

Research Article

Recurrent Neural Network-Based Model Predictive Control for Multiple Unmanned Quadrotor Formation Flight

Boyang Zhang , Xiuxia Sun , Shuguang Liu, and Xiongfeng Deng 

Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University, Xi'an 710051, China

Correspondence should be addressed to Boyang Zhang; boyang_530@163.com

Received 12 December 2018; Accepted 21 April 2019; Published 9 June 2019

Academic Editor: Paolo Castaldi

Copyright © 2019 Boyang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a dynamical recurrent neural network- (RNN-) based model predictive control (MPC) structure for the formation flight of multiple unmanned quadrotors. A distributed hierarchical control system with the translation subsystem and rotational subsystem is proposed to handle the formation-tracking problem for each quadrotor. The RNN-based MPC is proposed for each subsystem, where the RNN is introduced as the predictive model in MPC. And to improve the modeling accuracy, an adaptive updating law is developed to tune weights online for the RNN. Besides, the adaptive differential evolution (DE) algorithm is utilized to solve the optimization problem for MPC. Furthermore, the closed-loop stability is analyzed; meanwhile, the convergence of the DE algorithm is discussed as well. Finally, some simulation examples are provided to illustrate the validity of the proposed control structure.

1. Introduction

The unmanned quadrotor is a class of pilot-less aerial vehicles which enable to hover, take off, and land vertically with aerodynamic and propulsion characteristics. Due to the capabilities of simplicity, maneuverability, and payload [1, 2], the unmanned quadrotor has been gaining extensive attention and plays an important role in the military and civilian fields, such as real-time reconnaissance, surveillance, search and rescue missions, bush fire monitoring, agricultural crop dusting, and different airborne operations [3–5]. These flight missions not only impose stringent requirements over the modeling and control of quadrotor dynamics but also increasingly require multiple quadrotors to coordinate in a formation. Here, the basis of these flight missions is the trajectory tracking. However, the dynamics of the quadrotor are highly nonlinear and subject to uncertain disturbances, so it is a challenge to design a desirable control system for the unmanned quadrotor.

To solve the above problems, several control approaches have been investigated in recent years, such as proportional-integral-derivative (PID) control and linear quadratic regulator (LQR) control [6], sliding mode control [7], and

backstepping control [8]. However, these methods only consider the control effects and often neglect the actual flight constraints. Besides, the optimality of the path cannot be ensured, as the selection of reactive parameters is aimless or blind.

To solve the problems further, a new developing method, the model predictive control (MPC), has been proposed for the unmanned quadrotor system [9, 10], where the physical constraints of the system inputs and outputs could be easily solved by using the proper penalty terms in cost function. In [11], the MPC method was proposed to solve the path-tracking problem for the unmanned quadrotor, where a future control sequence was computed in a defined horizon, such that the prediction of the quadrotor output was driven close to the reference. In [12], a novel control strategy which combined the active disturbance rejection control and model predictive control was presented for the unmanned quadrotor. The proposed method improved the robustness for the modeling error and disturbances while performing a smooth trajectory tracking.

The performance of MPC highly depends on the accuracy of the described predictive model. Conventional techniques of system identification such as the maximum likelihood estimation method, modified maximum likelihood estimation

method, and Kalman filtering method [13–15] have been widely used in system identification. However, these methods require the known structure of the mathematical model, which is difficult for the unmanned quadrotor to know its accurate mathematical model, considering uncertain disturbances in the complex environment.

As a nonconventional method, the neural network (NN) does not explicitly need an accurate mathematical model [16–19]. Based on that, some works on NN-based MPC have improved modeling performance for many different nonlinear systems. In [20], Nikdel et al. presented a model predictive controller with the feedforward neural network (FNN) model for a single degree of freedom-rotary manipulator achieving well closed-loop performance. In [21], a model predictive controller based on the FNN model was utilized to improve the transient stability of a power system. However, the main drawback of FNNs considered in [20, 21] is that they essentially own static input-to-output maps and their capability of representing complex and time-varying nonlinear systems is limited. Besides, they can only provide predetermined finite predictions [22, 23]. To overcome the above drawbacks, a recurrent neural network (RNN) is introduced for MPC, which can capture the system behavior dynamically and provide long-range predictions even in the presence of disturbances [24–26]. In a RNN, the current control signal and state are given as the inputs to obtain the outputs which are fed back as new states. Therefore, RNN is more suitable to model the quadrotor system for MPC in this paper.

On the other hand, the performance of MPC also relies on the local optimization problem in MPC, which can be solved by either analytical or numerical solutions. The solution depends on the structure of the cost function. For the tracking problem of the multiquadrotor formation in this paper, the corresponding cost function is complex and non-differentiable. Traditional search techniques using the characteristics of the problem to determine the next sampling point (gradients, Hessians, and linearity) are not feasible, because the characteristics of tracking performance are affected by mismatched disturbances. Nevertheless, stochastic search techniques, such as the genetic algorithm (GA), particle swarm optimization (PSO) algorithm, and chaotic predator prey biogeography-based optimization (CPPBBO) algorithm [27–29], make no such solutions. Instead, the next point is determined by the stochastic decision rules. Therefore, this paper proposes a stochastic search technique based on the differential evolution (DE) algorithm [30] to solve the optimization problem in MPC, which has been successfully applied in many fields recently [31, 32].

Although the concept of NN-based MPC is not new [33, 34], the application of RNN-based MPC in multiquadrotor formation flight is rather scarce. Motivated by this fact, this paper presents the dynamical RNN-based MPC strategy for the multiquadrotor formation flight. The main contributions are summarized as follows:

- (i) A distributed hierarchical control system with the translation subsystem and rotational subsystem is proposed to simplify the control process for the unmanned quadrotor

- (ii) To overcome the drawbacks of conventional MPC, RNN is introduced as the predictive model in MPC to deal with the mismatched modeling for both subsystems
- (iii) The composite control structure based on RNN and MPC is novel for multiquadrotor formation flight, which shows a good control performance in simulation results as well as in theoretical analysis

The remainder of this paper is organized as follows. In Section 2, the problem formulation is introduced briefly. Section 3 models the MPC-based tracking problem. In Section 4, the RNN-based prediction model is presented. The control structure is provided in Section 5. In Section 6, the main results concerning stability of the proposed control scheme are analyzed. Simulation results are shown in Section 7. Section 8 includes the conclusion.

2. Problem Formulation

Suppose that there are N ($N > 1$) unmanned quadrotors with the same dynamic characteristics moving in \mathbb{R}^3 , which together compose a multiquadrotor formation system. In what follows, the unmanned quadrotor is a miniature four-rotor quadrotor [35]. Each quadrotor can be viewed as a 6 Degree-of-Freedom (DOF) rigid body with generalized coordinates $\mathbf{x}_i = [x_i, y_i, z_i, \phi_i, \theta_i, \psi_i]^T \in \mathbb{R}^6$, where $[x_i, y_i, z_i]^T$ denotes the i th quadrotor's positions with reference to the earth-fixed frame (E-frame) and $[\phi_i, \theta_i, \psi_i]^T$ represents the attitudes with reference to the body-fixed frame (B-Frame) ($i = 1, \dots, N$). Furthermore, the flight motion is controlled by a total thrust U_i^1 and three control torques U_i^2, U_i^3 , and U_i^4 . And the dynamical model of the i th quadrotor can be defined by the nonlinear state-space function $\dot{\mathbf{x}}_i = f_{\mathbf{x}_i}(\mathbf{x}_i, \dot{\mathbf{x}}_i, U_i^1, U_i^2, U_i^3, U_i^4)$. To simplify the control process, the quadrotor system can be decomposed into two subsystems in Figure 1: the translational subsystem and the rotational subsystem [36].

Define the system state of the translational subsystem as $\mathbf{x}_i^\zeta = [x_i, \dot{x}_i, y_i, \dot{y}_i, z_i, \dot{z}_i]^T \in \mathbb{R}^6$ and its control input as $\mathbf{u}_i^\zeta = [U_i^1, u_i^x, u_i^y]^T \in \mathbb{R}^3$ [37], where u_i^x and u_i^y are two intermediate control inputs, given by

$$\begin{cases} u_i^x = \cos \phi_i \sin \theta_i \cos \psi_i + \sin \phi_i \sin \psi_i, \\ u_i^y = \cos \phi_i \sin \theta_i \sin \psi_i - \sin \phi_i \cos \psi_i. \end{cases} \quad (1)$$

Then, the dynamics of translational subsystem can be described as

$$\begin{cases} \dot{\mathbf{x}}_i^\zeta = f_i^\zeta(\mathbf{x}_i^\zeta, \mathbf{u}_i^\zeta) + \mathbf{d}_i^\zeta, \\ \dot{\mathbf{y}}_i^\zeta = \mathbf{C}^\zeta \mathbf{x}_i^\zeta, \end{cases} \quad (2)$$

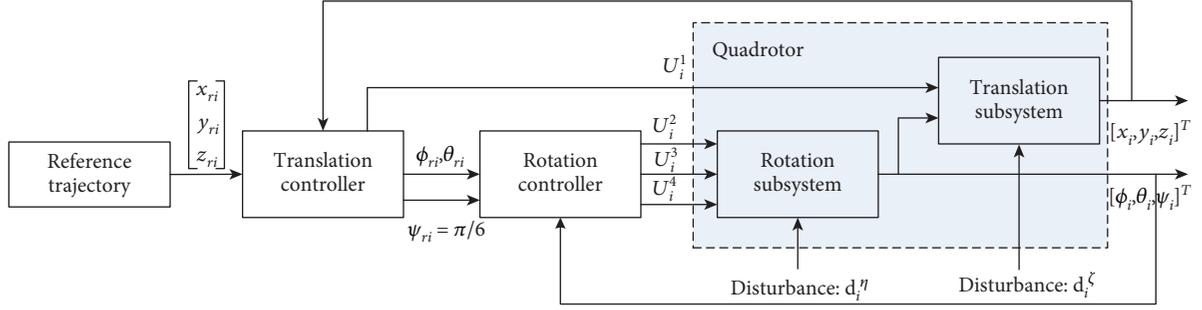


FIGURE 1: The distributed control scheme for each quadrotor.

where $f_i^\zeta: \mathbb{R}^6 \times \mathbb{R}^3 \rightarrow \mathbb{R}^6$ is a nonlinear function, $\mathbf{d}_i^\zeta \in \mathbb{R}^6$ is the bounded disturbance, $\mathbf{y}_i^\zeta = [x_i, y_i, z_i]^T$ is the translational subsystem output, and

$$\mathbf{C}^\zeta = \begin{bmatrix} 1, 0, 0, 0, 0, 0 \\ 0, 0, 1, 0, 0, 0 \\ 0, 0, 0, 0, 1, 0 \end{bmatrix}. \quad (3)$$

Similarly, the system state and control input of rotational subsystem are represented by $\mathbf{x}_i^\eta = [\phi_i, \dot{\phi}_i, \theta_i, \dot{\theta}_i, \psi_i, \dot{\psi}_i]^T \in \mathbb{R}^6$ and $\mathbf{u}_i^\eta = [U_i^2, U_i^3, U_i^4]^T \in \mathbb{R}^3$, respectively [37]. Then, the dynamic equation is described as

$$\begin{cases} \dot{\mathbf{x}}_i^\eta = f_i^\eta(\mathbf{x}_i^\eta, \mathbf{u}_i^\eta) + \mathbf{d}_i^\eta, \\ \mathbf{y}_i^\eta = \mathbf{C}^\eta \mathbf{x}_i^\eta, \end{cases} \quad (4)$$

where $f_i^\eta: \mathbb{R}^6 \times \mathbb{R}^3 \rightarrow \mathbb{R}^6$ is a nonlinear function, $\mathbf{d}_i^\eta \in \mathbb{R}^6$ is the bounded disturbance, $\mathbf{y}_i^\eta = [\phi_i, \theta_i, \psi_i]^T$ is the rotational subsystem output, and

$$\mathbf{C}^\eta = \begin{bmatrix} 1, 0, 0, 0, 0, 0 \\ 0, 0, 1, 0, 0, 0 \\ 0, 0, 0, 0, 1, 0 \end{bmatrix}. \quad (5)$$

Based on (2) and (4), both subsystems can be expressed in the discrete-time domain as follows:

$$\begin{cases} \mathbf{x}_i^\zeta(k+1) = f_i^\zeta(\mathbf{x}_i^\zeta(k), \mathbf{u}_i^\zeta(k), T) + \mathbf{d}_i^\zeta(k), \\ \mathbf{y}_i^\zeta(k) = \mathbf{C}^\zeta \mathbf{x}_i^\zeta(k), \\ \mathbf{x}_i^\eta(k+1) = f_i^\eta(\mathbf{x}_i^\eta(k), \mathbf{u}_i^\eta(k), T) + \mathbf{d}_i^\eta(k), \\ \mathbf{y}_i^\eta(k) = \mathbf{C}^\eta \mathbf{x}_i^\eta(k), \end{cases} \quad (6)$$

where f_i^ζ and f_i^η are discrete nonlinear functions and T is the sampling time.

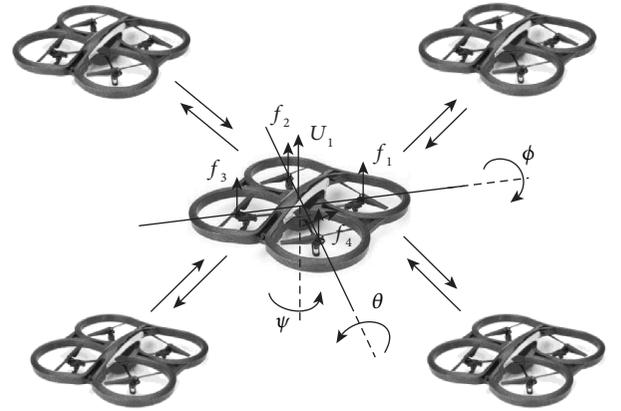


FIGURE 2: Multiquadrotor system.

As shown in Figure 2, the leader-follower structure is introduced for the multiquadrotor system. Define a quadrotor as the leader in the formation, and its state vector is represented by \mathbf{x}_L . Each quadrotor is equipped with a complete set of communication equipment and controllers. Assuming that the information exchange topology is undirected and connected, each follower receives the control commands from the leader directly or indirectly, then executes the corresponding maneuvers to keep the relative state to the leader, where $\Delta \mathbf{x}_{i,L} = (\mathbf{x}_i - \mathbf{x}_L)$ denotes the relative state of the i th follower.

The distance between the i th quadrotor and the r th quadrotor can be described as follows:

$$d_{ir}(t) = \sqrt{(x_i(t) - x_r(t))^2 + (y_i(t) - y_r(t))^2 + (z_i(t) - z_r(t))^2}. \quad (7)$$

In order to avoid collision among quadrotors, $d_{ir}(t)$ should be greater than the safe anti-collision distance D_{safe} , i.e.,

$$d_{ir}(t) \geq D_{\text{safe}}, \quad \forall t \in [0, T], \forall i \neq r, i, r \in \{1, \dots, N\}. \quad (8)$$

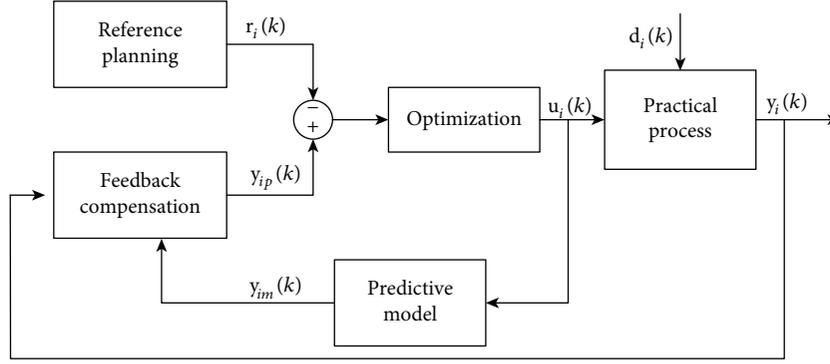


FIGURE 3: Flowchart of typical MPC.

Also, to ensure the real-time direct communication of adjacent quadrotors, $d_{ir}(t)$ would be less than the communication distance D_{com} , i.e.,

$$d_{ir}(t) \leq D_{com}, \quad \forall t \in [0, T], \exists i, r \in \{1, \dots, N\}. \quad (9)$$

3. Model Predictive Control

Generally, MPC is a control process based on predicting future outputs and obtaining nonlinear model outputs at each time horizon [38]. In this paper, a distributed MPC feedback system is designed for each quadrotor in Figure 3. It is mainly composed of the predictive model, reference planning, feedback compensation, and rolling optimization, where $\mathbf{u}_i(k)$, $\mathbf{y}_i(k)$, $\mathbf{r}_i(k)$, $\mathbf{y}_{im}(k)$, $\mathbf{y}_{ip}(k)$, and $\mathbf{d}_i(k)$ denote the control input, output state, reference, predictive output, revised predictive output, and disturbance input, respectively.

At each time horizon, the predictive output $\mathbf{y}_{im}(k)$ is obtained by the predictive model, to determine the revised predictive output $\mathbf{y}_{ip}(k)$ by feedback compensation. Then, the control input $\mathbf{u}_i(k)$ is optimally determined by comparing the revised predictive output $\mathbf{y}_{ip}(k)$ and the reference $\mathbf{r}_i(k)$. Based on that, the system output $\mathbf{y}_i(k)$ is obtained. Meanwhile, the next time horizon is starting and the new predictive output $\mathbf{y}_{im}(k+1)$ is predicted by the predictive model. The whole process will be online iterative until the output of system reaches the reference \mathbf{r}_i .

Due to the nonlinearity and time-variant characteristic, the predictive output $\mathbf{y}_{im}(k)$ is hard to match the real output $\mathbf{y}_i(k)$. The predictive errors between the actual output and predictive output are used to revise the predictive output. The formulation can be described as

$$\mathbf{y}_{ip}(k) = \mathbf{y}_{im}(k) + E(\mathbf{y}_i(k-1) - \mathbf{y}_{im}(k-1)), \quad (10)$$

where $E = I_{3 \times 3}$ is the weight coefficient matrix.

Considering that the quadrotor system decomposed into two subsystems, MPC is introduced separately for both subsystems. In the outer loop, the translational subsystem is controlled to follow a sequence of reference $[\mathbf{x}_{ri}, \mathbf{y}_{ri}, \mathbf{z}_{ri}]^T$. The optimal control inputs U_i^1 , u_i^x , and u_i^y are obtained by minimizing the trajectory tracking errors. With $\psi_{ri} = \pi/6$, the

desired attitudes θ_{ri} and ϕ_{ri} are calculated using (1). Then, the rotational subsystem's attitudes $[\phi_i, \theta_i, \psi_i]^T$ are adjusted to achieve the reference $[\phi_{ri}, \theta_{ri}, \psi_{ri}]^T$ in the inner loop. By minimizing attitude errors, the optimal controls U_i^2 , U_i^3 , and U_i^4 are calculated. Finally, the optimal control inputs U_i^1 , U_i^2 , U_i^3 , and U_i^4 are applied to the i th quadrotor.

Consider the MPC formulation based on the constrained finite-horizon optimization of tracking problem for translational subsystem [39]:

$$\begin{aligned} \min J_i^\zeta(k) = & \sum_{j=1}^{H_p} \left\| \mathbf{y}_{ip}^\zeta(k+j) - \mathbf{r}_i^\zeta(k+j) \right\|_{\mathbf{Q}^\zeta}^2 \\ & + \rho \sum_{j=1}^{H_u} \left\| \Delta \mathbf{u}_i^\zeta(k+j-1) \right\|_{\mathbf{R}^\zeta}^2, \end{aligned} \quad (11a)$$

subject to

$$\mathbf{y}_{\min}^\zeta \leq \mathbf{y}_{ip}^\zeta(k+j) \leq \mathbf{y}_{\max}^\zeta, \quad (11b)$$

$$\mathbf{u}_{\min}^\zeta \leq \mathbf{u}_i^\zeta(k+j-1) \leq \mathbf{u}_{\max}^\zeta, \quad (11c)$$

$$\Delta \mathbf{u}_{\min}^\zeta \leq \Delta \mathbf{u}_i^\zeta(k+j-1) \leq \Delta \mathbf{u}_{\max}^\zeta, \quad (11d)$$

where $\mathbf{y}_{ip}^\zeta(k+j)$ is the revised predictive output and $\mathbf{u}_i^\zeta(k+j-1)$ and $\Delta \mathbf{u}_i^\zeta(k+j-1)$ are control inputs and incremental control moves, respectively. $\mathbf{r}_i^\zeta(k+j) = [\mathbf{x}_{ri}(k+j), \mathbf{y}_{ri}(k+j), \mathbf{z}_{ri}(k+j)]^T$ is the reference position. $\mathbf{y}_{\min}^\zeta \geq \mathbf{y}_{\max}^\zeta$ and $\mathbf{u}_{\min}^\zeta \geq \mathbf{u}_{\max}^\zeta$ are the upper and lower bounds of the system states and control inputs, respectively. $\|\cdot\|_{\mathbf{Q}^\zeta}$ and $\|\cdot\|_{\mathbf{R}^\zeta}$ are two 2-norms weighted by positive definite matrices \mathbf{Q}^ζ and \mathbf{R}^ζ , respectively. H_p is the prediction horizon, H_u is the control prediction horizon ($H_p \geq H_u$), and ρ represents the penalty factor.

Remark 1. The tracking problem of the quadrotor formation is decomposed into two phases. Firstly, the leader tracks a given trajectory in a certain state. Then, the followers regard the leader as the trace point to track and further achieve keeping formation. Therefore, if the i th quadrotor is the

leader, $\mathbf{r}_i^\zeta(k+j)$ denotes reference trajectory; otherwise, it represents the position of the leader for the followers.

Similarly, for the rotational subsystem, the MPC optimization problem is described as

$$\begin{aligned} \min J_i^\eta(k) = & \sum_{j=1}^{H_p} \|\mathbf{y}_{ip}^\eta(k+j) - \mathbf{r}_i^\eta(k+j)\|_{\mathbf{Q}^\zeta}^2 \\ & + \rho \sum_{j=1}^{H_u} \|\Delta \mathbf{u}_i^\eta(k+j-1)\|_{\mathbf{R}^\zeta}^2, \end{aligned} \quad (12a)$$

subject to

$$\mathbf{y}_{\min}^\eta \leq \mathbf{y}_{ip}^\eta(k+j) \leq \mathbf{y}_{\max}^\eta, \quad (12b)$$

$$\mathbf{u}_{\min}^\eta \leq \mathbf{u}_i^\eta(k+j-1) \leq \mathbf{u}_{\max}^\eta, \quad (12c)$$

$$\Delta \mathbf{u}_{\min}^\eta \leq \Delta \mathbf{u}_i^\eta(k+j-1) \leq \Delta \mathbf{u}_{\max}^\eta, \quad (12d)$$

where $\mathbf{r}_i^\eta(k+j) = [\phi_{ri}(k+j), \theta_{ri}(k+j), \psi_{ri}(k+j)]^T$ represents the reference attitude.

In addition, the nonlinear constraints of (11a) and (12a) should be considered further. A feasible approach is to introduce a penalty cost, transforming the constrained problem to an alternative unconstrained form [39]. Taking the translational subsystem for example, we substitute $\mathbf{I}_1^\zeta(k+j) = \mathbf{y}_{\max}^\zeta - \mathbf{y}_{ip}^\zeta(k+j)$ and $\mathbf{I}_2^\zeta(k+j) = \mathbf{y}_{ip}^\zeta(k+j) - \mathbf{y}_{\min}^\zeta$ (inequality constraint (11b)), $\mathbf{m}_1^\zeta(k+j-1) = \mathbf{u}_{\max}^\zeta - \mathbf{u}_i^\zeta(k+j-1)$ and $\mathbf{m}_2^\zeta(k+j-1) = \mathbf{u}_i^\zeta(k+j-1) - \mathbf{u}_{\min}^\zeta$ (inequality constraint (11c)), and $\mathbf{h}_1^\zeta(k+j-1) = \Delta \mathbf{u}_{\max}^\zeta - \Delta \mathbf{u}_i^\zeta(k+j-1)$ and $\mathbf{h}_2^\zeta(k+j-1) = \Delta \mathbf{u}_i^\zeta(k+j-1) - \Delta \mathbf{u}_{\min}^\zeta$ (inequality constraint (11d)), then the penalty cost function can be defined as follows:

$$\begin{aligned} \bar{J}_i^\zeta(k) = & J_i^\zeta(k) + \mu \sum_{j=1}^{H_p} \left(\left\| \mathbf{I}_1^\zeta(k+j) \right\|_{\mathbf{S}(\mathbf{I}_1^\zeta(k+j))}^2 + \left\| \mathbf{I}_2^\zeta(k+j) \right\|_{\mathbf{S}(\mathbf{I}_2^\zeta(k+j))}^2 \right) \\ & + \lambda \sum_{j=1}^{H_u} \left(\left\| \mathbf{m}_1^\zeta(k+j-1) \right\|_{\mathbf{S}(\mathbf{m}_1^\zeta(k+j-1))}^2 + \left\| \mathbf{m}_2^\zeta(k+j-1) \right\|_{\mathbf{S}(\mathbf{m}_2^\zeta(k+j-1))}^2 \right) \\ & + \alpha \sum_{j=1}^{H_u} \left(\left\| \mathbf{h}_1^\zeta(k+j-1) \right\|_{\mathbf{S}(\mathbf{h}_1^\zeta(k+j-1))}^2 + \left\| \mathbf{h}_2^\zeta(k+j-1) \right\|_{\mathbf{S}(\mathbf{h}_2^\zeta(k+j-1))}^2 \right), \end{aligned} \quad (13)$$

where

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}, \quad (14)$$

with $\mathbf{a} = [a_1, a_2, a_3]^T$ and $s_i = 10^5$, if $a_i < 0$; otherwise, $s_i = 0$ and $i = 1, 2, 3$. The introduction of $\mathbf{S}(\mathbf{a})$ makes it possible to consider the inequality constraints during the optimization.

Similarly, the penalty cost function of the rotational subsystem can be given by

$$\begin{aligned} \bar{J}_i^\eta(k) = & J_i^\eta(k) + \mu \sum_{j=1}^{H_p} \left(\left\| \mathbf{I}_1^\eta(k+j) \right\|_{\mathbf{S}(\mathbf{I}_1^\eta(k+j))}^2 + \left\| \mathbf{I}_2^\eta(k+j) \right\|_{\mathbf{S}(\mathbf{I}_2^\eta(k+j))}^2 \right) \\ & + \lambda \sum_{j=1}^{H_u} \left(\left\| \mathbf{m}_1^\eta(k+j-1) \right\|_{\mathbf{S}(\mathbf{m}_1^\eta(k+j-1))}^2 + \left\| \mathbf{m}_2^\eta(k+j-1) \right\|_{\mathbf{S}(\mathbf{m}_2^\eta(k+j-1))}^2 \right) \\ & + \alpha \sum_{j=1}^{H_u} \left(\left\| \mathbf{h}_1^\eta(k+j-1) \right\|_{\mathbf{S}(\mathbf{h}_1^\eta(k+j-1))}^2 + \left\| \mathbf{h}_2^\eta(k+j-1) \right\|_{\mathbf{S}(\mathbf{h}_2^\eta(k+j-1))}^2 \right). \end{aligned} \quad (15)$$

Based on (8) and (9), the constraints of D_{safe} and D_{com} are also taken into account. The corresponding auxiliary optimization term is described as

$$\begin{aligned} L(k) = & \sigma \cdot \sum_{j=1}^{H_p} \sum_{i \neq r} [\max(0, D_{\text{safe}} - d_{ir}(k+j)) \\ & + \max(0, d_{ir}(k+j) - D_{\text{com}})]^2, \end{aligned} \quad (16)$$

where $\sigma = 10^5$ and $L(k)$ as a penalty term is added in (13) according to the flight conditions of the multiquadrotor.

Remark 2. The cost functions of (13) and (15) are complex and nondifferentiable. So the optimization problem using traditional search techniques is not feasible. In this study, the DE algorithm [30] as a probabilistic search technique is proposed to solve the optimization problem for MPC. The corresponding details will be presented in Section 5.

4. RNN-Based Prediction Models

Although the feedback mechanism of MPC tolerates some model mismatches for multiquadrotor system, MPC still demands that the prediction model be sufficiently accurate. In this section, RNN is considered as the prediction model for MPC, due to its superiority in comparison with other conventional modeling methods [40].

4.1. Dynamical RNN. Generally, in a RNN, there is usually a feedback from the hidden layer output to the hidden layer input. This recurrent connection allows the NN to detect and generate time-varying patterns [41]. In this paper, the RNN predictive model is separately developed for the translational subsystem and rotational subsystem. Both subsystems belong to the class of multi-input/multi-output square nonlinear dynamical systems (that is, the systems with as many inputs as outputs), which have the general form of

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \sum_{i=1}^m g_i(\mathbf{x}) u_i = f(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \mathbf{u}, \quad (17)$$

$$\mathbf{y} = \mathbf{C} \mathbf{x}, \quad (18)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the control input,

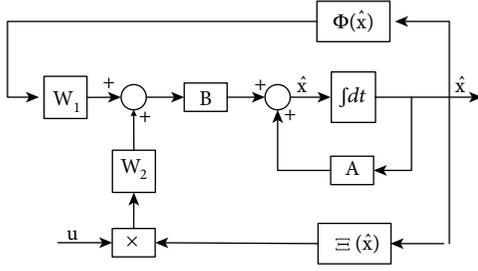


FIGURE 4: The block diagram of the dynamical RNN.

$\mathbf{y} \in \mathbb{R}^m$ is the system output, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a smooth function, \mathbf{C} is a $m \times n$ constant matrix, and $\mathbf{G}(\mathbf{x})$ is a matrix with columns g_i ; note that f and g_i contain parametric uncertainties.

This paper deals with the control of both subsystems described by (17), i.e., $m = 3$ and $n = 6$. However, due to the complex environment and disturbances, the system (17) is difficult to use if the functions f and g_i cannot be accurately known. Therefore, in order to provide a solution to this problem, a dynamical RNN is chosen to obtain a more accurate model of (17) in Figure 4.

This RNN contains $n = 6$ dynamical hidden neurons, i.e., a number of neurons equal to the dimension of the state vector of subsystem dynamics. And it can be described by the following system [42, 43]:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{W}_1\Phi(\hat{\mathbf{x}}) + \mathbf{B}\mathbf{W}_2\Xi(\hat{\mathbf{x}})\mathbf{u}, \quad (19)$$

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}}, \quad (20)$$

with the state $\hat{\mathbf{x}} \in \mathbb{R}^6$, the input $\mathbf{u} \in \mathbb{R}^3$, the output $\hat{\mathbf{y}} \in \mathbb{R}^3$, \mathbf{W}_1 is a 6×6 matrix of adjustable synaptic weights, \mathbf{A} is a 6×6 diagonal matrix with negative eigenvalues a_i , \mathbf{B} is a 6×6 diagonal matrix with scalar elements b_i , and \mathbf{W}_2 is a 6×6 diagonal matrix with adjustable synaptic weights. $\Phi(\hat{\mathbf{x}})$ is a 6-dimensional vector with elements of $\varphi(\hat{x}_i)$, and $\Xi(\hat{\mathbf{x}})$ is a 6×3 matrix with elements of $\xi(\hat{x}_i)$. $\varphi(\hat{x}_i)$ and $\xi(\hat{x}_i)$ are represented by sigmoids of the form

$$\begin{aligned} \varphi(\hat{x}_i) &= \frac{m_1}{1 + e^{-\delta_1 x \lambda_i}}, \\ \xi(\hat{x}_i) &= \frac{m_2}{1 + e^{-\delta_2 x \lambda_i}} + \kappa, \end{aligned} \quad (21)$$

$i = 1, \dots, 6,$

where m_1 , m_2 , δ_1 , and δ_2 are constants representing the bound and slope of the sigmoid's curvature and $\kappa > 0$ is a constant that shifts the sigmoid, such that $\xi(\hat{x}_i) > 0$ for all $i = 1, \dots, 6$. In this paper, the designed parameters are chosen as $m_1 = 0.5$, $m_2 = 1.3$, $\delta_1 = \delta_2 = 0.1$, and $\kappa = 0.4$.

4.2. Online Updating for RNN. Based on the above analysis, suppose that both subsystems of the quadrotor can be accurately described by the dynamical RNN (19) plus a modelling

error term $\omega(\mathbf{x}, \mathbf{u})$ as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{W}_1^*\Phi(\mathbf{x}) + \mathbf{B}\mathbf{W}_2^*\Xi(\mathbf{x})\mathbf{u} + \omega(\mathbf{x}, \mathbf{u}), \quad (22)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (23)$$

where \mathbf{W}_1^* and \mathbf{W}_2^* are the corresponding weight values.

Note that the optimal control input is applied to both the real system and the predictive model. Define the error between the predictive model (19) and the real system (22) as $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y} = \mathbf{C}(\hat{\mathbf{x}} - \mathbf{x})$. Assuming $\omega(\mathbf{x}, \mathbf{u})$ is zero, we obtain

$$\dot{\mathbf{e}} = \mathbf{C}\mathbf{A}\mathbf{C}^{-1}\mathbf{e} + \mathbf{C}\tilde{\mathbf{B}}\tilde{\mathbf{W}}_1\Phi(\mathbf{x}) + \mathbf{C}\tilde{\mathbf{B}}\tilde{\mathbf{W}}_2\Xi(\mathbf{x})\mathbf{u}, \quad (24)$$

where $\tilde{\mathbf{W}}_1 = \mathbf{W}_1 - \mathbf{W}_1^*$ and $\tilde{\mathbf{W}}_2 = \mathbf{W}_2 - \mathbf{W}_2^*$, which are substituted online through the appropriate updating laws.

To obtain the stable updating laws, the Lyapunov synthesis method is introduced. Consider a Lyapunov function candidate as

$$V = \left(\frac{1}{2}\right) \cdot \left(\mathbf{e}^T \mathbf{P} \mathbf{e} + \text{tr}\left\{\tilde{\mathbf{W}}_1^T \tilde{\mathbf{W}}_1\right\} + \text{tr}\left\{\tilde{\mathbf{W}}_2^T \tilde{\mathbf{W}}_2\right\}\right), \quad (25)$$

where $\mathbf{P} > 0$ satisfies the Lyapunov equation $\mathbf{P}\mathbf{C}\mathbf{A} + \mathbf{A}^T\mathbf{C}^T\mathbf{P} = -\mathbf{C}$.

Differentiating (25) along the solution of (24), there exists

$$\begin{aligned} \dot{V} = & -\frac{1}{2} \cdot \mathbf{e}^T \mathbf{e} + \Phi^T(\mathbf{x})\tilde{\mathbf{W}}_1^T \mathbf{B}\mathbf{C}^T \mathbf{P} \mathbf{e} + \mathbf{u}^T \Xi^T(\mathbf{x})\tilde{\mathbf{W}}_2 \mathbf{B}\mathbf{C}^T \mathbf{P} \mathbf{e} \\ & + \text{tr}\left\{\dot{\tilde{\mathbf{W}}}_1^T \tilde{\mathbf{W}}_1\right\} + \text{tr}\left\{\dot{\tilde{\mathbf{W}}}_2^T \tilde{\mathbf{W}}_2\right\}. \end{aligned} \quad (26)$$

By choosing

$$\begin{aligned} \text{tr}\left\{\dot{\tilde{\mathbf{W}}}_1^T \tilde{\mathbf{W}}_1\right\} &= -\Phi^T(\mathbf{x})\tilde{\mathbf{W}}_1^T \mathbf{B}\mathbf{C}^T \mathbf{P} \mathbf{e}, \\ \text{tr}\left\{\dot{\tilde{\mathbf{W}}}_2^T \tilde{\mathbf{W}}_2\right\} &= -\mathbf{u}^T \Xi^T(\mathbf{x})\tilde{\mathbf{W}}_2 \mathbf{B}\mathbf{C}^T \mathbf{P} \mathbf{e}, \end{aligned} \quad (27)$$

then

$$\dot{V} = -\frac{1}{2} \cdot \mathbf{e}^T \mathbf{e} = -\frac{1}{2} \cdot \|\mathbf{e}\|^2 \leq 0. \quad (28)$$

Consequently, based on (27), the weight matrices can be updated online over time horizons.

Theorem 1. *The updating laws (27) can guarantee the following properties:*

- (i) $\lim_{t \rightarrow \infty} \mathbf{e}(t) = 0$,
- (ii) $\lim_{t \rightarrow \infty} \dot{\tilde{\mathbf{W}}}_1(t) = 0$,
- (iii) $\lim_{t \rightarrow \infty} \dot{\tilde{\mathbf{W}}}_2(t) = 0$.

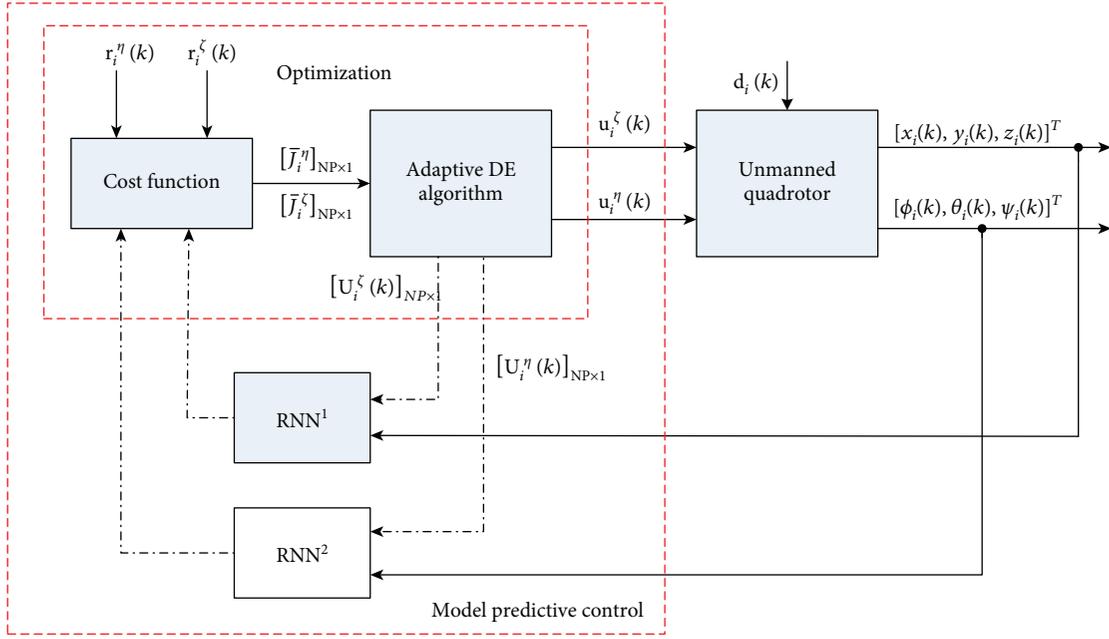


FIGURE 5: MPC structure based on RNN for each unmanned quadrotor.

Proof 1. Since \dot{V} in (28) is negative semidefinite, we have $V \in L_\infty$, which implies $\mathbf{e} \in L_\infty$, $\tilde{\mathbf{W}}_1 \in L_\infty$, and $\tilde{\mathbf{W}}_2 \in L_\infty$. Furthermore, $\tilde{\mathbf{x}} = \mathbf{x} + C^{-1}\mathbf{e}$ is bounded. Since V is a nonincreasing function of time and bounded from below, there exists $\lim_{t \rightarrow \infty} V(t) = V(\infty)$. By integrating \dot{V} from 0 to ∞ , we obtain

$$\int_0^\infty \frac{1}{2} \|\mathbf{e}\|^2 dt = V(\infty) - V(0) < \infty, \quad (29)$$

which implies $\mathbf{e} \in L_2$. By definition, $\varphi(x_i)$ and $\xi(x_i)$ are bounded for all x_i and all inputs to RNN are also bounded by assumption. From (24), we conclude $\dot{\mathbf{e}} \in L_\infty$. Since $\mathbf{e} \in L_2 \cap L_\infty$ and $\dot{\mathbf{e}} \in L_\infty$, based on Barbalat's lemma, there exists $\lim_{t \rightarrow \infty} \mathbf{e}(t) = 0$. Due to the boundedness of \mathbf{u} , $\Phi(\mathbf{x})$, $\Xi(\mathbf{x})$, and the convergence of $\mathbf{e}(t)$ to zero, we have that $\dot{\tilde{\mathbf{W}}}_1$ and $\dot{\tilde{\mathbf{W}}}_2$ also convergence to zero.

Remark 3. Combined with the MPC, updating online for RNN is executed over time horizons. With the actual outputs $\mathbf{y}_i^\zeta(k)$ and $\mathbf{y}_i^\eta(k)$ obtained at the k th time horizon, the weight matrices of the RNNs are updated online to predict the subsystem output for the next time horizon.

5. Control Structure

The composite control structure based on RNN and MPC is proposed in Figure 5 for each quadrotor, which consists of three main parts: the predictive model based on RNN, the designed cost function, and the evolutionary algorithm. In this figure, the RNN¹ and RNN² blocks represent the predictive models of the translational subsystem and the rotational subsystem, respectively. $\tilde{\mathbf{U}}_i^\zeta(k) \triangleq \arg \min \bar{J}_i^\zeta(k)$ and $\tilde{\mathbf{U}}_i^\eta(k) \triangleq$

$\arg \min \bar{J}_i^\eta(k)$ are the optimal control sequence of the translational subsystem and the rotational subsystem, respectively, where the first elements, i.e., $\tilde{\mathbf{u}}^\zeta(k)$ and $\tilde{\mathbf{u}}^\eta(k)$, are taken as the control inputs for both subsystems.

The evolutionary algorithm to introduce in this paper is the DE algorithm. It is an optimization algorithm which searches randomly in continuous space with three evolutionary operations, namely, mutation, recombination, and selection. Suppose that the initial population has NP individuals inside the given D -dimensional search space. The solution vector in continuous space can be represented by $\mathbf{v}_i^G = (v_{i1}, v_{i2}, \dots, v_{iD})$, $i = 1, \dots, \text{NP}$, $G = 1, \dots, \text{NC}$ [30]. The optimization goal of the DE algorithm is to find the proper control variable that minimizes the cost (13) and (15).

5.1. Mutation Operation. As shown in Figure 6, for each \mathbf{v}_i^G in the parent population, the mutation individual $\hat{\mathbf{v}}_i^{G+1}$ is produced by

$$\hat{\mathbf{v}}_i^{G+1} = \mathbf{v}_{r_1}^G + F \times (\mathbf{v}_{r_2}^G - \mathbf{v}_{r_3}^G), \quad (30)$$

where r_1 , r_2 , and r_3 are randomly chosen and different from the running index i and F is a real constant scaling factor within $[0, 1]$, which controls the amplification of the differential variation $(\mathbf{v}_{r_2}^G - \mathbf{v}_{r_3}^G)$.

5.2. Recombination Operation. For each \mathbf{v}_i^G , a trial individual $\bar{\mathbf{v}}_i^{G+1}$ is generated by coping components from the mutation vector $\hat{\mathbf{v}}_i^{G+1}$ and the target vector \mathbf{v}_i^G in dependence, as shown

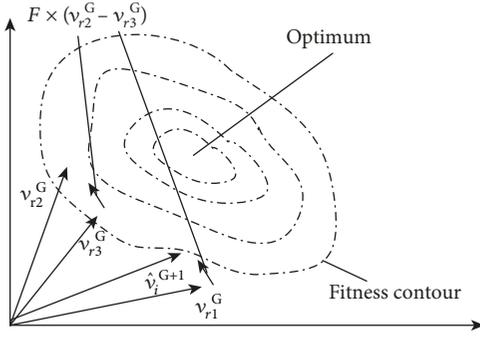


FIGURE 6: Mutation process of the DE algorithm.

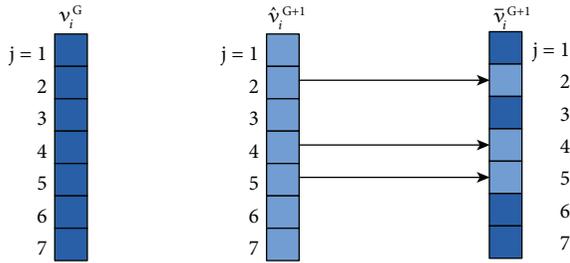


FIGURE 7: Recombination process of the DE algorithm.

in Figure 7. It can be formulated as follows:

$$\bar{v}_{ij}^{G+1} = \begin{cases} \hat{v}_{ij}^{G+1}, & \text{if randb} \leq \text{CR}, \\ v_{ij}^G, & \text{otherwise,} \end{cases} \quad (31)$$

$$i = 1, \dots, \text{NP}, j = 1, \dots, D,$$

where $\text{randb} \in [0, 1]$ is a uniform random number and the recombination control parameter CR is a constant in the interval $[0, 1]$.

5.3. Selection Operation. Offspring \bar{v}_i^{G+1} competes one to one with its parent v_i^G . The selection operation is expressed by

$$v_i^{G+1} = \begin{cases} \bar{v}_i^{G+1}, & \text{if } f(\bar{v}_i^{G+1}) \leq f(v_i^G), \\ v_i^G, & \text{otherwise.} \end{cases} \quad (32)$$

This selection scheme ensures that the best control variable with the minimum cost can be retained when moving from one generation to the next, which results in a fast convergence behavior.

Generally, the DE algorithm has been successfully applied in some fields due to its high search accuracy, robustness, and good convergence speed. However, when the optimization problem of multi-UAV tracking is complex and time-varying, selecting suitable mutation strategies and control parameters at different evolution stages is difficult. Therefore, in this paper, we adopt the ZEPDE algorithm [44], where the appropriate mutation strategy can be gradually adjusted by a roulette wheel, and suitable combinations of F and CR can be generated by zoning evolution. Each individual has

its own control parameter combination and mutation strategy. And the corresponding details can be found in [44].

The control algorithm can be summarized as follows:

Step 1. Load the state of the i th quadrotor and its corresponding reference. Initialize the solution and generate NP individuals randomly.

Step 2. The decoded values of each individual $U_i^\zeta(k)$ and $U_i^\eta(k)$ in the population carried out the RNN¹ and RNN² models to predict the values of $[x_i(k+j), y_i(k+j), z_i(k+j)]^T$ and $[\phi_i(k+j), \theta_i(k+j), \psi_i(k+j)]^T$, $j = 1, \dots, H_p$. Whereas, consecutive inputs to the RNN models can change their internal state in a detrimental way. Then, the DE algorithm is executed at fault due to this situation, leading the optimal control variables to not be found. To avoid this problem, the RNN models should return their initial condition before each individual carries out the RNN models.

Step 3. Calculate the fitness value of each individual using the predicted states of $[x_i(k+j), y_i(k+j), z_i(k+j)]^T$ and $[\phi_i(k+j), \theta_i(k+j), \psi_i(k+j)]^T$, $j = 1, \dots, H_p$.

Step 4. The individual which has the best fitness value is recorded. Perform the selection method based on the fitness of individuals, and in this way, the population remains better individuals.

Step 5. Execute mutation, recombination to create new individuals in the population.

Step 6. Produce the prediction values for each new individual in the population and calculate their fitness values based on procedures in Steps 2 and 3. The best individual that has been recorded previously is located in the new generation instead of the individual with the worst fitness value.

Step 7. Repeat Steps 4-7 until convergence criteria of the evaluation function are met.

Step 8. The first elements $\tilde{u}_i^\zeta(k)$ and $\tilde{u}_i^\eta(k)$ of the best individuals $\tilde{U}_i^\zeta(k)$ and $\tilde{U}_i^\eta(k)$ apply both the RNN models and the real quadrotor system. The corresponding outputs are obtained.

Step 9. Perform the feedback compensation in MPC and online updating for RNN. Then, go to Step 2 for the next time horizon.

6. Stability Analysis

In this section, we first give the stability of the proposed RNN-based MPC. Meanwhile, the convergence analysis of the DE algorithm is also discussed.

6.1. Stability Analysis of RNN-Based MPC. The quadrotor system in this paper is represented in state space, and the cost functions (11a) and (12a) are also based on states. Therefore,

the stability of the proposed predictive scheme is investigated by checking the monotonicity of the cost function with respect to time [39].

Problems (11a) and (12a) can be given in the following general form:

$$J_i(k) = \sum_{j=1}^{H_p} \left\| \mathbf{y}_{ip}(k+j) - \mathbf{r}_i(k+j) \right\|_{\mathbf{Q}}^2 + \rho \sum_{j=1}^{H_u} \left\| \Delta \mathbf{u}_i(k+j-1) \right\|_{\mathbf{R}}^2 \quad (33a)$$

$$= \sum_{j=1}^{H_p} \left\| \mathbf{e}_{ip}(k+j) \right\|_{\mathbf{Q}}^2 + \rho \sum_{j=1}^{H_u} \left\| \Delta \mathbf{u}_i(k+j-1) \right\|_{\mathbf{R}}^2,$$

$$\text{s.t. } \mathbf{y}_{\min} \leq \mathbf{y}_{ip}(k+i) \leq \mathbf{y}_{\max}, \quad (33b)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_i(k+j-1) \leq \mathbf{u}_{\max}, \quad (33c)$$

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u}_i(k+j-1) \leq \Delta \mathbf{u}_{\max}, \quad (33d)$$

$$\mathbf{e}_{ip}(k+H_p+j) = 0, \quad \forall j \geq 1, \quad (33e)$$

$$\Delta \mathbf{u}_i(k+H_u+j) = 0, \quad \forall j \geq 0. \quad (33f)$$

Assume that $\tilde{\mathbf{U}}_i(k) = [\tilde{\mathbf{u}}_i(k), \dots, \tilde{\mathbf{u}}_i(k+H_u-1)]^T$ is the optimal control sequence at time k . Then, the corresponding cost function is

$$\begin{aligned} \tilde{J}_i(k) &= \sum_{j=1}^{H_p} \left\| \tilde{\mathbf{y}}_{ip}(k+j) - \mathbf{r}_i(k+j) \right\|_{\mathbf{Q}}^2 + \rho \sum_{j=1}^{H_u} \left\| \Delta \tilde{\mathbf{u}}_i(k+j-1) \right\|_{\mathbf{R}}^2 \\ &= \sum_{j=1}^{H_p} \left\| \tilde{\mathbf{e}}_{ip}(k+j) \right\|_{\mathbf{Q}}^2 + \rho \sum_{j=1}^{H_u} \left\| \Delta \tilde{\mathbf{u}}_i(k+j-1) \right\|_{\mathbf{R}}^2. \end{aligned} \quad (34)$$

Then, at time $k+1$, based on the optimal control sequence of time k , a suboptimal control sequence $\tilde{\mathbf{U}}_i^*(k+1)$ is postulated as

$$\tilde{\mathbf{U}}_i^*(k+1) = \begin{bmatrix} \tilde{\mathbf{u}}_i^*(k+1) \\ \vdots \\ \tilde{\mathbf{u}}_i^*(k+H_u-1) \\ \tilde{\mathbf{u}}_i^*(k+H_u) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{u}}_i(k) \\ \vdots \\ \tilde{\mathbf{u}}_i(k+H_u-1) \\ \tilde{\mathbf{u}}_i(k+H_u-1) \end{bmatrix}. \quad (35)$$

For the suboptimal control $\tilde{\mathbf{U}}_i^*(k+1)$, the cost function

can be described as

$$\begin{aligned} \tilde{J}_i^*(k+1) &= \sum_{j=2}^{H_p+1} \left\| \tilde{\mathbf{y}}_{ip}^*(k+j) - \mathbf{r}_i(k+j) \right\|_{\mathbf{Q}}^2 \\ &\quad + \rho \sum_{j=1}^{H_u} \left\| \Delta \tilde{\mathbf{u}}_i^*(k+j) \right\|_{\mathbf{R}}^2 \\ &= \sum_{j=2}^{H_p+1} \left\| \tilde{\mathbf{e}}_{ip}^*(k+j) \right\|_{\mathbf{Q}}^2 + \rho \sum_{j=1}^{H_u} \left\| \Delta \tilde{\mathbf{u}}_i^*(k+j) \right\|_{\mathbf{R}}^2. \end{aligned} \quad (36)$$

Assuming that with the same control input, the predictions $\mathbf{y}_{ip}(k+j)$ at time $k+1$ are the same as ones derived at time k , the difference between the $\tilde{J}_i(k)$ and $\tilde{J}_i^*(k+1)$ can be defined as

$$\begin{aligned} \tilde{J}_i^*(k+1) - \tilde{J}_i(k) &= \left\| \tilde{\mathbf{e}}_{ip}^*(k+H_p+1) \right\|_{\mathbf{Q}}^2 \\ &\quad - \left\| \tilde{\mathbf{e}}_{ip}(k+1) \right\|_{\mathbf{Q}}^2 - \rho \left\| \Delta \tilde{\mathbf{u}}_i(k) \right\|_{\mathbf{R}}^2. \end{aligned} \quad (37)$$

Taking into account the terminal equality constraints (33e), it is obvious that $\tilde{\mathbf{e}}_{ip}^*(k+H_p+1) = 0$, then

$$\tilde{J}_i^*(k+1) - \tilde{J}_i(k) = -\left\| \tilde{\mathbf{e}}_{ip}(k+1) \right\|_{\mathbf{Q}}^2 - \rho \left\| \Delta \tilde{\mathbf{u}}_i(k) \right\|_{\mathbf{R}}^2 \leq 0. \quad (38)$$

Furthermore, introduce $\tilde{\mathbf{U}}_i(k+1)$ as the optimal solution further. It is obvious that $\tilde{J}_i^*(k+1) < \tilde{J}_i(k+1)$, so

$$\tilde{J}_i(k+1) - \tilde{J}_i(k) < -\left\| \mathbf{e}_{ip}(k+1) \right\|_{\mathbf{Q}}^2 - \rho \left\| \Delta \mathbf{u}_i(k) \right\|_{\mathbf{R}}^2. \quad (39)$$

It is clear that for $\rho \neq 0$, the cost is monotonically decreased with respect to time while the proposed control system is stable.

6.2. Convergence Analysis of DE Algorithm. It is known that the MPC method divides the global control problem into a series of local optimizations and each local optimization problem is solved by the adaptive DE algorithm. In this section, the asymptotic convergence of the local optimization is first analyzed; on this basis, the global convergence analysis is presented as well.

6.2.1. Local Convergence Analysis [45]. Suppose that at the k th time horizon, the optimal solution set of the cost function $\bar{J}(k)$ can be defined as follows:

$$S^*(k) = \{ \mathbf{v}^* : \bar{J}(\mathbf{v}^* | k) = \min \{ \bar{J}(\mathbf{v} | k) : \mathbf{v} \in S(k) \} \}, \quad (40)$$

where $S(k)$ is a measurable search space and \mathbf{v}^* represents the optimal solution individual.

However, condition (40) is hard to satisfy in the actual process. Thus, an expanded optimal solution set is

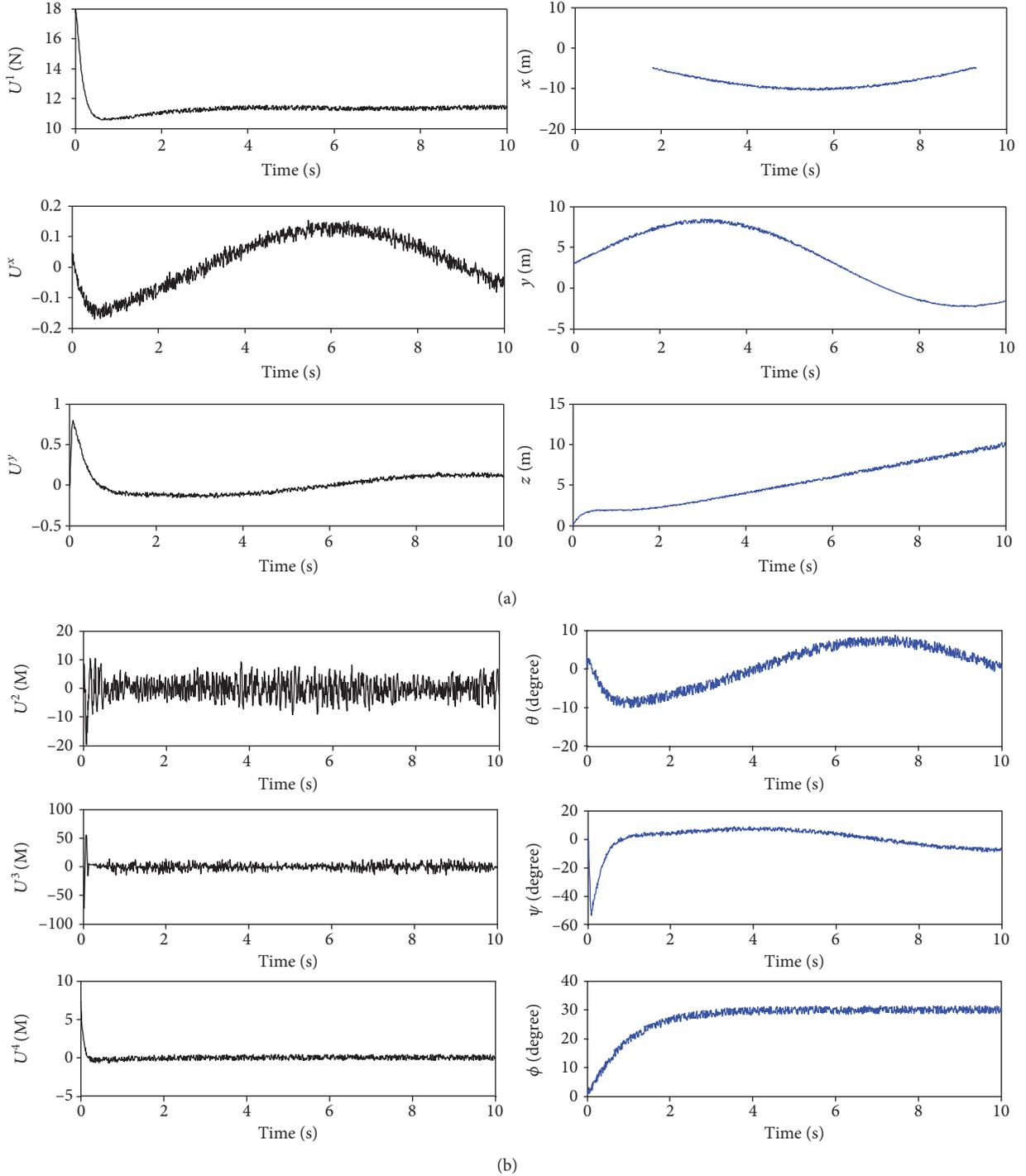


FIGURE 8: One sample of the training data sets: (a) control inputs and system outputs of translation subsystem; (b) control inputs and system outputs of the rotation subsystem.

considered as follows:

$$S_\varepsilon^*(k) = \{\tilde{\mathbf{v}} \in S(k): |\bar{J}(\tilde{\mathbf{v}} | k) - \bar{J}(\mathbf{v}^* | k)| < \varepsilon\}, \quad (41)$$

where $\varepsilon > 0$ is a small positive real number. Denoting μ as the Lebesgue measure, we suppose that the $\mu(S_\varepsilon^*(k)) > 0$ for each

ε . And the expanded optimal solution (41) can be approximately regarded as the optimal one.

Before analyzing the convergence of the local optimization, we need to introduce a concept of the convergence in probability as follows:

Define $\{V_G, G = 1, 2, \dots\}$ as a population sequence generated by the adaptive DE algorithm for the local optimization problem, where $V_G = \{\mathbf{v}_i^G, i = 1, 2, \dots, \text{NP}\}$. It is required

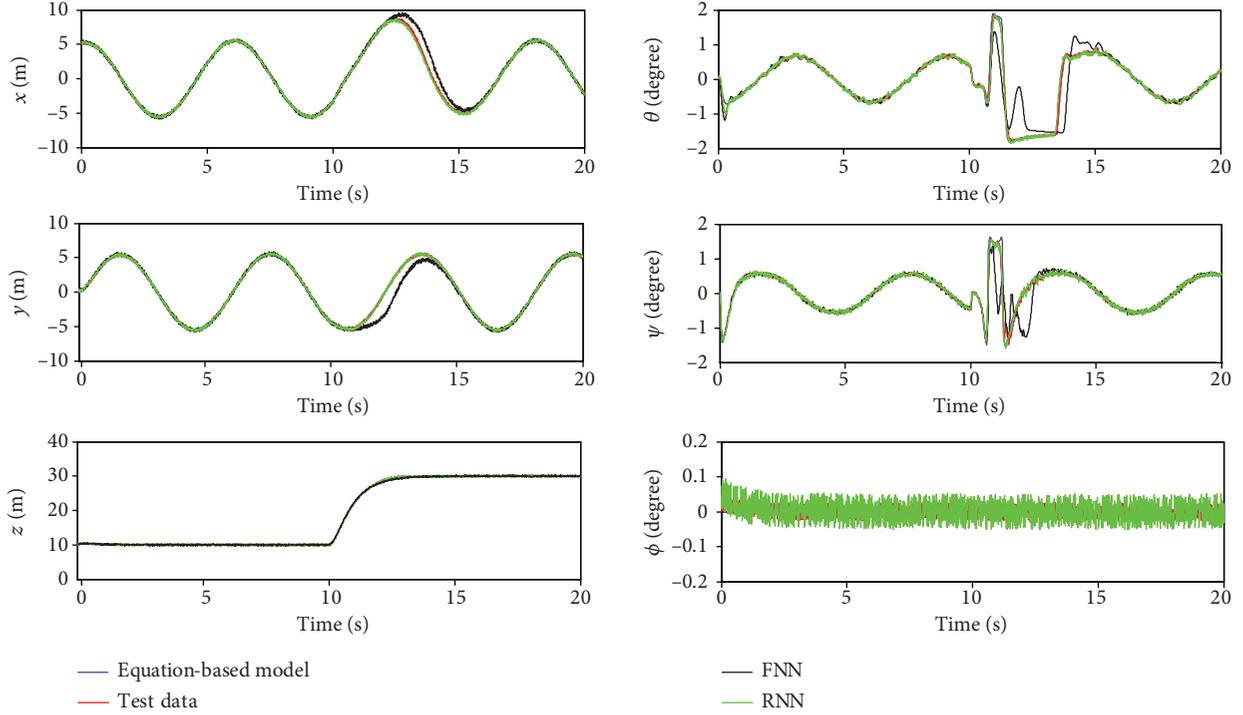


FIGURE 9: Test performance of RNN.

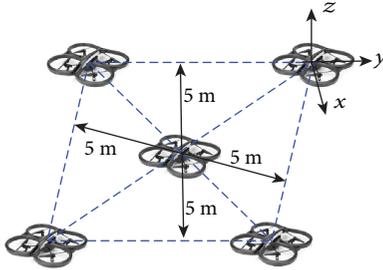


FIGURE 10: Initial formation of five quadrotors.

TABLE 1: Quadrotor dynamic parameters.

Parameters	Value
Mass m_s (kg)	2.14
Arm length l_a (m)	0.24
Inertia around x -axis I_{xx} ($\text{kg} \cdot \text{m}^2$)	1.25
Inertia around y -axis I_{yy} ($\text{kg} \cdot \text{m}^2$)	1.25
Inertia around z -axis I_{zz} ($\text{kg} \cdot \text{m}^2$)	2.5
Thrust factor b ($\text{N} \cdot \text{s}^2$)	0.3
Drag factor d ($\text{Nm} \cdot \text{s}^2$)	0.014

that the algorithm converges to the optimal solution set in probability if and only if

$$\lim_{G \rightarrow \infty} P\{V_G \cap S_\epsilon^*(k) \neq \emptyset\} = 1, \quad (42)$$

where P denotes the probability of an event.

TABLE 2: Constraints of the system states and control inputs.

Parameters	Value
Range of roll angle ϕ (rad)	$[-\pi/2, \pi/2]$
Range of pitch angle θ (rad)	$[-\pi/2, \pi/2]$
Range of yaw angle ψ (rad)	$[-\pi, \pi]$
Range of turn rates $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ (rad/s)	$[-1, 1]$
Range of the main thrust U_1 (N)	$[0, 60]$
Range of the torque U_2 (M)	$[-50, 50]$
Range of the torque U_3 (M)	$[-50, 50]$
Range of the torque U_4 (M)	$[-10, 10]$

TABLE 3: DE-run parameters and weights of the cost function.

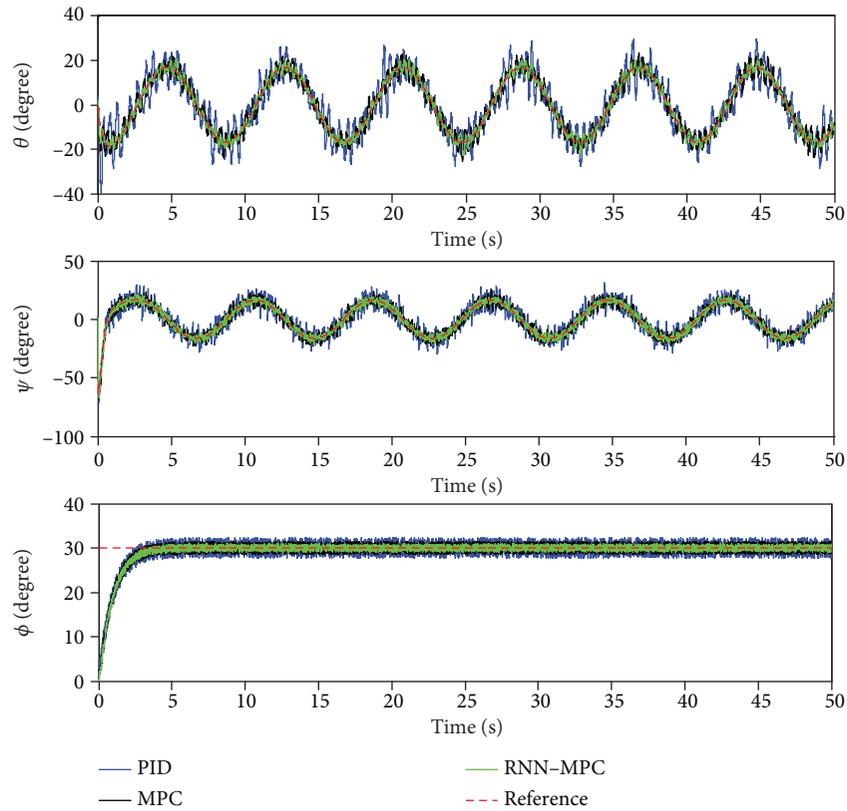
Parameters	NP	NC	μ	λ	α	ρ	\mathbf{Q}	\mathbf{R}
Value	50	50	1	1	1	0.5	$\mathbf{I}_{3 \times 3}$	$\mathbf{I}_{3 \times 3}$

Assumption 1. Based on the above definition, it is assumed that for each population V_G of the adaptive DE algorithm, there exists at least one individual \mathbf{v}^G such that

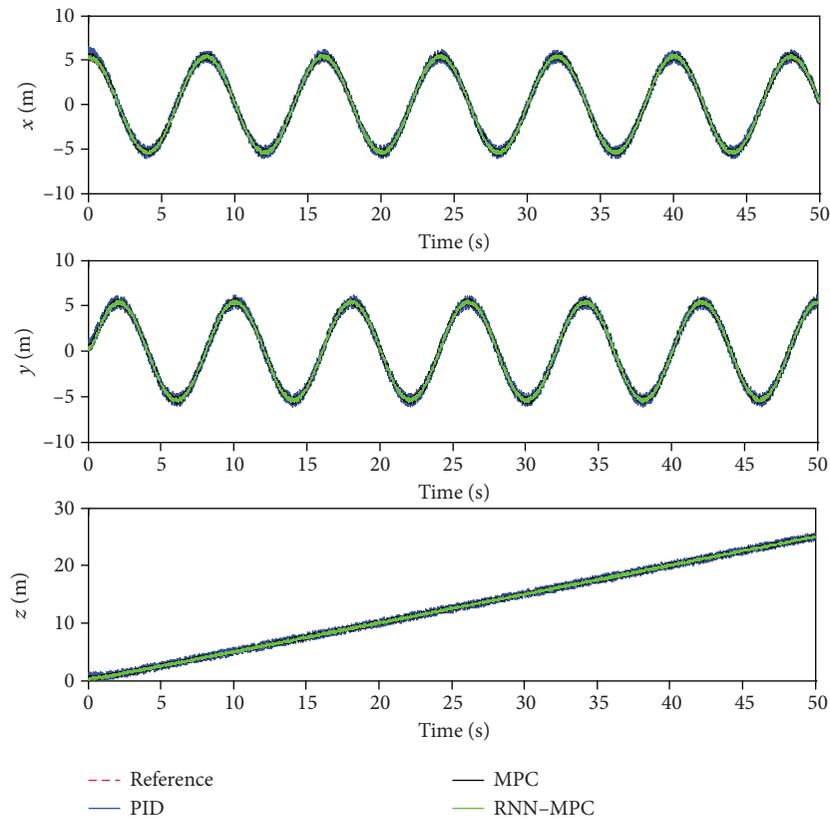
$$P\{\mathbf{v}^G \in S_\epsilon^*(k)\} \geq \alpha > 0, \quad (43)$$

where α is a small positive value.

Proof 2. Suppose that an individual $\mathbf{v}_{\text{rand}}^G$ is generated randomly in each population V_G . The probability that $\mathbf{v}_{\text{rand}}^G$ belongs to the optimal solution set is given by



(a)



(b)

FIGURE 11: Comparison results: (a) the variation of attitudes; (b) the variation of positions.

$$P\{\mathbf{v}_{\text{rand}}^G \in S_\varepsilon^*(k)\} = \frac{\mu(S_\varepsilon^*(k))}{\mu(S(k))} = \alpha > 0. \quad (44)$$

It means that $P\{\mathbf{v}_{\text{rand}}^G \notin S_\varepsilon^*(k)\} = 1 - \alpha > 0$. And the probability that the first d populations do not include an individual $\mathbf{v}^G \in S_\varepsilon^*(k)$ can be estimated by

$$\prod_{G=1}^d P\{V_G \cap S_\varepsilon^*(k) = \emptyset\} \leq (1 - \alpha)^d. \quad (45)$$

Based on basic ideas of the DE algorithm, the best individual in the population V_G has the same or better fitness value than the best one from all the previous populations, implying

$$\begin{aligned} \lim_{G \rightarrow \infty} P\{V_G \cap S_\varepsilon^*(k) = \emptyset\} &= \lim_{d \rightarrow \infty} \prod_{G=1}^d P\{V_G \cap S_\varepsilon^*(k) = \emptyset\} \\ &\leq \lim_{d \rightarrow \infty} (1 - \alpha)^d = 0, \end{aligned} \quad (46)$$

in other words,

$$\begin{aligned} \lim_{G \rightarrow \infty} P\{V_G \cap S_\varepsilon^*(k) \neq \emptyset\} &= 1 - \lim_{d \rightarrow \infty} \prod_{G=1}^d P\{V_G \cap S_\varepsilon^*(k) = \emptyset\} \\ &= 1 - 0 = 1, \end{aligned} \quad (47)$$

which proves (42).

6.2.2. Global Convergence Analysis. Based on the basic ideas of the MPC approach, the local optimizations follow the same control algorithm by using the adaptive DE algorithm. Therefore, the local convergence analysis of the k th time horizon can be applicable for the $(k + j)$ th time horizon, $j = 1, 2, \dots$. Furthermore, the local optimization of the $(k + j)$ th time horizon is executed on the basis of the optimization results of the $(k + j - 1)$ th time horizon. Finally, this local convergence of each time horizon leads to the global convergence of the entire control process.

7. Simulation Analysis

The performance of RNN-based MPC for quadrotor formation flight is evaluated by computer simulations. All the simulations are programmed with MATLAB, version 2015, and are run on a PC with a clock speed of 2.8 GHz and 4 GB RAM, in a Microsoft Windows 8.1 environment.

7.1. Offline Training for RNN. Before the experiment, the weights of RNN should be determined preliminarily. The input output data for RNN training is obtained by simulating a quadrotor model [35] with a PID controller. A large training data set spanning over 1000 seconds is generated by using the random step references ranging from 5 seconds to 15 sec-

TABLE 4: IAEs of the proposed RNN-based MPC and conventional MPC.

Method	RNN-based MPC	Conventional MPC
IAE	1.3983	2.3453

onds. All system states and control inputs are restricted within the corresponding ranges. To enable the RNN model to learn the noisy dynamics in the presence of sensor noises, wind gusts and aerodynamic coupling effects, additive disturbances are included for data generation during the simulation. The disturbances \mathbf{d}_i^ζ and \mathbf{d}_i^η are in the form of white noise with various intensity. And the network is adopted in the form of (19). Figure 8 shows one sample of the training data sets. Based on the test data, the training performance of RNN is tested in Figure 9, compared with the FNN model [20] and nonlinear partial-differential-equation-based model [35]. From the figures, the apparent advantage of the RNN model is the prediction accuracy. In other words, these results show the success of the dynamic RNN structure.

7.2. Results and Discussion. A scenario is constructed to simulate a certain flight condition for the multiquadrotor formation. Assuming that a square formation is composed of five quadrotors, the relative position of each quadrotor is described in Figure 10, and their initial positions are $[0, 5, 0]$ m, $[10, 5, 0]$ m, $[5, 0, 0]$ m, $[0, -5, 0]$ m, and $[10, -5, 0]$ m. Besides, the initial attitudes of all quadrotors are supposed to be $[\phi_i, \theta_i, \psi_i] = [0, 0, 0]$. Define the 3rd quadrotor as the leader flying along a reference trajectory called “elliptical”: $x = 5 \cdot \cos(\pi t/4)$, $y = 5 \cdot \sin(\pi t/4)$, $z = 0.5 \cdot t$, and $t \in [0, 50]$. And the other quadrotors as the followers fly with the following relative states: $\Delta \mathbf{x}_{1,3} = [-5, 5, 0, 0, 0, 0]$, $\Delta \mathbf{x}_{2,3} = [5, 5, 0, 0, 0, 0]$, $\Delta \mathbf{x}_{3,4} = [-5, -5, 0, 0, 0, 0]$, and $\Delta \mathbf{x}_{3,5} = [5, -5, 0, 0, 0, 0]$. To simulate the real flight environment, the bounded disturbances are introduced in rotation subsystem (white noise of intensity 0.3 rad/sec) and transition subsystem (white noise of intensity 0.5 m/sec). Moreover, the quadrotor’s safe anticollision and communication regions are $D_{\text{safe}} = 1$ m, $D_{\text{com}} = 50$ m, respectively.

The parameter settings of the proposed system condition are as follows: the dynamic parameters of unmanned quadrotor system in this paper are represented in Table 1. The limits on the system states including the range of ϕ , θ , and ψ ; the corresponding turn rates $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$; and the control inputs U_1 , U_2 , U_3 , and U_4 are shown in Table 2. Table 3 shows the DE algorithm-run parameters and the associated weight values of the fitness function.

In such a case, the MPC with the RNN online model is introduced for the formation-tracking problem over the prediction horizons, compared with the conventional MPC and traditional PID, where the parameters of the PID controller are $k_p = 106.3$, $k_I = 530.9$, and $k_D = 414.5$. For both conventional MPC and RNN-based MPC, the prediction horizon and control horizon are $H_p = 3$ and $H_u = 2$ and the sampling frequency f_s is 4 Hz. In Figure 11, taking the tracking

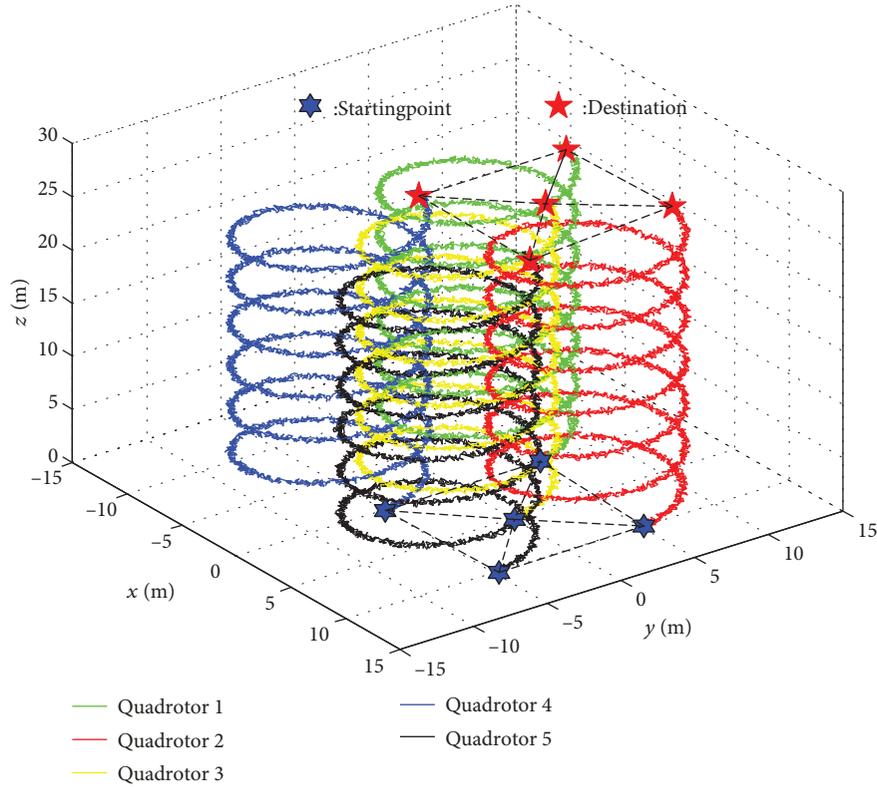


FIGURE 12: 3-D performance of formation flight.

performance of the leader as an example, the results of three different control methods are presented. It is obvious that the performances of the RNN-based MPC approach with smallest tracking errors are better than the other two approaches. In particular, the control inputs of the PID controller are out of constraints at some time horizons.

In order to reflect the control results intuitively, the tracking performance of the quadrotor formation can be measured using the integral of absolute error (IAE) [46].

$$\text{IAE} = \frac{1}{N \cdot H_T} \sum_{i=1}^N \sum_{k=1}^{H_T} \left(\left\| \mathbf{y}_i^\zeta(k) - \mathbf{r}_i^\zeta(k) \right\|_{\mathbf{I}_{3 \times 3}}^2 + \left\| \mathbf{y}_i^\eta(k) - \mathbf{r}_i^\eta(k) \right\|_{\mathbf{I}_{3 \times 3}}^2 \right), \quad (48)$$

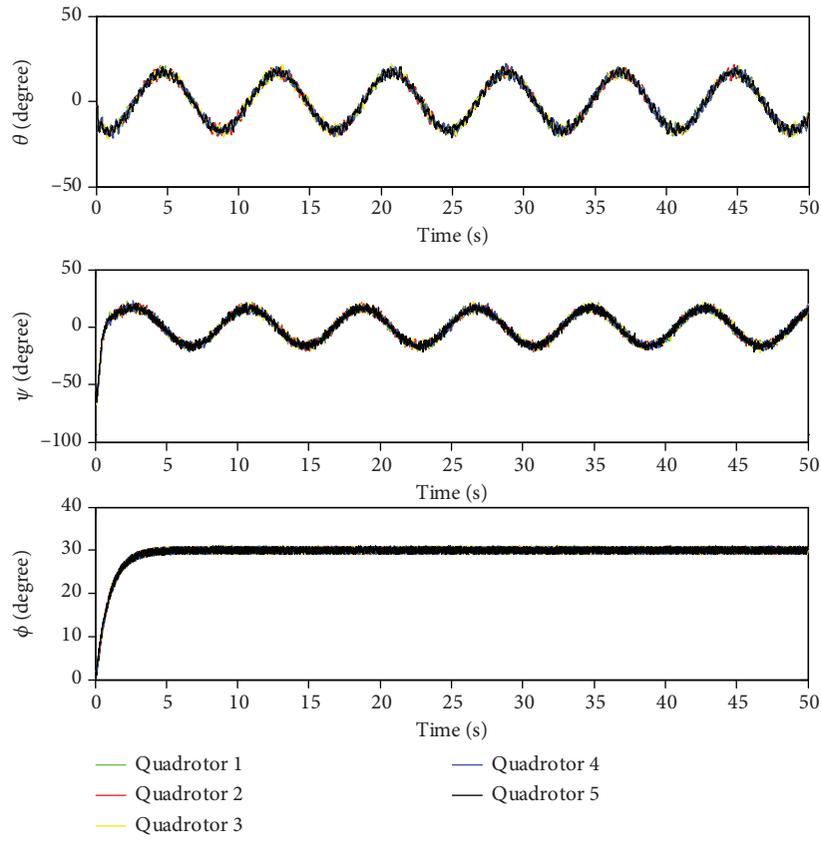
where N is the total number of quadrotors and H_T is the total number of time horizons.

The IAEs of the conventional MPC and RNN-based MPC are presented in Table 4. Based on that, the tracking performance controlled by RNN-based MPC is better with the smaller errors. This is because the RNN is trained based on the test data and has learned the influences of unknown disturbances before experiments. On the other hand, due to the updating online for RNN, the proposed MPC method could estimate in a timely manner the disturbances improving the control performance.

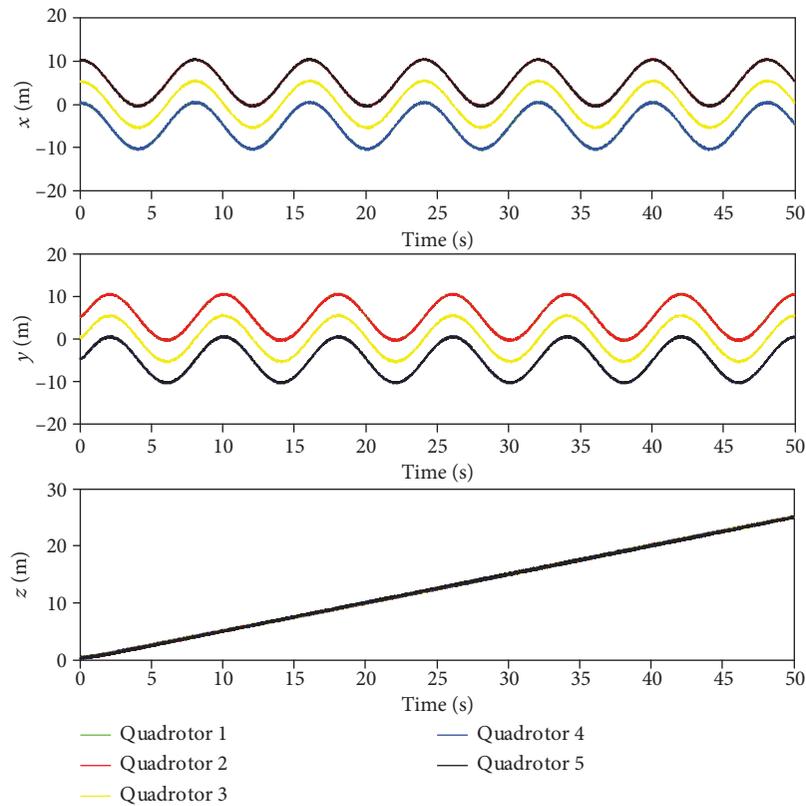
The 3-D performance of formation tracking with RNN-based MPC are depicted graphically in Figure 12. The overall variations of attitude and position are shown in Figure 13. The corresponding errors of relative attitude (Figure 14(a)) and position (Figure 14(b)) between the leader quadrotor and followers are within the anticipated range. Generally, the proposed control system manages to accurately achieve formation tracking with fast and precise responses and effectively attenuate the mismatched disturbances within the acceptable error range.

8. Conclusion

In this paper, a novel predictive control scheme based on RNNs was proposed for the multiquadrotor system to achieve trajectory tracking and formation keeping simultaneously. Different from the conventional MPC, the proposed control approach designed the RNNs as the predictive model and used the adaptive DE algorithm to solve the local optimization problem for both subsystems of each quadrotor. Moreover, the closed-loop stability was analyzed; meanwhile, the convergence of the DE algorithm was discussed as well. Finally, the effectiveness of the designed control strategy was illustrated through comparative simulation results. However, the applicability of the proposed control structure is limited to the high computation burden of receding optimization. Therefore, an appealing extension of this work could focus on the time efficiency of whole control process.

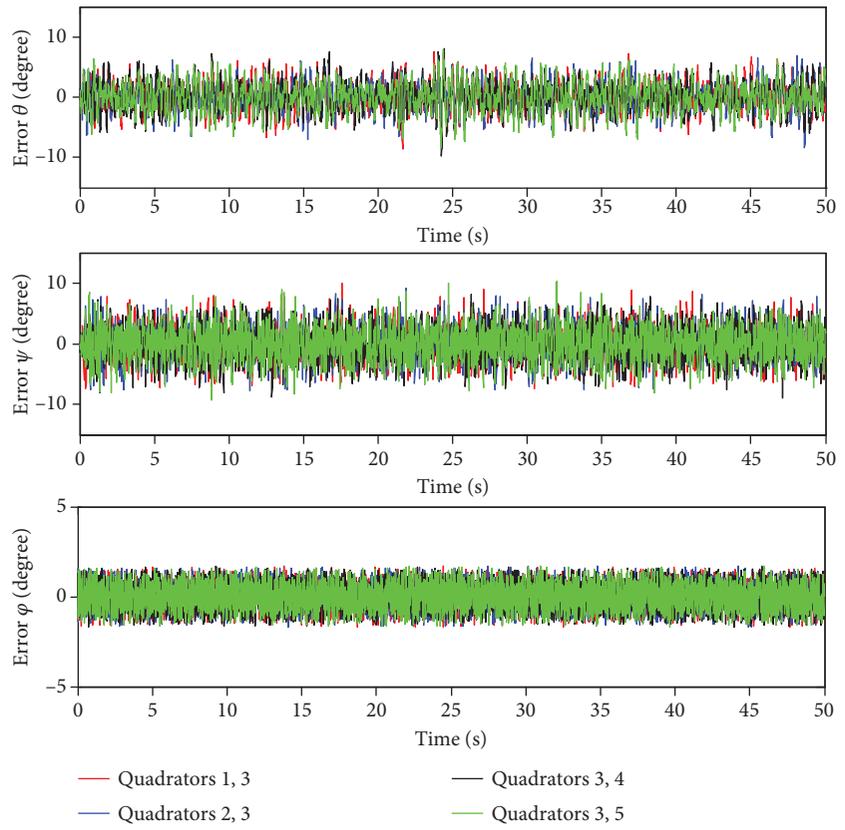


(a)

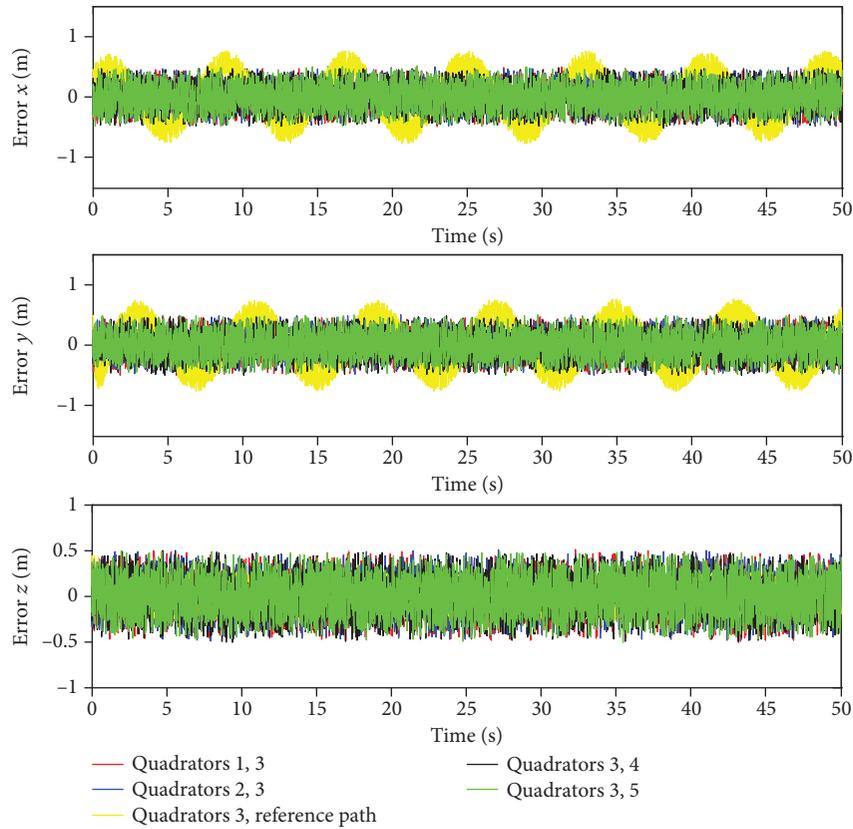


(b)

FIGURE 13: Flight process: (a) the variation of attitudes; (b) the variation of positions.



(a)



(b)

FIGURE 14: Tracking errors: (a) relative attitudes; (b) relative positions.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the support of the Aeronautical Science Foundation of China under Grant No. 20155896025.

References

- [1] L. Xiao, Y. F. Fang, N. Sun, and H. Lin, "Nonlinear hierarchical control for unmanned quadrotor transportation systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3395–3405, 2018.
- [2] B. Xiao and S. Yin, "A new disturbance attenuation control scheme for quadrotor unmanned aerial vehicles," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2922–2932, 2017.
- [3] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, vol. 33, no. 4, pp. 431–450, 2016.
- [4] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian Journal of Forest Research*, vol. 45, no. 7, pp. 783–792, 2015.
- [5] T. Bartelds, A. Capra, S. Hamaza, S. Stramigioli, and M. Fumagalli, "Compliant aerial manipulators: toward a new generation of aerial robotic workers," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 477–483, 2016.
- [6] C. L. Liu, J. Pan, and Y. F. Chang, "PID and LQR trajectory tracking control for an unmanned quadrotor helicopter: experimental studies," in *2016 35th Chinese Control Conference (CCC)*, pp. 10845–108503, Chengdu, China, 2016.
- [7] A. L'Afflitto, R. B. Anderson, and K. Mohammadi, "An introduction to nonlinear robust control for unmanned quadrotor aircraft: how to design control algorithms for quadrotors using sliding mode control and adaptive control techniques [focus on education]," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 102–121, 2018.
- [8] H. Lu, C. Liu, M. Coombes, L. Guo, and W.-H. Chen, "Online optimisation-based backstepping control design with application to quadrotor," *IET Control Theory & Applications*, vol. 10, no. 14, pp. 1601–1611, 2016.
- [9] J. J. Li, R. Li, Y. J. Shi, and J. X. Zou, "Design of attitude controller using explicit model predictive control for an unmanned quadrotor helicopter," in *2017 Chinese Automation Congress (CAC)*, pp. 2853–2857, Jinan, China, 2017.
- [10] A. Kuyumcu and I. Bayezit, "Augmented model predictive control of unmanned quadrotor vehicle," in *Asian Control Conference*, pp. 1626–1631, Gold Coast, QLD, Australia, 2017.
- [11] H. Cheng and Y. H. Yang, "Model predictive control and PID for path following of an unmanned quadrotor helicopter," in *IEEE Conference on Industrial Electronics and Applications*, pp. 768–773, Siem Reap, Cambodia, 2017.
- [12] D. Ma, Y. Xia, T. Li, and K. Chang, "Active disturbance rejection and predictive control strategy for a quadrotor helicopter," *IET Control Theory & Applications*, vol. 10, no. 17, pp. 2213–2222, 2016.
- [13] J. Zhao and L. Mili, "Sparse state recovery versus generalized maximum-likelihood estimator of a power system," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1104–1106, 2018.
- [14] P. Thune and G. Enzner, "Maximum-likelihood approach with Bayesian refinement for multichannel-Wiener postfiltering," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3399–3413, 2017.
- [15] G. Feng, C. Lai, and N. C. Kar, "Expectation-maximization particle-filter-and Kalman-filter-based permanent magnet temperature estimation for PMSM condition monitoring using high-frequency signal injection," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1261–1270, 2017.
- [16] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.
- [17] V. Muneeswaran and M. Pallikonda Rajasekaran, "Performance evaluation of radial basis function networks based on tree seed algorithm," in *International Conference on Circuit, Power and Computing Technologies*, pp. 530–534, Nagercoil, India, 2016.
- [18] J. Wang, C. Xu, X. Yang, and J. M. Zurada, "A novel pruning algorithm for smoothing feedforward neural networks based on group lasso method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 2012–2024, 2018.
- [19] J. Wang, L. Zhang, Q. Guo, and Z. Yi, "Recurrent neural networks with auxiliary memory units," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1652–1661, 2018.
- [20] N. Nikdel, P. Nikdel, M. A. Badamchizadeh, and I. Hassanzadeh, "Using neural network model predictive control for controlling shape memory alloy-based manipulator," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1394–1401, 2014.
- [21] S. Tiwari, R. Naresh, and R. Jha, "Neural network predictive control of UPFC for improving transient stability performance of power system," *Applied Soft Computing*, vol. 11, no. 8, pp. 4581–4590, 2011.
- [22] M. Fairbank, S. Li, X. Fu, E. Alonso, and D. Wunsch, "An adaptive recurrent neural-network controller using a stabilization matrix and predictive inputs to solve a tracking problem under disturbances," *Neural Networks*, vol. 49, pp. 74–86, 2014.
- [23] M. C. Phan and M. T. Hagan, "Error surface of recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1709–1721, 2013.
- [24] Z.-H. Ling, Y. Ai, Y. Gu, and L.-R. Dai, "Waveform modeling and generation using hierarchical recurrent neural networks for speech bandwidth extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 883–894, 2018.
- [25] Y. Wang, L. Cheng, Z.-G. Hou, J. Yu, and M. Tan, "Optimal formation of multirobot systems based on a recurrent neural

- network,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 322–333, 2016.
- [26] S. Li, J. He, Y. Li, and M. U. Rafique, “Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 2, pp. 415–426, 2017.
- [27] M. Gaur, H. Chauhary, S. Khatoon, and R. Singh, “Genetic algorithm based trajectory stabilization of quadrotor,” in *Conference on International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity*, pp. 29–33, Ghaziabad, India, 2016.
- [28] X. Y. Lu, X. Y. Zhang, S. M. Jia, and J. C. Shan, “Design of quadrotor hovering controller based on improved particle swarm optimization,” in *Conference on International Symposium on Computational Intelligence and Design*, pp. 414–417, Hangzhou, China, 2017.
- [29] X. H. Wang and H. B. Duan, “Biologically adaptive robust mean shift algorithm with Cauchy predator-prey BBO and space variant resolution for unmanned helicopter formation,” *Science China Mathematics*, vol. 57, no. 11, pp. 1–13, 2014.
- [30] R. Storn and K. Price, *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report, International Computer Science Institute, Berkeley, CA, USA, 1995.
- [31] Y. Mao, S. Niu, and Y. Yang, “Differential evolution-based multiobjective optimization of the electrical continuously variable transmission system,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2080–2089, 2018.
- [32] Y. Wang, H. Liu, H. Long, Z. Zhang, and S. Yang, “Differential evolution with a new encoding mechanism for optimizing wind farm layout,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1040–1054, 2018.
- [33] W. C. Liu, L. Cheng, C. Zhou, Z. G. Hou, and M. Tan, “Neural-network based model predictive control for piezoelectric-actuated stick-slip micro-positioning devices,” in *IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 1312–1317, Banff, AB, Canada, 2016.
- [34] L. Cheng, W. Liu, Z. G. Hou, J. Yu, and M. Tan, “Neural network based nonlinear model predictive control for piezoelectric actuators,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7717–7727, 2015.
- [35] K. Alexis, G. Nikolakopoulos, and A. Tzes, “Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances,” *Control Engineering Practice*, vol. 19, no. 10, pp. 1195–1207, 2011.
- [36] G. V. Raffo, M. G. Ortega, and F. R. Rubio, “An integral predictive/nonlinear control structure for a quadrotor helicopter,” *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.
- [37] G. Cao, E. M. K. Lai, and F. Alam, “Gaussian process model predictive control of an unmanned quadrotor,” *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 147–162, 2017.
- [38] Y. C. Lin, D. D. Chen, M. S. Chen, X. M. Chen, and J. Li, “A precise BP neural network-based online model predictive control strategy for die forging hydraulic press machine,” *Neural Computing and Applications*, vol. 29, no. 9, pp. 585–596, 2018.
- [39] K. Patan, “Neural network-based model predictive control: fault tolerance and stability,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1147–1155, 2015.
- [40] X. F. Bao, Z. Y. Sun, and N. Sharma, “A recurrent neural network based MPC for a hybrid neuroprosthesis system,” in *Annual Conference on Decision and Control*, pp. 4715–4720, Melbourne, VIC, Australia, 2017.
- [41] L. Jin, S. Li, and B. Hu, “RNN models for dynamic matrix inversion: a control-theoretical perspective,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 189–199, 2018.
- [42] E. Petre, D. Selisteanu, D. Sendrescu, and C. Ionete, “Neural networks-based adaptive control for a class of nonlinear bio-processes,” *Neural Computing and Applications*, vol. 19, no. 2, pp. 169–178, 2010.
- [43] E. Petre, D. Sendrescu, and D. Selisteanu, “Neural networks based model predictive control for a lactic acid production bio-process,” in *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 6884, pp. 388–398, 2011.
- [44] Q. Fan and X. Yan, “Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies,” *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 219–232, 2016.
- [45] R. Knobloch, J. Mlynek, and R. Srb, “The classic differential evolution algorithm and its convergence properties,” *Applications of Mathematics*, vol. 62, no. 2, pp. 197–208, 2017.
- [46] H. G. Han, L. Zhang, Y. Hou, and J. F. Qiao, “Nonlinear model predictive control based on a self-organizing recurrent neural network,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 402–415, 2016.

