of the controller.

### 1. Path planning of seeding robot

Path planning of a robot is to design an optimal path from the starting point to the end point for a robot under

various constraints, such as running speed, environmental obstacles and steering restrictions, by taking some performance index as a criterion [14]. The working environment of the seeding robot is usually relatively flat farmland, and the environmental parameters of the farmland are also required for planning the seeding path; therefore, the path planning of the seeding robot is path planning with a known environment.

The seeding robot needs to ensure that the planned path can achieve full area coverage of the sown area of the farmland considering the farmland seeding rule.

Fig. 1 shows the basic flow of planning the path that covers all the areas of farmland for a seeding robot.

Step 1. A planar coordinate system of the farmland is constructed using sensors and a positioning system. The obstacle area in the farmland is transformed into a rectangle (the horizontal side is parallel to the X-axis, and the vertical side is parallel to the Y-axis).

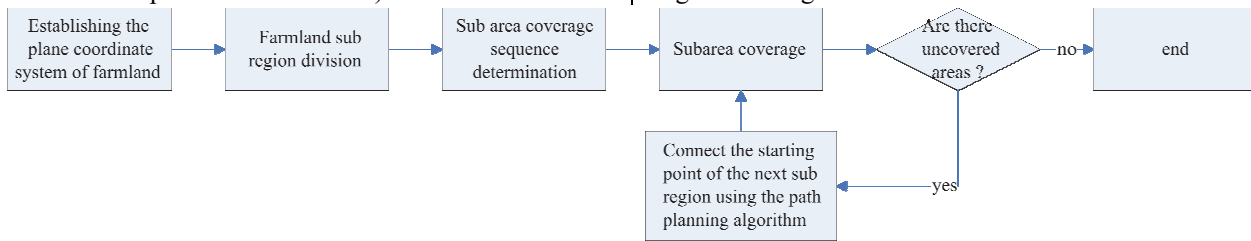


Fig. 1. Planning process for the path that covers all the farmland areas for a seeding robot

Step 4. The fold-back coverage method is used to achieve the full-area coverage of the sub-region, see the dashed paths in Fig. 2.

Step 5. Whether all sub-regions of the farmland are fully covered is determined. If they are fully covered, planning stops, and the path is output; if they are not fully covered, it goes from the end of the path of the current sub-region to the starting point of the next sub-region according to the coverage order of the sub-regions, and then step 4 is repeated.

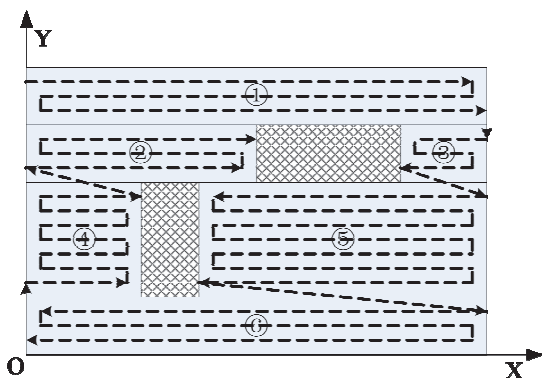


Fig. 2. Schematic diagram of farmland sub-region division and full-area coverage path

## 2. Motion control of seeding robot based on trajectory tracking

### 2.1. Trajectory tracking control law

When the path planning is completed, the seeding robot needs to advance along the designed path, but in prac-

Step 2. The sub-region of the farmland is divided. In short, the whole rectangular farmland is divided into multiple rectangles, and plural rectangles surround the obstacle area [15]. The division rules are as follows. The long side of the farmland is taken as the X-axis, and the short side as the Y-axis; a tangent line parallel to the X-axis is made by passing through the minimum and maximum vertical coordinates of every obstacle, and the tangent line ends at the farmland area boundary or obstacle. After division, all the sub-regions are numbered, as shown in Fig. 2.

Step 3. The coverage order of the sub-regions is determined. The robot covers the current sub-region first and then goes to the next sub-region, so it needs to determine the processing order of sub-regions. It starts from the initial sub-region and then visits the unvisited sub-regions adjacent to the current sub-region in turn until all sub-regions are visited. The visiting order is the sub-region coverage order.

tice, the actual forward trajectory of the robot will deviate under the influence of factors such as road surface, steering restrictions, and delayed control signals. In order to make the actual motion trajectory of the robot coincide with the planned path as much as possible, the planned path trajectory needs to be tracked, followed by the adjustment of the linear and angular velocities of the robot [16]. Fig. 3 is a schematic diagram of the trajectory tracking of the seeding robot, where  $p$  is the actual robot coordinate point of the robot,  $p_r$  is the coordinate point of the robot on the planned path (the reference coordinate point),  $\theta$  and  $\theta_r$  are the actual and reference angles. The differential equation of the pose error is:

$$\begin{cases} x'_e = -v + y_e\omega + v_r \cos \theta_e \\ y'_e = -x_e\omega + v_r \sin \theta_e \\ \theta'_e = \omega_r - \omega \end{cases}, \quad (1)$$

where  $x'_e$ ,  $y'_e$  and  $\theta'_e$  are the time differential of the robot's pose error in horizontal and vertical coordinates and angles,  $v$  and  $\omega$  are the true linear and angular velocities,  $v_r$  and  $\omega_r$  are the reference linear and angular velocities, and  $x_e$ ,  $y_e$  and  $\theta_e$  are the pose errors of the robot. Equation (1) converts the original pose tracking problem into a stabilization problem for the error system [17]. Parameters on the left side of the equation are target parameters; on the right side of the equation, only  $v$  and  $\omega$  are variables, and the other parameters have been determined by the planned path. In this paper, the stabilization problem of the error system is to solve variables  $v$  and  $\omega$  to make equation (1) asymptotically stable.

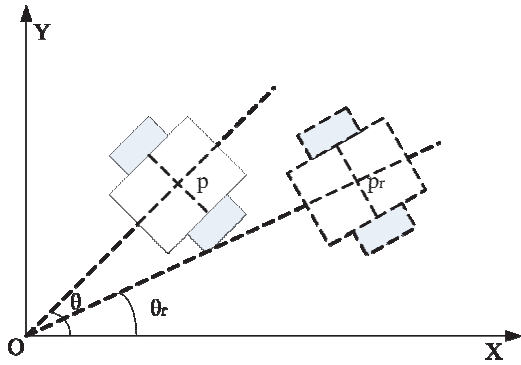


Fig. 3. Schematic diagram of trajectory tracking.

The backstepping method is adopted. The intermediate virtual control variable is introduced to construct the Liapunov function [18] to solve  $v$  and  $\omega$ . The corresponding equations are:

$$\begin{cases} \omega = \begin{cases} \omega_r + \varepsilon_{\omega} \text{sign}(\bar{\theta}_e) & |k_1 \bar{\theta}_e| \geq \varepsilon_{\omega} \\ \omega_r + k_1 \bar{\theta}_e & |k_1 \bar{\theta}_e| < \varepsilon_{\omega} \end{cases} \\ v = \begin{cases} v_r \cos \theta_e - \varepsilon_{v1} & -k_2 x_e \geq \varepsilon_{v1} \\ v_r \cos \theta_e + k_2 x_e & \varepsilon_{v2} < -k_2 x_e < \varepsilon_{v1} \\ v_r \cos \theta_e - \varepsilon_{v2} & -k_2 x_e \leq \varepsilon_{v2} \end{cases}, \\ \bar{\theta}_e = k_3 \theta_e + \frac{y_e}{\sqrt{1+x_e^2+y_e^2}} \\ \varepsilon_{\omega} = \omega_{\max} - \omega_{r\max} \\ \varepsilon_{v1} = v_r \cos \theta_e + v_{\max} \\ \varepsilon_{v2} = v_r \cos \theta_e - v_{\max} \end{cases} \quad (2)$$

where  $\bar{\theta}_e$  is the intermediate variable introduced,  $k_1$ ,  $k_2$  and  $k_3$  are adjustable control parameters,  $\varepsilon_{\omega}$ ,  $\varepsilon_{v1}$  and  $\varepsilon_{v2}$  are parameters limiting the difference between actual linear and angular velocities of the robot and the reference linear and angular velocities.

### 2.2. PID-based robot control algorithm

After  $v$  and  $\omega$  are obtained through calculation with Equation (2), the motion state of the robot is adjusted according to the gap between the current motion data of the robot and the motion data given by the control law. Fig. 3 shows the basic flow of the robot PID control algorithm [19] combined with the trajectory tracking control law.

Step 1. The robot's motion data in the plane coordinates of the farmland are collected using sensors, including the position coordinates, moving linear velocity and angular velocity of the robot.

Step 2. The reference trajectory of the robot when moving has been obtained during the path planning for the full-area coverage. Afterward, based on the actual robot motion data collected by the sensors and the reference trajectory, the linear and angular velocities required for the robot to achieve trajectory tracking are calculated using Equation (2).

Step 3. The linear and angular velocities in step 1 and step 2 are compared.

Step 4. The control equation for the difference is:

$$\Delta u(k) = k_p [e(k) - e(k-1)]e(k-1) + \frac{k_p T}{T_i} e(k) + \frac{k_p T_D}{T} [e(k) - 2e(k-1) + e(k-2)], \quad (3)$$

where  $\Delta u(k)$  is the incremental control signal for the current adjustment,  $e(k)$ ,  $e(k-1)$ ,  $e(k-2)$  are deviation signals for the current, previous and previous two adjustments, respectively,  $k_p$  is the proportional coefficient,  $T_i$  is the integral coefficient, and  $T_D$  is the differential coefficient [20].

Step 5. The robot adjusts its motion state according to  $\Delta u(k)$ ; then, it returns to step 1 until it traverses the planned path.

The collection, calculation, and adjustment of the motion state data in the above process take some time, and the adjustment of the motion state is inevitably lagged by multiple factors. In addition, the three control coefficients in the conventional PID control algorithm are often fixed, but the deviation between the actual motion state and the target motion state is dynamically changing in the process of adjusting the robot motion state, and the changing trend is not fixed, so the fixed control coefficients in the algorithm are prone to overshooting or insufficient adjustment in the face of the irregular change trend [21]. In order to improve the control effect of the incremental PID control algorithm, this paper divides control coefficients according to the characteristics of deviation changes to provide different control coefficients for different change characteristics to achieve targeted control degree changes. The formula of the improved incremental PID is:

$$\begin{cases} \Delta u(k) = \frac{k_{p1} T_{D1}}{T} [e(k) - 2e(k-1) + e(k-2)] & \{e(k) > 0 \cap e'(k) > 0 \cap e''(k) > 0\} \cup \{e(k) < 0 \cap e'(k) < 0 \cap e''(k) < 0\} \\ \Delta u(k) = k_{p2} [e(k) - e(k-1)] & \{e(k) > 0 \cap e'(k) > 0 \cap e''(k) < 0\} \cup \{e(k) < 0 \cap e'(k) < 0 \cap e''(k) > 0\} \\ \Delta u(k) = k_{p3} [e(k) - e(k-1) + \frac{T}{T_{i3}} e(k)] & \{e(k) > 0 \cap e'(k) < 0 \cap e''(k) > 0\} \cup \{e(k) < 0 \cap e'(k) > 0 \cap e''(k) < 0\}, \\ \Delta u(k) = \frac{k_{p4} T_{D4}}{T} [e(k) - 2e(k-1) + e(k-2)] & \{e(k) > 0 \cap e'(k) < 0 \cap e''(k) < 0\} \cup \{e(k) < 0 \cap e'(k) > 0 \cap e''(k) > 0\} \end{cases} \quad (4)$$

where  $e(k)$ ,  $e'(k)$  and  $e''(k)$  are the deviation, deviation change velocity and deviation change acceleration, respectively.

The improved control flow is nearly the same as Fig. 4, but the difference is that it will judge  $e(k)$ ,  $e'(k)$  and  $e''(k)$  when performing PID control on the devia-

tion before selecting the corresponding incremental PID formula.

### 3. Experiment

#### 3.1. Experimental setup

The specifications of the experimental farmland and the division of the sub-regions are shown in Fig. 5, where the well grid area is the obstacle area. The total length and width of the farmland were 24 m and 15 m, respectively. The specifications of sub-region ① and ② were both 12 m × 3 m. The specification of sub-region ③ was 6 m × 4.5 m. The specification of sub-region ④ was 6 m × 6 m. The specification of sub-region ⑤ was 15 m × 6 m. The specification of sub-region ⑥ was 24 m × 3 m.

Fig. 6 shows the appearance of the seeding robot. The robot had four wheels, was driven by the rear wheels, and was steered by the front wheels. The basic specification parameters of the seeding robot are as follows. The mass was 10 kg. The maximum linear speed was 3 m/s. The maximum angular speed was 2 rad/s. The length was 0.4 m. The width was 0.3 m. In addition, to strengthen the stability of the robot when moving on the uneven road surface, the front and rear wheels were equipped with shock absorbers. The robot was driven by the direct-current servo motor. The speed was controlled by voltage. The torque was controlled by current. The optical encoder, fiber optic gyroscope, inclination sensor and potentiometer were used as internal sensors for self-positioning.

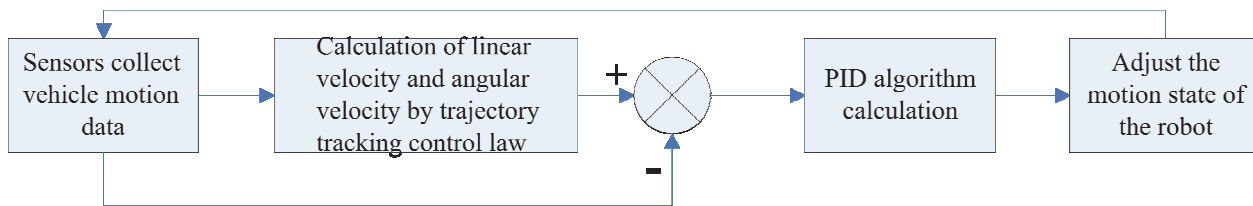


Fig. 4. Robot PID control algorithm flow combined with the trajectory tracking control law

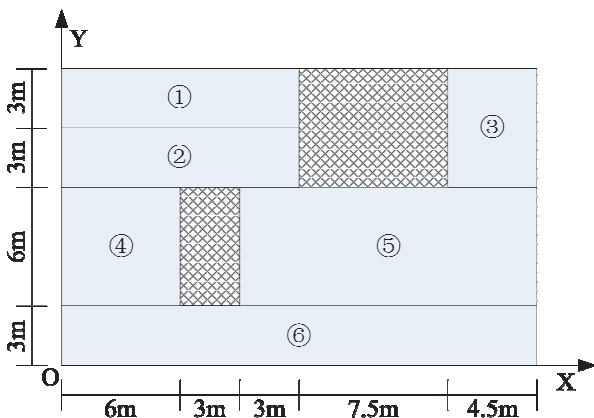


Fig. 5. Schematic diagram of the sub-regional division of the experimental farmland



Fig. 6. The seeding robot

#### 3.2. Algorithm parameters

For full-area coverage path planning, the fold-back coverage method was used in the sub-region. The path from the end of the current sub-region path to the start of

the next sub-region path was planned using the genetic algorithm.

After the full-area coverage path planning, the seeding robot used the improved PID control algorithm combined with the trajectory tracking control law to track the planned path. In the trajectory control law,  $k_1=8$ ,  $k_3=4$  and  $\varepsilon_0=2$ . In the improved PID control algorithm,  $T=1$ ,  $k_{P1}=k_{P2}=k_{P4}=5$ ,  $T_{D1}=T_{D4}=1$ ,  $k_{P3}=3$  and  $T_{I3}=0.5$ .

The improved PID control algorithm was compared with the traditional PID control algorithm. The desired signal required by the traditional PID control algorithm was given by the trajectory tracking control law, and the parameters of the trajectory tracking control law remained unchanged. In the traditional PID control algorithm,  $k_p=5$ ,  $T_I=0.5$ ,  $T_D=1$  and  $T=1$ .

#### 3.3. Experimental results

Since the overall area of the experimental farmland was large and the moving path of the seeding robot covered the whole area, only the planned path in sub-region ④ and the path tracking trajectories under the improved and non-improved PID control algorithms are shown in Fig. 7. It was seen from Fig. 7 that the seeding robot used a fold-back path to achieve full-area coverage in the sub-region and adopted a circular steering path in order to ensure the stability of the robot when turning back. The coordinate of the starting point of the fold-back path in sub-region ④ was (0.0 m, 8.85 m), and the coordinate of the end point was (0.0 m, 3.15 m). The comparison of the path tracking trajectories under the improved and non-improved PID control algorithms in Fig. 7 showed that the trajectories under both control algorithms nearly overlapped with the planned path when tracking the straight

path; the trajectory under the traditional PID control algorithm had very obvious deviations when facing the steering path, while the trajectory under the improved PID control algorithm nearly fit the planned steering path. The comparison of the trajectories of the two control algorithms showed that the robot trajectory obtained by the improved PID control algorithm combined with the trajectory tracking control law was closer to the planned path.

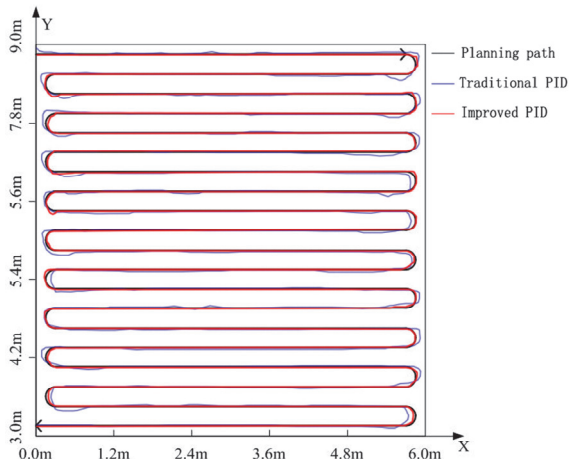


Fig. 7. The planned path in sub-region ④ and tracking trajectories under improved and non-improved PID control algorithms

Due to the large area of the experimental farmland, the path that covered the whole area was long, so the time spent by the seeding robot was long. Therefore, only the tracking path at the first 15 s was selected to compare the displacement deviation and angle deviation of the tracking trajectory of the improved and non-improved PID control algorithms, as shown in Fig. 8. It was seen from Fig. 8 that under both control algorithms, the tracking trajectory of the robot effectively converged to the planned path and eventually stabilized. It was concluded that the PID control algorithm combined with the trajectory control law could achieve the robot's trajectory tracking. In addition, the displacement deviation and angle deviation under the improved PID control algorithm converged to stability faster; when the convergence was roughly stabilized at 0, the traditional PID control algorithm still had relatively obvious deviation.

Tab. 1 shows the length of the tracking trajectory and the running time of the seeding robot under the improved and non-improved PID control algorithms. Although the planned path tracked by the robot was the same under both control algorithms, the performance of the robot in tracking the path through the control algorithm was different, and the tracking trajectory did not coincide with the planned path completely; therefore, the length of the tracking trajectory was different, which in turn affected the running time of the robot. Under the traditional PID control algorithm, the length of the tracking trajectory was 1010 m, and the running time was 31 min; under the improved PID control algorithm, the length of the track-

ing trajectory was 990, and the running time was 27 min. It was found that the robot had a shorter tracking trajectory and less running time under the improved PID control algorithm. The reason for the above result is that the improved PID control algorithm had better tracking performance when facing the steering path, and the tracking trajectory of the traditional PID control algorithm deviated a lot when steering, which made the robot take more time and distance to return to the planned path.

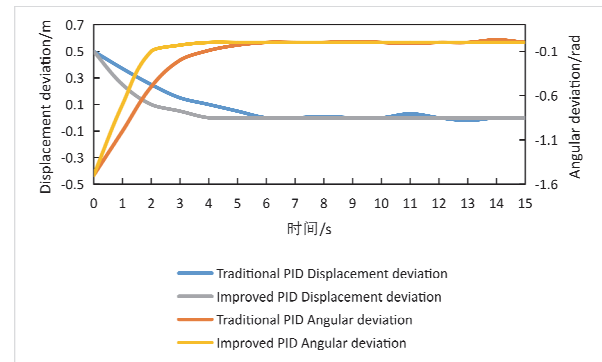


Fig. 8. Displacement deviation and angle deviation of tracking trajectories of the non-improved and improved PID control algorithms

Tab. 1. Length of tracking trajectory and robot running time under improved and non-improved PID control algorithms

	Length of the tracking path /m	Running time of the robot /min
The traditional PID control algorithm	1010	31
The improved PID control algorithm	990	27

**Conclusion**

This paper introduced the full-area path planning of the seeding robot and the control algorithm for path tracking, performed experiments on the improved PID control algorithm combined with the trajectory control law using the seeding robot, and compared it with the traditional PID control algorithm. The results are as follows. When tracking the planned folding path, both control algorithms had a good tracking effect for the straight path; however, when tracking the steering path, the tracking trajectory of the traditional PID algorithm had apparent deviations, but the tracking trajectory of the improved PID algorithm nearly overlapped with the planned path. The tracking trajectory of the improved PID control algorithm converged the deviation between the trajectory and the planned path to 0 faster and remained stable. Under the traditional PID control algorithm, the tracking trajectory had a length of 1010 m, and the robot ran for 31 min; under the improved PID control algorithm, the tracking trajectory had a length of 990 m, and the robot ran for 27 min.

**References**

[1] Scaglia G, Serrano E, Rosales A, Albertos P. Linear interpolation based controller design for trajectory tracking

- under uncertainties: Application to mobile robots. *Control Eng Pract* 2015; 45: 123-132.
- [2] Guo J, Li K, Luo Y. Coordinated control of autonomous four wheel drive electric vehicles for platooning and trajectory tracking using a hierarchical architecture. *J Dyn Syst Meas Control* 2015; 137(10): 1-18.
- [3] Li Z, Deng J, Lu R, Xu Y, Bai J, Su C. Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach. *IEEE Trans Syst Man Cybern Syst* 2017; 46(6): 740-749.
- [4] Rijnen M, Saccon A, Nijmeijer H. On optimal trajectory tracking for mechanical systems with unilateral constraints. 2015 54th IEEE Conf on Decision and Control (CDC) 2015: 2561-2566.
- [5] Yu C, Chen X. Trajectory tracking of wheeled mobile robot by adopting iterative learning control with predictive, current, and past learning items. *Robotica* 2015; 33(7): 1393-1414.
- [6] Klauer C, Schwabe M, Mobalegh H. Path tracking control for urban autonomous driving. *IFAC-PapersOnLine* 2020; 53(2): 15705-15712.
- [7] Ostafew CJ, Collier J, Schoellig AP, Barfoot TD. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *J Field Robot* 2016; 33(1): 133-152.
- [8] Oberherber M, Gattringer H, A Müller. Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking. *Mech Sci* 2015; 6(2): 245-254.
- [9] Zhao YM, Lin Y, Xi F, Guo S. Calibration-based iterative learning control for path tracking of industrial robots. *IEEE Trans Ind Electron* 2015; 62(5): 2921-2929.
- [10] Chen X, Watanabe K, Kiguchi K, Izumi K. Translational crawl and path tracking of a quadruped robot. *J Robot Syst* 2016; 19(12): 569-584.
- [11] Kapania NR, Gerdes JC. Path tracking of highly dynamic autonomous vehicle trajectories via iterative learning control. *American Control Conf* 2015: 2753-2758.
- [12] Zhao Y, Zhou F, Li Y, Wang Y. A novel iterative learning path-tracking control for nonholonomic mobile robots against initial shifts. *Int J Adv Robot Syst* 2017; 14(3). DOI: 10.1177/1729881417710634.
- [13] Wang K, Liu Y, Li L. Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement. *IEEE Trans Robot* 2017; 30(4):1026-1035.
- [14] Wang N, Qian C, Sun J, Liu Y. Adaptive robust finite-time trajectory tracking control of fully actuated marine surface vehicles. *IEEE Trans Control Syst Technol* 2016; 24(4): 1454-1462.
- [15] Zhang L, Chen W, Liu J, Wen C. A robust adaptive iterative learning control for trajectory tracking of permanent-magnet spherical actuator. *IEEE Trans Ind Electron* 2015; 63(1): 291-301.
- [16] Aguilar-Avelar C, Moreno-Valenzuela J. A composite controller for trajectory tracking applied to the Furuta pendulum. *ISA Trans* 2015; 57: 286-294.
- [17] Jian X, Man W, Lei Q. Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. *Ocean Eng* 2015; 105: 54-63.
- [18] Kayacan E, Kayacan E, Ramon H, Saeys W. Towards agrobots: Identification of the yaw dynamics and trajectory tracking of an autonomous tractor. *Comput Electron Agric* 2015; 115: 78-87.
- [19] Zhang CL, Li JM. Adaptive iterative learning control of non-uniform trajectory tracking for strict feedback nonlinear time-varying systems with unknown control direction. *Appl Math Model* 2015; 39(10-11): 2942-2950.
- [20] Radac MB, Precup RE, Petriu EM. Model-free primitive-based iterative learning control approach to trajectory tracking of MIMO systems with experimental validation. *IEEE Trans Neural Netw Learn Syst* 2017; 26(11): 2925-2938.
- [21] Naldi R, Furci M, Sanfelice RG, Marconi L. Robust global trajectory tracking for underactuated VTOL aerial vehicles using inner-outer loop control paradigms. *IEEE Trans Automat Contr* 2016; 62(1): 97-112.

#### Author's information

**Linlin Chen** (b. 1987) He received the master's degree from Zhejiang University of Technology in January, 2013, P.R. China. Now, he is a lecturer. His research interests include mechanical engineering, robot motion control and machinevision and application. E-mail: [cen2rt@163.com](mailto:cen2rt@163.com).

*Received June 2, 2022. The final version – September 13, 2022.*