

Received 29 July 2023, accepted 26 August 2023, date of publication 5 September 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312174

RESEARCH ARTICLE

Formation Control of Multiple Mobile Robots Based on Iterative Learning Distributed Model Predictive Control

WEI SHANG^{1,2,3}, MENG LIU¹, DAODE ZHANG¹, AND HANZONG ZHU¹

¹School of Mechanical Engineering, Hubei University of Technology, Wuhan, Hubei 430068, China

²Ningbo Cixing Co., Ltd., Ningbo 315000, China

³Ningbo University, Ningbo 315211, China

Corresponding author: Meng Liu (1510523498@qq.com)

This work was supported in part by the Collaborative Innovation Center of Intelligent Green Manufacturing Technology and Equipment, Shandong, under Grant IGSD-2020-007; in part by the Natural Science Foundation under Grant 5207053225; and in part by the Hubei Provincial Department of Science and Technology under Grant 2020BIB012.

ABSTRACT This paper proposes an iterative learning distributed model predictive control (ILDMPC) to control the formation of multiple mobile robots under uncertainty. Specifically, we design a general performance index constructed from the system's state variables and coupling parameters to replace the traditional cost function, promoting the system's control efficiency and the ability to seek the optimal solution. Furthermore, when dealing with and calculating the robot information, such as state variables and coupling parameters, information from the previous iteration is employed to construct closed-loop constraints in the optimization problem. Then, the results of the next iteration are calculated by learning the previous optimization trajectory and improving the overall system performance. Under the closed-loop constraints of the optimization problem, we analyze our system's feasible solution and iterative performance, demonstrating its effectiveness through several simulation experiments.

INDEX TERMS Iterative learning distributed MPC, general performance index, closed-loop constraints, model uncertainty.

I. INTRODUCTION

The formation control of multi-mobile robots has been widely used in military, production, and service fields [1], [2], [3], [4]. Compared with a single mobile robot, multiple mobile robots promote the system's task completion efficiency, fault tolerance, and robustness due to their cooperation. Considering their formation information exchange strategy, the distributed control strategy stands out due to its strong adaptability, expansibility, and fault tolerance. However, solving the conformance protocol problem in distributed formation control is challenging.

To date, researchers have proposed the leader-follower method [5], [6], the behavior-based method [7], [8], and the virtual structure method [9], [10] to overcome the conformance protocol problem. Nevertheless, with the introduction

of model predictive control (MPC), the existing methods for this problem have been expanded because MPC overcomes the constraints by adopting optimization theory and the rolling optimization strategy. Therefore, researchers applied MPC to the distributed control and suggested the distributed model predictive control (DMPC) method for the distributed formation control [11], [12], which discards the strong coupling relationship among the mobile robots. However, multi-mobile robot setups utilizing DMPC suffer from substantial uncertainty. It should be noted that when DMPC is used in formation control, it requires an accurate model, which in practice is almost impossible. Hence, when studying the DMPC, the model of the multi-mobile robot system is often defined precisely, and the model's uncertainty is ignored. Nevertheless, this strategy imposes a too-optimistic DMPC prediction process, which forces the robot information prediction result to deviate from the actual effect and interfere with the distributed system [13], [14]. Therefore, resolving

The associate editor coordinating the review of this manuscript and approving it for publication was Nasim Ullah¹.

the uncertainty in a distributed system is still challenging. Moreover, the coupling relationship in traditional DMPC still suffers from various problems. For instance, if uncertainty is considered, the impact of imprecise coupling parameters on a single repeating DMPC system is not allowed to change, degrading the performance of the distributed formation system.

Although few studies on DMPC formation control settle the uncertainty, existing studies focus on improving the MPC, i.e., the core of DMPC, and combining robust control [15], [16] and adaptive control [17], [18] methods to manage the influence of uncertainty. While the two approaches tend to rely on cost functions and model structures, with the development of intelligence, the Iterative Learning Control (ILC) algorithm has gained significance due to its ability to overcome uncertainty. Indeed, ILC can control the non-linear dynamic system with high uncertainty by employing a simple algorithm within a given time range and tracking the given desired trajectory with high precision [19], [20]. Considering the advantages of ILC, researchers applied ILC in MPC and obtained appealing results. Indeed, adding ILC [21] constrains the optimization problem's objective as a general performance index. Hence, only a simple quadratic cost term reduces the burden of seeking the optimal solution. By designing an iterative learning MPC [22] strategy, the system iteratively learns how to deal with the unknown polyhedron state constraints. In this strategy, the estimated constraint set gets ascension by iteration, promoting the system's ability to resist uncertainty. In [23], the authors considered real-time trajectory tracking of a pressure-point nano-positioning, where the Iterative Learning MPC scheme accurately tracked the trajectory and overcame the measurement noise and model uncertainty. In [29], the authors developed a global fast terminal attractor-based flight trajectory tracking control that compensates matched uncertainties.

Besides, RMPC combines the advantages of robust control and model predictive control, which affords the optimal control quantity for changing state parameters and solves the poor robustness problem caused by model uncertainty and system disturbance in traditional MPC algorithms [30]. Robust model predictive control is generally divided into two important solution methods, the min-max and the tube-based. The basic idea of minimax predictive control is similar to the traditional MPC. However, it differs regarding the optimal solution of the objective function of robust model predictive control by minimizing the maximum value, i.e., the min-max method obtains the optimal control law. The core idea of the min-max method is to consider that if the system can maintain a stable state to achieve control of the controlled object under the "worst" situation, then the system can still ensure smooth operation under normal conditions or unexpected situations [31]. According to the correlation between the control sequence and the system state, the min-max RMPC can be divided into two types: open-loop and closed-loop, which contain the following four types [32]:

- (1) Open-loop min-max robust model predictive control
- (2) Constant feedback min-max robust model predictive control
- (3) Dynamic feedback min-max robust model predictive control
- (4) Dual-mode enumeration min-max robust model predictive control

The above studies indicate that iterative learning MPC effectively solves the shortcomings of the MPC-based methods. Specifically, a control system integrating iterative learning and MPC is robust, even when influenced by various bounded disturbances and uncertainties, the system's iterative trajectory converges to the desired trajectory. However, in the distributed MPC field, each mobile robot has its bounded disturbances and uncertainties, and therefore it is challenging to design the algorithm reasonably while considering all mobile robots simultaneously.

Considering calculating the robot's information, ILC is quite efficient. Indeed, in the Ventilator Flow Tracking System [24], the ILC module collects previous flow information and evaluates the information for errors through iterative learning to correct the flow to the expected values. By proposing a learning model involving a predictive controller for iterative tasks [25], the non-reference controller enhances its performance by learning from previous iterations. Given the velocity fluctuation of the power system of semi-submersible ships [26], the periodic torque ripple of the propulsion motor is effectively reduced by using the error trend and previous error information in the ILC algorithm. According to the above research, ILC exploits more information than the single-repetition system, and therefore the information of the previous iteration can be integrated into the problem formula of the next iteration to increase the system's performance [27]. Although ILC is widely used, it should be improved further to enhance the robustness of DMPC. Specifically, to successfully implement the designed algorithm, it is necessary to analyze the robot information reasonably, construct the coupling constraint, and apply this constraint to the iterative learning DMPC.

An alternative method considers a neural network-based method that combines a fast integral terminal sliding mode control (FIT-SMC), a robust exact differentiator (RED) observer, and a feedforward neural network (FFNN) estimator [28]. This method was applied on a multi-DoF anthropomorphic manipulator demonstrating low overshoot and settling time for all joints.

This paper considers a distributed multi-robot system where we aim to control the agents' formation during their motion. Solving such a control problem is important as multi-robot systems are extensively used in numerous domains, and thus controlling their formation is important. Nevertheless, current methods cannot handle the related uncertainties despite their importance. Besides, the existing robust methods that combine ILC and MPC are constrained to a single platform rather than a multi-platform configuration

operating in a distributed organization, which is the focus of this paper. To our knowledge, extending the ILC and MPC to optimize the distributed multi-robot formation control has not been investigated yet.

Hence, spurred by the proven capabilities of the ILC and MPC methods and the poor performance of the existing DMPC schemes, this paper proposes ILDMPC. Specifically, this paper's contributions are as follows:

- 1) Developing ILDMPC, which combines ILC and MPC and extends their application for a distributed multi-robotic formation control.
- 2) Overcoming the deficiencies of current DMPC schemes, such as uncertainty and accurate system model requirements, by exploiting and extending the ILC strategy into the distributed multi-robotic formation control domain.
- 3) Transforming the objective function of the optimization problem into a general performance index by utilizing the total relative position state error and synchronization path parameter.
- 4) Exploiting the information from the previous iteration to construct closed-loop constraints. This optimizes the results of the next iteration by learning the previous iteration's decoupling results and optimization trajectory. Therefore, as the number of iterations increases, the distributed system's convergence curve improves, and the convergence effect is optimized.

The remainder of this paper is organized as follows. Section II describes the control problem. Section III introduces the proposed iterative learning distributed model predictive control (ILDMPC), proves the recursive feasibility, and demonstrates that the iterative performance is improved as the number of iterations increases under close-loop constraints. Section IV presents two simulation examples to illustrate our algorithm's effectiveness, and finally, Section V concludes this work.

II. PROBLEM FORMATION

This study considers a finite number of mobile robots under distributed formation control. A finite index is set to mark each mobile robot node as $1, 2, \dots, n$, and the relationship between mobile robot i and j is represented by a weighted adjacency matrix $\Lambda = [a_{ij}]$. If the i -th mobile robot can obtain the information from the j -th robot, then $a_{ij} = a_{ji} = 1$. Otherwise $a_{ij} = a_{ji} = 0$. Next, we analyze the relationship between the two mobile robots (Section II-A) and obtain the system of mobile robot i by summing the variables (Section II-B).

A. TWO MOBILE ROBOTS

Let mobile robots i and j have the position states $p_i = [x_{pi} \ y_{pi} \ \theta_{pi}]^T$ and $p_j = [x_{pj} \ y_{pj} \ \theta_{pj}]^T$ on their path Γ_i and Γ_j , where (x_p, y_p) and θ_p are the robot's coordinates and orientation, respectively. The desired relative position state difference of mobile robot j relative to mobile robot i is

represented as $p_{ij} = [x_{ij}^r \ y_{ij}^r \ \theta_{ij}^r]$, where (x_{ij}^r, y_{ij}^r) and θ_{ij}^r are the desired coordinates and orientation difference. The linear and angular velocity of the j -th mobile robot is expressed as $[v_j \ \omega_j] = [v_i - v_{ij}^r \ \omega_i - \omega_{ij}^r]$, where v_i and ω_i represent the linear and angular velocity of the i -th mobile robot, which can be obtained from the sensor. v_{ij}^r and ω_{ij}^r are the desired relative linear and angular velocity differences between the i -th and j -th mobile robots, respectively. Then the kinematic models of these robots are established as follows:

$$\begin{bmatrix} \dot{x}_{pi} \\ \dot{y}_{pi} \\ \dot{\theta}_{pi} \end{bmatrix} = \begin{bmatrix} \cos \theta_{pi} & 0 \\ \sin \theta_{pi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad (1)$$

$$\begin{bmatrix} \dot{x}_{pj} \\ \dot{y}_{pj} \\ \dot{\theta}_{pj} \end{bmatrix} = \begin{bmatrix} \cos \theta_{pj} & 0 \\ \sin \theta_{pj} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i - v_{ij}^r \\ \omega_i - \omega_{ij}^r \end{bmatrix}. \quad (2)$$

Considering the desired relative position state difference between the i -th and j -th mobile robot, the relative position state error can be expressed as:

$$\begin{aligned} \begin{bmatrix} x_{e,ij} \\ y_{e,ij} \\ \theta_{e,ij} \end{bmatrix} &= \begin{bmatrix} x_{ij} - x_{ij}^r \\ y_{ij} - y_{ij}^r \\ \theta_{ij} - \theta_{ij}^r \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta_{pi} & \sin \theta_{pi} & 0 \\ -\sin \theta_{pi} & \cos \theta_{pi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{pi} - x_{pj} - x_{ij}^r \\ y_{pi} - y_{pj} - y_{ij}^r \\ \theta_{pi} - \theta_{pj} - \theta_{ij}^r \end{bmatrix}. \end{aligned} \quad (3)$$

Taking the derivative on both sides of (3) provides:

$$\begin{cases} \dot{x}_{e,ij} = (\omega_i - \omega_{ij}^r) (y_{ij} - y_{ij}^r) - v_i \\ \quad + (v_i - v_{ij}^r) \cos (\theta_{ij} - \theta_{ij}^r) \\ \dot{y}_{e,ij} = -\omega_i (x_{ij} - x_{ij}^r) + (v_i - v_{ij}^r) \sin (\theta_{ij} - \theta_{ij}^r) \\ \dot{\theta}_{e,ij} = \omega_i - \omega_j - \omega_{ij}^r \end{cases} \quad (4)$$

Besides, the path-tracking problem of a mobile robot can be described as follows. In the initial position state, an appropriate bounded control input $U_{e,ij} = f(v_i - v_{ij}^r, \omega_i - \omega_{ij}^r)$ is sought to make the relative position state error $[x_{e,ij} \ y_{e,ij} \ \theta_{e,ij}]^T$ convergent and bounded. $[x_{e,ij} \ y_{e,ij} \ \theta_{e,ij}]^T$ should satisfy:

$$\lim_{t \rightarrow \infty} [x_{e,ij} \ y_{e,ij} \ \theta_{e,ij}]^T = 0.$$

Let the relative position state error be $X_{e,ij} = [x_{e,ij} \ y_{e,ij} \ \theta_{e,ij}]^T$, and the control input be $U_{e,ij} = [-v_i + (v_i - v_{ij}^r) \cos (\theta_{ij} - \theta_{ij}^r) \ \omega_i - \omega_j - \omega_{ij}^r]^T$. The linearization equation of the system between the i -th and j -th mobile robot is obtained from (4) as:

$$\dot{X}_{e,ij} = A_{0i} X_{e,ij} + B_{0i} U_{e,ij}, \quad (5)$$

$$\text{where } A_{0i} = \begin{bmatrix} 0 & (\omega_i - \omega_{ij}^r) & 0 \\ -\omega_i & 0 & (v_i - v_{ij}^r) \\ 0 & 0 & 0 \end{bmatrix}, B_{0i} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

B. THE SYSTEM OF MOBILE ROBOTS I

Assumption 1: The relationship between any mobile robot and robot i can be described as presented in Section II-A. Moreover, considering all mobile robots, the system of mobile robot i is constructed by accumulating the variables. By defining the total relative position state error as $X_i = \sum_{j=1, j \neq i}^n a_{ij} X_{e,ij}$, the total control input is $U_i = \sum_{j=1, j \neq i}^n a_{ij} U_{e,ij}$. Therefore, when ignoring the errors, the linearization equation of the system of the i -th mobile robot is:

$$\dot{X}_i = A_{0i} X_i + B_{0i} U_i. \quad (6)$$

Let $X_i = [x_i \ y_i \ \theta_i]^T$, where (x_i, y_i) and θ_i are the total coordinates errors and total orientation errors. For the system of mobile robot i , its task can be described as follows. Considering the uncertainty effect, the actual increment will be different from the ideal value due to the error of the state, and the non-linear model of the system affected by uncertainty is described in the following way.

$$\dot{X}_i = \sigma_i (A_{0i} X_i + B_{0i} U_i)$$

An appropriate bounded control input is sought in the initial position state to make the total relative position state error $X_i = [x_i \ y_i \ \theta_i]^T$ convergent and bounded. Therefore, $X_i = [x_i \ y_i \ \theta_i]^T$ should satisfy the following:

$$\lim_{t \rightarrow \infty} X_i = [x_i \ y_i \ \theta_i]^T = 0.$$

Through the zero-order hold, system (6) is discretized as:

$$X_i(k+1) = A_i X_i(k) + B_i U_i(k), \quad (7)$$

where $A_i = e^{A_{0i} \Delta t}$, $B_i = \int_0^{\Delta t} e^{A_{0i} s} B_{0i} ds$, and Δt is the sampling period. A polyhedron set Σ_i is defined to describe the errors during discretization and the system uncertainty, satisfying:

$$\Sigma_i = \{ [A_i \ B_i] \mid [A_i \ B_i] = [A_i \ B_i] \delta_i \},$$

where δ_i is the uncertainty weight matrix, with the uncertainty weight threshold determined by the user. The uncertainty matrix in this paper is defined as:

$$\delta_i = \text{diag} \{ \lambda_1 I, \lambda_2 I \}, \lambda_s \in [-0.5, 0.5], s = 1, 2.$$

By adding the bounded external interference $d_i(k) \in R^{3 \times 3}$, the system of the mobile robot i becomes:

$$X_i(k+1) = \Sigma_i \begin{bmatrix} X_i(k) \\ U_i(k) \end{bmatrix} + d_i(k). \quad (8)$$

III. ILDMPC CONTROLLER DESIGN AND SYSTEM ANALYSIS

A. ITERATIVE LEARNING DISTRIBUTED MPC CONTROLLER DESIGN

In the formation system, before each sampling, the information of each mobile robot will be exchanged to establish the relative position state error $X_{e,ij}$ between the mobile robots.

Although the coordinates and orientation of each robot are obtained, different robots use the same variables to describe their position state. This clearly and easily describes their position-state difference and constructs the coupling problem between the mobile robots. Therefore, a synchronous path parameter ρ_i is applied to describe the path $\Gamma_i = [x_{\Gamma_i}(\rho_i) \ y_{\Gamma_i}(\rho_i) \ \theta_{\Gamma_i}(\rho_i)]$ of the i -th mobile robot, which is a function of the control input and the linear and angular velocities:

$$\rho_i(k+1) = \rho_i(k) + C_i U_i(k) + D_i [v_i \ \omega_i]^T. \quad (9)$$

where C_i and D_i are the weight matrices in the linear formula that are heuristically determined. This synchronous path parameter is used to form the coupling parameter in the system. Although it has an independent linear relationship, it is also associated with the position state by the control input. In the control process, the path of the j -th mobile robot is guaranteed by controlling the path parameter difference between it and the i -th mobile robot to converge to the expected value. The path parameter difference (coupling parameter) is defined as $\rho_{ij} = \rho_i - \rho_j$, and the expected value is ρ_{ij}^r . The formation shape is guaranteed by applying this definition to all mobile robots.

In traditional DMPC, the cost function, which comprises state variables, control inputs, and coupling parameters, is used to solve quadratic programming problems. The traditional cost function is:

$$J_i(k) = \sum_{t=k}^{k+M-1} \left(\|X_{MPCi}(t|k)\|_{S_i}^2 + \|U_{MPCi}(t|k)\|_{Q_i}^2 + \sum_{j=1, j \neq i}^n a_{ij} \left| \rho_{MPCi}(t|k) - \rho_{MPCj}(t|k) - \rho_{ij}^r \right|_{w_{ij}}^2 \right).$$

where S_i , Q_i , and w_{ij} are the weight matrices, X_{MPCi} , ρ_{MPCi} , and U_{MPCi} represent the total position state error, path parameter, and the total control input calculated in DMPC, and M is the predictive horizon, when $M+k > T$. We consider $M = T - k$.

In the ILDMPC, only the state variables and the coupling parameter are applied to construct the general performance index, and the result trajectories of the previous iteration are used to construct the close-loop constraints to solve the optimization problem. The cost function changes enhance the optimization process and reduce the computational burden when dealing with the coupling and optimal problems. For the q -th iteration and at time T , the optimization problem is:

$$\begin{aligned} \min J_i^q(k) &= \sum_{t=k}^{k+M-1} \psi_i^q (X_{MPCi}^q(t|k), \rho_{MPCi}^q(t|k)) \\ &= \sum_{t=k}^{k+M-1} \left(\|X_{MPCi}^q(t|k)\|^2 + \sum_{j=1, j \neq i}^n a_{ij} \left| \rho_{MPCi}^q(t|k) \right. \right) \end{aligned}$$

$$\begin{aligned}
 & -\rho_{MPCj}^q(t|k) - \rho_{ij}^r|^2) \\
 \min J_i^q(k) &= \sum_{t=k}^{k+M-1} \psi_i^q(X_{MPCi}^q(t|k), \rho_{MPCi}^q(t|k)) \\
 &= \sum_{t=k}^{k+M-1} \left(\|X_{MPCi}^q(t|k)\|^2 + \sum_{j=1, j \neq i}^n a_{ij} |\rho_{MPCi}^q(t|k) \right. \\
 & \quad \left. - \rho_{MPCj}^q(t|k) - \rho_{ij}^r|^2) \right) \\
 \text{s.t.} \quad & (8), (9) \\
 & X_{MPCi}^q(k|k) = X_i^q(k) \\
 & \rho_{MPCi}^q(k|k) = \rho_i^q(k) \\
 & X_{MPCi}^q(k+M|k) = X_i^{q-1}(k+M) \\
 & \rho_{MPCi}^q(k+M|k) = \rho_i^{q-1}(k+M) . \\
 \text{s.t.} \quad & (8), (9) \\
 & X_{MPCi}^q(k|k) = X_i^q(k) \\
 & \rho_{MPCi}^q(k|k) = \rho_i^q(k) \\
 & X_{MPCi}^q(k+M|k) = X_i^{q-1}(k+M) \\
 & \rho_{MPCi}^q(k+M|k) = \rho_i^{q-1}(k+M) . \quad (10)
 \end{aligned}$$

where X_{MPCi}^q , ρ_{MPCi}^q , and U_{MPCi}^q represent the total position state error, path parameter, and the total control input calculated by the iterative learning MPC controller of the q -th iteration.

Fig. 1 illustrates the schematic block diagram of the iterative learning distributed MPC. Each mobile robot has the same internal structure, and robots $j1$ and $j2$ communicate with robot i . The communication content includes the robot's position state and synchronization path parameter. It should be noted that a memory module in the robot structure stores the state variables and synchronization path parameters calculated during each iteration. The required robot information is extracted from the memory module to construct the closed-loop constraint and optimization problem in the iterative calculation. After the machines interact, the optimal control input is obtained through rolling optimization, which will be applied to the robotic system. After receiving the formation task, the robots conduct finite iterations for the distributed system and then output their final control sequences to act on their plants.

The rolling optimization involves scroll optimization. Indeed, the classic optimal control theory is usually for a certain and invariant global optimization goal, where solving an optimization problem offline to obtain a globally optimal solution can be regarded as "offline optimization" and "one-time online implementation". Conversely, MPC does not use a constant global optimization goal but Rolling (usually finite time domain) optimization goals. Thus, at every moment (within time steps), MPC solves the optimization problem of finite dimensionality. At the next moment, the "window" of optimizing the time domain continues to move forward while keeping the length of the optimized time domain constant.

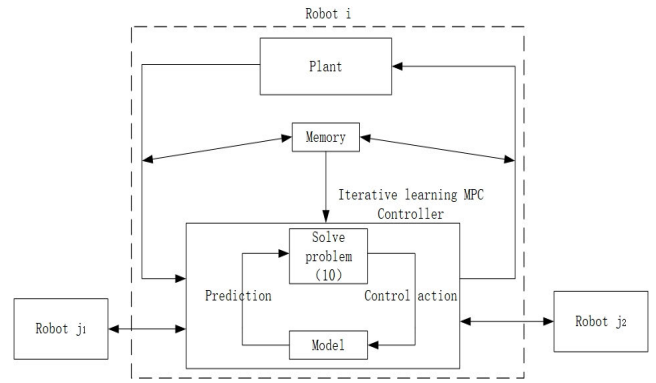


FIGURE 1. Schematic block diagram of ILDMPC.

The same optimization process is repeated online, constituting MPC's "rolling optimization." At its core, it is the process of repeatedly performing the optimal solution at different moments according to the set objective function and the established constraints and controlling the vehicle system more accurately and reducing the tracking error with the advancement of time and the change of state information.

The rolling optimization process predicts the future state variables at every time step using (8) and (9), and the closed-loop constraint is constructed using the results of the last iteration or the initial information. Given this constraint, the iterative learning MPC controller solves the convex optimization problem online, so the system can obtain the optimal control sequence when considering the uncertainty and external interference. After solving this optimization problem, $U_i^{q*}(k) = U_{MPCi}^q(k|k)$ and the optimal solution is applied to the controlled object.

It should be noted that although our method relies on DMPC, it has two major differences.

1) Traditional DMPC trades computation time for global optimality, resulting in steady-state errors in the system. Therefore, this study iteratively reduces the feasible domain constraints of DMPC to obtain a more accurate optimal solution sequence. This strategy improves the accuracy of the system's dynamic response process for a certain path point and reduces the total optimization cost.

2) Traditional DMPC does not involve any methods or preventive measures to deal with uncertainty. Opposing, our method uses an iterative strategy to refer to the results affected by uncertainty from the previous calculation used for the next solution, offering the system a certain number of correction opportunities and indirectly avoiding the direct impact of several uncertainty factors on the system. The related iterations are stopped if the number of iterations reaches a preset upper limit or the overall transient performance metric converges to a fixed value.

B. SYSTEM ANALYSIS

Next, we analyze the algorithm for each mobile robot. At the beginning of the iteration, two feasible data sets X_i^0, ρ_i^0 are

available for the iterative calculations. The values at time T are defined as X_{Ti}^0 and ρ_{Ti}^0 . Each iteration considers the same task $X_i^q = X_{Ti}^0$, $\rho_i^q = \rho_{Ti}^0$.

Theorem 1: When the solutions of the optimization problem are feasible in the $q-1$ -th iteration, the solutions of the q -th iteration optimization problem are feasible.

PROOF: The X_{MPCi}^q and ρ_{MPCi}^q calculated by ILDMPC are expressed as follows:

$$\begin{aligned} & X_{MPCi}^q(k+1|k), \dots, X_{MPCi}^q(k+M|k) \\ & \rho_{MPCi}^q(k+1|k), \dots, \rho_{MPCi}^q(k+M|k). \end{aligned}$$

According to the previous definition of the prediction horizon, X_{MPCi}^q and ρ_{MPCi}^q are calculated from the solutions U_{MPCi}^q , and the optimization problem is divided into two cases:

If $M+k \leq T$, we notice that

$$\begin{aligned} X_{MPCi}^q(k+M|k) &= X_i^{q-1}(k+M) \\ \rho_{MPCi}^q(k+M|k) &= \rho_i^{q-1}(k+M), \end{aligned}$$

with the feasible solutions in the $q-1$ -th iteration, $X_i^{q-1}(k+M)$ and $\rho_i^{q-1}(k+M)$ being feasible. In this case, the final result $X_{MPCi}^q(k+M|k)$ and $\rho_{MPCi}^q(k+M|k)$ is feasible. Therefore the solutions U_{MPCi}^q to the optimization problem in the q -th iteration are feasible.

If $M+k > T$, $M=T-k$, so there is $M+k=T$.

$$\begin{aligned} X_{MPCi}^q(k+M|k) &= X_i^{q-1}(k+M) = X_i^{q-1}(T) \\ &= X_i^{q-2}(T) = \dots = X_{T0}^q \\ \rho_{MPCi}^q(k+M|k) &= \rho_i^{q-1}(k+M) = \rho_i^{q-1}(T) \\ &= \rho_i^{q-2}(T) = \dots = \rho_{T0}^q, \end{aligned}$$

where X_{Ti}^0 and ρ_{Ti}^0 are feasible values given before the iteration at time T . So the final results $X_{MPCi}^q(k+M|k)$ and $\rho_{MPCi}^q(k+M|k)$ are feasible, proving that the solutions U_{MPCi}^q to the optimization problem in the q -th iteration are feasible.

Theorem 2: The iteration performance is defined as the sum of all X_i^q and $\rho_i^q - \rho_j^q$ errors at time T :

$$\Phi_i^q = \sum_{t=0}^T \Psi_i^q(X_{MPCi}^q(t|k), \rho_{MPCi}^q(t|k))$$

where ρ_j^q is obtained from the j -th mobile robot through a Bluetooth or wireless network, so ρ_j^q can be treated as known. In ILDMPC, the iteration performance improves for each mobile robot i as the number of iterations increases.

PROOF: For the time T , the iteration performance can be analyzed from two cases:

If $M+k \leq T$, $J_i^q(k)$ should satisfy the following relationship:

$$\begin{aligned} & J_i^q(k+1) \\ & \leq \sum_{t=k+1}^{k+M} \Psi_i^q(X_{MPCi}^q(t|k), \rho_{MPCi}^q(t|k)) \end{aligned}$$

$$\begin{aligned} & = J_i^q(k) + \Psi_i^q(X_{MPCi}^q(k+M|k), \rho_{MPCi}^q(k+M|k)) \\ & \quad - \Psi_i^q(X_{MPCi}^q(k|k), \rho_{MPCi}^q(k|k)) \\ & = J_i^q(k) + \Psi_i^q(X_i^{q-1}(k+M|k), \rho_i^{q-1}(k+M|k)) \\ & \quad - \Psi_i^q(X_{MPCi}^q(k), \rho_{MPCi}^q(k)). \end{aligned} \quad (11)$$

When summing (11) from $k=0$ to $k=T-M$, the result is as follows:

$$\begin{aligned} & J_i^q(T+M-1) - J_i^q(0) \\ & \leq \sum_{m=N_k}^T \Psi_i^q(X_i^q(m), \rho_i^q(m)) \\ & \quad - \sum_{n=0}^{T-M} \Psi_i^q(X_{MPCi}^q(n), \rho_{MPCi}^q(n)). \end{aligned} \quad (12)$$

If $M+k > T$, take $M=T-k$, and $J_i^q(k)$ should satisfy the following relationship:

$$\begin{aligned} J_i^q(k+1) & \leq \sum_{t=k+1}^T \Psi_i^q(X_{MPCi}^q(t|k), \rho_{MPCi}^q(t|k)) \\ & = J_i^q(k) - \Psi_i^q(X_{MPCi}^q(k|k), \rho_{MPCi}^q(k|k)). \end{aligned} \quad (13)$$

When summing (13) from $k=T-M+1$ to $k=T-1$, the result is as follows:

$$\begin{aligned} & J_i^q(T) - J_i^q(T-M+1) \\ & \leq - \sum_{n=T-M+1}^{T-1} \Psi_i^q(X_{MPCi}^q(n), \rho_{MPCi}^q(n)). \end{aligned} \quad (14)$$

At moment T , $J_i^q(T) = \Psi_i^q(X_i^q(T), \rho_i^q(T))$. By combining (12) and (14), we obtain the following:

$$\begin{aligned} J_i^q(T) - J_i^q(0) & = \Psi_i^q(X_i^q(T), \rho_i^q(T)) - J_i^q(0) \\ & \leq \sum_{m=M}^T \Psi_i^q(X_i^q(m), \rho_i^q(m)) \\ & \quad - \sum_{n=0}^{T-M} \Psi_i^q(X_{MPCi}^q(n), \rho_{MPCi}^q(n)) \\ & \quad - \sum_{n=T-M+1}^{T-1} \Psi_i^q(X_{MPCi}^q(n), \rho_{MPCi}^q(n)) \\ & = \sum_{m=M}^T \Psi_i^q(X_i^{q-1}(m), \rho_{MPCi}^{q-1}(m)) \\ & \quad - \sum_{m=0}^{T-1} \Psi_i^q(X_{MPCi}^q(m), \rho_{MPCi}^q(m)), \end{aligned} \quad (15)$$

After the transfer, there is:

$$\begin{aligned} \Phi_i^q & = \Psi_i^q(X_{MPCi}^q(T), \rho_{MPCi}^q(T)) \\ & \quad + \sum_{n=0}^{T-1} \Psi_i^q(X_{MPCi}^q(n), \rho_{MPCi}^q(n)) \leq J_i^q(0), \end{aligned} \quad (16)$$

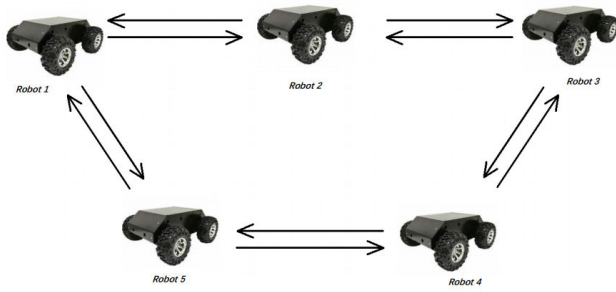


FIGURE 2. Communication topology.

Besides $J_i^q(0)$ should satisfy $J_i^q(0) \leq \sum_{m=0}^{M-1} \Psi_i^q(X_i^{q-1}(m), \rho_i^{q-1}(m))$. Therefore (16) is expressed as follows:

$$\begin{aligned} \Phi_i^q &\leq J_i^q(0) + \sum_{m=M}^T \Psi_i^q(X_i^{q-1}(m), \rho_i^{q-1}(m)) \\ &= \sum_{m=0}^{M-1} \Psi_i^q(X_i^{q-1}(m), \rho_i^{q-1}(m)) \\ &\quad + \sum_{m=M}^T \Psi_i^q(X_i^{q-1}(m), \rho_i^{q-1}(m)) \\ &= \Phi_i^{q-1}. \end{aligned} \tag{17}$$

Remark 1: Theorem 1 analyzes the feasibility of the algorithm solution, and theorem 2 proves that the controller’s performance after each iteration is better than in the previous iteration. Therefore, the algorithm provides feasible X_i and ρ_i and converges them to the control target during the iteration process. Furthermore, if theory 1 and theory 2 hold for each mobile robot, they hold for the distributed formation system.

IV. SIMULATION RESULTS

This section evaluates the formation control of five robots using the proposed ILDMPC method on two simulated scenarios. In the second scenario, we challenge ILDMPC against RMPC to demonstrate our method’s effectiveness. It should be noted that we only compare our method against RMPC as, to our knowledge, existing methods are not open-source and thus were unavailable during our trials. Furthermore, implementing these methods based on the corresponding papers might lead to a code that underperforms compared to its original implementation, affecting the performance evaluation trials. Hence, we compare our scheme only against RMPC.

Regarding the robotic setup, the neighbor robot of robot i is the communication object with robot i , with Fig. 2 presenting the corresponding oriented communication topology graph. The vertex array is defined as:

$$[Robot1 \ Robot2 \ Robot3 \ Robot4 \ Robot5],$$

the adjacency matrix is:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

A. SIMULATION 1

In this simulation, the lane-changing path is considered the formation shape of the five mobile robots. The path function is:

$$\Gamma = \begin{cases} x_\Gamma = -\rho \\ y_\Gamma = -2/1 + e^{-2(\rho-2)} \end{cases}.$$

The path parameter differences of the five mobile robots are $\rho_{12}^r = -10, \rho_{23}^r = -10, \rho_{34}^r = 15, \rho_{45}^r = -10, \rho_{51}^r = 15$, and the path error between the i -th and j -th mobile robot is $x_{\Gamma ij}(\rho_{ij}^r) = -\rho_{ij}^r$ and $y_{\Gamma ij} = 2$. Two groups of feasible data¹ are used as the values of the state error X_i and the path parameter ρ_i of each mobile robot for the initial iteration learning. The initial values $X_i^q(0)$ before each iteration are:

$$\begin{aligned} X_1^q(0) &= [-0.2 \ 0.5 \ 0.25]^T, \\ X_2^q(0) &= [0.1 \ 0.03 \ 0.15]^T, \\ X_3^q(0) &= [0.15 \ -0.02 \ 0.1]^T, \\ X_4^q(0) &= [0.15 \ -0.02 \ 0.1]^T, \\ X_5^q(0) &= [0.15 \ -0.02 \ 0.1]^T. \end{aligned}$$

We set the sampling period to $\Delta t = 0.1s$, and the coefficients are set to $C_i = D_i = [-0.2 \ 0.1]$. Each robot’s angle and line speed are $\omega_i^r = 1$ and $v_i^r = 1$, respectively. The uncertainty weight matrix δ_i is set as $\text{diag}\{-0.3I, 0.5I\}$. The system performs 10 iterations, each lasting $T = 20s$. During each iteration, every mobile robot moves along $\rho_1 = 10, \rho_2 = 20, \rho_3 = 30, \rho_4 = 15, \rho_5 = 25$. The bounded external disturbance of each mobile robot is as follows:

$$\begin{aligned} d_1(k) &= \begin{bmatrix} (\frac{0.2}{2k+1} + \frac{2}{1000}) \cos(0.1k) \\ (\frac{0.2}{k+1} + \frac{2}{1000}) \sin(0.1k) \\ (\frac{0.1}{k+1} + \frac{1}{1000}) \cos(0.2k) \end{bmatrix} \\ d_2(k) &= \begin{bmatrix} (\frac{0.1}{2k+1} + \frac{1}{1000}) \cos(0.1k) \\ (\frac{0.2}{k+1} + \frac{2}{1000}) \sin(0.1k) \\ (\frac{0.1}{2k+1} + \frac{1}{1000}) \cos(0.2k) \end{bmatrix} \\ d_3(k) &= \begin{bmatrix} (\frac{0.1}{2k+1} + \frac{1}{1000}) \cos(0.1k) \\ (\frac{0.15}{k+1} + \frac{1}{1000}) \sin(0.1k) \\ (\frac{0.2}{k+1} + \frac{2}{1000}) \cos(0.2k) \end{bmatrix} \\ d_4(k) &= \begin{bmatrix} (\frac{0.2}{2k+1} + \frac{2}{1000}) \cos(0.1k) \\ (\frac{0.15}{k+1} + \frac{1}{1000}) \sin(0.1k) \\ (\frac{0.1}{k+1} + \frac{1}{1000}) \cos(0.2k) \end{bmatrix} \\ d_5(k) &= \begin{bmatrix} (\frac{0.1}{2k+1} + \frac{1}{1000}) \cos(0.1k) \\ (\frac{0.2}{k+1} + \frac{2}{1000}) \sin(0.1k) \\ (\frac{0.1}{k+1} + \frac{1}{1000}) \cos(0.2k) \end{bmatrix} \end{aligned}$$

¹The data can be provided by contacting the corresponding author.

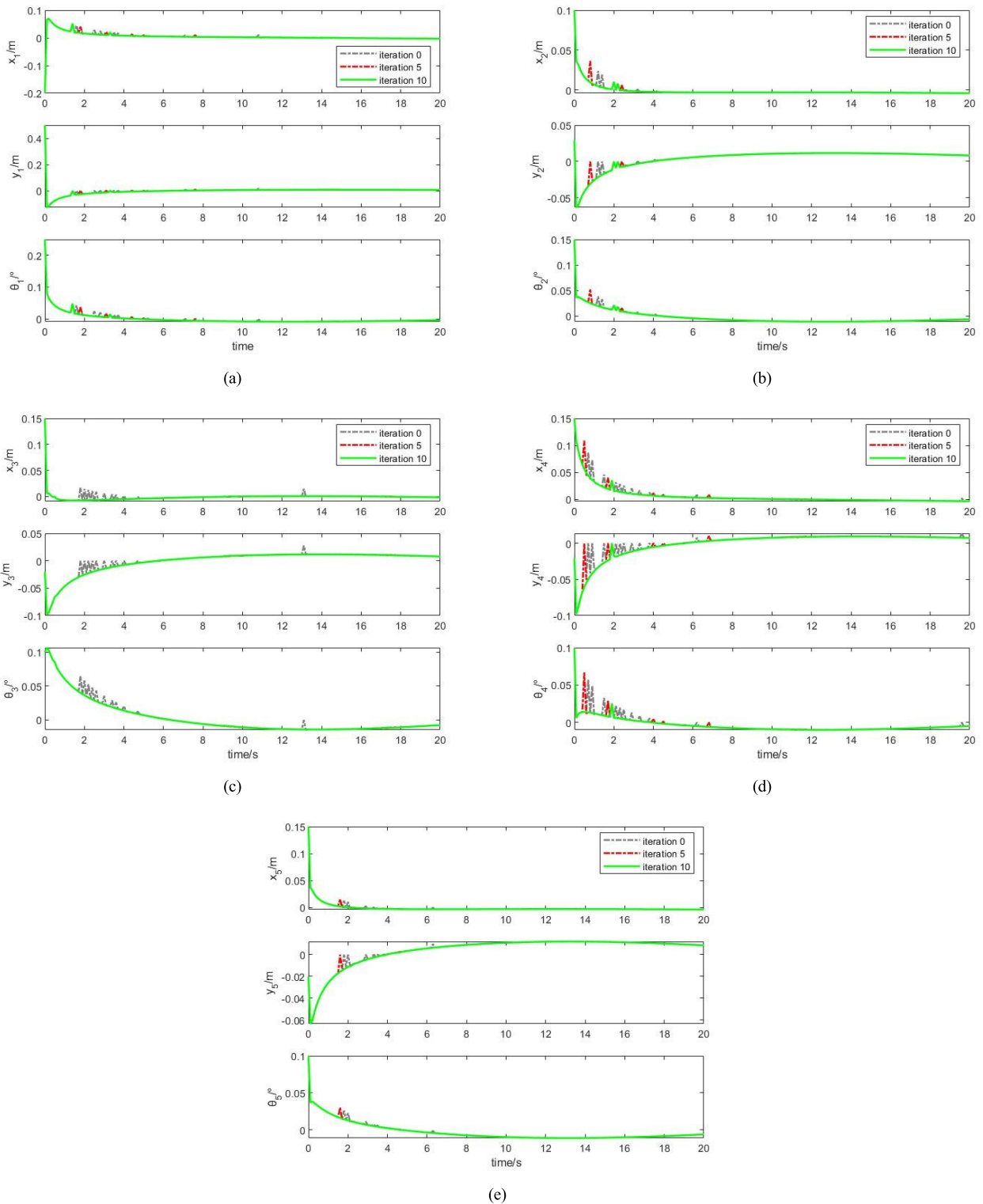


FIGURE 3. State error X_i of each mobile robot (a)-(e) denoting robot 1-5, respectively.

Fig. 3 presents the state error X_i of each mobile robot in the iterative process, highlighting that the convergence curve improves as the number of iterations increases. Moreover, Fig. 4 compares ILDMPC and RMPC. Specifically,

Fig 4 compares the state quantity obtained by the tenth iteration with the RDMPC result under the same initial conditions. The comparative results infer that ILDMPC enhances the convergence speed of the system because of its

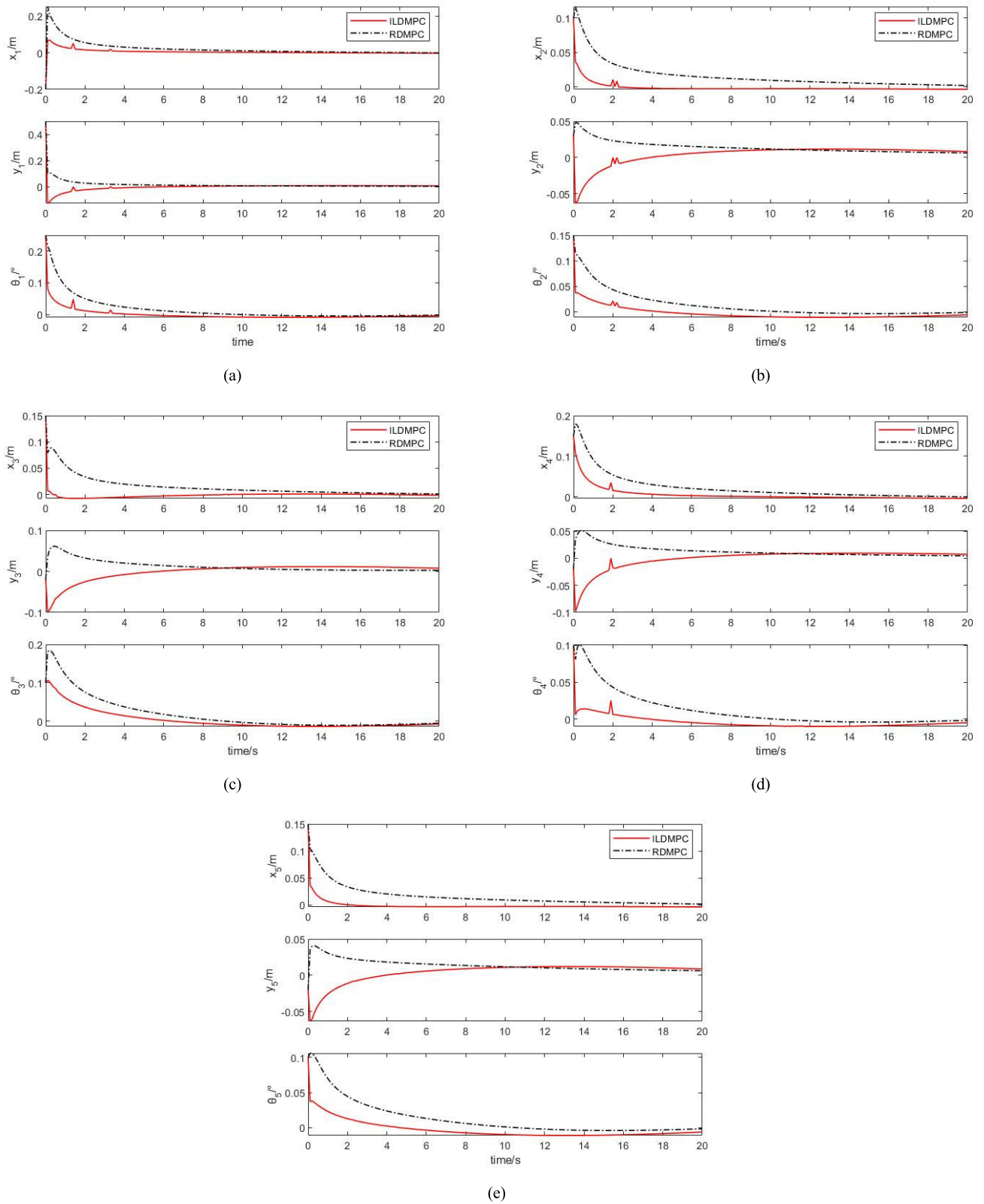


FIGURE 4. Comparing ILDMPC and RMPC per mobile robot (a)-(e) denoting robot 1-5, respectively.

learning ability and ameliorates the overshoot effect in the convergence process.

Fig. 5 illustrates the results of the path parameter difference tracking on the desired values. This figure reveals that

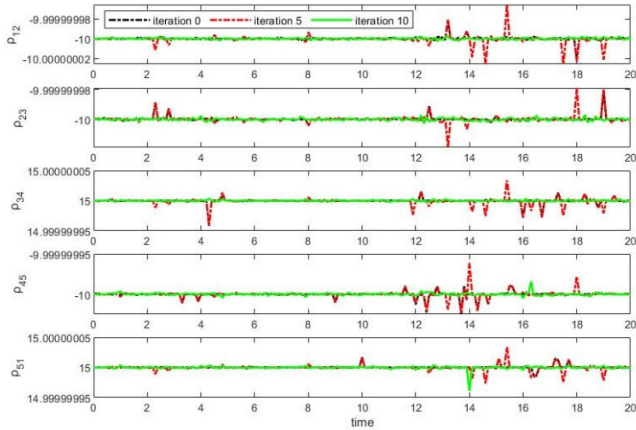


FIGURE 5. Path parameter differences.

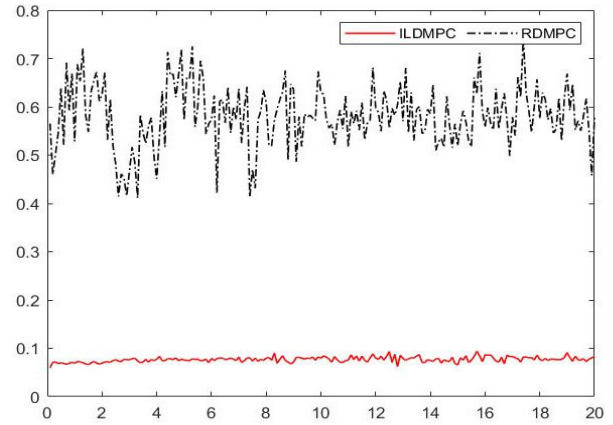


FIGURE 7. Computing speed of the controller.

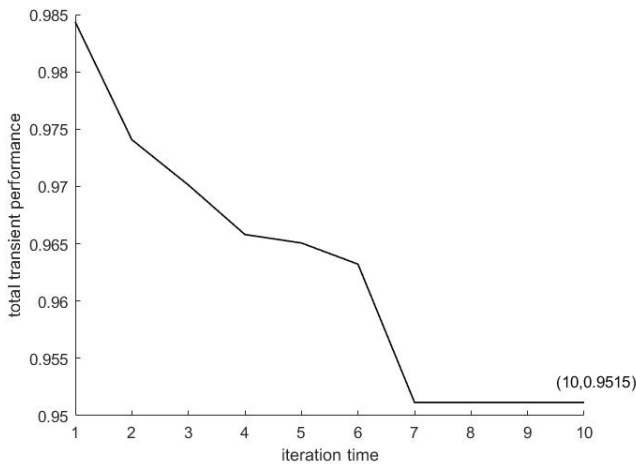


FIGURE 6. Total transient performance indicators over 10 iterations.

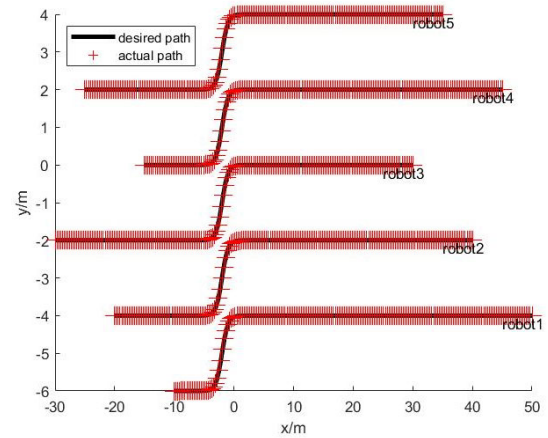


FIGURE 8. Path of each robot.

as the number of iterations increases, the iterative learning algorithm effectively reduces the tracking error and promotes tracking accuracy. The total transient performance is defined as $\sum_{i=1}^n \sum_{k=0}^T \Psi_i^q(X_i(k), \rho_i(k))$. Besides, Fig. 6 illustrates the total transient performance, highlighting the ability of X_i and ρ_i of the formation system to track the control target. The figure demonstrates that the total transient performance decreases along the learning process, and the total transient performance converges to a fixed value of 0.9515. Fig. 7 depicts the computing speed of the controller, where the execution time of the 10th iteration is compared against the robust control. The results infer that the cost function improvement and constraints increase the computing speed by at least 4 times. Finally, Fig. 8 presents the actual path of each robot following the desired path, suggesting that the mobile robots keep their queue shape and track the desired path well.

B. SIMULATION 2

In this simulation, the five mobile robots aim to preserve the formation of concentric circles and move half an arc. The path

function is expressed as a polar coordinate equation:

$$\Gamma = \begin{cases} x_\Gamma = r_\Gamma \cos(\rho) \\ y_\Gamma = r_\Gamma \sin(\rho) \end{cases}$$

The path parameter differences of five mobile robots are $\rho_{12}^r = \rho_{23}^r = \rho_{34}^r = \rho_{45}^r = \rho_{51}^r = 0$. The path error between the i -th and j -th mobile robots is expressed as $r_{\Gamma ij} = -0.25$ and $r_{\Gamma 1} = 0.25$. Two groups of feasible data are used as the values of the state error X_i and the path parameter ρ_i of each mobile robot for the initial learning of iteration. The initial values $X_i^q(0)$ of the mobile robots before each iteration are as follows:

$$\begin{aligned} X_1^q(0) &= [0.5 \ 0.2 \ 0.25]^T, \\ X_2^q(0) &= [-0.1 \ 0.2 \ 0.05]^T, \\ X_3^q(0) &= [0.05 \ -0.2 \ 0.3]^T, \\ X_4^q(0) &= [-0.15 \ -0.1 \ 0.15]^T, \\ X_5^q(0) &= [-0.15 \ -0.2 \ 0.15]^T. \end{aligned}$$

We chose a sampling period of $\Delta t = 0.1s$, and the coefficients are set to $C_i = D_i = [0 \ pi/200]$. Each robot's angle and linear speed are $\omega_i^r = pi/200$ and

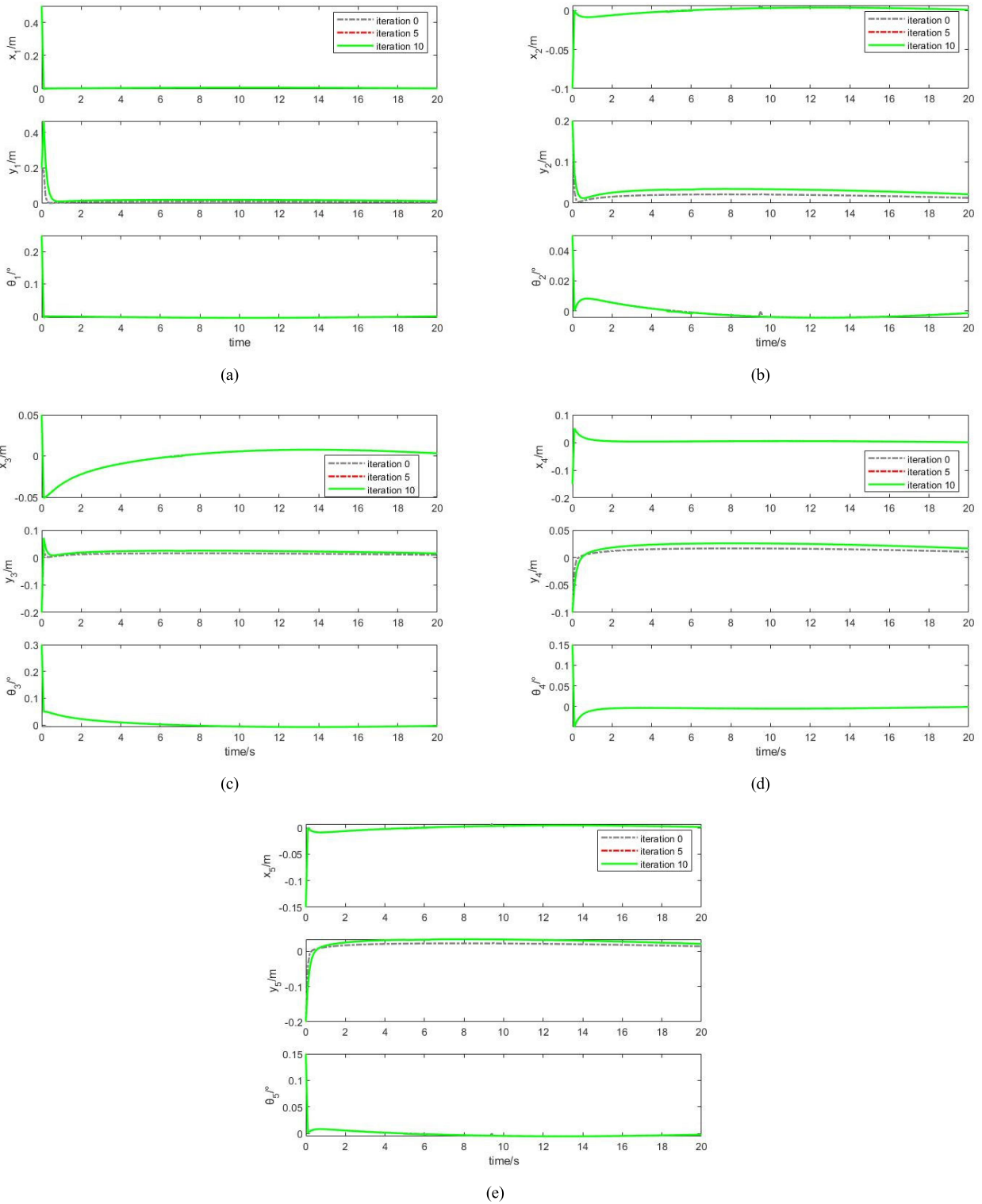


FIGURE 9. State error X_i of each mobile robot (a)-(e) denoting robot 1-5, respectively.

$v_i^r = r_{\Gamma_i} \omega_i^r$, respectively. The uncertainty weight matrix δ_i is set as $\text{diag}\{0.5I, -0.5I\}$. The system conducts 10 iterations, each lasting $T=20s$, during which each mobile robot moves along $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5 = 0$.

The bounded external disturbance of each mobile robot is the same as in simulation 1. Moreover, it should be noted that simulation 2 resets the system's uncertainty weight matrix.

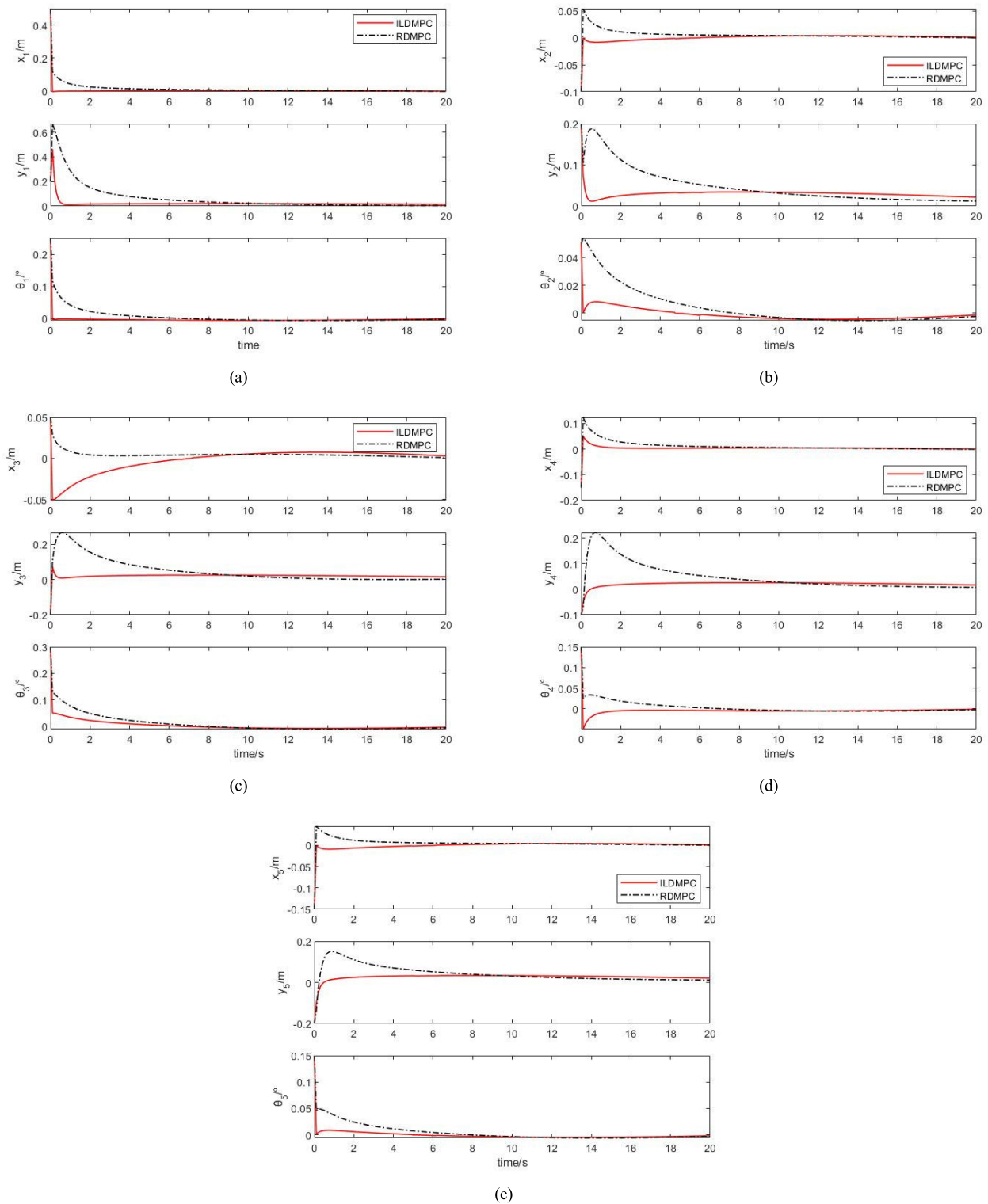


FIGURE 10. Comparing ILDMPC and RMPC per mobile robot (a)-(e) denoting robot 1-5, respectively.

Fig. 9 expresses the state error X_i of each mobile robot in the iterative process, demonstrating similar results to simulation 1. Indeed, the system forces the state variables to converge even under uncertainty. Fig. 10 compares ILDMPC and

RMPC, highlighting that the mitigation effect of ILDMPC on overshoot is more obvious than in simulation 1. Moreover, Fig. 11 presents the results of path parameter differences tracking on the desired values. This figure demonstrates that

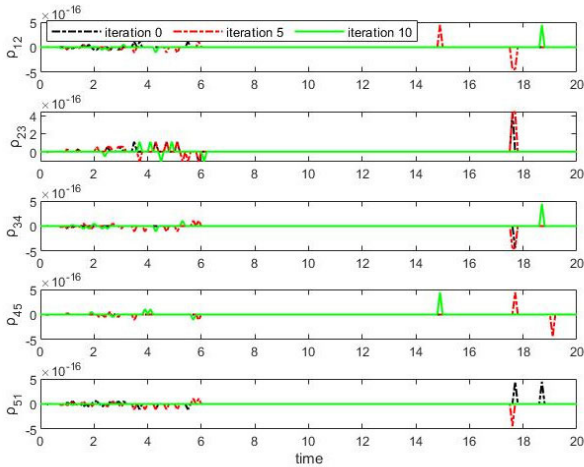


FIGURE 11. Path parameter differences.

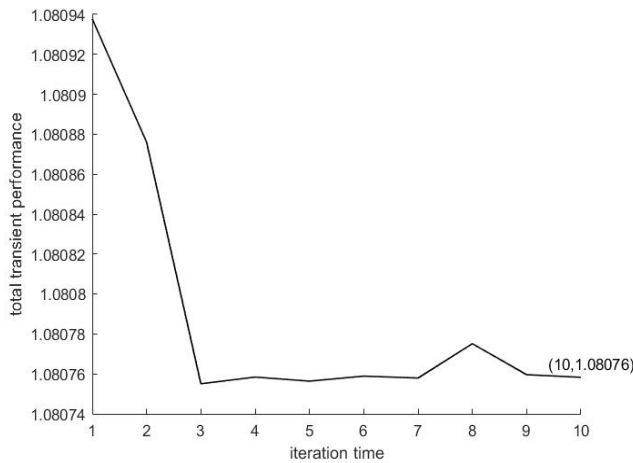


FIGURE 12. Total transient performance indicators over 10 iterations.

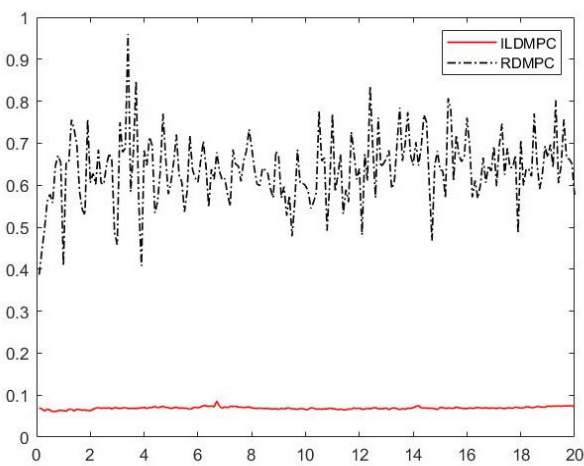


FIGURE 13. Computing speed of the controller.

as the number of iterations increased, the tracking accuracy significantly improved compared to simulation 1. The main reason is that the path parameters are set to be consistent, which is more conducive to applying the algorithm. Besides,

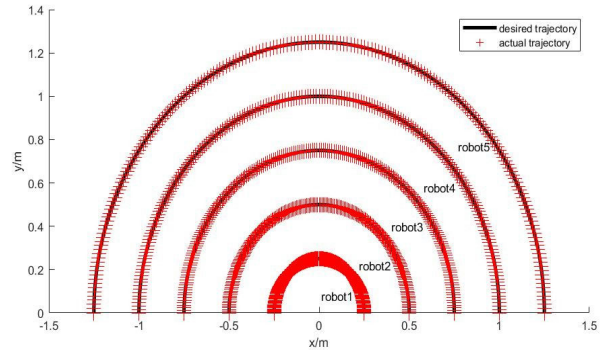


FIGURE 14. Path of each robot.

Fig. 12 illustrates the total transient performance, which illustrates the ability of X_i and ρ_i of the formation system to track the control target. The total transient performance converges to a fixed value of 1.08076. Fig. 13 depicts the computing speed of the controller, where the calculation speed is increased by at least 4 times. Finally, Fig. 14 illustrates the actual path of each robot following the desired path, i.e., the mobile robots move half an arc synchronously and track the desired paths well.

V. CONCLUSION

This paper designs an iterative learning distributed predictive model controller for multiple robots under model uncertainties. Specifically, in the optimization problem, the proposed method considers the total relative position state error and coupling parameter to construct the general performance index and replace the traditional cost function. This strategy improves the system’s efficiency and ability to find the optimal solution. The closed-loop constraints are based on historical information when dealing with and calculating the robot information, such as total relative position error and coupling parameters. Therefore, during each iteration, the system learns the information obtained in the previous iteration and continuously improves the results of the next iteration. Overall, this strategy improves the convergence curve and convergence effect. Through extensive analysis, it is proven that our method seeks feasible solutions to the optimization problem and thus improves the system’s iterative performance as the number of iterations increases under the close-loop constraints. Our algorithm’s effectiveness is demonstrated through two groups of simulation experiments.

Nevertheless, the proposed method iteratively updates the system model and establishes non-repetitive constraints, limiting its effectiveness. Hence, future studies will aim to overcome these limitations. Furthermore, future studies will focus on overcoming our method’s design difficulties by updating the system model iteratively and iteratively establishing distinct constraints.

AVAILABILITY OF DATA AND MATERIALS

All data generated or analyzed during this study are included in this published article. The source codes used during the

research are available from the corresponding author upon reasonable request.

AUTHOR CONTRIBUTIONS

All authors contributed to the study's conception and design. Wei Shang, Hanzong Zhu, Daode Zhang, and Xiuhong Li performed material preparation, data collection, and analysis. Hanzong Zhu and Xiuhong Li wrote the first draft of the manuscript. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

DECLARATIONS

ETHICS APPROVAL

Approval was obtained from the Hubei University of Technology ethics committee.

CONSENT TO PARTICIPATE

Informed consent was obtained from all individual participants included in the study.

CONSENT FOR PUBLICATION

The participant consented to submit the research manuscript to the journal.

COMPETING INTERESTS

The authors have no conflicts of interest to declare relevant to this article's content.

ACKNOWLEDGMENT

The authors would like to express their gratitude to EditSprings (<https://www.editsprings.cn>) for the expert linguistic services provided.

REFERENCES

- N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, Sep. 2012, doi: [10.1109/TCST.2011.2167331](https://doi.org/10.1109/TCST.2011.2167331).
- Q. Jiang and V. Kumar, "The inverse kinematics of cooperative transport with multiple aerial robots," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 136–145, Feb. 2013, doi: [10.1109/TRO.2012.2218991](https://doi.org/10.1109/TRO.2012.2218991).
- Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, "Vision-based model predictive control for steering of a nonholonomic mobile robot," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 553–564, Mar. 2016, doi: [10.1109/TCST.2015.2454484](https://doi.org/10.1109/TCST.2015.2454484).
- F. Ke, Z. Li, H. Xiao, and X. Zhang, "Visual servoing of constrained mobile robots based on model predictive control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1428–1438, Jul. 2017, doi: [10.1109/TSMC.2016.2616486](https://doi.org/10.1109/TSMC.2016.2616486).
- H. Cai and G. Hu, "Distributed tracking control of an interconnected leader–follower multiagent system," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3494–3501, Jul. 2017, doi: [10.1109/TAC.2017.2660298](https://doi.org/10.1109/TAC.2017.2660298).
- N. Chen, Y. Wang, and L. Jia, "Multi-Mobile robot leader–follower formation distributed control under switching topology," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 2673–2678, doi: [10.1109/CCDC49329.2020.9164773](https://doi.org/10.1109/CCDC49329.2020.9164773).
- H. Liang, Y. Fu, F. Kang, J. Gao, and N. Qiang, "A behavior-driven coordination control framework for target hunting by UAV intelligent swarm," *IEEE Access*, vol. 8, pp. 4838–4859, 2020, doi: [10.1109/ACCESS.2019.2962728](https://doi.org/10.1109/ACCESS.2019.2962728).
- M. A. Nursyeha, M. Rivai, and D. Purwanto, "LiDAR equipped robot navigation on behavior-based formation control for gas leak localization," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Jul. 2020, pp. 89–94, doi: [10.1109/ISITIA49792.2020.9163758](https://doi.org/10.1109/ISITIA49792.2020.9163758).
- Y. Qingkai, C. Ming, F. Hao, C. Jie, and H. Jie, "Distributed formation stabilization for mobile agents using virtual tensegrity structures," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 447–452, doi: [10.1109/ChiCC.2015.7259678](https://doi.org/10.1109/ChiCC.2015.7259678).
- E. Abbasi, M. Ghayour, M. Danesh, P. Amiri, and M. H. Yoosefian, "Formation flight control and path tracking of a multi-quadrotor system in the presence of measurement noise and disturbances," in *Proc. 6th RSI Int. Conf. Robot. Mechatronics (ICRoM)*, Oct. 2018, pp. 273–279, doi: [10.1109/ICRoM.2018.8657620](https://doi.org/10.1109/ICRoM.2018.8657620).
- A. Liu, W.-A. Zhang, L. Yu, H. Yan, and R. Zhang, "Formation control of multiple mobile robots incorporating an extended state observer and distributed model predictive approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4587–4597, Nov. 2020, doi: [10.1109/TSMC.2018.2855444](https://doi.org/10.1109/TSMC.2018.2855444).
- C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 604–611, Apr. 2020, doi: [10.1109/LRA.2020.2964159](https://doi.org/10.1109/LRA.2020.2964159).
- L. Chen, H. Hopman, and R. R. Negenborn, "Distributed model predictive control for vessel train formations of cooperative multi-vessel systems," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 101–118, Jul. 2018.
- H. Ebel, E. S. Ardakani, and P. Eberhard, "Distributed model predictive formation control with discretization-free path planning for transporting a load," *Robot. Auto. Syst.*, vol. 96, pp. 211–223, Oct. 2017.
- H. Wei, Q. Sun, J. Chen, and Y. Shi, "Robust distributed model predictive platooning control for heterogeneous autonomous surface vehicles," *Control Eng. Pract.*, vol. 107, Feb. 2021, Art. no. 104655.
- H. Wang, J. Jiang, W. Wu, L. Liu, D. Wang, and Z. Peng, "Robust distributed guidance and control of multiple autonomous surface vehicles based on extended state observers and finite-set model predictive control," in *Proc. 5th Int. Conf. Autom., Control Robot. Eng. (CACRE)*, Sep. 2020, pp. 235–239, doi: [10.1109/CACRE50138.2020.9230166](https://doi.org/10.1109/CACRE50138.2020.9230166).
- S. J. Yoo and B. S. Park, "Connectivity-preserving approach for distributed adaptive synchronized tracking of networked uncertain nonholonomic mobile robots," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2598–2608, Sep. 2018, doi: [10.1109/TCYB.2017.2743690](https://doi.org/10.1109/TCYB.2017.2743690).
- A. Marino, "Distributed adaptive control of networked cooperative mobile manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1646–1660, Sep. 2018, doi: [10.1109/TCST.2017.2720673](https://doi.org/10.1109/TCST.2017.2720673).
- J. Zhang and D. Meng, "Robust tracking problems for continuous-time iterative learning systems with nonrepetitive model uncertainties," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 5855–5860, doi: [10.1109/CAC51589.2020.9327033](https://doi.org/10.1109/CAC51589.2020.9327033).
- R. Chi and Z. Hou, "Adaptive ILC for a discrete-time nonlinear system with both parametric and non-parametric uncertainties," in *Proc. Chin. Control Decis. Conf.*, Jun. 2009, pp. 1722–1727, doi: [10.1109/CCDC.2009.5192270](https://doi.org/10.1109/CCDC.2009.5192270).
- H. Khan, J. K. Tar, and K. Széll, "On replacing Lagrange's 'reduced gradient algorithm' by simplified fixed point iteration in adaptive model predictive control," in *Proc. IEEE 23rd Int. Conf. Intell. Eng. Syst. (INES)*, Apr. 2019, pp. 201–206, doi: [10.1109/INES46365.2019.9109504](https://doi.org/10.1109/INES46365.2019.9109504).
- M. Bujarbaruah, C. Vallon, and F. Borrelli, "Learning to satisfy unknown constraints in iterative MPC," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 6204–6209, doi: [10.1109/CDC42340.2020.9303765](https://doi.org/10.1109/CDC42340.2020.9303765).
- S. Xie and J. Ren, "Iterative learning-based model predictive control for precise trajectory tracking of piezo nanopositioning stage," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 2922–2927, doi: [10.23919/ACC.2018.8430854](https://doi.org/10.23919/ACC.2018.8430854).
- H. Hazarika and A. Swarup, "Application of an optimal ILC algorithm for flow rate tracking of a ventilator system," in *Proc. 1st IEEE Int. Conf. Meas., Instrum., Control Autom. (ICMICA)*, Jun. 2020, pp. 1–6, doi: [10.1109/ICMICA48462.2020.9242747](https://doi.org/10.1109/ICMICA48462.2020.9242747).
- U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. A data-driven control framework," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1883–1896, Jul. 2018, doi: [10.1109/TAC.2017.2753460](https://doi.org/10.1109/TAC.2017.2753460).
- W. Yao, R. Chi, B. Li, and A. Chen, "Propulsion motor vector control based on ILC for dynamic positioning system," in *Proc. 6th Data Driven Control Learn. Syst. (DDCLS)*, May 2017, pp. 341–345, doi: [10.1109/DDCLS.2017.8068094](https://doi.org/10.1109/DDCLS.2017.8068094).

- [27] X.-S. Dai, X.-Y. Zhou, S.-P. Tian, and H.-T. Ye, "Iterative learning control for MIMO singular distributed parameter systems," *IEEE Access*, vol. 5, pp. 24094–24104, 2017, doi: [10.1109/ACCESS.2017.2768480](https://doi.org/10.1109/ACCESS.2017.2768480).
- [28] K. Ali, S. Ullah, A. Mehmood, H. Mostafa, M. Marey, and J. Iqbal, "Adaptive FIT-SMC approach for an anthropomorphic manipulator with robust exact differentiator and neural network-based friction compensation," *IEEE Access*, vol. 10, pp. 3378–3389, 2022.
- [29] S. Ullah, Q. Khan, A. Mehmood, S. A. M. Kirmani, and O. Mechali, "Neuro-adaptive fast integral terminal sliding mode control design with variable gain robust exact differentiator for under-actuated quadcopter UAV," *ISA Trans.*, vol. 120, pp. 293–304, Jan. 2022.
- [30] S. J. Fesharaki, M. Kamali, F. Sheikholeslam, and H. A. Talebi, "Robust model predictive control with sliding mode for constrained non-linear systems," *IET Control Theory Appl.*, vol. 14, no. 17, pp. 2592–2599, Nov. 2020.
- [31] Z. Xuan, "Research on collaborative adaptive cruise control system based on robust model predictive control," Chongqing Univ. Technol., Chongqing, China, Tech. Rep.
- [32] H. Xie, L. Dai, Y. Luo, and Y. Xia, "Robust MPC for disturbed nonlinear discrete-time systems via a composite self-triggered scheme," *Automatica*, vol. 127, May 2021, Art. no. 109499.



WEI SHANG received the Ph.D. degree in aeronautical and astronautical science and technology from the Beijing Institute of Technology, China, in 2017. He is currently a Lecturer with the Hubei University of Technology. His research interests include control theory of multi-agent systems and flight control.



MENG LIU is currently pursuing the degree with the Hubei University of Technology (HBUT), Wuhan, under the supervision of Dr. Wei Shang. He is conducting research and learning in the direction of path planning.



DAODE ZHANG received the Ph.D. degree in mechanical and electronic engineering from the Huazhong University of Science and Technology, China, in 2012. He is currently a Professor with the Hubei University of Technology. His research interests include mechanical and electronic engineering, intelligent control, and embedded system applications.



HANZONG ZHU is currently pursuing the degree with the Hubei University of Technology (HBUT), Wuhan, under the supervision of Dr. Wei Shang. He has completed a few projects in the fields of mobile robots and stability analysis. His research interests include distributed control, iterative learning, and machine learning.

...