

Article

Design Hybrid Iterative Learning Controller for Directly Driving the Wheels of Mobile Platform against Uncertain Parameters and Initial Errors

Lijun Qiao ¹, Luo Xiao ^{2,*}, Qingsheng Luo ¹, Minghao Li ³ and Jianfeng Jiang ¹

¹ School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China; Qiao_Lijun_123@163.com (L.Q.); luqsh@bit.edu.cn (Q.L.); xiangyang_ersheng@163.com (J.J.)

² School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

³ School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; swd_16@163.com

* Correspondence: luox@bit.edu.cn

Abstract: In this paper, we develop a hybrid iterative learning controller (HILC) for a non-holonomic wheeled mobile platform to achieve trajectory tracking with actual complex constraints, such as physical constraints, uncertain parameters, and initial errors. Unlike the traditional iterative learning controller (ILC), the control variable selects the rotation speed of two driving wheels instead of the forward speed and the rotation speed. The hybrid controller considers the physical constraints of the robot's motors and can effectively handle the uncertain parameters and initial errors of the system. Without the initial errors, the hybrid controller can improve the convergence speed for trajectory tracking by adding other types of error signals; otherwise, the hybrid controller achieves trajectory tracking by designing a signal compensation for the initial errors. Then, the effectiveness of the proposed hybrid controller is proven by the relationship between the input, output, and status signals. Finally, the simulations demonstrate that the proposed hybrid iterative learning controller effectively tracked various trajectories by directly controlling the two driving wheels under various constraints. Furthermore, the results show that the controller did not significantly depend on the system's structural parameters.

Keywords: non-holonomic wheeled mobile platform; hybrid iterative learning controller; trajectory tracking; uncertain parameters; initial errors

check for
updates

Citation: Qiao, L.; Xiao, L.; Luo, Q.; Li, M.; Jiang, J. Design Hybrid Iterative Learning Controller for Directly Driving the Wheels of Mobile Platform against Uncertain Parameters and Initial Errors. *Appl. Sci.* **2021**, *11*, 8181. <https://doi.org/10.3390/app11178181>

Academic Editor: Giovanni Boschetti

Received: 10 August 2021

Accepted: 30 August 2021

Published: 3 September 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the simple structure design and strong movement ability, mobile platforms have been extensively applied in various fields, including agricultural production [1], environmental detection [2], and industrial production [3]. Due to the differences in the number of motors and the structure shape, differential mobile robots controlled by two driving wheels have become the mainstream mobile robot. In the movement of the mobile platform, there are two actual input control variables: the speeds of the driving wheels on the left side and the right side [4].

However, the actual positioning of the mobile platform requires three parameters: the position of the xy plane and the rotation angle of the z axis [5]. Therefore, a differential mobile robot with two driving wheels is a typical non-holonomic constraint system [6]. In solving various problems associated with this mobile robot, trajectory tracking is an important research problem with a high application value in many fields.

With mobile robots, uncertain parameters are unavoidable, which are caused by many factors, such as physical error in the machining [7], external disturbances in the environment [8], equipment wear during working [9], and disturbances in the control signals [10]. To deal with these uncertainties parameters, various controllers have been proposed, such as the back-stepping controller [11], model predictive controller [12,13], sliding

mode controller [14,15], fuzzy controller [16,17], adaptive tracking controller [18–20], and the neural network controller [21–23].

Compared with the kinematic controller, the dynamic controller has more uncertain parameters, a more complex construction, and worse adaptability to different environments. The kinematic controller also needs to be designed before the dynamic controller can be designed for realizing trajectory tracking. However, the design of most kinematic controllers is based on the structural parameters of the mobile robot. Therefore, designing a simple kinematic controller and reducing the disturbance due to uncertain parameters remains a significant research challenge.

In various kinematics controllers, an iterative learning controller can ignore the robot's structure parameters and realizes high-precision trajectory tracking by continuously adjusting the input control signals and optimizing the output result. Furthermore, iterative learning controllers mainly focus on specific task requirements and movement characteristics instead of the basic parameters of the system.

As an advanced control theory, iterative learning controllers have been widely applied to various robots, such as a parallel Delta robot [24], manipulator robot [25], mobile platform, and self-balancing mobile robot [26]. Additionally, iterative learning controllers have demonstrated effectiveness in solving many research problems, such as trajectory tracking problems [27,28], and trajectory planning problems [29,30]. Currently, the typical iterative learning controllers mainly include the P-type, PD-type, and PID-type [31,32]. In addition, iterative learning controllers have widely used in solving the trajectory tracking problems of mobile platforms.

The authors in [33] first applied the P-type iterative learning controller to realize the sine-type trajectory tracking of a mobile platform. Another article [34] revised the design of P-type iterative control into the typical PD-type control design for realizing rectangular trajectory tracking. Chong [35] proposed different error signals to optimize the P-type iterative learning controller for completing various types of trajectory tracking control with mobile platforms. Then, researchers [36] added the PID coefficient in the iterative learning controller to improve the effectiveness of arc-shaped trajectory tracking. At the same time, researchers [37] changed the forgetting factor to reduce the correction of the error parameters at the initial moment of each iteration process.

Due to the many limitations not considered in the actual modeling and controller design, these iterative learning controllers produced in the above studies are not fully suitable for trajectory tracking with mobile platforms. These shortcomings can be summarized as follows:

- (1) The discrete kinematics model does not conform to the actual physical model of the mobile robot. The gravity center may not wholly coincide with the midpoint of the driving wheels in the actual mobile robot. Therefore, the previous modeling methods and iterative learning controllers based on the gravity center are different from the actual model and control of the mobile platform.
- (2) The output variables of the iterative controller cannot directly control the motion of the mobile platform. The actual control variable of the mobile platform is the rotation speed of the driving wheels, whereas the output variables in these literatures are the forward speed and rotation speed of the mobile robot. The output variables produced by the above controllers need to be converted into the rotation speed of the driving wheels.
- (3) The solution speed may not meet the actual physical constraints of the motor. Since the selected control parameters are the mobile robot's forward speed and rotation speed in these literatures, they cannot be effectively limited by the driving wheels. Therefore, the solution in the above-described studies cannot be converted into the actual rotation speed of two wheels considering the actual physical constraints.
- (4) The above-mentioned studies did not analyze the influence of the various parameters of the iterative controller on the trajectory tracking task of the mobile robot. These researchers only analyzed and verified the effectiveness of the designed controller,

but did not study the influence of the controller parameters, such as the parameter uncertainty, speed limitation, structural parameter uncertainty, and initial error. It is necessary to conduct in-depth research so that the iterative controller can be more widely used in trajectory tracking.

To overcome the shortcomings in earlier studies, we established a new discrete kinematic model of a mobile platform to directly choose the speed of the driving wheels as the control variable for trajectory tracking, instead of controlling the forward speed and angular rotation speed. Furthermore, we explicitly analyzed multiple constraints in actual trajectory tracking tasks, and established a new type of iterative learning controller in the presence or absence of the initial error. With the initial error, a corresponding compensatory mechanism was designed for the learning controller to correct the error at the start time; otherwise, the other error signals were added in the iterative learning law to improve the effectiveness of the controller.

According to the theoretical analysis, the proposed controller has convergence and stability under certain conditions. In the simulation experiment, the effectiveness of the proposed controller was verified using various typical trajectories, such as a circular trajectory, a combination trajectory of arc and straight, a cardioid-like trajectory, and a spiral-like trajectory. Then, in-depth research on the iterative learning controller was specifically conducted by comparing the performance of the circular trajectory tracking task with different control parameters, which mainly included the selection of the control coefficient, the choice of various error signals, and the impact of speed constraints.

This paper is organized as follows: Section 2 describes the specific structure and basic assumptions of a non-holonomic wheeled mobile platform, establishes the discrete kinematics of the differential drive mobile platform, and analyzes the control constraints of the drive wheels. In Section 3, we explicitly design the iterative learning controller and prove the effectiveness and robustness of the new controller through theoretical analysis. In Section 4, numerical simulations illustrate the effectiveness of the new controller's trajectory tracking with a mobile platform by controlling the driving wheels, and we describe our in-depth research on the new iterative learning controller. Finally, Section 5 summarizes the content of the article and explains future work.

2. Problem Formulation

In this section, a discrete kinematic model of a non-holonomic wheeled mobile robot is presented, which was used in designing a new type of iterative learning controller for completing trajectory tracking tasks. The parametric uncertainty and other practical constraints are formulated at the end of this section.

2.1. System Description

Currently, most mobile platforms use different wheels as driving and supporting mechanisms. However, mobile platforms with many driving wheels require more complicated synchronization control, are more expensive, and have fewer applications in daily life. Therefore, a typical mobile robot only uses the front wheels or the rear wheels as the driving wheels. The mobile robot is controlled by the rotation speed of the driving wheels. In general, the mobile platform's rotational movement and forward movement can be precisely controlled by the driving wheels' rotation speed. However, the basic movement of the mobile robot needs to avoid forward sliding and lateral slipping during movement.

Some coordinate frames need to be established in the working environment and on the mobile platform for positioning and trajectory tracking. These frames are as follows:

- (1) The world frame O_W : Establish the system's global coordinate system at a fixed position in the inertial coordinate system; it is used to describe the position and posture of the mobile robot.
- (2) The driving-fixed frame O_D : Select the midpoint of the driving wheels as the origin of this local coordinate system, the x -axis direction is the same as the forward movement direction, the z -axis direction is the vertical direction of the movement plane,

and the corresponding y -axis direction is established according to the right-hand rule direction.

- (3) The gravity-fixed frame O_C : The origin of the gravity-fixed frame is established at the gravity center of the mobile platform, the x -axis direction is the same as the forward direction, the z -axis direction is the vertical direction of the movement plane, and the corresponding y -axis direction is established according to the right-hand rule direction.

During movement, some structural parameters must be determined by measurement instead of using the values provided by the manufacturer to solve the positioning problem and achieve trajectory tracking with a mobile platform including the following:

- (1) Basic structure parameters: The length of the mobile platform is L_c , the width of the mobile platform is B_c , the radius of the wheels of the mobile platform is R_c , and the distance from the midpoint of the driving wheels to the gravity center of the mobile platform is (D_{cx}, D_{cy}, D_{cz}) .
- (2) Vehicle positioning parameters: The initial position and initial posture of the midpoint of the driving wheels are $(P_{dx0}, P_{dy0}, Q_{dz0})$.

In the trajectory tracking task, the structural parameters are fixed values, and the position and posture of the mobile platform are constantly changing. However, the distance from the midpoint of the driving wheels to the gravity center of the mobile platform is also a fixed value. The specific description of the structural parameters is shown in Figure 1.

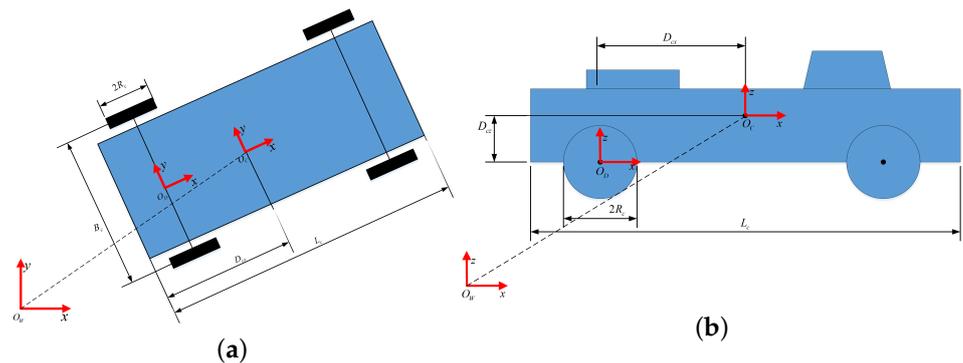


Figure 1. The basic parameters of the mobile platform. (a) The top view and corresponding parameters. (b) The side view and corresponding parameters.

2.2. Kinematic Model

2.2.1. Discrete kinematics Model of Mobile Platform

(1) Ideal kinematics discrete model

For an ideal mobile platform, the midpoint of the driving wheels coincides with the gravity center of the mobile platform on the xy plane. Assuming the robot is moving on the xy plane, the linear velocity and angular velocity can be expressed as follows:

$$\begin{cases} \dot{P}_{xc}(t) = V_c(t) \cos(q_{zc}(t)) \\ \dot{P}_{yc}(t) = V_c(t) \sin(q_{zc}(t)) \\ \dot{\theta}_{zc}(t) = W_c(t) \end{cases} \Leftrightarrow \begin{bmatrix} \dot{P}_{xc}(t) \\ \dot{P}_{yc}(t) \\ \dot{\theta}_{zc}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta_{zc}(t)) & 0 \\ \sin(\theta_{zc}(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_c(t) \\ W_c(t) \end{bmatrix} \quad (1)$$

where $\dot{P}_{xc}, \dot{P}_{yc}$ is the linear velocity of the mobile platform along the x axis or the y axis of the world frame, $\dot{\theta}_{zc}$ is the angular velocity of rotation around the z axis of the world frame, V_c is the linear velocity of the mobile platform, and W_c is the rotation angle velocity of the mobile platform.

Considering control by two driving wheels, the linear velocity and angular velocity of the system can be directly described by the rotation speed of two driving wheels, which is explicitly expressed as follows:

$$\begin{cases} V_c(t) = \frac{R_c(W_L(t)+W_R(t))}{2} \\ W_c(t) = \frac{R_c(W_L(t)-W_R(t))}{2B_c} \end{cases} \Leftrightarrow \begin{bmatrix} V_c(t) \\ W_c(t) \end{bmatrix} = \frac{R_c}{2} \begin{bmatrix} 1 & 1 \\ 1/B_c & -1/B_c \end{bmatrix} \begin{bmatrix} W_L(t) \\ W_R(t) \end{bmatrix} \quad (2)$$

where W_L and W_R are the actual rotational speed of the left and right driving wheels, respectively; R_c is the radius of two driving wheels; and B_c is the distance between two driving wheels of the mobile platform.

To control the mobile platform, we performed the kinematic modeling by combining Formulas (1) and (2) for directly driving the wheels, which can be expressed as follows:

$$\begin{bmatrix} \dot{p}_{xc}(t) \\ \dot{p}_{yc}(t) \\ \dot{\theta}_{zc}(t) \end{bmatrix} = \frac{R_c}{2} \begin{bmatrix} \cos(\theta_{zc}(t)) & \cos(\theta_{zc}(t)) \\ \sin(\theta_{zc}(t)) & \sin(\theta_{zc}(t)) \\ 1/B_c & -1/B_c \end{bmatrix} \begin{bmatrix} W_L(t) \\ W_R(t) \end{bmatrix} \quad (3)$$

In the actual movement, we need to convert continuous motion into discrete motion for controlling the mobile platform. In this case, the rotation speeds of the left wheel and the right wheel are regarded as the control variables; the linear velocity along the x axis and y axis of the world frame is denoted as $\dot{p}_{xc}(k)$ and $\dot{p}_{yc}(k)$, respectively; and the angular velocity around the z axis of the world frame is $\dot{q}_{zc}(k)$. Thus, the linear velocity and the angular velocity of the mobile platform can be expressed as follows:

$$\begin{bmatrix} \dot{p}_{xc}(k) \\ \dot{p}_{yc}(k) \\ \dot{q}_{zc}(k) \end{bmatrix} = \frac{R_c}{2} \begin{bmatrix} \cos(q_{zc}(k)) & \cos(q_{zc}(k)) \\ \sin(q_{zc}(k)) & \sin(q_{zc}(k)) \\ 1/B_c & -1/B_c \end{bmatrix} \begin{bmatrix} \dot{q}_{cl}(k) \\ \dot{q}_{cr}(k) \end{bmatrix} \quad (4)$$

where q_{zc} is the rotation angle of the mobile platform at this time; \dot{q}_{cl} and \dot{q}_{cr} are the rotation velocity of the left and right wheel at this time, respectively.

When the rotation velocity of the driving wheels is regarded as the mobile platform's control variable, p_{xc} and p_{yc} are the position of the mobile platform along the x axis and the y axis of the world frame, respectively; and q_{zc} is the rotation angle of the z axis. Therefore, the discrete kinematics model of the mobile platform at $(k + 1)$ time can be described as follows:

$$\begin{bmatrix} p_{xc}(k + 1) \\ p_{yc}(k + 1) \\ q_{zc}(k + 1) \end{bmatrix} = \begin{bmatrix} p_{xc}(k) \\ p_{yc}(k) \\ q_{zc}(k) \end{bmatrix} + d_k \cdot \frac{R_c}{2} \begin{bmatrix} \cos(q_{zc}(k)) & \cos(q_{zc}(k)) \\ \sin(q_{zc}(k)) & \sin(q_{zc}(k)) \\ 1/B_c & -1/B_c \end{bmatrix} \begin{bmatrix} \dot{q}_{cl}(k) \\ \dot{q}_{cr}(k) \end{bmatrix} \quad (5)$$

where d_k is the sampling time during the movement of the mobile platform.

(2) Practical kinematics discrete model

For a practical mobile platform, the midpoint of the two driving wheels cannot wholly coincide with the actual gravity center of the mobile platform. As shown in Figure 1, an offset may be created between the midpoint of the driving axis and the gravity center of the mobile platform, D_{cz} is the offset at the vertical height (the z axis of the driving-fixed frame), D_{cx} is the offset at the horizontal direction (the x axis of the driving-fixed frame), and D_{cy} is the offset at the y axis of the driving-fixed frame.

Assuming that the mobile platform is moving in the horizontal plane, the impact of D_{cx} needs to be considered, but that of D_{cz} should be ignored in the kinematic modeling of the mobile platform. Therefore, if the gravity center coincides with the midpoint of the driving wheels, Formula (6) can be used to describe the position and posture of the gravity center of the mobile platform at $(k + 1)$ time, which is expressed as follows:

$$\begin{bmatrix} p_{xd}(k+1) \\ p_{yd}(k+1) \\ q_{zd}(k+1) \end{bmatrix} = \begin{bmatrix} p_{xd}(k) \\ p_{yd}(k) \\ q_{zd}(k) \end{bmatrix} + d_k \frac{R_c}{2} \begin{bmatrix} \cos(q_{zd}(k)) & \cos(q_{zd}(k)) \\ \sin(q_{zd}(k)) & \sin(q_{zd}(k)) \\ 1/B_c & -1/B_c \end{bmatrix} \begin{bmatrix} \dot{q}_{cl}(k) \\ \dot{q}_{cr}(k) \end{bmatrix} \quad (6)$$

where p_{xd} and p_{yd} indicate the midpoint location of the driving wheel axis in the xy plane, and q_{zd} indicates the angle rotation around the z axis of the driving-fixed frame.

Suppose the gravity center does not coincide with the midpoint of the two driving wheels. In this case, to the actual position of the gravity center, the corresponding offset needs to be added to the kinematics model, which is expressed as follows:

$$\begin{bmatrix} p_{xc}(k) \\ p_{yc}(k) \\ q_{zc}(k) \end{bmatrix} = \begin{bmatrix} p_{xd}(k) \\ p_{yd}(k) \\ q_{zd}(k) \end{bmatrix} + \begin{bmatrix} \cos(q_{zd}(k)) & -\sin(q_{zd}(k)) & 0 \\ \sin(q_{zd}(k)) & \cos(q_{zd}(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_{cx} \\ D_{cy} \\ 0 \end{bmatrix} \quad (7)$$

where p_{xc} and p_{yc} are the position of the gravity center of the mobile robot in the xy plane, and q_{zc} is the rotation angle of the mobile robot.

Remark 1. When selecting the position and posture of the gravity center as the output variable, the offset between the midpoint of driving wheels and the gravity center of the mobile platform needs to be considered. To reduce the complexity of the trajectory tracking problem, the position and posture of the midpoint of driving wheels in the world frame are regarded as the desired output variables.

2.2.2. State Equation of the Mobile Platform

(1) Ideal state equation without disturbance

Converting the discrete kinematics model into the state equation, we redefine various variables of the mobile platform, where $u(k) = [\dot{q}_{cl}(k), \dot{q}_{cr}(k)]^T$ is the input control variable, $y(k) = [p_{xc}(k), p_{yc}(k), q_{zc}(k)]^T$ is the output variable, and $q(k)$ is the state variable of the system. Therefore, the discrete state equation of the mobile platform can be expressed as follows:

$$\begin{cases} q(k+1) = q(k) + d_k \cdot B(q_k) \cdot u(k) \\ y(k) = q(k) \end{cases} \quad (8)$$

where $B(q_k)$ is the the control matrix of the mobile platform, which can be expressed as follows:

$$B(q_k) = \frac{R_c}{2} \begin{bmatrix} \cos(q_{zc}(k)) & \cos(q_{zc}(k)) \\ \sin(q_{zc}(k)) & \sin(q_{zc}(k)) \\ 1/B_c & -1/B_c \end{bmatrix} \quad (9)$$

(2) Practical state equation with disturbance

The primary parameters of the state equation include the wheel radius and the width of the mobile platform. However, the above parameters can only be obtained through actual measurement, and the specific error between the measured parameters and the actual parameter cannot be ignored. Therefore, the exact state equation of the mobile platform can be expressed as follows:

$$\begin{cases} q(k+1) = q(k) + d_k \cdot \tilde{B}(q_k) \cdot u(k) \\ y(k) = q(k) \end{cases} \quad (10)$$

where $\tilde{B}(q_k)$ is the estimated parameter of the corresponding mobile platform.

The external disturbances of various signals need to be added to the actual state equation, where $\beta(k)$ is the random disturbance in the state parameters of the system, and $\alpha(k)$ is the random disturbances in the initial parameters of the system. Considering the above disturbance parameters in Formula (10), we can express the practical state equation with disturbance as follows:

$$\begin{cases} q(k+1) = q(k) + d_k \cdot B(q_k) \cdot u(k) + \tilde{\beta}(k) \\ y(k) = q(k) + \tilde{\alpha}(k) \end{cases} \quad (11)$$

where $\tilde{\beta}(k) = \beta(k) + d_k \cdot (\tilde{B}(q_k) - B(q_k)) \cdot u(k)$ is the sum of the state disturbance and the system, and $\tilde{\alpha}(k) = \alpha(k)$ is the combination of the output disturbance and the measurement error of the system.

Remark 2. The kinematics model is mainly used to construct the basic control model of the mobile platform. The control signal needs to be directly input to the actual mobile platform in trajectory tracking. Therefore, we did not need to perform high-precision tests on the above-mentioned physical structural parameters.

2.3. Actual Limitations

To obtain the actual physical limitations of the mobile platform, some constraints on the input variables and output variables must be described by formulas before performing the trajectory tracking task. Therefore, we specifically express various constraints as follows:

(1) Structure parameter limitations: The measurement error cannot be ignored when obtaining the radius of the driving wheels and the width of the mobile platform, which may be caused by the limitation of the measurement tool and the surveyor’s experience. Therefore, the measurement error must be in a particular range, which can be expressed as follows:

$$\begin{cases} \max \|Bc - \tilde{B}c\| < b_{Bc} \\ \max \|Rc - \tilde{R}c\| < b_{Rc} \end{cases} \Rightarrow \max \|B(q_k)\| < b_B \quad (12)$$

where b_{Bc}, b_{Rc} is the maximum error of the actual estimated parameters, and b_B is the maximum error of the control object.

(2) Signal disturbance limitations: The external environment is one of the key factors that cause other disturbances in the system. Therefore, the random disturbance of the state parameters and the output variable of the system are also within a specific constraint range, defined as follows:

$$\begin{cases} \max \|\beta(k)\| < b_\beta \\ \max \|\alpha(k)\| < b_\alpha \end{cases} \Rightarrow \begin{cases} \max \|\beta(k) + d_k(\tilde{B}(q_k) - B(q_k)) \cdot u(k)\| < \tilde{b}_\beta \\ \max \|\tilde{\alpha}_i(k)\| = \max \|\alpha(k)\| < \tilde{b}_\alpha \end{cases} \quad (13)$$

where b_β is the maximum disturbance of the system state variable, and b_α is the maximum disturbance of the output signal.

(3) Motor physical limitations: Due to the limitations of the motor and the voltage of the driving wheels, the rotation speed of the driving wheels must have a maximum speed limitation, expressed as follows:

$$\begin{cases} \max \|\dot{q}_{cl}(k)\| \leq b_w \\ \max \|\dot{q}_{cr}(k)\| \leq b_w \end{cases} \Leftrightarrow \max \|u_i(k)\| \leq b_{u_d} \quad (14)$$

where b_w is the maximum speed of the two driving wheels, and b_{u_d} is representing the maximum norm of the two-wheel drive speed matrix. The driving wheels were produced as one model to improve the control accuracy of the mobile platform.

(4) Initial error limitations: During the trajectory tracking task, the initial position of the robot and the starting position of the given trajectory are kept within a certain error range, which can be expressed as follows

$$\|q_d(0) - q_k(0)\| \leq C_{q_0} \quad (15)$$

where, $q_k(0)$ represents the initial position and posture of the robot in each iteration learning, $q_d(0)$ represents the starting position and posture of a given trajectory, and C_{q_0} is the maximum norm of the initial error matrix of the system.

(5) Kinematics limitations: In the robot state Equation (11), the B matrix needs to meet $\max \|B(q_k)\| < b_B$. Meanwhile, the function needs to satisfy the global Lipschitz condition

$$\|d_k \cdot B(q_{i+1}(k)) - d_k \cdot B(q_i(k))\| \leq C_B \|q_{i+1}(k) - q_i(k)\| \tag{16}$$

where d_k is a constant that represents the sampling time, and C_B is the maximum value.

3. Design of the Controller

3.1. Design of the Hybrid Learning Iteration Controller for Mobile Platform

The hybrid controller has a two-layer hybrid structure that involves an iterative learning controller without initial error and an iterative learning controller with initial error; the basic structure of the hybrid controller is shown in Figure 2. To determine the choice of the corresponding controller, we designed the switch function based on the initial error of the mobile platform in different trajectory tracking tasks. When the initial error of the robot is equal to zero, the hybrid controller first uses the iterative controller without initial error to complete the assigned task. When the initial error exists in the system, it proceeds to the second type of iterative learning controller through the switching function. The specific design steps are as follows:

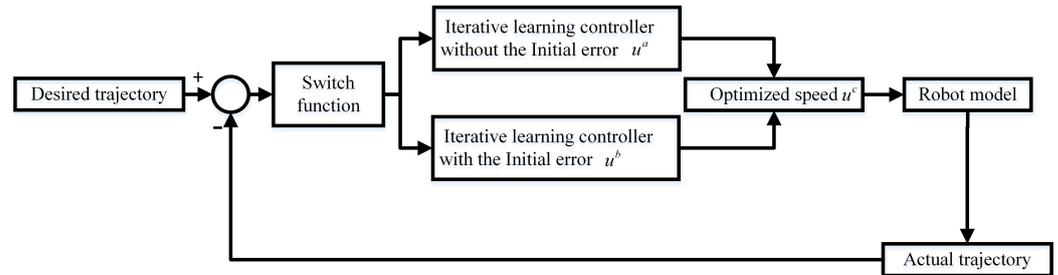


Figure 2. Framework for the hybrid iterative learning controller (“+” represents the input, “−” represents the feedback).

Step 1: Design the learning iteration controller without initial error. The basic aim of the iterative learning controller is to correct the actual input signal by iteratively learning the error between the ideal output signal and the actual input signal to perform the specified trajectory tracking task. Due to the different types of error parameters obtained, we can optimize the typical P-type learning control law by increasing different types of error signals. The optimized iterative learning controller can be expressed as follows:

$$u_{i+1}^a(k) = u_i^a(k) + L_1(k)e_i(k + 1) + L_2(k)e_{i+1}(k) + L_3(k)e_{i+1}(k - 1) \tag{17}$$

where $u_{i+1}^a(k)$ indicates the control speeds at k time in the next iteration, $e_i(k + 1)$ indicates the error at $(k + 1)$ time in the current iteration, $e_{i+1}(k)$ indicates the error at k time in the next iteration, $e_{i+1}(k - 1)$ indicates the error at $(k - 1)$ time in the next iteration, and $L_1(k), L_2(k), L_3(k)$ indicate the gain matrix of iterative learning in different error categories, and these gain matrices need to meet

$$\|L_1(k)\| \leq b_{L_1}, \|L_2(k)\| \leq b_{L_2}, \|L_3(k)\| \leq b_{L_3} \tag{18}$$

where $b_{L_1} > 0, b_{L_3} > 0, b_{L_2} > 0, k \in N$.

Theorem 1. For the mobile robot described by Formula (11), in addition to satisfying the given kinematic characteristics and the specified constraint, the robot still satisfies the following conditions:

$$\begin{cases} \|I - [L_1(k) + L_2(k) + L_3(k)]B(q_i, k)\| \leq \rho < 1 \\ y_i(0) = y_d(0), (i = 0, 1, 2, \dots) \end{cases} \tag{19}$$

By using the iterations learning law (17), the output signal of the mobile system can be converted to the desired trajectory $y_d(k) (0 \leq k \leq n)$, namely

$$\lim_{k \rightarrow \infty} y_s(k) \rightarrow y_d(k) (0 \leq k \leq n) \tag{20}$$

Proof. The proof of Theorem 1 is shown in Section 3.2 of the article. \square

Step 2: Design the learning iteration controller with initial error. To achieve trajectory tracking, the initial error of the typical iterative control law (17) needs to be equal to zero. In the actual movement of the mobile platform, error may exist between the initial position of the mobile platform and the given trajectory. In this case, the iterative learning controller needs to compensate for the corresponding error in the initial moment to achieve trajectory tracking. Therefore, the iterative learning controller can be expressed as follows:

$$u_{i+1}^b(k) = u_i^b(k) + L_1(k)e_i(k+1) + e^{-kdt}\theta(k)L_1(k)e_i(0) \tag{21}$$

where $e_i(0)$ represents the initial error of each iteration learning process, and $\theta(k)$ represents the corresponding compensation function, which can be expressed as follows:

$$\theta(k) = \begin{cases} \frac{2}{d_k * h} (1 - \frac{k}{h}), & (0 \leq k \leq h) \\ 0, & (h < k \leq n) \end{cases} \tag{22}$$

Theorem 2. For the mobile robot described by Formula (11), in addition to satisfying the given kinematic characteristics and the specified constraint, the robot still satisfies the following conditions:

$$\begin{cases} y_i(0) = y_d(0) \\ \|I - L_1(k)B(q_i, k)\| \leq \rho < 1 \end{cases} \tag{23}$$

By using the iterations learning law (21), the output signal of the mobile system can be converted to the desired trajectory $y_d(k) (0 \leq k \leq n)$, namely

$$y_s(k) = \begin{cases} y_d(k) + e^{-kdt}L_1(k) \sum_{j=h}^k \theta(k)(y_d(0) - y_i(0)) & (0 \leq k \leq h) \\ y_d(k) & (h < k \leq n) \end{cases} \tag{24}$$

Proof. The proof of Theorem 2 is very similar as the proof of Theorem 1, but it is simpler than Section 3.2. The proof process can refer to the literature [38]. \square

Step 3: Optimize the control speeds. The input control signals of the driving wheels can be obtained through the above two types of iterative controllers. Due to the limitation on the voltage and structural design of the drive motor in the actual robot system, the maximum speed of the motor movement is a given value, which can be expressed as b_w . Then, we need to optimize the control speeds solved by the iterative controller. To ensure the rotation movement of the mobile platform, the optimization signal needs to ensure the rotation speed ratio of the two driving wheels is the same as the solution result. Therefore, the optimized control speeds can be expressed as follows:

$$u^c(k) = \begin{cases} \left[\frac{\dot{q}_{cl}(k)}{\max(\dot{q}_{cl}(k), \dot{q}_{cr}(k))} b_w, \frac{\dot{q}_{cr}(k)}{\max(\dot{q}_{cl}(k), \dot{q}_{cr}(k))} b_w \right]^T, & \text{if } (\max(\dot{q}_{cl}(k), \dot{q}_{cr}(k)) \geq b_w); \\ \left[\dot{q}_{cl}(k), \dot{q}_{cr}(k) \right]^T, & \text{others;} \end{cases} \tag{25}$$

where $u^c(k)$ represents the optimized system control signal, b_w represents the maximum speed of the two driving wheels, $\dot{q}_{cl}(k)$ represents the speed of the left drive wheel, and $\dot{q}_{cr}(k)$ represents the speed of the right drive wheel of the system.

3.2. Theoretical Proof

To prove Theorem 1, we need to solve the error between the desired control signal and the actual control signal and analyze the error between the expected trajectory and the actual trajectory. The convergence of the controller is proved as follows:

Step 1: In solving the error between the desired control signal and the actual control signal, we need to use the state error of the system. Assuming q_d represents the expected motion state of the system, the state error can be expressed as follows:

$$\begin{aligned} \Delta q_i(k+1) &= q_d(k+1) - q_i(k+1) \\ &= [q_d(k) + d_k B_d u_d(k)] - (q_i(k) + d_k B_i u_i(k) + \tilde{\beta}_i(k)) \\ &= \Delta q_i(k) + (B_d - B_i)u_d(k) - B_i(u_d(k) - u_i(k)) - \tilde{\beta}_i(k) \end{aligned} \tag{26}$$

According to the formula constraint expression, the state error of the system can be effectively scaled, which can be expressed as follows:

$$\|\Delta q_i(k+1)\| \leq \|\Delta q_i\| + C_B b_{u_d} \|\Delta q_i\| + b_B \|\Delta u_i\| + \tilde{b}_\beta \tag{27}$$

If we set $h_2 = (1 + C_B b_{u_d})$, the state error (27) can be simplified again, which can be expressed as follows:

$$\|\Delta q_i(k+1)\| \leq h_2 \|\Delta q_i\| + b_B \|\Delta u_i\| + \tilde{b}_\beta \tag{28}$$

Solving the state error by iteration method from the state error at the initial moment to the current moment, we obtain the state error expression:

$$\begin{aligned} \|\Delta q_i\| &\leq (\|h_2 b_B \Delta u_1\| + \|h_2^2 b_B \Delta u_2\| + \dots + \|h_2^{i-1} b_B \Delta u_{i-1}\|) \\ &\quad + h_2 (\tilde{b}_\beta + \tilde{b}_\beta + \dots + \tilde{b}_\beta) + h_2^k b_{q_0} \\ &\leq \sum_{j=0}^{k-1} (h_2^{k-1-j} [b_B \|\Delta u_i\| + \tilde{b}_\beta]) + h_2^k b_{q_0} \end{aligned} \tag{29}$$

where b_B represents the maximum error of the control object, \tilde{b}_β is the maximum disturbance of the state variable, and b_{q_0} indicates the error of the system at the initial time.

Step 2: Assuming that u_d as the desired control signal can control the mobile platform to track the desired trajectory. The error between the actual control signal and the desired control signal can be expressed as follows:

$$\begin{aligned} \Delta u_{i+1}(k) &= u_d(k) - u_{i+1}(k) \\ &= u_d(k) - u_i(k) - L_1 e_i(k+1) - L_2 e_{i+1}(k) - L_3 e_{i+1}(k-1) \\ &= u_d(k) - u_i(k) - L_1 [y_d(k+1) - y_i(k+1)] \\ &\quad - L_2 [y_d(k) - y_{i+1}(k)] - L_3 [y_d(k-1) - y_{i+1}(k-1)] \\ &= \|\Delta u_i\| - L_1 (q_d(k+1) - q_i(k+1) - \alpha_i(k+1)) \\ &\quad - L_2 (q_d(k) - q_{i+1}(k) - \alpha_{i+1}(k+1)) \\ &\quad - L_3 (q_d(k-1) - q_{i+1}(k-1) - \alpha_{i+1}(k-1)) \\ &= \|\Delta u_i\| - L_1 [\|\Delta q_i\| + (B_d - B_i)u_d + B_i \|\Delta u_i\| - \alpha_i(k)] \\ &\quad - L_2 (\|\Delta q_{i+1}\| - \alpha_{i+1}(k+1)) - L_3 (\|\Delta q_{i-1}\| - \alpha_{i-1}(k-1)) \\ &= \|\Delta u_i\| - L_1 \|\Delta q_i\| - L_1 (B_d - B_i)u_d - L_1 B_i \|\Delta u_i\| + L_1 \alpha_i(k) \\ &\quad - L_2 \|\Delta q_{i+1}\| + L_2 \alpha_{i+1}(k+1) - L_3 \|\Delta q_{i-1}\| + L_3 \alpha_{i-1}(k-1) \\ &= (1 - L_1 B_i) \|\Delta u_i\| - L_1 \|\Delta q_i\| - L_1 (B_d - B_i)u_d + L_1 \alpha_i(k) \\ &\quad - L_2 \|\Delta q_{i+1}\| + L_2 \alpha_{i+1}(k+1) - L_3 \|\Delta q_{i-1}\| + L_3 \alpha_{i-1}(k-1) \end{aligned} \tag{30}$$

Considering physical constraints, Formula (30) can be simplified to the following:

$$\begin{aligned} \|\Delta u_{i+1}(k)\| &\leq \|I - L_1 B_i\| \|\Delta u_i\| + b_{L1} \|\Delta q_i\| + b_{L1} c_B b_u \|\Delta q_i\| + b_{L1} b_\alpha \\ &\quad + b_{L2} \|\Delta q_{i+1}\| + b_{L2} b_\alpha + b_{L3} \|\Delta q_{i-1}\| + b_{L3} b_\alpha \\ &= \|I - L_1 B_i\| \|\Delta u_i\| + (b_{L1} + b_{L1} c_B b_u) \|\Delta q_i\| + b_{L2} \|\Delta q_{i+1}\| \\ &\quad + b_{L3} \|\Delta q_{i-1}\| + (b_{L1} b_\alpha + b_{L2} b_\gamma + b_{L3} b_\alpha) \end{aligned} \tag{31}$$

If we set $h_1 = b_{L1} + b_{L1} c_B b_u = b_{L1} (1 + c_B b_u)$, $b_1 = (b_{L1} b_\alpha + b_{L2} b_\alpha + b_{L3} b_\alpha)$, Formula (31) can be simplified again as follows:

$$\|\Delta u_{i+1}(k)\| \leq \|I - L_1 B_i\| \|\Delta u_i\| + h_1 \|\Delta q_i\| + b_{L2} \|\Delta q_{i+1}\| + b_{L3} \|\Delta q_{i-1}\| + b_1 \tag{32}$$

The Formula (32) contains the system output error function and the system state error function; however, we need to eliminate the system state error to obtain the output error function. Substitute the status error (29) into the above system Formula (32) to obtain the following:

$$\begin{aligned} \|\Delta u_{i+1}(k)\| &\leq \|I - L_1 B_i\| \|\Delta u_i\| + h_1 \left[\sum_{j=0}^{k-1} [h_2^{k-1-j} (b_B \|\Delta u_i\| + b_\beta) + h_2^k b_{q_0}] \right] \\ &\quad + b_{L2} \left[\sum_{j=0}^{k-1} [h_2^{k-1-j} (b_B \|\Delta u_{i+1}\| + b_\beta) + h_2^k b_{q_0}] \right] \\ &\quad + b_{L3} \left[\sum_{j=0}^{k-2} [h_2^{k-2-j} (b_B \|\Delta u_{i-1}\| + b_\beta) + h_2^{k-1} b_{q_0}] \right] + b_1 \\ &= \|I - L_1 B_i\| \|\Delta u_i\| + h_1 h_2^k b_{q_0} + b_{L2} h_2^k b_{q_0} + b_{L3} h_2^{k-1} b_{q_0} + b_1 \\ &\quad + h_1 \sum_{j=0}^{k-1} h_2^{k-1-j} (b_B \|\Delta u_i\| + b_\beta) + b_{L2} \sum_{j=0}^{k-1} h_2^{k-1-j} (b_B \|\Delta u_{i+1}\| + b_\beta) \\ &\quad + b_{L3} \sum_{j=0}^{k-2} h_2^{k-2-j} (b_B \|\Delta u_{i-1}\| + b_\beta) \end{aligned} \tag{33}$$

Multiply the left and right sides of the above Formula (33) by $\frac{1}{\lambda^k}$, which can, thus, be solved to obtain the following:

$$\begin{aligned} \|\Delta u_{i+1}(k)\| \left(\frac{1}{\lambda^k}\right) &\leq \|I - L_1 B_i\| \|\Delta u_i\| \left(\frac{1}{\lambda^k}\right) + h_1 \frac{h_2^k}{\lambda^k} b_{q_0} + b_{L2} \frac{h_2^k}{\lambda^k} b_{q_0} + b_{L3} \frac{h_2^{k-1}}{\lambda^k} b_{q_0} \\ &\quad + b_1 \left(\frac{1}{\lambda^k}\right) + h_1 \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^k} (b_B \|\Delta u_i\| + b_\beta) + b_{L2} \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^k} (b_B \|\Delta u_{i+1}\| + b_\beta) \\ &\quad + b_{L3} \sum_{j=0}^{k-2} \frac{h_2^{k-2-j}}{\lambda^k} (b_B \|\Delta u_{i+1}\| + b_\beta) \\ &= \|I - L_1 B_i\| \|\Delta u_i\| \left(\frac{1}{\lambda^k}\right) + h_1 \frac{h_2^k}{\lambda^k} b_{q_0} + b_{L2} \frac{h_2^k}{\lambda^k} b_{q_0} + b_{L3} \frac{h_2^{k-1}}{\lambda^k} b_{q_0} \left(\frac{1}{\lambda^k}\right) \\ &\quad + b_1 \left(\frac{1}{\lambda^k}\right) + \frac{h_1}{\lambda} \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^{k-1-j}} (b_B \frac{\|\Delta u_i\|}{\lambda^j} + \frac{b_\beta}{\lambda^j}) + \frac{b_{L2}}{\lambda} \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^{k-1-j}} (b_B \frac{\|\Delta u_{i+1}\|}{\lambda^j} + \frac{b_\beta}{\lambda^j}) \\ &\quad + \frac{b_{L3}}{\lambda^2} \sum_{j=0}^{k-2} \frac{h_2^{k-2-j}}{\lambda^{k-2-j}} (b_B \frac{\|\Delta u_{i+1}\|}{\lambda^j} + \frac{b_\beta}{\lambda^j}) \\ &= \rho \|\Delta u_i\|_\lambda + h_1 b_{q_0} + b_{L2} b_{q_0} + b_{L3} b_{q_0} \left(\frac{1}{\lambda^k}\right) + b_1 \left(\frac{1}{\lambda^k}\right) \\ &\quad + \frac{h_1}{\lambda} (b_B \|\Delta u_i\|_\lambda + b_\beta) \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^{k-1-j}} + \frac{b_{L2}}{\lambda} (b_B \|\Delta u_{i+1}\|_\lambda + b_\beta) \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^{k-1-j}} \\ &\quad + \frac{b_{L3}}{\lambda^2} (b_B \|\Delta u_{i-1}\|_\lambda + b_\beta) \sum_{j=0}^{k-2} \frac{h_2^{k-2-j}}{\lambda^{k-2-j}} \\ &\leq \rho \|\Delta u_i\|_\lambda + h_1 b_{q_0} + b_{L2} b_{q_0} + b_{L3} b_{q_0} + b_1 \\ &\quad + (b_B \|\Delta u_i\|_\lambda + b_\beta) \left(\frac{h_1 [1 - (h_2/\lambda)^n]}{\lambda - h_2}\right) + (b_B \|\Delta u_{i+1}\|_\lambda + b_\beta) \left(\frac{b_{L2} [1 - (h_2/\lambda)^n]}{\lambda - h_2}\right) \\ &\quad + (b_B \|\Delta u_{i+1}\|_\lambda + b_\beta) \left(\frac{b_{L3} [1 - (h_2/\lambda)^n]}{\lambda - h_2}\right) \\ &= \left(\rho + \frac{b_\beta h_1}{\lambda - h_2}\right) \|\Delta u_i\| + \frac{(b_B b_{L2} + b_B b_{L3}) [1 - (h_2/\lambda)^n]}{\lambda - h_2} \|\Delta u_{i+1}\|_\lambda \\ &\quad + h_1 b_{q_0} + b_{L2} b_{q_0} + b_{L3} b_{q_0} + b_1 + \frac{b_\beta (h_1 + b_{L2} + b_{L3}) [1 - (h_2/\lambda)^n]}{\lambda - h_2} \end{aligned} \tag{34}$$

If we set $\lambda > \max\{1, h_2, b_B(b_{L2} + b_{L3})\}$, we can obtain the following:

$$\left[1 - \frac{b_B(b_{L2} + b_{L3})[1 - (h_2/\lambda)^n]}{\lambda - h_2}\right] > 0 \Rightarrow \begin{cases} \alpha - h_2 \neq 0 \\ \alpha > h_2 + b_B(b_{L2} + b_{L3}) \\ \alpha - h_2 > b_B(b_{L2} + b_{L3})[1 - (h_2/\lambda)^n] \end{cases} \quad (35)$$

Then, the above inequality (35) can be simplified as follows:

$$\|\Delta u_{i+1}(k)\|_\lambda \leq A\|\Delta u_i(k)\|_\lambda + B \quad (36)$$

where

$$A = \frac{\left[\rho + \frac{b_B h_1}{\lambda - h_2}\right]}{\left[1 - \frac{b_B(b_{L2} + b_{L3})[1 - (h_2/\lambda)^n]}{\lambda - h_2}\right]}, B = \frac{\left[h_1 b_{q_0} + b_{L2} b_{q_0} + b_{L3} b_{q_0} + b_1 + \frac{b_\beta(h_1 + b_{L2} + b_{L3})[1 - (h_2/\lambda)^n]}{\lambda - h_2}\right]}{\left[1 - \frac{b_B(b_{L2} + b_{L3})[1 - (h_2/\lambda)^n]}{\lambda - h_2}\right]} \quad (37)$$

Iterating the output error from the initial moment to the current moment, the output error at any moment can be obtained:

$$\|\Delta u_{i+1}(k)\|_\lambda \leq A\|\Delta u_i(k)\|_\lambda + B = A^i\|\Delta u_1(k)\|_\lambda + \frac{B(1 - A^i)}{1 - A} \quad (38)$$

If we set λ to be very large and the system parameter $A \approx \rho < 1$, Formula (38) can be simplified as follows:

$$\lim_{i \rightarrow \infty} \|\Delta u_i(k)\|_\lambda \leq \frac{B}{1 - A} \quad (39)$$

When the output disturbance $b_\alpha = 0$, the initial error $b_{q_0} = 0$ and the state disturbance signal $b_\beta = 0$ are brought into Formulas (37) and (39), the output error of the controller is proved to approach zero asymptotically. Namely, the actual output of the controller is infinitely close to the desired control signal.

Step 3: In analyzing the error between the actual trajectory and the expected trajectory, we need to use the state error of the system. According to Formulas (29) and (39), we can find the expression of the state error. To analyze the output error of the system, we also need to multiply the state error by $(\lambda)^{-k}$. Then, the expression can be simplified as follows:

$$\begin{aligned} \|\Delta q_i\| \left(\frac{1}{\lambda^k}\right) &\leq \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^{k-1-j}} \left(b_B \frac{\|\Delta u_i\|}{\lambda^j} + b_\beta \frac{1}{\lambda^j}\right) + \frac{h_2^k}{\lambda^k} b_{q_0} \\ &\leq (b_B \|\Delta u_i\|_\lambda + b_\beta) \sum_{j=0}^{k-1} \frac{h_2^{k-1-j}}{\lambda^{k-1-j}} + b_{q_0} \\ &= b_B \frac{[1 - (h_2/\lambda)^k]}{\lambda - h_2} \|\Delta u_i\|_\lambda + b_\beta \frac{[1 - (h_2/\lambda)^k]}{\lambda - h_2} + b_{q_0} \end{aligned} \quad (40)$$

After infinite iterations, the state error of system can be simplified to the following:

$$\lim_{i \rightarrow \infty} \|\Delta q_i\|_\lambda \leq b_B \frac{[1 - (h_2/\lambda)^n]}{\lambda - h_2} \frac{B}{1 - A} + b_\beta \frac{[1 - (h_2/\lambda)^n]}{\lambda - h_2} + b_{q_0} \quad (41)$$

where b_B represents the maximum norm of the control matrix, b_β represents the maximum norm of the state disturbance, and b_{q_0} represents the initial error of the system. Assuming the initial error in the Formula (41) is $b_\alpha = 0, b_\beta = 0, b_{q_0} = 0$, the actual state output of the system is consistent with the expected state output.

Step 4: To prove that the controller can track the desired trajectory, we need to analyze the output error of the system, which can be expressed as follows:

$$\begin{aligned} \|\Delta y_i(k)\| &= y_d(k) - y_i(k) = q_d(k) - (q_i(k) + \alpha_i(k)) \\ &= \|\Delta q_i(k)\| - b_\alpha \end{aligned} \quad (42)$$

After infinite iterations, the output error can be simplified to the following:

$$\lim_{i \rightarrow \infty} \|\Delta y_i\|_\lambda \leq \lim_{i \rightarrow \infty} \|\Delta q_i\|_\lambda - b_\alpha \quad (43)$$

where b_α represents the maximum norm of the output disturbance signal of the system. If the maximum norm of output disturbance signal is $b_\alpha = 0$, the output error of the system is close to the state error. When the state error of the third step is $\lim_{i \rightarrow \infty} \|\Delta q_i\|_\lambda = 0$, the actual trajectory of the robot can follow the expected trajectory.

Comparing the output error signal of the controller (39), the state error (41), and the output error (43) of the system, we can find that: (1) When the output disturbance $b_\alpha = 0$, the initial error $b_{q_0} = 0$ and the state disturbance signal $b_\beta = 0$, the actual control signal will converge to the desired control signal. (2) When the output disturbance $b_\alpha = 0$, the initial error $b_{q_0} = 0$ and the state disturbance signal $b_\beta = 0$, the output signal of the system will converge to the expected trajectory.

4. Experimental Results

To validate the performance of the hybrid controller, we designed some experiments with various tracking trajectories by directly control the driving wheels. To enable the general use of the controller, we selected a personal computer to simulate the performance. The computer was equipped with an Intel(R) Core(TM) i5-7500, CPU @3.4HZ, RAM @8G, and the basic configuration was the a medium type in terms of computing performance. The basic parameters of the mobile platform are specifically expressed as follows:

$$\begin{aligned} L_c &= 1.104 \text{ m}; & B_c &= 0.518 \text{ m}; & R_c &= 0.200 \text{ m}; \\ D_{cx} &= 0.340 \text{ m}; & D_{cy} &= 0.00 \text{ m}; & D_{cz} &= 0.119 \text{ m}; & u_{max} &= 180 \text{ rad/s}; \end{aligned} \quad (44)$$

The following subsection summarizes the processes and the results of these typical simulations.

4.1. Basic Performance Test

To test the basic performance of the iterative controller proposed in this paper, we conducted the first simulation experiment for various trajectory tracking tasks by directly driving the two wheels of the mobile robot shown in Figure 3. Four typical trajectory tracking tasks were used in the first simulation: circular trajectory, combination trajectory of arc and straight, cardioid-like trajectory, and spiral-like trajectory.

Trajectory 1: The circular trajectory is a closed path composed of a single arc. The circle's center is $P_{C1}(0, 0)$, the radius length is $R_1 = 10$ m, and the initial point is $P_0(0, 10)$. The mobile platform needs to return to the initial position by rotating 360° in the counterclockwise direction of the circular trajectory, the total time of the trajectory is $t_s = 2$ s, the sampling time is $d_t = 0.001$ s, and the initial position and posture of the mobile platform is $(0, 10, \pi/2)$.

Trajectory 2: The combination trajectory of arc and straight is also a closed path composed of two arcs and two straight lines. In this path, the center of first arc is $P_{C1}(0, 0)$, and the radius is $R_1 = 5$ m; the center of second arc is $P_{C2}(0, -10)$, the radius is $R_2 = 5$ m; the two straight lines connect the upper points and the lower points of the two arcs. The mobile platform needs to rotate 360° in the counterclockwise direction along the combination trajectory and return to the initial position. The time of each arc is $t_{C1} = t_{C2} = 0.5$ s, the time of each straight line is $t_{L1} = t_{L2} = 1$ s, the total time of the combination trajectory is $t_s = 3$ s, the sampling time is $d_t = 0.001$ s, and the initial position and posture of the mobile platform are $(0, 5, \pi/2)$.

Trajectory 3: The cardioid-like trajectory is a closed path composed of multiple semicircles. In this path, the center of the first semicircle is $P_{C1}(0, 0)$, the radius is $R_1 = 2.5$ m, and the starting point of the first semicircle is $(2.5, 0)$. The mobile platform arrives at the endpoint of the first semicircle trajectory by rotating 180° in the counterclockwise direction.

The starting point of the second semicircle is set as the first endpoint, the center of the second semicircle is $P_{C2}(2.5, 0)$, and the radius is $R_2 = 5$ m. The mobile robot rotates 180° in the counterclockwise direction to the endpoint of the second semicircle trajectory. The starting point of the third semicircle is set as the second endpoint, the center of the third semicircle is $P_{C3}(5, 0)$, and the radius is $R_3 = 2.5$ m. The mobile robot returns to the initial position of the cardioid-like trajectory by rotating 180° in the counterclockwise direction. The duration for first and third semicircle is $t_{C1} = t_{C3} = 0.5$ s, the time of the second semicircle is $t_{C2} = 1$ s, the total time is $t_s = 2$ s, the sampling time is $d_t = 0.001$ s, and the initial position and posture of the mobile platform are $(2.5, 0, \pi/2)$.

Trajectory 4: The spiral-like trajectory is an open path composed of multiple semicircles. In this path, the center of the first semicircle is $P_{C1}(0, 0)$, the radius is $R_1 = 3$ m, and the starting point of first semicircle is $(2.5, 0)$. The mobile platform arrives at the endpoint of the first semicircle trajectory by rotating 180° in the counterclockwise direction. The starting point of the second semicircle is set as the first endpoint, the center of the second semicircle is $P_{C2}(0.5, 0)$, and the radius is $R_2 = 3.5$ m. The mobile platform rotates 180° in the counterclockwise direction to the endpoint of the second semicircle trajectory. The starting point of the third semicircle is set as the second endpoint, the center position of the third semicircle is $P_{C3}(0, 0)$, and the radius is $R_3 = 4$. Then, the mobile platform needs to rotate 180° in the counterclockwise direction to the endpoint of the third semicircle trajectory. The starting point of the fourth semicircle is set as the third endpoint, the center position of the fourth semicircle is $P_{C4}(0, 0)$, and the radius is $R_4 = 4$ m.

Then, the mobile robot reaches the endpoint of the spiral trajectory by rotating 180° in the counterclockwise direction. The time of each semicircle is $t_{C1} = t_{C2} = t_{C3} = t_{C4} = 0.5$ s, the total time of the spiral trajectory is $t_s = 2$ s, the sampling time is $d_t = 0.001$ s, and the initial position and posture of the mobile platform is $(3, 0, \pi/2)$.

4.1.1. Performance Test of the Hybrid Controller without Initial Error

Assuming there is no initial error in the various trajectories, the typical iterative learning controller (17) directly controls the rotation speed of the two wheels to solve the trajectory tracking problem.

Situation 1: The control law of the corresponding gain matrix is described as follows:

$$L_1(k) = \frac{1}{R_{rs}} \begin{bmatrix} \cos(q) & \sin(q) & B_{rs} \\ \cos(q) & \sin(q) & -B_{rs} \end{bmatrix}; L_2(k) = 0; \quad L_3(k) = 0; \quad (45)$$

where $R_{rs} = R_c$ represents the actual radius of the wheels, and $B_{rs} = B_c$ represents the actual distance between the driving wheels.

The trajectory tracking process and the final result of the mobile platform after 100 iteration process are shown in Figure 3. By comparing the tracking results, we found the following:

- (1) The system can directly control two driving wheels to track the specified trajectories without initial error using the iterative learning controller.
- (2) Selecting the midpoint of the two driving wheels as the control variable can effectively achieve different trajectory tracking tasks.

4.1.2. Performance Test of Hybrid Controller with Initial Error

The above tasks were simulated without the initial error; however, the initial position of the mobile platform maybe not completely coincide with the starting of the given trajectory during actual movement. To test the novel-type controller in the initial error state, we effectively compared the previous control law and controller against the circular trajectory's initial error. We assumed the initial position and posture of the mobile platform are expressed as $(9, 0, \pi/2)$, which does not coincide with the initial position of the trajectory in this task. The specific controllers were as follows:

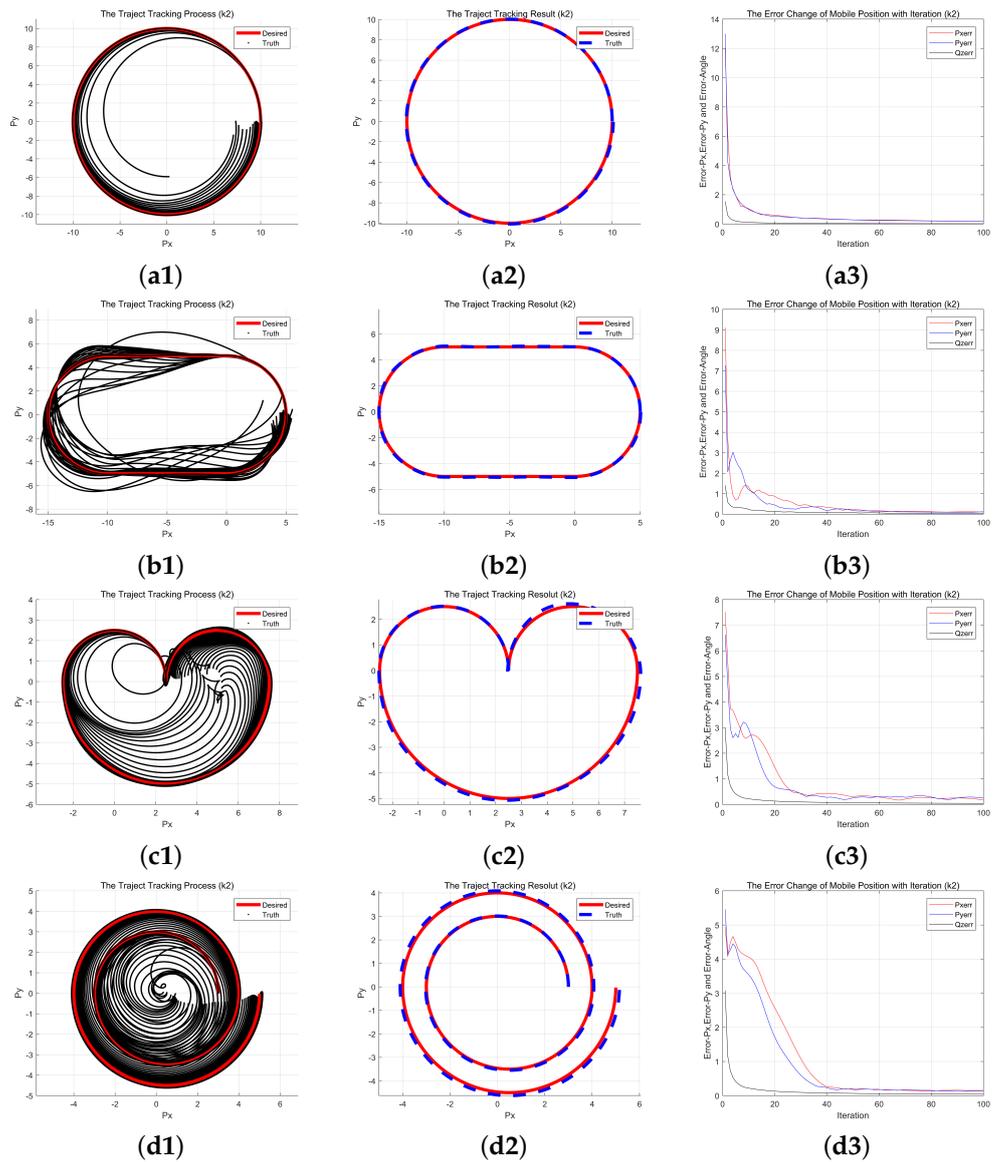


Figure 3. Typical trajectory tracking results of the mobile platform using the P–type iterative learning controller. (a1) Path change in single circular trajectory. (a2) Final path in single circular trajectory. (a3) Maximum tracking error change in single circular trajectory. (b1) Path change in mixed arc and line closed trajectory. (b2) Final path in mixed arc and line closed trajectory. (b3) Maximum tracking error change in mixed arc and line closed trajectory. (c1) Path change in cardioid–like trajectory. (c2) Final path in cardioid–like trajectory. (c3) Maximum tracking error change in cardioid–like trajectory. (d1) Path change in spiral–like trajectory. (d2) Final path in spiral–like trajectory. (d3) Maximum tracking error change in spiral–like trajectory.

Situation 2: There is an uncompensated control law with a speed constraint, which can be expressed explicitly as follows:

$$L_1^2(k) = 0.5 \cdot L_1(k); \quad L_2^2(k) = 0.25 \cdot L_1(k); \quad L_3^2(k) = 0.25 \cdot L_1(k); \quad (46)$$

Situation 3: The iterative learning controller (21) directly controls the rotation speed of the two wheels to solve the trajectory tracking problem, which can be expressed as follows:

$$u_{i+1}(k) = u_i(k) + L_1(k)e_i(k+1) + e^{-k \cdot d_t} \theta(k)L_1(k)e_i(0)$$

$$\theta(k) = \begin{cases} \frac{2}{d_k \cdot h} \left(1 - \frac{k}{h}\right), & (0 \leq k \leq h) \\ 0, & (h < k \leq n) \end{cases} \quad (47)$$

where $L_1^{s2}(k)$ is the gain of the controller, which can be expressed as $L_1^{s2}(k) = 0.5 \cdot L_1(k)$; the coefficient of h was chosen as 200; and the corresponding trajectory tracking effect was obtained, as shown in Figure 4.

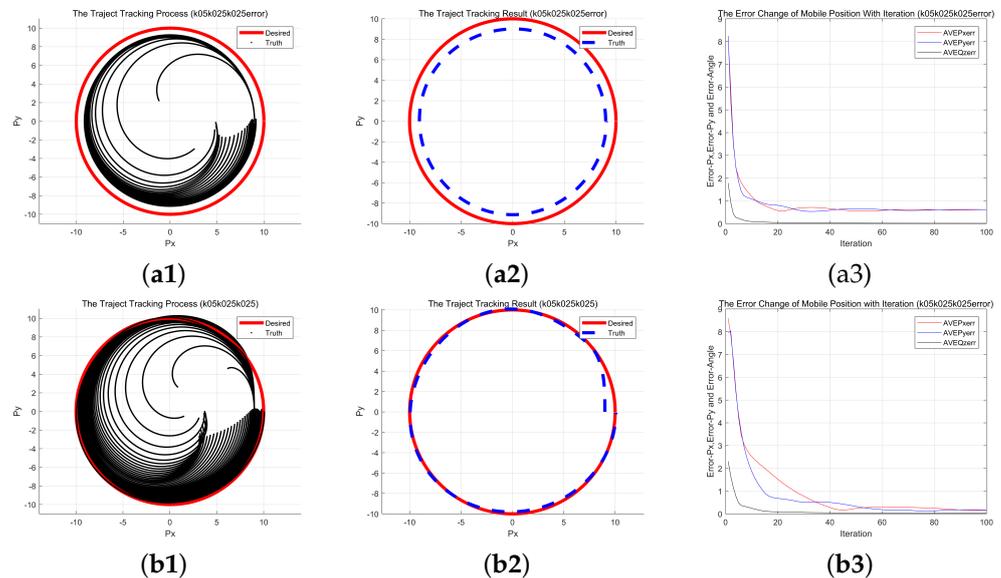


Figure 4. Circular trajectory tracking results of the mobile platform using the HILC controller with initial error. (a1) Path change tracking using HILC law (46). (a2) Final path tracking using HILC law (46). (a3) Average tracking error change using HILC law (46). (b1) Path change tracking using HILC law (47). (b2) Final path tracking using HILC law (47). (b3) Average tracking error change using HILC law (47).

To illustrate the adaptability of the novel-type controller, we also tested the novel-type controller on other trajectories with initial errors. For example, the initial position and posture of the robot in the combination trajectory of arc and straight task is defined as $(4, 0, \pi/2)$, the initial position and posture of the robot in the cardioid-like trajectory task is defined as $(2, 0, 14\pi/15)$, and the initial position and posture of the robot in the spiral-like trajectory task is defined as $(2.5, 0, 31\pi/30)$. Using the novel-type controller with a speed constraint, the following experimental results were obtained, as shown in Figure 5.

Comparing the above experimental results, we found the following:

- (1) Without the initial error compensation in Figure 4a1–a3, the control law (17) can achieve closed-loop motion, but the actual trajectory has a relatively fixed deviation from the given trajectory. Therefore, the previous controller cannot control the tracking of the various trajectories with the initial error.
- (2) With the initial error compensation in Figure 4a2,a3,b2,b3, the control law (21) can effectively achieve the circular trajectory tracking with initial error. Additionally, the controller with a velocity constraint must meet the actual physical limitations of the mobile platform.
- (3) According to various trajectory tracking results with initial error in Figure 5, the error compensation controller designed in this paper has good adaptability to various trajectories.

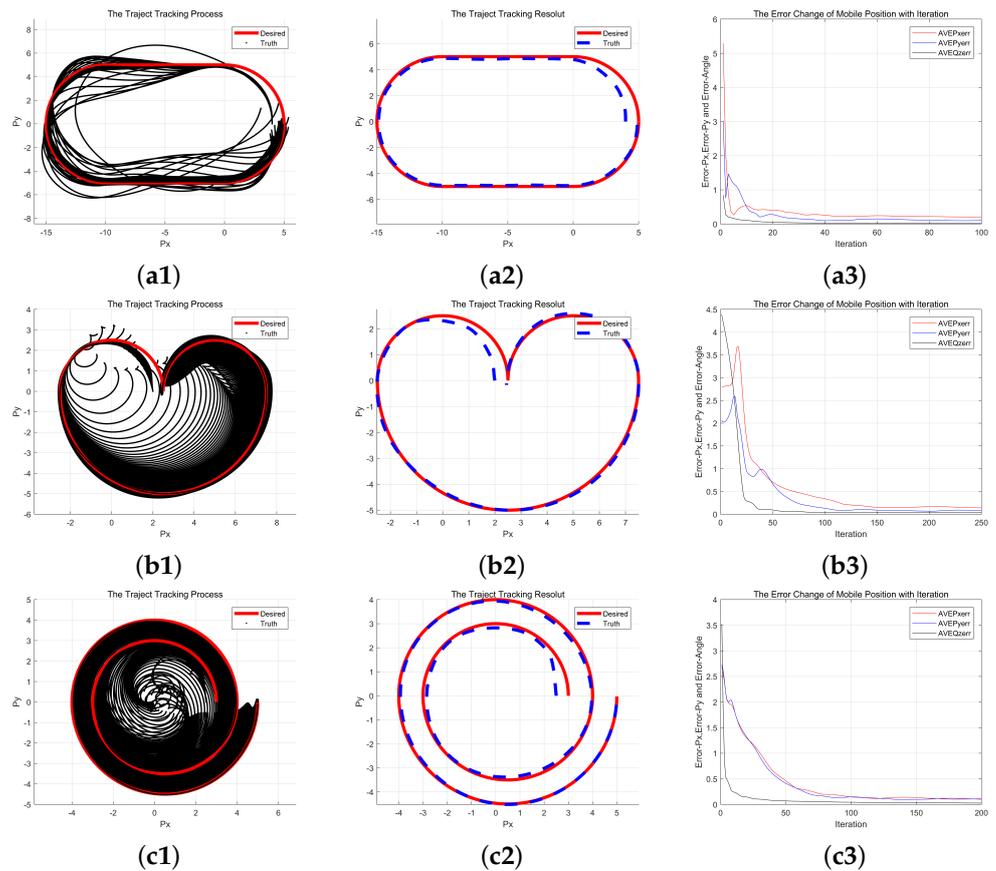


Figure 5. Other typical trajectories tracking results of mobile platform using the HILC law (47) with Initial error. (a1) Path change of mixed the arc and line closed trajectory tracking using HILC law (47). (a2) Final path of mixed the arc and line closed trajectory tracking using HILC law (47). (a3) Average error of the arc and line mixed closed trajectory tracking using HILC law (47). (b1) Path change of the cardioid-like trajectory tracking using HILC law (47). (b2) Final path of the cardioid-like trajectory tracking using HILC law (47). (b3) Average error of the cardioid-like trajectory tracking using HILC law (47). (c1) Path change of the spiral-like trajectory tracking using HILC law (47). (c2) Final path of the spiral-like trajectory tracking using HILC law (47). (c3) Average error of the spiral-like trajectory tracking using HILC law (47).

4.2. The Influence of Various Parameters

4.2.1. The Influence of the Control Coefficients and the Error Signals

To study the influence of the error signals and the control coefficients on the trajectory tracking problem, we selected the circular path as the research task for the trajectory tracking with the mobile platform. In the second task, the modified iterative learning control law was selected as the basic controller, and different error signals and the control coefficients were taken as variable parameters. The specific controller settings were as follows:

Situation 4: The $e_i(k + 1)$ only uses as the feedback error signal, and the control law (17) is selected as the basic controller. The corresponding gain parameters are described as follows:

$$\begin{aligned} L_1^{s4a}(k) &= 0.1 \cdot L_1(k); \\ L_1^{s4b}(k) &= 1.0 \cdot L_1(k); \\ L_1^{s4c}(k) &= 10 \cdot L_1(k); \end{aligned} \tag{48}$$

Situation 5: The $e_i(k + 1)$ and $e_{i+1}(k)$ are used as the feedback error signal, and the control law (17) is selected as the basic controller. The corresponding gain parameters are specifically expressed as follows:

$$\begin{aligned}
 L_1^{s5a}(k) &= 0.1 \cdot L_1(k); & L_2^{s5a}(k) &= 0.5 \cdot L_1(k); \\
 L_1^{s5b}(k) &= 0.1 \cdot L_1(k); & L_2^{s5b}(k) &= 1.0 \cdot L_1(k); \\
 L_1^{s5c}(k) &= 0.1 \cdot L_1(k); & L_2^{s5c}(k) &= 10 \cdot L_1(k);
 \end{aligned}
 \tag{49}$$

Situation 6: The feedback error signal includes $e_i(k + 1)$, $e_{i+1}(k)$, and $e_{i+1}(k - 1)$ and the control law (17) is selected as the basic controller. The corresponding gain parameters are specifically expressed as follows:

$$\begin{aligned}
 L_1^{s6a}(k) &= 0.1 \cdot L_1(k); & L_2^{s6a}(k) &= 1 \cdot L_1(k); & L_3^{s6a}(k) &= 0.1 \cdot L_1(k); \\
 L_1^{s6b}(k) &= 0.1 \cdot L_1(k); & L_2^{s6b}(k) &= 1 \cdot L_1(k); & L_3^{s6b}(k) &= 1.0 \cdot L_1(k); \\
 L_1^{s6c}(k) &= 0.1 \cdot L_1(k); & L_2^{s6c}(k) &= 1 \cdot L_1(k); & L_3^{s6c}(k) &= 20 \cdot L_1(k);
 \end{aligned}
 \tag{50}$$

By selecting different controller coefficients and different error parameters, we obtained different dynamic iterative trajectory graphics and the final trajectory result of the mobile platform with 100 iterations, which are shown in Figure 6. To determine the influence of the control coefficient and the error parameter, we compared the various dynamic iterative trajectory graphics, as shown in Figure 6.

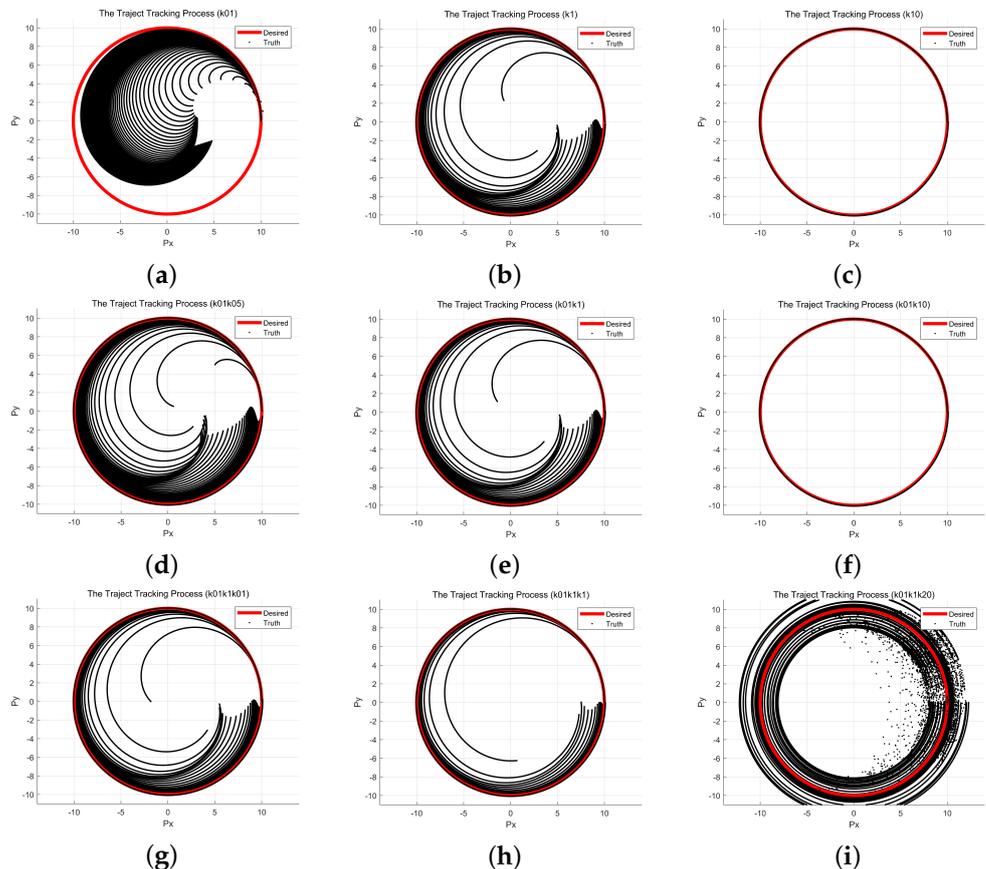


Figure 6. Circular trajectory tracking results of the mobile platform using the HILC controller with different parameters. (a) Using the HILC law with $k_1 = 0.1$. (b) Using the HILC law with $k_1 = 1$. (c) Using the HILC law with $k_1 = 10$. (d) Using the HILC law with $k_1 = 0.1$, $k_2 = 0.5$. (e) Using the HILC law with $k_1 = 0.1$, $k_2 = 1$. (f) Using the HILC law with $k_1 = 0.1$, $k_2 = 10$. (g) Using the HILC law with $k_1 = 0.1$, $k_2 = 1$, $k_3 = 0.1$. (h) Using the HILC law with $k_1 = 0.1$, $k_2 = 1$, $k_3 = 1$. (i) Using the HILC law with $k_1 = 0.1$, $k_2 = 1$, $k_3 = 20$.

(1) Comparing a, b, and c in Figure 6 with different controller coefficients, the number of the iterations required for trajectory tracking can be continuously reduced by increasing the control coefficient within a reasonable range (0.1 to 10). However, if the control

coefficient is too small, the controller cannot effectively track the trajectory in a shorter time; if the value of the control coefficient is too large (>20), the controller cannot effectively track the given trajectory caused by increasing the error. Therefore, the system may be unstable.

(2) Comparing a, d, e, g, and h in Figure 6 with different error parameters, if the error at $(k + 1)$ time in the current iteration remains unchanged, the other error parameters can compensate for the error parameter to improve the convergence speed of the new-type controller. The tracking performance may be improved by increasing the other error parameter.

(3) Comparing a, d, e, g, and h in Figure 6 with different error parameters and different controller coefficients, increasing the control coefficients of each error parameter can reduce the number of iterations required for the trajectory tracking by the mobile platform; however, the control coefficients of each error parameter still affect the stability of the system.

4.2.2. The Influence of the Physical Speed Limitation

The rotation speed of the driving wheel has physical limitations. However, previous studies did not analyze the influence of the speed constraint on the actual trajectory tracking process. Therefore, we deliberately added the speed constraint to the novel-type iterative learning controller to study the impact of the physical speed limitation on the trajectory tracking process. The specific controllers were as follows:

Situation 7: Hybrid iterative learning controller without speed limitation:

$$L_1^{s7}(k) = 0.1 \cdot L_1(k); \quad L_2^{s7}(k) = 0.25 \cdot L_1(k); \quad L_3^{s7}(k) = 0.25 \cdot L_1(k); \quad (51)$$

Situation 8: Hybrid iterative learning controller with speed limitation:

$$\begin{cases} L_1^{s8}(k) = 0.1 \cdot L_1(k); & L_2^{s8}(k) = 0.25 \cdot L_1(k); & L_3^{s8}(k) = 0.25 \cdot L_1(k); \\ u_{cr}(k) < u_{\max}, u_{cl}(k) < u_{\max} \end{cases} \quad (52)$$

Situation 9: Hybrid iterative learning controller without speed limitation:

$$L_1^{s9}(k) = 1 \cdot L_1(k); \quad L_2^{s9}(k) = 0.25 \cdot L_1(k); \quad L_3^{s9}(k) = 0.25 \cdot L_1(k); \quad (53)$$

Situation 10: Hybrid iterative learning controller with speed limitation:

$$\begin{cases} L_1^{s10}(k) = 0.5 \cdot L_1(k); & L_2^{s10}(k) = 0.25 \cdot L_1(k); & L_3^{s10}(k) = 0.25 \cdot L_1(k) \\ u_{cr}(k) < u_{\max}, u_{cl}(k) < u_{\max} \end{cases} \quad (54)$$

We used the above controller law to describe the tracking result and error of the specified trajectory. Comparing the above controllers, we found the following:

- (1) Comparing Figure 7a2,c2, the speed solved by the single controller does not meet the physical limitations; however, adding the speed constraint to the controller can meet the physical limitations of the mobile platform.
- (2) Comparing Figure 7a3,b3 or Figure 7c3,d3, the controller with a velocity limitation can reduce the number of system iterations required and improve the convergence speed of the system.
- (3) Comparing Figure 7a2,a3,b2,b3 or Figure 7c2,c3,d2,d3, it can be found that the control signal is not smooth with the speed constraint since the control speed at the next moment can be reached considering the speed limitation. However, the velocity limitation can reduce the tracking error to significantly enhance the performance of the controller.

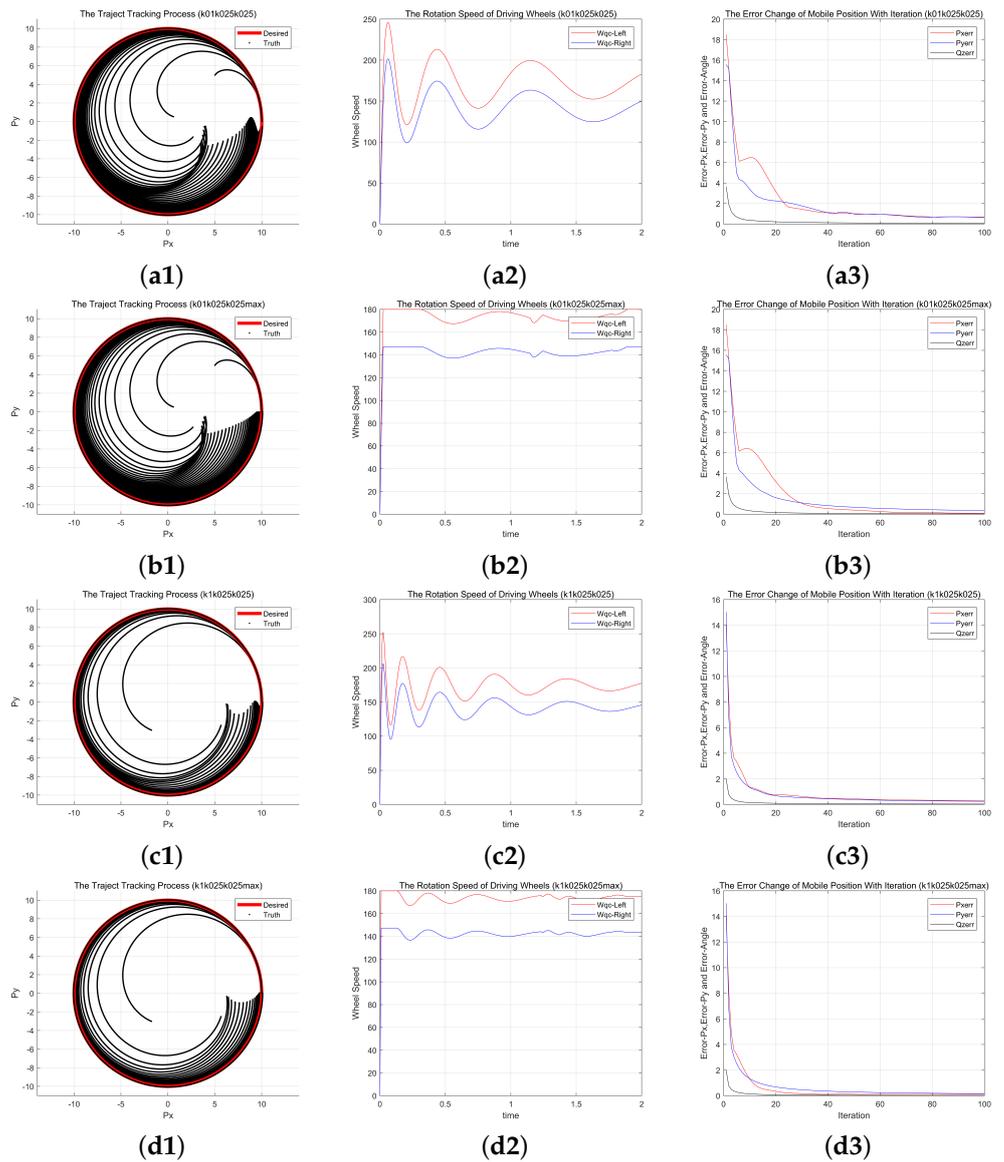


Figure 7. Circular trajectory tracking results of the mobile platform using the HILC controller with a speed limit. (a1) Path change tracking using HILC law (51). (a2) Final speed of the two-wheels tracking using HILC law (51). (a3) Maximum tracking error change tracking using HILC law (51). (b1) Path change tracking using HILC law (52). (b2) Final speed of the two-wheels tracking using HILC law (52). (b3) Maximum tracking error change tracking using HILC law (52). (c1) Path change tracking using HILC law (53). (c2) Final speed of the two-wheels tracking using HILC law (53). (c3) Maximum tracking error change tracking using ILC law (53). (d1) Path change in tracking using HILC law (54). (d2) Final speed of the two-wheels tracking using HILC law (54). (d3) Maximum tracking error change tracking using HILC law (54).

4.2.3. The Influence of Basic Structure Parameters

To study the influence of the actual structure parameters of the mobile platform on the controller, we compared the controller with the system parameters to that without the system parameters. The specific parameters were as follows:

Situation 11: The novel hybrid controller without error in the parameters of the mobile platform:

$$L_1^{s11}(k) = 0.10 \cdot L_1(k); \quad L_2^{s11}(k) = 0.25 \cdot L_1(k); \quad L_3^{s11}(k) = 0.25 \cdot L_1(k); \quad (55)$$

Situation 12: The novel hybrid controller with error in the parameters of the mobile platform:

$$L_1^{s12}(k) = \begin{bmatrix} \cos(q) & \sin(q) & 1 \\ \cos(q) & \sin(q) & -1 \end{bmatrix}; \tag{56}$$

$$L_2^{s12}(k) = 0.5 \cdot L_1^{s12}(k); \quad L_3^{s12}(k) = 0.5 \cdot L_1^{s12}(k);$$

Situation 13: The other controller without the control matrix parameters of mobile platforms:

$$L_1^{s13}(k) = 0.1 \begin{bmatrix} 0.5 & 0.2 & B_{rs}/R_{rs} \\ \cos(q) & \sin(q) & -B_{rs}/R_{rs} \end{bmatrix}; \tag{57}$$

$$L_2^{s13}(k) = 2.5 \cdot L_1^{s13}(k); \quad L_3^{s13}(k) = 2.5 \cdot L_1^{s13}(k);$$

By selecting different Iterative learning controllers, we obtained the different dynamic iterative trajectory graphics and the final trajectory result of the mobile platform, as shown in Figure 8.

(1) Comparing Figure 8a1–a3,b1–b3, the controller without the basic system parameters can still track the circular trajectory, which proves that the iterative controller designed in this article does not depend on the actual parameters.

(2) Comparing Figure 8b1–b3,c1–c3, the designed controller cannot effectively track the given trajectory when the gain matrix $L_1(k)$ does not conform to the movement characteristics of mobile platforms,.

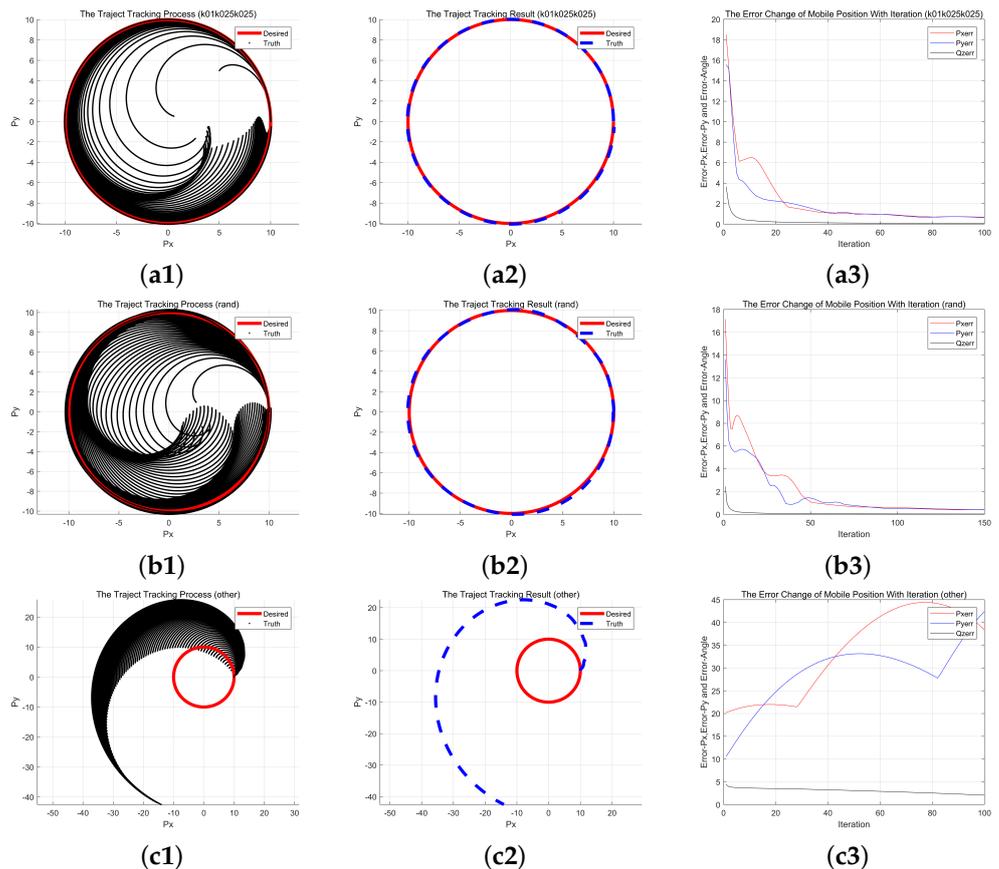


Figure 8. Circular trajectory tracking results of the mobile platform using the HILC controller with different structure parameters. (a1) Path change tracking using HILC law (55). (a2) Final path tracking using HILC law (55). (a3) Maximum tracking error change using HILC law (55). (b1) Path change tracking using HILC law (56). (b2) Final path tracking using HILC law (56). (b3) Maximum tracking error change using HILC law (56). (c1) Path change tracking using HILC law (57). (c2) Final path tracking using HILC law (57). (c3) Maximum tracking error change using HILC law (57).

5. Conclusions

In this study, a new kinematic model of the mobile platform was designed to achieve trajectory tracking by directly controlling the rotation speed of the driving wheels, and the model considers the actual structure and physical constraints of the mobile robot. Different from the typical iterative learning controller of mobile robots, the hybrid controller can directly select the rotation speed of the wheels as the control signal and consider the actual physical constraints of the system.

We theoretically proved the basic performance of the controller, and designed various simulation experiments to analyze the influence of different parameters on the trajectory tracking task, including a speed constraint, the control coefficient, the different error parameters, basic physical parameters, and initial error. Experiments showed that the reference trajectories with or without the initial error can be tracked with the designed hybrid controller, and the system control performance was improved by increasing the other types of error signal in the trajectory tracking task without initial error.

In the future, we aim to develop an iterative controller for mobile platforms to achieve dynamic motion control in various complex environments. In the meantime, we will study the iterative learning method to achieve path planning for mobile platforms with physical limitations in complex environments.

Author Contributions: Conceptualization, L.Q. and Q.L.; methodology, L.Q.; software, L.X. and Q.L.; validation, L.Q., L.X., J.J. and Q.L.; formal analysis, L.Q.; investigation, L.X. and Q.L.; resources, L.Q.; data curation, L.Q., J.J. and M.L.; writing—original draft preparation, L.Q.; writing—review and editing, L.X. and Q.L.; visualization, L.Q., J.J. and M.L.; supervision, L.X. and Q.L.; project administration, L.X. and Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Guo, P.; Liang, Z.; Wang, X.; Zheng, M. Adaptive trajectory tracking of wheeled mobile robot based on fixed-time convergence with uncalibrated camera parameters. *ISA Trans.* **2020**, *99*, 1–8. [[CrossRef](#)] [[PubMed](#)]
2. Stančić, I.; Musić, J.; Grujić, T. Gesture recognition system for real-time mobile robot control based on inertial sensors and motion strings. *Eng. Appl. Artif. Intell.* **2017**, *66*, 33–48. [[CrossRef](#)]
3. Zein, Y.; Darwiche, M.; Mokhiamar, O. GPS tracking system for autonomous vehicles. *Alexandria Eng. J.* **2018**, *57*, 3127–3137. [[CrossRef](#)]
4. Chan, R.P.M.; Stol, K.A.; Halkyard, C.R. Review of modelling and control of two-wheeled robots. *Annu. Rev. Control.* **2013**, *37*, 89–103. [[CrossRef](#)]
5. Luo, S.; Wu, S.; Liu, Z.; Guan, H. Wheeled Mobile Robot RBFNN Dynamic Surface Control Based on Disturbance Observer. *ISRN Appl. Math.* **2014**, *2014*, 634936. [[CrossRef](#)]
6. Scaglia, G.; Rosales, A.; Quintero, L.; Mut, V.; Agarwal, R. A linear-interpolation-based controller design for trajectory tracking of mobile robots. *Control Eng. Pract.* **2010**, *18*, 318–329. [[CrossRef](#)]
7. Shojaei, K.; Shahri, A.M.; Tarakameh, A. Adaptive feedback linearizing control of nonholonomic wheeled mobile robots in presence of parametric and nonparametric uncertainties. *Robot. Comput. Integr. Manuf.* **2011**, *27*, 194–204. [[CrossRef](#)]
8. Wu, D.; Cheng, Y.; Du, H.; Zhu, W.; Zhu, M. Finite-time output feedback tracking control for a nonholonomic wheeled mobile robot. *Aerosp. Sci. Technol.* **2018**, *78*, 574–579. [[CrossRef](#)]
9. Alakshendra, V.; Chiddarwar, S.S. Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dyn.* **2017**, *87*, 2147–2169. [[CrossRef](#)]
10. Begnini, M.; Bertol, D.W.; Martins, N.A. A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot: Simulation and experimental results. *Control Eng. Pract.* **2017**, *64*, 27–43. [[CrossRef](#)]
11. Fierro, R.; Lewis, F.L. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. *J. Robot. Syst.* **1997**, *14*, 149–163. [[CrossRef](#)]
12. He, N.; Qi, L.; Li, R.; Liu, Y. Design of a Model Predictive Trajectory Tracking Controller for Mobile Robot Based on the Event-Triggering Mechanism. *Math. Probl. Eng.* **2021**, *2021*, 5573467. [[CrossRef](#)]

13. Liu, K.; Gong, J.; Chen, S.; Zhang, Y.; Chen, H. Model predictive stabilization control of high-speed autonomous ground vehicles considering the effect of road topography. *Appl. Sci.* **2018**, *8*, 5. [[CrossRef](#)]
14. Matraji, I.; Al-Durra, A.; Haryono, A.; Al-Wahedi, K.; Abou-Khousa, M. Trajectory tracking control of Skid-Steered Mobile Robot based on adaptive Second Order Sliding Mode Control. *Control Eng. Pract.* **2017**, *11*, 167–176. [[CrossRef](#)]
15. Goswami, N.K.; Padhy, P.K. Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance. *Comput. Electr. Eng.* **2018**, *72*, 307–323. [[CrossRef](#)]
16. Aouf, A.; Boussaid, L.; Sakly, A. TLBO-Based Adaptive Neurofuzzy Controller for Mobile Robot Navigation in a Strange Environment. *Comput. Intell. Neurosci.* **2018**, *2018*, 3145436. [[CrossRef](#)] [[PubMed](#)]
17. Omrane, H.; Masmoudi, M.S.; Masmoudi, M. Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation. *Comput. Intell. Neurosci.* **2016**, *2016*, 9548482. [[CrossRef](#)]
18. Dixon, W.E.; Dawson, D.M.; Zergeroglu, E.; Behal, A. Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* **2001**, *31*, 341–352. [[CrossRef](#)]
19. Xin, L.; Wang, Q.; She, J.; Li, Y. Robust adaptive tracking control of wheeled mobile robot. *Rob. Auton. Syst.* **2016**, *78*, 36–48. [[CrossRef](#)]
20. Cui, M.; Sun, D.; Liu, W.; Zhao, M.; Liao, X. Adaptive tracking and obstacle avoidance control for mobile robots with unknown sliding. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 1–14. [[CrossRef](#)]
21. Hoang, N.B.; Kang, H.J. Neural network-based adaptive tracking control of mobile robots in the presence of wheel slip and external disturbance force. *Neurocomputing* **2016**, *188*, 12–22. [[CrossRef](#)]
22. Li, S.; Ding, L.; Gao, H.; Chen, C.; Liu, Z.; Deng, Z. Adaptive neural network tracking control-based reinforcement learning for wheeled mobile robots with skidding and slipping. *Neurocomputing* **2018**, *283*, 20–30. [[CrossRef](#)]
23. Boukens, M.; Boukabou, A.; Chadli, M. Robust adaptive neural network-based trajectory tracking control approach for nonholonomic electrically driven mobile robots. *Rob. Auton. Syst.* **2017**, *92*, 30–40. [[CrossRef](#)]
24. Boudjedir, C.E.; Bouri, M.; Boukhetala, D. Iterative learning control for trajectory tracking of a parallel Delta robot. *At-Automatisierungstechnik* **2019**, *67*, 145–156. [[CrossRef](#)]
25. Dong, J.; He, B.; Zhang, C.; Li, G. Open-Closed-Loop PD Iterative Learning Control with a Variable Forgetting Factor for a Two-Wheeled Self-Balancing Mobile Robot. *Complexity* **2019**, *2019*, 5705126. [[CrossRef](#)]
26. Lu, X.; Fei, J. Velocity Tracking Control of Wheeled Mobile Robots by Iterative Learning Control. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1–10. [[CrossRef](#)]
27. Son, T.D.; Ahn, H.S.; Moore, K.L. Iterative learning control in optimal tracking problems with specified data points. *Automatica* **2013**, *49*, 1465–1472. [[CrossRef](#)]
28. Li, G. Adaptive iterative learning control for nonlinear systems with unknown control direction. *Nanjing Li Gong Daxue Xuebao/J. Nanjing Univ. Sci. Technol.* **2015**, *39*, 2942–2950. [[CrossRef](#)]
29. Bouakrif, F.; Zasadzinski, M. High order iterative learning control to solve the trajectory tracking problem for robot manipulators using Lyapunov theory. *Trans. Inst. Meas. Control* **2018**, *40*, 4105–4114. [[CrossRef](#)]
30. Bouakrif, F.; Boukhetala, D.; Boudjema, F. Velocity observer-based iterative learning control for robot manipulators. *Int. J. Syst. Sci.* **2013**, *44*, 214–222. [[CrossRef](#)]
31. Ahn, H.S.; Chen, Y.Q.; Moore, K.L. Iterative learning control: Brief survey and categorization. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 1099–1121. [[CrossRef](#)]
32. Wang, Y.; Gao, F.; Doyle, F.J. Survey on iterative learning control, repetitive control, and run-to-run control. *J. Process Control* **2009**, *19*, 1589–1600. [[CrossRef](#)]
33. Kang, M.K.; Lee, J.S.; Han, K.L. Kinematic path tracking of mobile robot using iterative learning control. *J. Robot. Syst.* **2005**, *22*, 111–121. [[CrossRef](#)]
34. Han, K.L.; Lee, J.S. Iterative path tracking of an omni-directional mobile robot. *Adv. Robot.* **2011**, *25*, 1817–1838. [[CrossRef](#)]
35. 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*, 1393–1414. [[CrossRef](#)]
36. Wang, H.; Dong, J.; Wang, Y. Discrete PID-Type Iterative Learning Control for Mobile Robot. *J. Control Sci. Eng.* **2016**, *2016*, 2320746. [[CrossRef](#)]
37. Wang, H.; Dong, J.; Wang, Y. Research on Open-Closed-Loop Iterative Learning Control with Variable Forgetting Factor of Mobile Robots. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 6452179. [[CrossRef](#)]
38. 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*. [[CrossRef](#)]