

## Article

# Flatness-Based Aggressive Formation Tracking Control of Quadrotors with Finite-Time Estimated Feedforwards

Zhicheng Hou <sup>1,\*</sup>, Gong Zhang <sup>1</sup>, Wenlin Yang <sup>1</sup>, Weijun Wang <sup>1</sup> and Changsoo Han <sup>2</sup>

<sup>1</sup> Guangzhou Institute of Advanced Technology, Chinese Academy of Sciences, Guangzhou 511458, China; gong.zhang@giat.ac.cn (G.Z.); wl.yang@giat.ac.cn (W.Y.); wj.wang@giat.ac.cn (W.W.)

<sup>2</sup> Department of Robot Engineering, ERICA Campus, Hanyang University, Gyeonggi-do 426-791, Korea; cshan@hanyang.ac.kr

\* Correspondence: zc.hou@giat.ac.cn

**Abstract:** In this paper we address a decentralized neighbor-based formation tracking control of multiple quadrotors with leader–follower structure. Different from most of the existing work, the formation tracking controller is given in one loop without distinguishing the motion control and attitude control by means of the theory of flatness. In order to achieve an aggressive formation tracking, the high-order states of the neighbors motion are estimated by using a proposed extended finite-time observer for each quadrotor. Then the estimated motion states are used as feedforwards in the formation controller design. Simulation and experimental results show that the proposed formation controller improves the formation performance, i.e., the formation pattern of the quadrotors is better maintained than that using the formation controller without high-order feedforwards, when tracking an aggressive reference formation trajectory.

**Keywords:** quadrotors; aggressive formation tracking control; extended finite-time observer; flatness



**Citation:** Hou, Z.; Zhang, G.; Yang, W.; Wang, W.; Han, C. Flatness-Based Aggressive Formation Tracking Control of Quadrotors with Finite-Time Estimated Feedforwards. *Appl. Sci.* **2021**, *11*, 792. <https://doi.org/10.3390/app11020792>

Received: 12 December 2020

Accepted: 12 January 2021

Published: 15 January 2021

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



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

## 1. Introduction

The cooperative control of multi-quadrotor systems has progressively attracted attention of researchers in both civil and military areas [1]. Especially, in the application of object transportation, the formation of multiple quadrotors has potential application significance [2–6].

The formation control problem can be categorized into two types. Namely, the formation producing problem and the formation tracking problem [7]. The former problem refers to the algorithm design for a group of vehicles to reach a predefined geometric pattern without a group reference [8–10], while the latter problem refers to the same task and meanwhile following a predefined trajectory [11–13]. The object transformation is a typical formation tracking problem, which is considered in this paper. An external reference formation trajectory (RFT) is given to the leading quadrotor(s) to guide the formation. We note that in the case of multiple leaders, they share the same RFT. According to [14,15], the formation tracking problem can be attributed as a consensus problem.

In most aggressive maneuver cases, the quadrotor is controlled in tracking a desired aggressive moving trajectory that is prescribed a priori, for instance, by polynomials w.r.t time. A widely used method is the control of geometry on Lie group ( $SE(3)$  for example). The aggressive formation problem of multiple quadrotors is investigated in [16], where the quadrotors are permitted to move quickly in 3-D environment with a tight formation. The formation of quadrotors is based on proper trajectory generation process, thereby, the full information of the trajectory (including each order derivatives) are known. However, in a multi-agent system with neighbor-based decentralized control, it is impossible to enforce the motion of quadrotor by generating polynomials-based trajectories.

The flatness-based control is frequently used in the trajectory planning for high-order dynamical system. For instance, the time-optimal trajectory generation problem is in-

vestigated for helicopter landing in [17] and for multi-rotor in [18]. Especially, in [18], the trajectory is planned in real-time. The agile maneuvers for a quadrotor with a cable-suspended load are investigated in [19]. In this work, by means of flatness-based planning, the air vehicle is able to behave in aggressive maneuver, real-time collision avoidance, and time optimal flight. However, the planned trajectory's high-order derivatives are required for the quadrotor. In the decentralized formation tracking control of quadrotors, each quadrotor (including the leader) has to behave according to the state of neighbors, which are in general not able or difficult to get the full information of these states.

In multi-agent systems, the consensus speed of agents with linear dynamics depends on the minimum non-singular eigenvalue of the interaction matrix (or Laplacian) [20] and the absolute value of the maximum pole in left half plane of agent dynamics [21]. For the formation of quadrotors, the idea in this paper is to propose a nonlinear formation controller, in order to use these existing results.

One of the difficulties in the control of multi-quadrotor systems is that the quadrotor has under-actuated, nonlinear, and coupling dynamics. The existing consensus algorithms in the literature, which are mostly developed for single-integrator or double-integrator agent dynamics, have not satisfactory formation performance for the formation control of quadrotors in practice, especially in aggressive formation tracking where the small angle assumption should be removed. Furthermore, lacking velocities and its higher derivatives of the neighbors' states and the RFT makes aggressive formation non-trivial. Nevertheless, in very recent years, the researchers start to consider the nonlinearities/disturbances in formation control [1,22].

In this paper, we aimed at proposing an aggressive control framework for a class of systems with high-order dynamics (in special, quadrotor). Instead of using inner-outer loop structure, we directly design the torques by using the flatness theory, where the high-order cooperative errors are used. The formation controller is decentralized, since the quadrotors move based on the measurement to their neighbors (in special, positions. We suppose that the velocities and higher-order derivatives are not available). Then, the main contributions of this paper are three-fold. A flatness-based formation control with estimated feedforwards is proposed such that the aggressive formation is attained. In order to estimate the velocities and higher-order derivatives of the neighbors states, an extended finite-time observer is proposed. To avoid probably instability caused by the great initial deviation of the estimated states, the saturated feedforwards are developed to improve the formation controller, the convergence of the formation tracking error is then analyzed.

We note that our proposed controller can be combined with the existing methods by adding some nonlinear compositions into the controller, for instance, to deal with the bounded nonlinearities (uncertainties), to achieve asymptotic convergence, to improve the formation speed. These methods can be listed such as robust compensation [20,22], composite nonlinear feedback [13], artificial neural network estimation [23].

The paper is organized as follows. Some preliminaries in graph theory are given in Section 2. The aggressive control of quadrotors with finite-time observer is presented in Section 3. The formation control design and the stability analysis are shown in Section 4. Some simulation and experimental results are given in Section 5. Finally, some conclusions are stated in Section 6. Some useful notations and definitions are given as follows.

**Notations:** The real vectorial field is represented by  $\mathbb{R}^m$ . Notation  $\mathbb{R}^+$  represents real positive numbers. The saturation function  $\sigma_b : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is defined as follows, if  $n = 1$ ,  $\sigma_b(a) = \text{sign}(a) \cdot \min\{\|a\|, b\}$ , where  $\text{sign}(\cdot)$  represents the sign function. If  $n > 1$  and  $a = [a_1, \dots, a_n]^T$ , then,  $\sigma_b(a) = [\sigma_b(a_1), \dots, \sigma_b(a_n)]^T$ . In special, if  $b = \infty$ , the saturation function is equal to a linear function without saturation, i.e.,  $\sigma_\infty(a) = a$ , for any  $a \in \mathbb{R}$ . The notation  $C^m(\mathbb{R}, \mathbb{R}^n)$  represents the ensemble of  $m$ -order continuous manifolds in  $\mathbb{R} \rightarrow \mathbb{R}^n$ . The notations  $I_n$  and  $\mathbf{0}_{n \times n}$  represent identical and zero matrices in  $\mathbb{R}^{n \times n}$ . In special,  $\mathbf{1}_n$  and  $\mathbf{0}_n$  represent the vectors with all entries equal to one and zero in  $\mathbb{R}^n$ , respectively. Function  $\lambda_i(\cdot)$  represents the  $i$ -th eigenvalue of matrix inside the parenthesis, in special,

$\lambda_{\max}(\cdot)$  and  $\lambda_{\min}(\cdot)$  represent the maximum and minimum eigenvalue of matrix inside the parenthesis, respectively.

## 2. Preliminaries

Some basic knowledge in graph theory and ‘interaction matrix’ is given in this section. Some propositions and corollaries about the interaction matrix are introduced, which are useful in the analysis of the consensus of quadrotors in Section 4.

### 2.1. Graph Theory

Graph theory is helpful to represent the interaction relations of multi-agent systems. A graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is normally with the sets of vertices  $\mathcal{V}$  and edges  $\mathcal{E}$ . The set of vertices  $\mathcal{V} = \{1, 2, \dots, n\}$  is composed of the indices of agents.  $|\mathcal{V}|$  represents the cardinality of the set  $\mathcal{V}$ , which satisfies  $|\mathcal{V}| = n$ . The set of edges is represented by  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . If an edge exists between two vertices, the two vertices are called adjacent. In this work, simple and undirected graphs are considered.

The adjacency matrix of  $\mathcal{G}$  is denoted by  $G^A = [a_{ij}] \in R^{n \times n}$ , where  $a_{ij}$  represents the connection of nodes  $i$  and  $j$  on a graph. Since the simple graph is considered, we have  $a_{ii} = 0$ . Since the graph is undirected, we have  $a_{ij} = a_{ji}$  and  $a_{ij} = 1$  if  $(i, j) \in \mathcal{E}$ , otherwise,  $a_{ij} = 0$ . The degree matrix of  $\mathcal{G}$  is denoted by  $G^D = \text{diag}\{\sum_{j=1}^n a_{1j}, \dots, \sum_{j=1}^n a_{nj}\}$ . The neighbor set  $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$  of agent  $i$ , is composed of the indices of the agents  $j$ , which has interaction with the agent  $i$ . In other words, if  $a_{ij} > 0$ , then, agent  $j$  is a neighbor of agent  $i$ . The number of neighbors of agent  $i$  is equal to  $|\mathcal{N}_i|$ .

We define a diagonal matrix  $G^L = \text{diag}\{l_1, \dots, l_n\}$ , which indicates the knowledge of reference formation trajectory (RFT) of each agent. If  $l_i = 1$ , agent  $i$  can obtain the RFT (by sensing/detecting), i.e., a leader. Otherwise, if  $l_i = 0$ , agent  $i$  is a follower, for  $i \in \mathcal{V}$ . Then, the leader set is defined as  $\mathcal{V}_L = \{i \in \mathcal{V} : l_i > 0\}$ . The leader set  $\mathcal{V}_L \subset \mathcal{V}$  is a subset of  $\mathcal{V}$ , which contains the indices of the leaders. Particularly, all the quadrotors are leaders, when  $\mathcal{V}_L = \mathcal{V}$ . The indices of the followers are contained in the complementary set of  $\mathcal{V}_L$ , namely,  $\mathcal{V} - \mathcal{V}_L$ .

### 2.2. Interaction Matrix

We propose an interaction matrix  $G$  for representing the interconnecting relations of quadrotors with leader–follower (L–F) configuration as follows.

$$G = G^D - G^A + G^L \tag{1}$$

**Remark 1.** Let us note that the first two terms  $G^D - G^A$  are normally called the Laplacian in graph theory.

**Proposition 1.** Let  $\mathcal{G}$  be an undirected simple graph, then the interconnection matrix  $G$  in Equation (1), is positive-definite, if (i)  $\mathcal{G}$  is connected; (ii)  $\mathcal{V}_L \neq \Phi$ , where  $\Phi$  represents a null set.

**Proof.** Since  $\mathcal{G}$  is connected, we have  $G^D - G^A \geq 0$ . Considering the definition of  $G^L$ , we have  $G^L \geq 0$ . We prove this proposition by contradiction. Firstly, we suppose that there exists a nonzero vector  $x \in \mathbb{R}^n$ , which renders

$$x^T G x = x^T (L + G^L) x = x^T L x + x^T G^L x = 0$$

Therefore, we must have  $x^T L x = 0$  and  $x^T G^L x = 0$ .

Since  $x^T L x = 0$ , we obtain that  $x = \alpha \mathbf{1}_n$ , where  $\alpha$  is a nonzero scalar. According to the fact that  $\mathcal{V}_L \neq \Phi$ , then,  $G^L \neq 0$ . As a result,  $x^T G^L x = \alpha^2 \mathbf{1}_n^T G^L \mathbf{1}_n > 0$ , which contradicts  $x^T G^L x = 0$ . Therefore, such a nonzero vector  $x$  does not exist. Thus, for any nonzero vector  $x$ ,  $x^T G x > 0$ , namely,  $G$  is positive-definite.  $\square$

When condition (i) is not satisfied such that  $\mathcal{G}$  is not connected, the graph can be divided into  $s$  connected sub-graphs  $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_s$ . In that case, according to Proposition 1, we have the following corollary.

**Corollary 1.** *The interaction matrix  $G$  is positive-definite, if the leaders set satisfies  $\mathcal{V}_L \geq s$ , and  $\mathcal{V}_{iL} \neq \Phi$ , where  $i = \{1, 2, \dots, s\}$ .*

**Proof.** If a multi-agent system contains  $s$  connected sub-groups of agents, then, the interaction matrix is block diagonal, which has  $s$  blocks on the diagonal. Since each sub-group  $\mathcal{G}_i, i = \{1, 2, \dots, s\}$  is connected and  $\mathcal{V}_{iL} \neq \Phi$ , then, the block in the interaction matrix is positive definite. Hence, the interaction matrix is positive definite.  $\square$

The normalized interaction matrix is defined by

$$\bar{G} = (G^D + G^L)^{-1}G \tag{2}$$

The normalized interaction matrix  $\bar{G}$  is not symmetric, but it has the following property.

**Proposition 2.** *The normalized interaction matrix  $\bar{G}$  has real eigenvalues, if  $\mathcal{G}$  is undirected.*

**Proof.** Since  $\bar{G}$  is a normalized interaction matrix, then,  $\bar{G} = (G^D + G^L)^{-1} \cdot G$ . We recall that  $G$  is symmetric.

Using  $(G^D + G^L)^{-\frac{1}{2}}$ , we apply similarity transformation to  $\bar{G}$ , then, we obtain that  $(G^D + G^L)^{\frac{1}{2}} \cdot \bar{G} \cdot (G^D + G^L)^{-\frac{1}{2}} = (G^D + G^L)^{-\frac{1}{2}} \cdot G \cdot (G^D + G^L)^{-\frac{1}{2}}$  is symmetric, because  $G^D + G^L$  are diagonal and  $G$  is symmetric.

Since  $\bar{G}$  is similar to  $(G^D + G^L)^{-\frac{1}{2}} \cdot G \cdot (G^D + G^L)^{-\frac{1}{2}}$ , the eigenvalues of  $\bar{G}$  are real, moreover, they are equal to the eigenvalues of  $(G^D + G^L)^{-\frac{1}{2}} \cdot G \cdot (G^D + G^L)^{-\frac{1}{2}}$ .  $\square$

**Corollary 2.** *The normalized interaction matrix  $\bar{G}$  is positive-definite, if conditions in Proposition 1 or Corollary 1 are satisfied.*

**Proof.** According to the conditions in Proposition 1 or Corollary 1, we know that  $G$  is invertible. Since  $\bar{G}$  is similar to  $(G^D + G^L)^{-\frac{1}{2}} \cdot G \cdot (G^D + G^L)^{-\frac{1}{2}}$  according to Proposition 2, we obtain that  $\bar{G}$  is also positive-definite.  $\square$

### 3. Aggressive Control of Quadrotors with Extended Finite-Time Observer

#### 3.1. Quadrotor Model

The dynamics of a quadrotor is modelled as the motion of a rigid body in 3-D space under a thrust force and three moments, which are generated by the thrust forces of the four rotors. The notations used in the quadrotor’s model are shown in Table 1. The orientation of the quadrotor with respect to the inertial frame is represented by the rotation matrix  $R_i \in \text{SO}(3)$ , where  $\text{SO}(3)$  represents a special orthogonal group, and whose determinant is one. Note that  $R_i^T R_i = I$ , where  $I$  represents an identical matrix. The dynamics of a quadrotor  $i$  is shown as follows

$$\begin{cases} m\ddot{\mathcal{X}}_i = -mg\mathbf{e}_3 + R_i F_{T_i} \mathbf{e}_3 \\ \dot{R}_i = R_i \mathbf{S}(\Omega_i) \\ J\dot{\Omega}_i + \mathbf{S}(\Omega_i)J\Omega_i = \tau_i \end{cases} \tag{3}$$

where  $\mathcal{X}_i = [X_i, Y_i, Z_i]^T$  represents the coordinates of the center of mass of a quadrotor in the fixed inertial frame  $x_e y_e z_e$ . The Euler angles (pitch, roll, and yaw) are represented by the vector  $\Theta_i = [\phi_i, \theta_i, \psi_i]^T$ . The first equation of (3) represents the translational dynamics in the inertial frame, where  $m$  represents the mass of a quadrotor and  $g$  represents the gravity. The rotation matrix  $R_i$  in the second equation of (3) can transform the coordinates of a point from the body-fixed frame to the inertial frame. The third equation of (3) represents the

rotational dynamics of quadrotor  $i$ . The inertia matrix  $J$  of a quadrotor  $i$  is represented in the body-fixed frame, according to the assumptions on the physical structure of a quadrotor, we can obtain that  $J = \text{diag}\{I_{x_b}, I_{y_b}, I_{z_b}\}$  is diagonal, where scalars  $I_{x_b}$ ,  $I_{y_b}$  and  $I_{z_b}$  represent the moments of inertia with respect to  $x_b$ ,  $y_b$  and  $z_b$  respectively. Additionally, we assume that all the quadrotors in the flock have the same physical coefficients such as the mass and the inertia matrix. The angular velocity of the quadrotor  $i$  in the body-fixed frame is represented by  $\Omega_i \in \mathbb{R}^3$ . The function  $S(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$  represents an operation that transforms a vector in  $\mathbb{R}^3$  to a skew-symmetric matrix  $\mathbb{R}^{3 \times 3}$ . Given two arbitrary vectors  $v_1, v_2 \in \mathbb{R}^3$ , the function  $S(\cdot)$  satisfies the property  $S(v_1) \cdot v_2 = v_1 \times v_2$ . Then, according to the definition of cross product and the operation  $S(\cdot)$ , we conclude that  $S(\Omega_i) \in \mathbb{R}^{3 \times 3}$  is skew-symmetric matrix in terms of the members of  $\Omega_i = [p_i, q_i, r_i]^T$  as follows

$$S(\Omega_i) = \begin{bmatrix} 0 & -r_i & q_i \\ r_i & 0 & -p_i \\ -q_i & p_i & 0 \end{bmatrix} \tag{4}$$

where  $p_i, q_i$  and  $r_i$  represent the angular velocities with respect to the body-fixed frame of the quadrotor  $i$ .

**Table 1.** The denotations of the variables in the dynamic model of quadrotor.

Symbol	Description
$x_e y_e z_e$	Fixed inertial frame
$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$	Basis vector of inertial frame $\mathbf{e}_1 = [1, 0, 0]^T$ , $\mathbf{e}_2 = [0, 1, 0]^T$ , $\mathbf{e}_3 = [0, 0, 1]^T$
$\mathcal{X}_i = [X_i, Y_i, Z_i]^T$	Coordinates of the center of mass of a quadrotor in $x_e y_e z_e$
$\Theta_i = [\phi_i, \theta_i, \psi_i]^T$	Euler angles (pitch, roll and yaw)
$R_z(\psi_i), R_y(\theta_i), R_x(\phi_i)$	Rotation matrices in yaw, pitch and roll
$R_i$	Rotation matrix $R_i = R_z(\psi_i)R_y(\theta_i)R_x(\phi_i)$
$J = \text{diag}\{I_{x_b}, I_{y_b}, I_{z_b}\}$	Inertia matrix represented in the body-fixed frame $x_b y_b z_b$
$\Omega_i = [p_i, q_i, r_i]^T$	Angular velocity of the quadrotor $i$ in the body-fixed frame
$S(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$	Operation from vector in $\mathbb{R}^3$ to skew-symmetric matrix $\mathbb{R}^{3 \times 3}$
$\tau_i = [\tau_{\phi_i}, \tau_{\theta_i}, \tau_{\psi_i}]^T \in \mathbb{R}^3$	The moments of roll, pitch and yaw
$F_{T_i}$	Total thrust force of quadrotor $i$
$\mathcal{X}_i = [\mathcal{X}_i^T, \dot{\mathcal{X}}_i^T]^T$	State of quadrotor $i$
$u_i = [F_{T_i}, \tau_i^T]^T \in \mathbb{R}^4$	Control input of the quadrotor $i$

We refer to  $\phi_i, \theta_i, \psi_i$  as the roll, pitch, and yaw angles,  $\tau_i = [\tau_{\phi_i}, \tau_{\theta_i}, \tau_{\psi_i}]^T \in \mathbb{R}^3$  represents the roll, pitch, and yaw moments. The thrust force is represented by  $F_{T_i}$ . We note that  $\mathbf{e}_3 = [0, 0, 1]^T$  is a constant vector. The attitude dynamics of each quadrotor is generated by moments  $\tau_{\phi_i}, \tau_{\theta_i}$  and  $\tau_{\psi_i}$  and thrust force  $F_{T_i}$ .

The rotation matrix  $R_i$  from the body-fixed frame to the inertial frame is the sequence of roll-pitch-yaw (namely  $\phi - \theta - \psi$ ), i.e.,  $R_i = R_z(\psi_i)R_y(\theta_i)R_x(\phi_i)$ , where  $R_z(\psi_i), R_y(\theta_i), R_x(\phi_i)$  represents the rotation matrices of yaw, pitch, and roll.

$$R_i = \begin{bmatrix} \cos \psi_i \cos \theta_i & \cos \psi_i \sin \theta_i \sin \phi_i - \cos \phi_i \sin \psi_i & \sin \psi_i \sin \phi_i + \cos \psi_i \cos \phi_i \sin \theta_i \\ \cos \theta_i \sin \psi_i & \cos \psi_i \cos \phi_i + \sin \psi_i \sin \theta_i \sin \phi_i & \cos \phi_i \sin \psi_i \sin \theta_i - \cos \psi_i \sin \phi_i \\ -\sin \theta_i & \cos \theta_i \sin \phi_i & \cos \theta_i \cos \phi_i \end{bmatrix}$$

Thus, according to Equation (3), we obtain the translational dynamics as follows

$$\begin{aligned} \ddot{X}_i &= (\sin \psi_i \sin \phi_i + \cos \psi_i \cos \phi_i \sin \theta_i) \frac{F_{T_i}}{m} \\ \ddot{Y}_i &= (\cos \phi_i \sin \psi_i \sin \theta_i - \cos \psi_i \sin \phi_i) \frac{F_{T_i}}{m} \\ \ddot{Z}_i &= -g + (\cos \theta_i \cos \phi_i) \frac{F_{T_i}}{m} \end{aligned} \tag{5}$$

According to the second equation in (3), we can obtain

$$\Omega_i = \begin{bmatrix} p_i \\ q_i \\ r_i \end{bmatrix} = R_i^T \dot{\psi}_i \mathbf{e}_3 + \dot{\theta}_i R_x^T(\phi_i) R_y^T(\theta_i) \mathbf{e}_2 + \dot{\phi}_i R_x^T(\phi_i) \mathbf{e}_1 = T_i \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} \tag{6}$$

where

$$T_i = \begin{bmatrix} R_x^T(\phi_i) \mathbf{e}_1 & R_x^T(\phi_i) R_y^T(\theta_i) \mathbf{e}_2 & R_i^T \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta_i \\ 0 & \cos \phi_i & \cos \theta_i \sin \phi_i \\ 0 & -\sin \phi_i & \cos \theta_i \cos \phi_i \end{bmatrix}$$

Let us denote

$$\tilde{T}_i = \begin{bmatrix} \tan \theta_i & 0 & \cos \theta_i + \tan \theta_i \sin \theta_i \\ 0 & -\cos \theta_i & 0 \\ \sec \theta_i & 0 & \tan \theta_i \end{bmatrix}$$

According to (6) and using  $\tilde{T}_i$ , we can obtain the rotational dynamics as follows

$$\begin{bmatrix} \ddot{\phi}_i \\ \ddot{\theta}_i \\ \ddot{\psi}_i \end{bmatrix} = (JT_i)^{-1} \tau_i + \tilde{T}_i \cdot \begin{bmatrix} \dot{\phi}_i \dot{\theta}_i \\ \dot{\phi}_i \dot{\psi}_i \\ \dot{\theta}_i \dot{\psi}_i \end{bmatrix} + (JT_i)^{-1} \mathbf{S}^T \left( T_i \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} \right) JT_i \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} \tag{7}$$

The quadrotor is known as an under-actuated system. When the quadrotor tilts, a component of the total thrust is directed sideways and the aircraft translates horizontally. Therefore, the translational motion of quadrotor is generated by implicitly controlling its attitude (pitch and roll). This fact can be observed from (5), where the translation is controlled by the rotational angles.

In general, the control of the quadrotor is composed of inner and outer loop controllers, in which the outer loop is the translation control and the inner loop is the rotation control. Within this control framework, the control output of the outer loop is treated as an input of the rotational control loop. Nevertheless, in this paper, the idea is to get rid of the structure of inner-outer-loop control, we use the theory of flatness to directly design the torques of the quadrotor.

We assume that  $\theta \neq \pm \frac{\pi}{2}$ . According to (3), (5), and (7) the quadrotor dynamics in inertial frame is represented in state space as Equation (8). We denote the state of a quadrotor  $i$  by  $\mathbf{x}_i \in \mathbb{R}^{12}$ , which is defined by  $\mathbf{x}_i = [\mathcal{X}_i^T, \dot{\mathcal{X}}_i^T, \Theta_i^T, \dot{\Theta}_i^T]^T$ . We denote the control input of the quadrotor  $i$  by  $u_i = [F_{T_i}, \tau_i^T]^T \in \mathbb{R}^4$ . The state space quadrotor  $i$  dynamics is given by

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i) + \mathbf{g}(\mathbf{x}_i) u_i \tag{8}$$

where  $\mathbf{f} : \mathbb{R}^{12} \rightarrow \mathbb{R}^{12 \times 12}$  and  $\mathbf{g} : \mathbb{R}^{12} \rightarrow \mathbb{R}^{12 \times 4}$  satisfy

$$\mathbf{f}(\mathbf{x}_i) = \begin{bmatrix} \dot{\mathcal{X}}_i \\ \mathbf{g} \mathbf{e}_3 \\ \dot{\Theta}_i \\ \tilde{T}_i \cdot \begin{bmatrix} \dot{\phi}_i \dot{\theta}_i \\ \dot{\phi}_i \dot{\psi}_i \\ \dot{\theta}_i \dot{\psi}_i \end{bmatrix} + (JT_i)^{-1} \mathbf{S}^T (T_i \cdot \dot{\Theta}_i) JT_i \cdot \dot{\Theta}_i \end{bmatrix} \quad \mathbf{g}(\mathbf{x}_i) = \begin{bmatrix} \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\ -\frac{1}{m} R_i \mathbf{e}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 1} & (JT_i)^{-1} \end{bmatrix}$$

The output vector  $y_i$  of the quadrotor  $i$  is composed by the positions, linear velocities, angles, and the angular velocities, which will be used in the controller design. Then,  $y_i$  is given as follows

$$y_i = C_i \mathbf{x}_i \tag{9}$$

where  $C_i = \begin{bmatrix} C^t \\ C_i^r \end{bmatrix}$ . Matrix  $C_i$  is divided into two parts as follows

$$C^t = \left[ \begin{array}{c|c} I_3 & \mathbf{0}_{6 \times 9} \\ \hline \mathbf{0}_{3 \times 3} & \end{array} \right] \text{ and } C_i^r = \left[ \begin{array}{cc|c} \mathbf{0}_{6 \times 6} & I_3 & \mathbf{0}_{3 \times 3} \\ \hline \mathbf{0}_{3 \times 3} & T_i & \end{array} \right]$$

It is important to note that the translational dynamics are modeled in the inertial frame while the rotational dynamics are modeled in the body-fixed frame, although the states ( $\mathcal{X}_i, \dot{\mathcal{X}}_i, \Theta_i$  and  $\dot{\Theta}_i$ ) of the quadrotor model (8) are represented in the inertial frame. Indeed, in the body-fixed frame, the inertia matrix  $J$  is diagonal and then the rotational dynamics are easier to calculate than in the inertial frame, where  $J$  is not diagonal and time-variant. Note that in this work, the dynamics of the motors are omitted for the sake of simplicity.

The identification of the quadrotor parameters is not detailed here. Therefore, in the sequel, the parameters of the quadrotor, such as mass, inertias, and physical size are supposed to be known.

For a quadrotor system, the flat output can be selected according to the following proposition.

**Proposition 3.** *The vector  $F_i = [\mathcal{X}_i^T, \psi_i]^T$  is a flat output of system (8).*

**Proof.** See Appendix A.1.  $\square$

### 3.2. Nonlinear-Flatness Based Decoupling Control

We firstly define the aggressive formation of quadrotors as follows.

**Assumption 1** (Aggressive formation assumption). *The attitude angles pitch  $\theta_i$  and roll  $\phi_i$  of each quadrotor can be greater than  $15^\circ$  (but smaller than  $30^\circ$ ) in aggressive formation modes.*

According to Proposition 3, we conclude that the desired motion of a quadrotor can be uniquely specified by the 3D coordinates  $\mathcal{X}_i$  and the yaw angle  $\psi_i$ . Therefore, the decentralized formation tracking problem is to find the guidance vector  $[X_i^d, Y_i^d, Z_i^d, \psi_i^d]^T$  for each quadrotor. The guidance vector is composed by the neighbors states, which will be designed in Section 4.

The control input  $u_i = [F_{T_i}, \tau_{\phi_i}, \tau_{\theta_i}, \tau_{\psi_i}]^T$  is designed in order to minimize the tracking errors of  $\mathcal{X}_i - \mathcal{X}_i^d$  and  $\psi_i - \psi_i^d$ , where  $\mathcal{X}_i^d$  and  $\psi_i^d$  represent the desired 3D position and the heading direction (yaw angle). According to Equation (A1), the controller for the altitude is designed as follows

$$F_{T_i} = \frac{(\ddot{Z}_i^d - k_{2Z}(\dot{Z}_i - \dot{Z}_i^d) - k_{1Z}(Z_i - Z_i^d) + g)m}{\cos \theta_i \cos \phi_i} \tag{10}$$

where  $Z_i^d$  represents the desired value of the altitude. Let us denote by  $e_{Z_i} = Z_i - Z_i^d$  the tracking error of the altitude and substitute (10) into the third equation in (5). Then, we have

$$\ddot{e}_{Z_i} = -k_{2Z}\dot{e}_{Z_i} - k_{1Z}e_{Z_i}$$

Obviously, the altitude can exponentially track the given desired altitude trajectory  $Z_i^d$  for some selected positive scalars  $k_{1z}$  and  $k_{2z}$ . The desired altitude trajectory should satisfy  $Z_i^d \in C^3(\mathbb{R}, \mathbb{R})$ . Then, the altitude dynamics are decoupled with the other states of the quadrotor.

We rewrite the dynamic of the attitude angles in (7) as follows

$$\ddot{\Theta}_i = \tilde{T}_i \cdot \begin{bmatrix} \dot{\phi}_i \dot{\theta}_i \\ \dot{\phi}_i \dot{\psi}_i \\ \dot{\theta}_i \dot{\psi}_i \end{bmatrix} + (JT_i)^{-1} S^T (T_i \cdot \dot{\Theta}_i) JT_i \cdot \dot{\Theta}_i + (JT_i)^{-1} \tau_i \tag{11}$$

The inputs  $\tau_i = [\tau_{\phi_i}, \tau_{\theta_i}, \tau_{\psi_i}]^T$  are designed as follows

$$\tau_i = JT_i \bar{\tau}_i + J\tilde{T}_i \begin{bmatrix} \dot{\phi}_i \dot{\theta}_i \\ \dot{\phi}_i \dot{\psi}_i \\ \dot{\theta}_i \dot{\psi}_i \end{bmatrix} - S^T (T_i \cdot \dot{\Theta}_i) JT_i \cdot \dot{\Theta}_i \tag{12}$$

Substituting (12) into (11), we obtain

$$\ddot{\Theta}_i = \bar{\tau}_i \tag{13}$$

where  $\bar{\tau}_i = \ddot{\Theta}_i^d - k_{2\Theta_i}(\dot{\Theta}_i - \dot{\Theta}_i^d) - k_{1\Theta_i}(\Theta_i - \Theta_i^d)$ . Notations  $k_{1\Theta_i}$  and  $k_{2\Theta_i}$  represent two diagonal gain matrices.

If  $\Theta_i^d, \dot{\Theta}_i^d$  and  $\ddot{\Theta}_i^d$  are calculated according to the equations in (A4), namely, replacing  $F_i, \dot{F}_i, \ddot{F}_i, \ddot{\ddot{F}}_i$ , and  $\ddot{\ddot{F}}_i$  by  $F_i^d, \dot{F}_i^d, \ddot{F}_i^d, \ddot{\ddot{F}}_i^d$ , and  $\ddot{\ddot{\ddot{F}}}_i^d$ , it is trivial to observe that the attitude angles can exponentially track the desired attitude angles that are in terms of  $F_i$  and its derivatives. However, the controller (12) is open-loop on the translational dynamics, because the position feedback is not used in the controller design. If the dynamics of the quadrotor is precisely modeled and no external disturbances and sensors noise exist, this controller can perfectly perform. However, in practice, the dynamics of the system can be never precisely modeled, additionally, the noise and disturbance always exist. Therefore, the closed-loop control should be used, such that the feedback of the positions and translational velocities are required.

We observe from the equations in (A4) that the yaw angle is a component of the flat output  $F_i$ , then, the desired values of the yaw angle ( $\psi_i^d, \dot{\psi}_i^d$  and  $\ddot{\psi}_i^d$ ) are trivial to obtain. However, the desired pitch and roll angles are not explicitly given by the components of the flat output  $F_i^d$  and its derivatives. We will discuss how to obtain the terms  $\phi_i^d, \dot{\phi}_i^d, \ddot{\phi}_i^d, \theta_i^d, \dot{\theta}_i^d$  and  $\ddot{\theta}_i^d$  as follows.

As analyzed before, the design of  $\tau_{\psi_i}$  is simple. Normally, in order to represent the control input by using the flatness output, the high-order derivatives of  $F_i$  are calculated. Specifically, in our case, in order to represent the torques  $\tau_{\phi_i}$  and  $\tau_{\theta_i}$ , two supplementary derivatives of  $\ddot{X}_i$  and  $\ddot{Y}_i$  in Equation (A2) are necessary and we obtain

$$\begin{bmatrix} X_i \\ Y_i \end{bmatrix}^{(4)} = A \begin{bmatrix} \ddot{\phi}_i \\ \ddot{\theta}_i \end{bmatrix} + \zeta(\Theta_i, \dot{\Theta}_i, \ddot{\psi}_i, \ddot{Z}_i, \ddot{\ddot{Z}}_i, \ddot{\ddot{Z}}_i) = v_i$$

where matrix  $A$  satisfies

$$A = (\ddot{Z}_i + g) \times \begin{bmatrix} \sin \psi_i \sec^2 \phi_i \sec \theta_i & \cos \psi_i \sec^2 \theta_i + \sin \psi_i \tan \phi_i \tan \theta_i \sec \theta_i \\ -\cos \psi_i \sec^2 \phi_i \sec \theta_i & \sin \psi_i \sec^2 \theta_i - \cos \psi_i \tan \phi_i \tan \theta_i \sec \theta_i \end{bmatrix}$$

The function  $\zeta$  represents the terms of the second-order derivative of the right side of Equation (A2) except the term that contains  $\ddot{\phi}_i$  and  $\ddot{\theta}_i$ .

The vector  $v_i = [v_{i1} \ v_{i2}]^T$  is composed by the new control inputs  $v_{i1}$  and  $v_{i2}$ , which are decoupled on axes  $X$  and  $Y$ . Then, the new trajectory tracking controllers in closed loop on the plane  $X - Y$  are given as follows

$$\begin{aligned} v_{i1} &= \ddot{\ddot{X}}_i^d - k_{4X}(\ddot{\ddot{X}}_i - \ddot{\ddot{X}}_i^d) - k_{3X}(\ddot{\ddot{X}}_i - \ddot{\ddot{X}}_i^d) - k_{2X}(\ddot{\ddot{X}}_i - \ddot{\ddot{X}}_i^d) - k_{1X}(X_i - X_i^d) \\ v_{i2} &= \ddot{\ddot{Y}}_i^d - k_{4Y}(\ddot{\ddot{Y}}_i - \ddot{\ddot{Y}}_i^d) - k_{3Y}(\ddot{\ddot{Y}}_i - \ddot{\ddot{Y}}_i^d) - k_{2Y}(\ddot{\ddot{Y}}_i - \ddot{\ddot{Y}}_i^d) - k_{1Y}(Y_i - Y_i^d) \end{aligned} \tag{14}$$



Observe that  $\det A = (\ddot{Z}_i + g)\sec^2 \phi_i \sec^3 \theta_i$ . Therefore, when  $\ddot{Z}_i \neq -g$  and  $|\phi_i| < \frac{\pi}{2}$ ,  $|\theta_i| < \frac{\pi}{2}$ ,  $\det A$  is finite and positive such that  $A$  is invertible. In fact, if  $|\phi_i| < \frac{\pi}{2}$ ,  $|\theta_i| < \frac{\pi}{2}$ , we have  $\ddot{Z}_i > -g$ .  $\ddot{Z}_i$  will be never equal to  $-g$ , unless that all the rotors of the quadrotor stop rotating. This case is not considered in this work.

Then, according to Equation (13), we can design the torque  $\bar{\tau}_i = [\bar{\tau}_{\phi_i} \quad \bar{\tau}_{\theta_i} \quad \bar{\tau}_{\psi_i}]^T$  as follows

$$\begin{bmatrix} \bar{\tau}_{\phi_i} \\ \bar{\tau}_{\theta_i} \\ \bar{\tau}_{\psi_i} \end{bmatrix} = \begin{bmatrix} A^{-1}(v_i - \zeta) \\ \ddot{\psi}_i^d - k_{2\psi_i}(\dot{\psi}_i - \dot{\psi}_i^d) - k_{1\psi_i}(\psi_i - \psi_i^d) \end{bmatrix} \tag{15}$$

Then, substituting (15) into (12), we obtain the three torques.

In practice, the velocities  $\dot{X}_i, \dot{Y}_i$  and its higher-order derivatives  $\ddot{X}_i, \ddot{X}_i, \ddot{Y}_i$  and  $\ddot{Y}_i$  are not directly measurable (not contained in the output  $y_i$  (9)), they are obtained by using Equation (A2) and its derivatives. We will present in the following subsection how to obtain the these derivatives.

Under Assumption 1, we can assume that  $\tan \theta_i \approx \theta_i, \tan \phi_i \approx \phi_i, \cos \theta_i \approx 1$ , and  $\cos \phi_i \approx 1$ . The desired yaw angle is zero. Using controller (11),  $\psi_i \approx 0$ . Then,  $\cos \psi_i \approx 1$  and  $\sin \psi_i \approx 0$ . We also assume that the altitude is stabilized and keep constant, such that  $Z_i \approx Z_i^d$  and  $\dot{Z}_i \approx 0$ . Then we have,  $\ddot{Z}_i \approx 0$ . Therefore, according to the analysis in Appendix A.2, we consider

$$\begin{aligned} \ddot{X}_i &= \theta_i g \\ \ddot{Y}_i &= -\phi_i g \end{aligned} \tag{16}$$

Then, the controllers in (14) becomes

$$\begin{aligned} v_{i1} &= \ddot{X}_i^d - k_{4X}(\dot{\theta}_i g - \ddot{X}_i^d) - k_{3X}(\theta_i g - \dot{X}_i^d) - k_{2X}(\dot{X}_i - \dot{X}_i^d) - k_{1X}(X_i - X_i^d) \\ v_{i2} &= \ddot{Y}_i^d - k_{4Y}(-\dot{\phi}_i g - \ddot{Y}_i^d) - k_{3Y}(-\phi_i g - \dot{Y}_i^d) - k_{2Y}(\dot{Y}_i - \dot{Y}_i^d) - k_{1Y}(Y_i - Y_i^d) \end{aligned} \tag{17}$$

Then, we obtain

$$\begin{bmatrix} \bar{\tau}_{\phi_i} \\ \bar{\tau}_{\theta_i} \\ \bar{\tau}_{\psi_i} \end{bmatrix} = \begin{bmatrix} -\frac{1}{g}v_{i2} \\ \frac{1}{g}v_{i1} \\ \ddot{\psi}_i^d - k_{2\psi_i}(\dot{\psi}_i - \dot{\psi}_i^d) - k_{1\psi_i}(\psi_i - \psi_i^d) \end{bmatrix} \tag{18}$$

The control strategy in (18) is based on the knowledge of high-order derivatives of the planar positions  $X_i^d$  and  $Y_i^d$  in guidance vector, which are normally not measurable. Note that, for each quadrotor, the guidance vector depends on neighbors states (and the reference formation trajectory (RFT), if the quadrotor is a leader). If the quadrotors move slowly and the RFT varies slowly, we can assume that the higher-order derivatives are null. However, the performance will not be satisfactory for aggressive formations. The fast convergence of the observer is crucial in aggressive formation. Therefore, we will propose in the next subsection an extended finite-time observer to estimate these high-order derivatives.

### 3.3. Extended Finite-Time Observer

An observer of the high-order derivatives guidance vector for each quadrotor is proposed to get the feedforwards. We take  $X_i^d$  for example, according to  $v_{i1}$  in (17), we have to estimate  $\dot{X}_i^d, \ddot{X}_i^d, \ddot{X}_i^d$  and  $\ddot{X}_i^d$ . Therefore, we construct an observer to estimate the high-order derivatives of the positions in guidance vector.

Before designing the finite-time observer, we introduce the following lemma about finite-time stability.

**Lemma 1.** [24] Consider the system  $\dot{x} = f(x, u)$ . Suppose that there exist continuous differentiable functions  $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}^+$ , real numbers  $c > 0$ , and  $0 < \alpha < 1$ , such that  $V(0) = 0, V(x) > 0$  for any  $x \neq 0$ , and  $\dot{V}(x) \leq -cV^\alpha(x)$ . Then, for all  $x(t_0) = x_0$  in the neighborhood  $\mu$  of the origin, there exists settling time function  $T(x_0) : \mu \rightarrow \mathbb{R}^+,$  such that  $T(x_0) \leq \frac{V^{1-\alpha}(x_0)}{c(1-\alpha)}$ .

Denoting that  $x_i^d = [X_i^d, \dot{X}_i^d, \ddot{X}_i^d, \ddot{X}_i^d]^T$ , the dynamics of  $x_i^d$  yields

$$\begin{aligned} \dot{x}_i^d &= A_t x_i^d + B_t \ddot{X}_i^d \\ y_i &= C_t x_i^d \end{aligned} \tag{19}$$

where

$$A_t = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B_t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ and } C_t = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T$$

The extended state is  $(x_i^d)_E = [(x_i^d)^T, \ddot{X}_i^d]^T$  and its estimation  $\hat{x}_i^d = [\hat{X}_i^d, \hat{\dot{X}}_i^d, \hat{\ddot{X}}_i^d, \hat{\ddot{X}}_i^d]^T$ . Then, we propose the extended observing model as follows

$$\dot{\hat{x}}_i^d = \begin{bmatrix} A_t & B_t \\ \mathbf{0}_4^T & 0 \end{bmatrix} \hat{x}_i^d - L_t (\hat{X}_i^d - X_i^d) \tag{20}$$

where  $L_t$  represents the gain matrix which satisfies  $L_t = [L_{t1} \ L_{t2}]^T$ , where  $L_{t1} = [k_{o1} \ k_{o2} \ k_{o3} \ k_{o4}]$  and  $L_{t2} = k_{o5}$ . Let us denote  $\tilde{x}_i^d = \hat{x}_i^d - (x_i^d)_E$ , then, according to (19) and (20),  $\dot{\tilde{x}}_i^d$  satisfies

$$\dot{\tilde{x}}_i^d = \begin{bmatrix} A_t & B_t \\ \mathbf{0}_4^T & 0 \end{bmatrix} \tilde{x}_i^d - \begin{bmatrix} \mathbf{0}_4 \\ 1 \end{bmatrix} (X_i^d)^{(5)} - L_t [C_t, 0] \tilde{x}_i^d \tag{21}$$

Denoting

$$A_o = \begin{bmatrix} A_t - L_{t1}^T C_t & B_t \\ -L_{t2}^T C_t & 0 \end{bmatrix}, B_o = \begin{bmatrix} \mathbf{0}_4 \\ 1 \end{bmatrix}$$

Suppose that  $J$  is the Jordan standard form of matrix  $A_o$  by implementing a Jordan transformation using matrix  $M$ .

**Theorem 1.** For each agent, the extended guidance vector observer is given in (20). If  $X_i^d \in C^5(\mathbb{R}^+, \mathbb{R}^2)$ , then, (i) the matrix  $A_o$  is stabilizable by proper selected gain matrix  $L_t$ ; (ii) the observing error  $\tilde{x}_i^d$  converge to the compact sphere  $\|\tilde{x}_i^d\| \leq \beta$ , where  $\beta > \frac{\lambda \rho}{\lambda(\gamma-1)}$ , within finite-time

$$T_{converge} \leq \frac{\|\tilde{x}_i^d(0)\|}{(\gamma-1)\beta\lambda/\bar{\lambda}-\rho} \tag{22}$$

where we denote by  $\underline{\lambda} = \min\{|\lambda_i(M)|\}$  and  $\bar{\lambda} = \max\{|\lambda_i(M)|\}$ , scalar  $\gamma > 1$ . The scalar  $\rho$  represents the bound of  $|(X_i^d)^{(5)}|$ .

**Proof.** (i). The matrix  $A_o$  can be rewritten in the following form.

$$A_o = \begin{pmatrix} -L_{t1}^T & I_4 \\ -L_{t2} & \mathbf{0}_4^T \end{pmatrix}$$

which is stable, for some gains  $k_{oi} > 0, i = \{1, \dots, 5\}$  in  $L_t$ .

(ii). The control gains in  $L_t$  are selected such that the nearest pole of  $A_o$  to imaginary axis is placed to  $-\gamma$ , where  $\gamma > 1$ . The Equation (21) can be rewritten as  $\dot{\tilde{x}}_i^d = A_o \tilde{x}_i^d - B_o (X_i^d)^{(5)}$ , where  $|(X_i^d)^{(5)}| < \rho$  is bounded, since  $X_i^d(t) \in C^5(\mathbb{R}^+, \mathbb{R}^2)$ . Choose an invertible matrix  $M$ , which transforms  $A_o$  to Jordan standard form  $J_o = M A_o M^{-1}$ . We process the variable transformation  $\epsilon = M \tilde{x}_i^d$ , then we have  $\dot{\epsilon} = J_o \epsilon - M B_o (X_i^d)^{(5)}$ . The Lyapunov function candidate is chosen by  $V(\epsilon) = \frac{1}{2} \epsilon^T \epsilon$ . Its derivative yields,  $\dot{V}(\epsilon) = \epsilon^T J_o \epsilon - \epsilon^T M B_o (X_i^d)^{(5)}$ . We suppose that the Jordan matrix  $J_o$  has  $l$  Jordan blocks, the size of the

blocks is  $m_i, i = \{1, \dots, l\}$ . The eigenvalue of each block is represented by  $\lambda_i(J_o)$ . The Lyapunov function candidate can be rewritten by

$$V(\epsilon) = \sum_{i=1}^{m_1} \frac{1}{2} \epsilon_i^T \epsilon_i + \dots + \sum_{j=m_1+\dots+m_{l-1}+1}^{m_1+\dots+m_l} \frac{1}{2} \epsilon_j^T \epsilon_j = \sum_{i=1}^l V_i(\epsilon)$$

The quadratic term  $\epsilon^T J_o \epsilon$  in  $\dot{V}(\epsilon)$  yields,

$$\epsilon^T J_o \epsilon = \lambda_1(J_o) \sum_{j=1}^{m_1} \epsilon_j^2 + \sum_{j=1}^{m_1-1} \epsilon_j \epsilon_{j+1} + \dots + \lambda_l(J_o) \sum_{j=m_1+\dots+m_{l-1}+1}^{m_1+\dots+m_l} \epsilon_j^2 + \sum_{j=m_1+\dots+m_{l-1}+1}^{m_1+\dots+m_l-1} \epsilon_j \epsilon_{j+1} \quad (23)$$

where the intersection term  $\sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i-1} \epsilon_j \epsilon_{j+1}$  is null, when  $m_i = 1$ .

We note that  $\sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i} \epsilon_j^2 - \sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i-1} \epsilon_j \epsilon_{j+1} = (1/2) \epsilon_{m_1+\dots+m_{i-1}+1}^2 + (1/2) \sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i-1} (\epsilon_j - \epsilon_{j+1})^2 + (1/2) \epsilon_{m_1+\dots+m_i}^2 \geq 0$ . Thus,  $\sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i} \epsilon_j^2 \geq \sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i-1} \epsilon_j \epsilon_{j+1}$ , i.e.,  $\sum_{j=m_1+\dots+m_{i-1}+1}^{m_1+\dots+m_i-1} \epsilon_j \epsilon_{j+1} \leq 2V_i(\epsilon)$ . Add both side from  $i = 1$  to  $l$ , then we have

$$\sum_{j=1}^{m_1-1} \epsilon_j \epsilon_{j+1} + \dots + \sum_{j=m_1+\dots+m_{l-1}+1}^{m_1+\dots+m_l-1} \epsilon_j \epsilon_{j+1} \leq 2V(\epsilon) \quad (24)$$

According to (23) and (24), we have

$$\epsilon^T J_o \epsilon \leq \sum_{i=1}^l 2\lambda_i(J_o) V_i(\epsilon) + 2V(\epsilon) \leq 2(1 - \gamma) V(\epsilon) \quad (25)$$

Then, the derivative of the Lyapunov function candidate yields,

$$\dot{V}(\epsilon) = \epsilon^T J_o \epsilon - \epsilon^T M B_o (X_i^d)^{(5)} \leq 2(1 - \gamma) V(\epsilon) - \epsilon^T M B_o (X_i^d)^{(5)}$$

Therefore,

$$\begin{aligned} \dot{V}(\epsilon) &\leq 2(1 - \gamma) V(\epsilon) + \bar{\lambda} |(X_i^d)^{(5)}| \|\epsilon\| = (1 - \gamma) \|\epsilon\|^2 + \bar{\lambda} \rho \|\epsilon\| = [(1 - \gamma) \|\epsilon\| + \bar{\lambda} \rho] \|\epsilon\| \\ &= \sqrt{2} [(1 - \gamma) \|\epsilon\| + \bar{\lambda} \rho] V^{\frac{1}{2}}(\epsilon) = -\sqrt{2} [(\gamma - 1) \|\epsilon\| - \bar{\lambda} \rho] V^{\frac{1}{2}}(\epsilon) \end{aligned} \quad (26)$$

When  $\|\tilde{x}_i^d\| \geq \beta$ , then,  $\|\epsilon\| = \|M\tilde{x}_i^d\| \geq \underline{\lambda} \|\tilde{x}_i^d\| \geq \underline{\lambda} \beta$ . Note that  $\gamma > 1$ , thus,  $(\gamma - 1) \|\epsilon\| - \bar{\lambda} \rho \geq (\gamma - 1) \underline{\lambda} \beta - \bar{\lambda} \rho$ . Thus,  $\dot{V}(\epsilon) \leq -\sqrt{2} [(\gamma - 1) \underline{\lambda} \beta - \bar{\lambda} \rho] V^{\frac{1}{2}}(\epsilon)$ . Hence,  $\dot{V}(\epsilon)$  is negative definite, since  $\beta > \frac{\bar{\lambda} \rho}{\underline{\lambda}(\gamma - 1)}$ . In addition,  $\dot{V}(\epsilon) = 0$ , if and only if  $V(\epsilon) = 0$ . Note that  $V(\epsilon) = \frac{1}{2} \epsilon^T \epsilon = 0$ , only if  $\epsilon = 0$  which is equivalent to that the observer error  $\tilde{x}_i^d = 0$ . Therefore, the observer error is asymptotic stable.

According to Lemma 1,  $\epsilon$  will converge into the compact sphere within the finite-time  $T_{converge}$ .

$$T_{converge} \leq \frac{\sqrt{2} V^{\frac{1}{2}}(\epsilon(0))}{(\gamma - 1) \underline{\lambda} \beta - \bar{\lambda} \rho} = \frac{\|M\tilde{x}_i^d(0)\|}{(\gamma - 1) \underline{\lambda} \beta - \bar{\lambda} \rho} \leq \frac{\|\tilde{x}_i^d(0)\|}{(\gamma - 1) \beta \underline{\lambda} / \bar{\lambda} - \rho}$$

The proof is completed.  $\square$

**Remark 2.** The observing error  $\tilde{x}_i^d$  is ultimately bounded, since  $|(X_i^d)^{(5)}|$  is bounded by  $\sqrt{2} \rho$ . According to (26), the observing error will converge to the invariant set  $\|\epsilon\| \leq \frac{\bar{\lambda} \rho}{\gamma - 1}$ . Therefore, by increasing observer gains, which renders a greater  $\gamma$ , the invariant set will become smaller. The observing error will smaller as well. Nevertheless, the gains cannot be chosen arbitrarily great in practice, since the sensor noise will become significant and lead to instability.

**Remark 3.** We note that since a decoupling quadrotor dynamics control law is derived in (16), the observer for the derivatives of  $Y_i^d$  can be designed independent with  $X_i^d$ .

The high-order derivatives are replaced by the estimated values using the observer (20). Then, the decoupled control law (17) becomes

$$\begin{aligned} v_{i1} &= (\hat{\ddot{X}}_i^d) - k_{4X}(\hat{\theta}_i g - (\hat{\ddot{X}}_i^d)) - k_{3X}(\theta_i g - (\hat{\dot{X}}_i^d)) - k_{2X}(\dot{X}_i - \hat{\dot{X}}_i^d) - k_{1X}(X_i - X_i^d) \\ v_{i2} &= (\hat{\ddot{Y}}_i^d) - k_{4Y}(-\hat{\phi}_i g - (\hat{\ddot{Y}}_i^d)) - k_{3Y}(-\phi_i g - (\hat{\dot{Y}}_i^d)) - k_{2Y}(\dot{Y}_i - \hat{\dot{Y}}_i^d) - k_{1Y}(Y_i - Y_i^d) \end{aligned} \tag{27}$$

where the scalars  $k_{iX}$  and  $k_{iY}$ ,  $i \in \{1, 2, 3, 4\}$  represent some selected positive gains.

In the following section, we will show how to find the  $X_i^d$  and  $Y_i^d$  in the guidance vector (since the control of the altitude and yaw is decoupled in double integrators, it is omitted in the following section).

### 4. Formation Controller Design

Since the control of altitude (10) and rotation (12) is decoupled with the planar motion control, they are assumed to be controlled in constant, without loss of generality.

#### 4.1. Available Guidance Vector

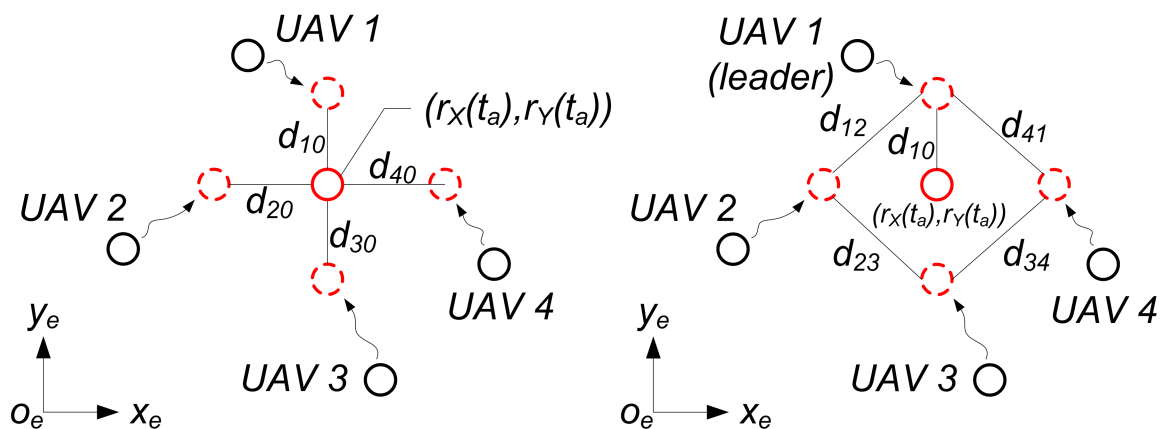
The formation controller of each quadrotor uses the relative states (positions, velocities, and the high-order derivatives with respect to its neighbors). If the quadrotor is a leader, besides the foregoing measurements, its formation controller also uses the relative states with respect to the RFT  $r(t) = [r_X(t), r_Y(t), r_Z(t), r_\psi(t)]^T$ , where  $r_Z$  and  $r_\psi$  are assumed to be constant. Then, for the sake of simplicity, the formation control in  $(X, Y)$  space is considered.

The objective of the formation tracking control is to guarantee that all the quadrotors track the RFT with some constant biases  $d_{i0} = [d_{i0}^X, d_{i0}^Y] \in \mathbb{R}^2$ , such that the quadrotors keep a formation pattern. In detail, the formation objective is to achieve the following equations.

$$\lim_{t \rightarrow \infty} \left\| \begin{pmatrix} X_i - r_X - d_{i0}^X \\ Y_i - r_Y - d_{i0}^Y \end{pmatrix} \right\| = 0 \quad \text{where } i = 1, \dots, n \tag{28}$$

for some initial condition  $x_i(0)$ ,  $i = 1, \dots, n$ .

An example of the formation tracking task of four quadrotors is shown in Figure 1 (left), where the solid red circle represents the RFT at time  $t_a$ . The dashed red circles represent the desired positions  $(\bar{X}_i^d(t_a), \bar{Y}_i^d(t_a))$  for the quadrotors  $i \in \mathcal{V}$  at  $t_a$ , where we denote by  $\bar{X}_i^d = r_X + d_{i0}^X$  and  $\bar{Y}_i^d = r_Y + d_{i0}^Y$ . The solid black circles represent the quadrotors' real positions at  $t_a$ , i.e.,  $(X_i(t_a), Y_i(t_a))$ ,  $i \in \mathcal{V}$ .



**Figure 1.** Left: Intuitive desired motion trajectories for quadrotors. Right: Available desired motion trajectories using neighbor-based states.

However, in decentralized formation control, the RFT  $r(t)$  is not available for the followers, therefore, it is not possible to design the formation tracking controller using the errors  $\bar{X}_i^d - X_i$  and  $\bar{Y}_i^d - Y_i$  for quadrotor  $i$ . For the followers, only the neighbors states are available for the formation controller design, as shown in Figure 1 (right). Now, the formation problem becomes: how to find the available guidance vector for quadrotors  $i \in \mathcal{V}$  in order to attain the formation task.

Let us make a sum of the relative position state vectors. Note that we drop the explicit expression of time in the expressions for the sake of simplicity. Similar to the foregoing analysis, we still take  $X$  for example.

$$\sum_{j \in \mathcal{N}_i} (X_i - X_j - d_{ij}^X) \quad \text{if } i \in \mathcal{V} - \mathcal{V}_L$$

$$\sum_{j \in \mathcal{N}_i} (X_i - X_j - d_{ij}^X) + X_i - r_X - d_{i0}^X \quad \text{if } i \in \mathcal{V}_L \tag{29}$$

The inter-distance satisfies  $d_{ij}^X = d_{i0}^X - d_{j0}^X$ . Then, Equation (29) can be rewritten as follows

$$\sum_{j \in \mathcal{N}_i} (X_i - r_X - d_{i0}^X - (X_j - r_X - d_{j0}^X)), \text{ if } i \in \mathcal{V} - \mathcal{V}_L$$

$$\sum_{j \in \mathcal{N}_i} (X_i - r_X - d_{i0}^X - (X_j - r_X - d_{j0}^X)) + X_i - r_X - d_{i0}^X, \text{ if } i \in \mathcal{V}_L$$

Then, the guidance vector for each quadrotor is given as follows

$$X_i^d = \begin{cases} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (X_j + d_{ij}^X), & \text{if } i \in \mathcal{V} - \mathcal{V}_L \\ \frac{1}{|\mathcal{N}_i|+1} \left( \sum_{j \in \mathcal{N}_i} (X_j + d_{ij}^X) + r_X + d_{i0}^X \right), & \text{if } i \in \mathcal{V}_L \end{cases} \tag{30}$$

We then observe that  $X_i^d$  is available for quadrotor  $i$ , since  $X_i^d$  is composed only by neighbors states.

$$X_i - X_i^d = \begin{cases} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \left( (X_i - \bar{X}_i^d) - (X_j - \bar{X}_j^d) \right), & \text{if } i \in \mathcal{V} - \mathcal{V}_L \\ \frac{1}{|\mathcal{N}_i|+1} \left( \sum_{j \in \mathcal{N}_i} \left( (X_i - \bar{X}_i^d) - (X_j - \bar{X}_j^d) \right) + X_i - \bar{X}_i^d \right), & \text{if } i \in \mathcal{V}_L \end{cases} \tag{31}$$

Then, according to definition of normalized interaction matrix in (2), we can rewrite Equation (31) in matrix form for all the quadrotors as follows

$$\begin{bmatrix} X_1 - X_1^d \\ \vdots \\ X_n - X_n^d \end{bmatrix} = \bar{G} \begin{bmatrix} X_1 - \bar{X}_1^d \\ \vdots \\ X_n - \bar{X}_n^d \end{bmatrix} \tag{32}$$

where  $\bar{G}$  represents the normalized interaction matrix.

According to Corollary 2 and the definition in Equation (2), we know that  $\bar{G}$  is invertible if each connected sub-graph of the multi-quadrotor system has at least one leader. Therefore, the formation task is achieved, i.e.,  $[X_1 - \bar{X}_1^d, \dots, X_n - \bar{X}_n^d]^T \rightarrow \mathbf{0}$ , as long as each quadrotor can precisely track the guidance vector  $X_i^d(t)$  and  $Y_i^d(t)$ , i.e., vector  $[X_1 - X_1^d, \dots, X_n - X_n^d] \rightarrow \mathbf{0}$ , and  $[Y_1 - Y_1^d, \dots, Y_n - Y_n^d] \rightarrow \mathbf{0}$ .

#### 4.2. Convergence Analysis

Now, the guidance vector  $[X_i^d, Y_i^d, Z_i^d, \psi_i^d]^T$  for each quadrotor is obtained (we note that  $Z_i^d$  and  $\psi_i^d$  are set constant, so omitted here). Their higher-order derivatives are estimated by using finite-time observer (20) in an aggressive formation. Then, these estimated derivatives are used as feedforwards in Equation (17), composing controller (12).

In practice, the estimation errors are usually high at  $t = 0$  s, when the initial states of the observer may greatly deviate from the real values. In order to avoid the instability caused by the big initial deviation of the observer, we implement saturation functions to the estimated high-order derivatives to prevent great observation errors at the beginning.

To avoid the instability of quadrotors caused by the great initial observation error, we propose a modified controller in Equation (33) using the saturation function as follows

$$\begin{aligned} v_{i1} &= \sigma_{b4}(\hat{X}_i^d) - k_{4X}(\dot{\theta}_i g - \sigma_{b3}(\hat{X}_i^d)) - k_{3X}(\theta_i g - \sigma_{b2}(\hat{X}_i^d)) \\ &\quad - k_{2X}(\dot{X}_i - \sigma_{b1}(\hat{X}_i^d)) - k_{1X}(X_i - X_i^d) \\ v_{i2} &= \sigma_{b4}(\hat{Y}_i^d) - k_{4Y}(-\dot{\phi}_i g - \sigma_{b3}(\hat{Y}_i^d)) - k_{3Y}(-\phi_i g - \sigma_{b2}(\hat{Y}_i^d)) \\ &\quad - k_{2Y}(\dot{Y}_i - \sigma_{b1}(\hat{Y}_i^d)) - k_{1Y}(Y_i - Y_i^d) \end{aligned} \tag{33}$$

The analysis of the convergence is given by the following theorem.

**Theorem 2.** Consider the quadrotors with dynamics (8) are controlled by the formation controller (12) composed by (33). The bounds are chosen as follows,

$$b_j = \begin{cases} \hbar_j, & \text{when } t \leq T_{convergence} \\ \infty & \text{when } t > T_{convergence} \end{cases} \tag{34}$$

where  $j = \{1, 2, 3, 4\}$  and  $\hbar_j \in \mathbb{R}^+$ . Then, the formation error for each quadrotor is ultimately bounded and converge to the invariant set as follows.

$$S_b = \{e_{X_i} \mid \|e_{X_i}\| \leq \frac{2\bar{\kappa}\beta\lambda_{\max}(P)}{\lambda_{\min}(Q)}\} \tag{35}$$

where  $\bar{\kappa} > 0$ . Notations  $P$  and  $Q$  represent some positive-definite matrices.

**Proof.** Using (13), (16), (18) and (33), the closed-loop planar dynamics of a quadrotor yields the following decoupled form.

$$\begin{aligned} \ddot{X}_i &= \sigma_{b4}(\hat{X}_i^d) - k_{4X}(\ddot{X}_i - \sigma_{b3}(\hat{X}_i^d)) - k_{3X}(\dot{X}_i - \sigma_{b2}(\hat{X}_i^d)) - k_{2X}(\dot{X}_i - \sigma_{b1}(\hat{X}_i^d)) - k_{1X}(X_i - X_i^d) \\ \ddot{Y}_i &= \sigma_{b4}(\hat{Y}_i^d) - k_{4Y}(\ddot{Y}_i - \sigma_{b3}(\hat{Y}_i^d)) - k_{3Y}(\dot{Y}_i - \sigma_{b2}(\hat{Y}_i^d)) - k_{2Y}(\dot{Y}_i - \sigma_{b1}(\hat{Y}_i^d)) - k_{1Y}(Y_i - Y_i^d) \end{aligned}$$

Take  $X_i$  for example and denote by  $e_{X_i} = [X_i - X_i^d, \dot{X}_i - \dot{X}_i^d, \ddot{X}_i - \ddot{X}_i^d, \ddot{X}_i - \ddot{X}_i^d]^T$ . Then, we can write the dynamics of  $e_{X_i}$  in state-space form as follows

$$\dot{e}_{X_i} = A_c e_{X_i} + \delta_i \tag{36}$$

where

$$A_c = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_{1X} & -k_{2X} & -k_{3X} & -k_{4X} \end{pmatrix}$$

and  $\delta = [0, 0, 0, \Delta_i]^T$ .

$$\Delta_i = \sigma_{b4}(\hat{X}_i^d) - \ddot{X}_i^d + k_{4X}(\sigma_{b3}(\hat{X}_i^d) - \ddot{X}_i^d) + k_{3X}(\sigma_{b2}(\hat{X}_i^d) - \dot{X}_i) + k_{2X}(\sigma_{b1}(\hat{X}_i^d) - \dot{X}_i) \tag{37}$$

- When  $t \leq T_{convergence}$ , it is trivial to prove that  $\|\delta_i\| = |\Delta| \leq \hat{h}_4 + |\ddot{\hat{X}}_i^d| + k_{4X}(\hat{h}_3 + |\ddot{\hat{X}}_i^d|) + k_{3X}(\hat{h}_2 + |\dot{\hat{X}}_i|) + k_{2X}(\hat{h}_1 + |\dot{\hat{X}}_i|) \leq \mathbf{d} < \infty$  is bounded, since  $r(t) \in C^5(\mathbb{R}^+, \mathbb{R}^2)$ .
- When  $t > T_{convergence}$ ,  $b_j = \infty$ , the saturations are removed, such that  $\Delta = \hat{\hat{X}}_i^d - \ddot{\hat{X}}_i^d + k_{4X}(\hat{\hat{X}}_i^d - \ddot{\hat{X}}_i^d) + k_{3X}(\hat{\hat{X}}_i^d - \ddot{\hat{X}}_i) + k_{2X}(\hat{\hat{X}}_i^d - \dot{\hat{X}}_i) = \kappa^T \tilde{x}_i^d$ , where  $\kappa = [0, k_{2X}, k_{3X}, k_{4X}, 1]^T$ , its norm is  $\|\kappa\| = \bar{\kappa}$ . According to theorem 1,  $\|\tilde{x}_i^d\| \leq \beta$  when  $t > T_{convergence}$ . Then, we conclude that  $\|\delta_i\| = |\Delta_i| = \bar{\kappa}\|\tilde{x}_i^d\| \leq \bar{\kappa}\beta < \infty$ .

Therefore, the perturbed term  $\delta$  in (36) is bounded and satisfying  $\|\delta_i\| \leq \delta_{max} = \max\{\mathbf{d}, \bar{\kappa}\beta\}$ , for all  $t \geq t_0$ .

The control gains are selected such that  $A_c$  is Hurwitz. Since  $A_c$  is Hurwitz, there exists a positive-definite matrix  $P$ , which solves the Lyapunov equation  $A_c^T P + P A_c = -Q$ , for a positive-definite matrix  $Q$ . A Lyapunov function is selected as  $V = e_{X_i}^T P e_{X_i}$ . Then, its time derivative yields

$$\begin{aligned} \dot{V} &= e_{X_i}^T (A_c^T P + P A_c) e_{X_i} + 2e_{X_i}^T P \delta_i \leq -e_{X_i}^T Q e_{X_i} + 2\lambda_{max}(P)\|e_{X_i}\|\|\delta_i\| \\ &\leq -\lambda_{min}(Q)\|e_{X_i}\|^2 + 2\lambda_{max}(P)\|e_{X_i}\|\|\delta_i\| = \|e_{X_i}\|(2\delta_{max}\lambda_{max}(P) - \lambda_{min}(Q)\|e_{X_i}\|) \end{aligned}$$

Then,  $\dot{V} < 0$ , if  $\|e_{X_i}\| > \frac{2\delta_{max}\lambda_{max}(P)}{\lambda_{min}(Q)}$ . When  $t > T_{convergence}$ , according to LaSalle’s invariance principle, for any initial condition  $x(t_0)$ , the state  $x$  will finally stay in the set  $S_b = \{e_{X_i} \mid \|e_{X_i}\| \leq \frac{2\bar{\kappa}\beta\lambda_{max}(P)}{\lambda_{min}(Q)}\}$ . The error  $e_{X_i}$  in (36) is ultimately bounded. The same result on  $Y$  can be obtained similarly.  $\square$

**Remark 4.** According to Theorem 2, the tracking error of quadrotor is proportional to the bound  $\beta$  of the observer error after finite time  $T_{convergence}$ . If the observer error asymptotically converges to zero, then the formation tracking error will also converge to zero asymptotically.

**Remark 5.** In practice, before the time instant  $T_{convergence}$ , the bounds  $b_j$  are chosen small enough in order to prevent the unstable dynamics, since the observer error probably enormous, which is generated by great initial deviation.

### 5. Simulation and Experimental Results

In this subsection, we first use numerical simulation by means of MATLAB to qualitatively illustrate that the formation controller with estimated feedforwards has better performance than that without them. By doing this, we are able to show that our proposed formation controller makes the multi-quadrotor system performing a high bandwidth, such that the aggressive formation is achieved.

Then in the second subsection, the proposed control laws are implemented on the simulator-experiment platform comparing with that using the formation controller without high-order feedforwards.

#### 5.1. Simulation Results

We reconsider the formation of four quadrotors in Figure 1 (left). The reference formation trajectory RFT is given by  $r(t) = [3 \sin(0.1t), 3 \cos(0.1t), 1, 0]^T$ . The four quadrotors should keep a desired biases w.r.t. RFT defined by  $d_{i0}$ ,  $i \in \{1, 2, 3, 4\}$ . As shown in Figure 1 (right), quadrotor 3 has two neighbors, i.e., 2 and 4. The quadrotor 3 is expected to keep inter-distances  $d_{32} = [\sqrt{2}, -\sqrt{2}]^T$  and  $d_{34} = [-\sqrt{2}, -\sqrt{2}]^T$  with respect to 2 and 4. It is easy to verify that  $d_{32} = d_{30} - d_{20}$  and  $d_{34} = d_{30} - d_{40}$ .

The tuning process of the controller gains in Table 2 is given as follows.

- **Controller gains tuning on simulator:** Let  $\dot{X}_i^d, \ddot{X}_i^d, \dddot{X}_i^d, \ddot{X}_i^d$ , and  $\dot{Y}_i^d, \ddot{Y}_i^d, \dddot{Y}_i^d, \ddot{Y}_i^d$  be null. Give some small  $X_i^d$  and  $Y_i^d$ , for example a small step signal, then, tune the gains  $k_{jX}$  and  $k_{jY}, j = \{1, 2, 3, 4\}$ . Observe the oscillation of the rotation angles during translational motion, adjust the gains  $k_{3X}, k_{4X}$  and  $k_{3Y}, k_{4Y}$  to reduce the oscillation.
- **Implement the controller gains on real quadrotor:** Test on single quadrotor, slightly tune the parameters  $k_{1X}, k_{2X}$  if necessary. Then, if the performance is satisfied, implement them on the formation tracking control.

**Table 2.** The controllers parameters.

$k_{4X}$	$k_{3X}$	$k_{2X}$	$k_{1X}$	$k_{4Y}$	$k_{3Y}$	$k_{2Y}$	$k_{1Y}$	$k_{2\psi}$	$k_{1\psi}$	$k_{2Z}$	$k_{1Z}$
1.0	2.5	1.8	1.3	1.0	2.5	1.8	1.3	17.3	10.2	3.1	2.2

The observer gain matrix is selected according to the pole assignment as follows

- **Pole assignment:** The gains of the observer are dominated by the finite time  $T_{converge}$  that we want to have. Once it is fixed, the gains of the observer can be calculated by the technique of pole assignment, such that the nearest pole of  $A_o$  to imaginary axis is placed to  $-\gamma$ .

In the following simulation and experiments, we assume that the position and velocity in the guidance vector are measurable, i.e.,

$$C_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

the observe gains are selected as follows,

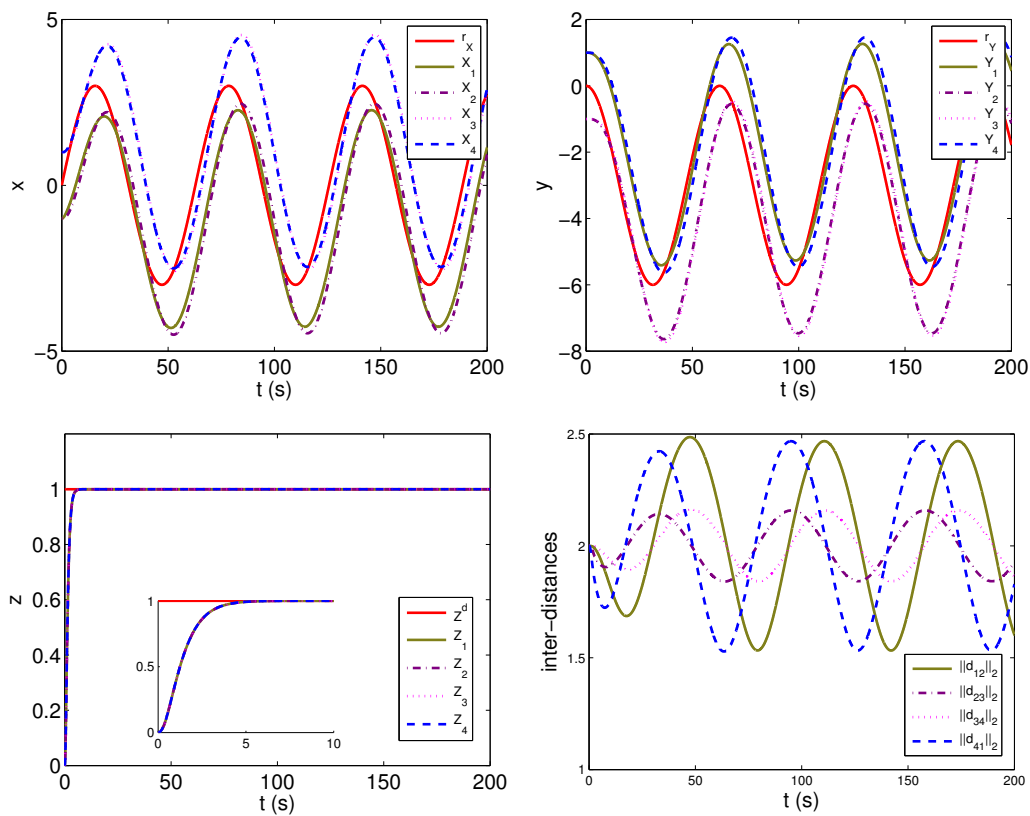
$$L_t = \begin{bmatrix} 6 & 1 & 1 & 1.02 & 0.198 \\ 1.804 & 9.02 & 14.432 & 6.675 & 0.934 \end{bmatrix}^T \tag{38}$$

Some constant bounds of the saturation functions are selected as  $b_4 = 0.000035$ ,  $b_3 = 0.00035$ ,  $b_2 = 0.0035$ , and  $b_1 = \infty$ . The saturation functions are introduced to regulate the large initial deviation of the observer. There bounds are tuned progressively until a satisfactory performance is achieved.

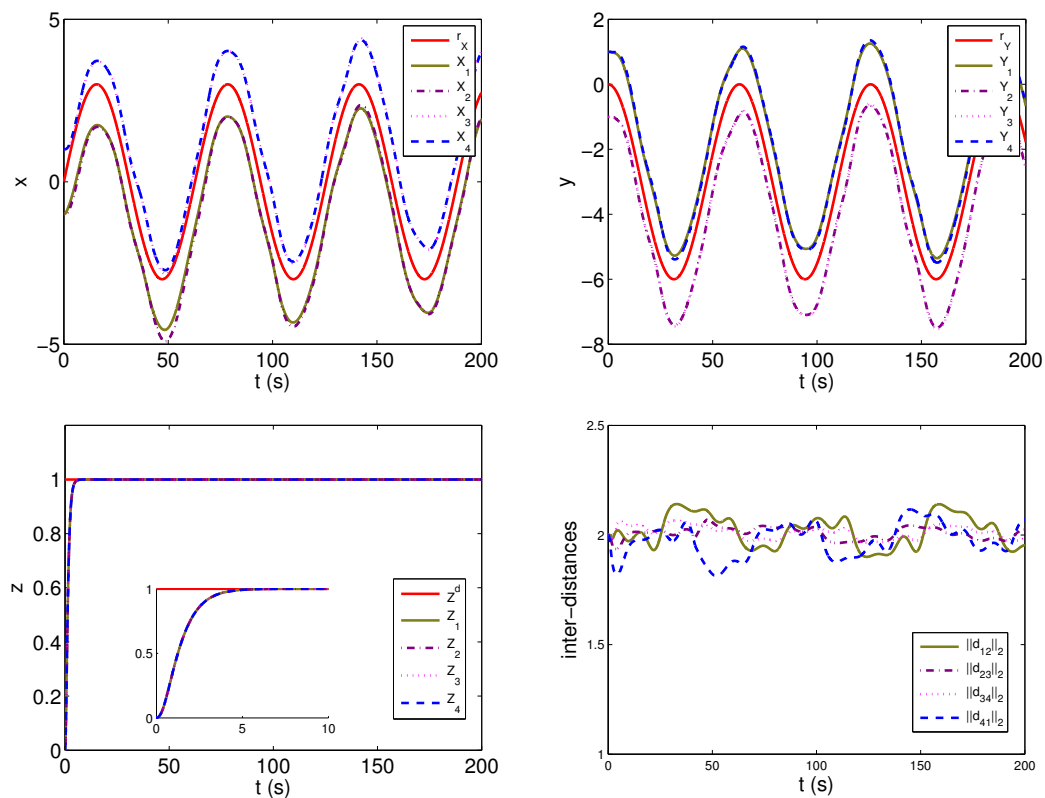
We respectively show in Figures 2 and 3 the simulation results (i) without estimation of high-order derivatives and (ii) with estimation of high-order derivatives. We observe from Figure 2 that the quadrotors track the RFT with delays and the inter-distances are not well maintained. On the contrary, in Figure 3, the delays are small and the quadrotors keep around the desired inter-distances.

In the following subsection, we will implement the formation control protocol on the real-time simulator-experiment platform.





**Figure 2.** Rigid formation of 4 quadrotors without high-order derivatives estimation, the objective is to track the RFT  $r(t)$  and meanwhile keep inter-distance between quadrotors.



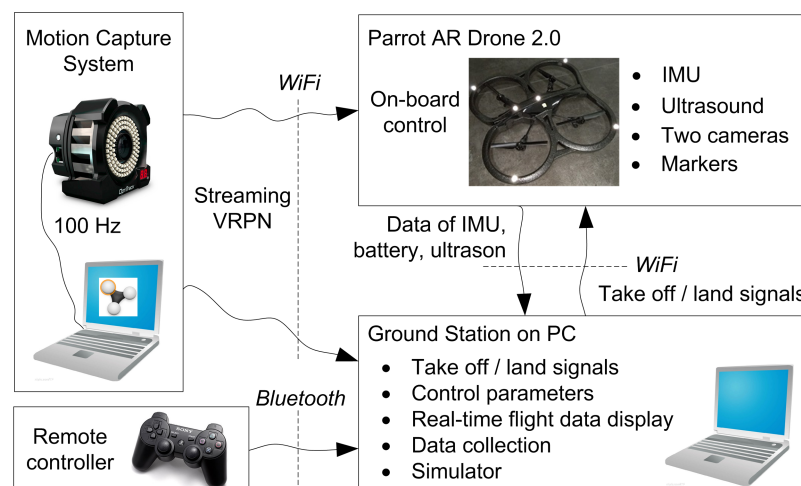
**Figure 3.** Rigid formation of 4 quadrotors with high-order derivatives estimation, the objective is to track the RFT  $r(t)$  and meanwhile keep inter-distance between quadrotors.

## 5.2. Experimental Setup

*Heudiasyc laboratory* (In the Université de Technologie de Compiègne) has developed a PC-based simulator-experiment framework (This framework is primarily designed and developed by engineer Guillaume Sanahuja under the help of other colleagues and PhD students in Heudiasyc laboratory) for controlling quadrotors in real-time. The programs (written in C++) running in the quadrotors are the same, both in the simulator and in the embedded processors of real quadrotors. The goal of the framework is to firstly validate a program on the simulator before implementing on real quadrotors, in order to avoid destroying the real drones. It is important to note that within this framework, the control algorithms are implemented on the quadrotors rather than on a PC. There does not exist a central controller that sends control signals to the quadrotors.

In the simulator, the complete quadrotor dynamics are used. The flight of the quadrotors are animated in a virtual 3D environment, which permits us to observe the behavior of the quadrotors under some formation control laws. This framework permits the simulation to reflect better the real-time experiment. Both the simulator and the real-time experiment share the ground station interface on the PC, which is responsible for displaying and sending instructions such as taking off and landing.

The experimental setup is shown in Figure 4. In the experiments, the motion capture system Optitrack is used to localize the quadrotors in the formation. The ArDrone2 quadrotors manufactured by *Parrot* are used for real-time experiments.



**Figure 4.** The experimental setup of the real-time experiments.

The procedure of using the simulator-experiment framework is stated as follows:

- Program the algorithms by using C++.
- Compile the program into executable files for the quadrotors in the simulator and for the real quadrotors.
- Test the algorithm in the simulator, adjust the parameters of the controller.
- Send the executable file to each quadrotor.
- Carry out the real-time experiment.

**Remark 6.** Although the system Optitrack is a global localization system, we just use it to obtain the coordinates or inter-distances of the quadrotors. For instance, a quadrotor can use its on-board cameras to detect other quadrotors and calculate their inter-distances. Additionally, the self-localization can be realized by using GPS or other sensors (laser, etc.). The system Optitrack is used here to simplify the study.

### 5.3. Real-Time Experiments on Simulator-Experiment Platform

In these experiments are the formation task of four quadrotors, where quadrotor 1 is a leader. The objective is to track the trajectory  $r(t)$  with constant biases  $d_{10} = [0, \sqrt{2}]^T$ ,  $d_{20} = [-\sqrt{2}, 0]^T$ ,  $d_{30} = [0, -\sqrt{2}]^T$  and  $d_{40} = [\sqrt{2}, 0]^T$ .

Two tests are carried out on the simulator for the sake of comparison. In both tests, the objective is to track a circular trajectory (radius: 2 m; maximum linear velocity: 1.3 m/s; linear acceleration 0.1 m/s<sup>2</sup>).

In the formation control without high-order feedforwards, the gains are chosen by  $k_p = 1.5$  and  $k_d = 0.1$ . We note that these gains are not arbitrarily chosen. For the fairness of comparison, we tune the parameters on the simulator to obtain the gains with best performance.

We proposed a nonlinear-flatness-based formation controller in (12), where the high-order derivatives are estimated by using (20). In the simulator and the experiment, the control frequency is 50 Hz. Then, the states of the observer (20) are updated as follows

$$\hat{X}_i^d(k+1) = (I_5 + T \cdot A_t)\hat{X}_i^d(k) - T \cdot L_t \cdot C_t(\hat{X}_i^d(k) - X_i^d(k))$$

where  $T = \frac{1}{50} = 0.02$  s. The observer gains are given by (38).

In Figure 5, we observe from Figure 5 that even at the beginning (when  $t < 20$  s), the quadrotors are able to follow the given trajectory, after 20 s, the performance becomes not satisfactory, although the formation errors keep stable.

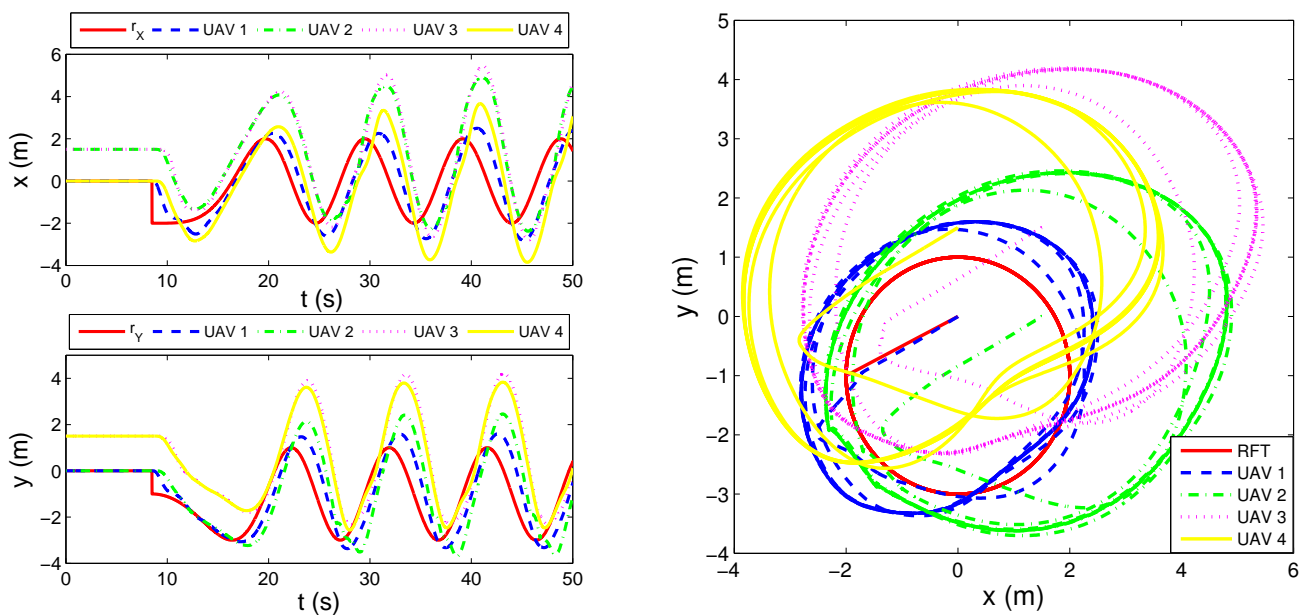


Figure 5. Formation control without high-order feedforwards.

On the contrary, by using the proposed formation control with the same controller gains, the performance is greatly improved and the quadrotors can always follow the given trajectory (see Figure 6).

The desired altitude is 1 m for all the quadrotors. The curves of the altitudes of the quadrotors are given in Figure 7. We can observe from Figure 7 (left) that the altitude of the quadrotor 2 oscillates after  $t = 25$  s. It is important to note that this phenomena is caused by the actuator saturation of the quadrotors. Considering this problem, we have proposed several bounded formation controllers such as in [13], which will not be detailed in the current paper.

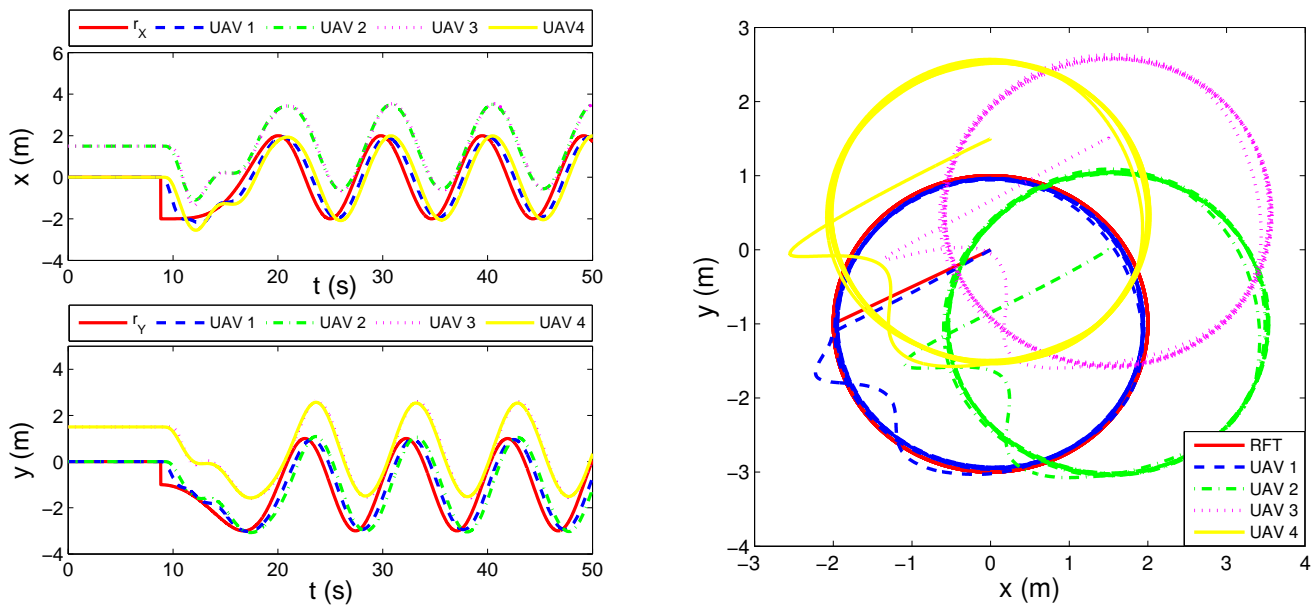


Figure 6. Our proposed formation control.

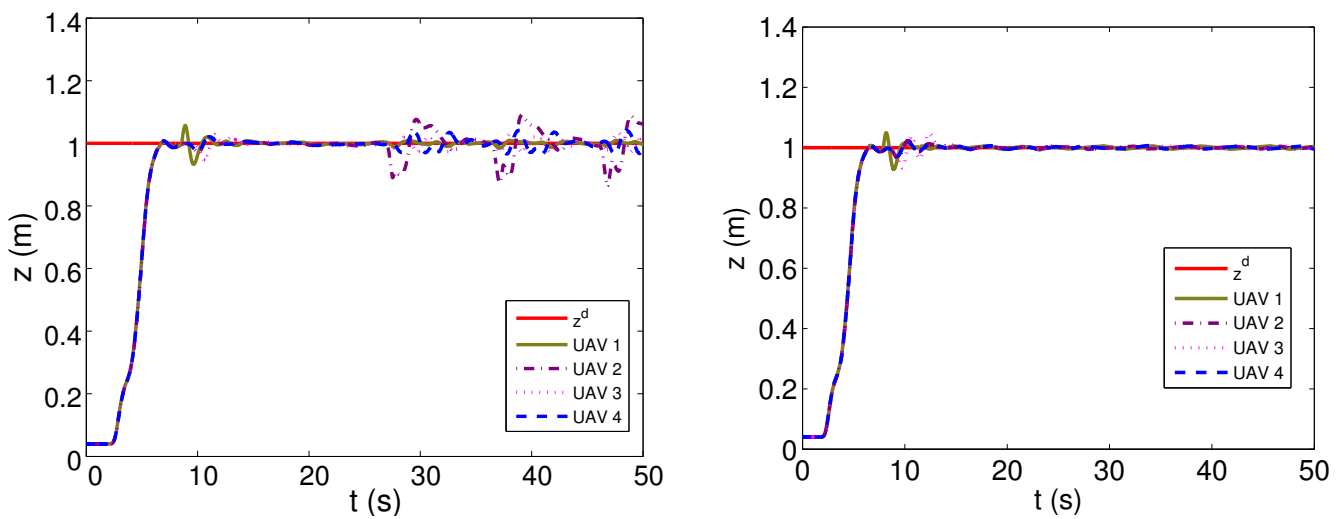
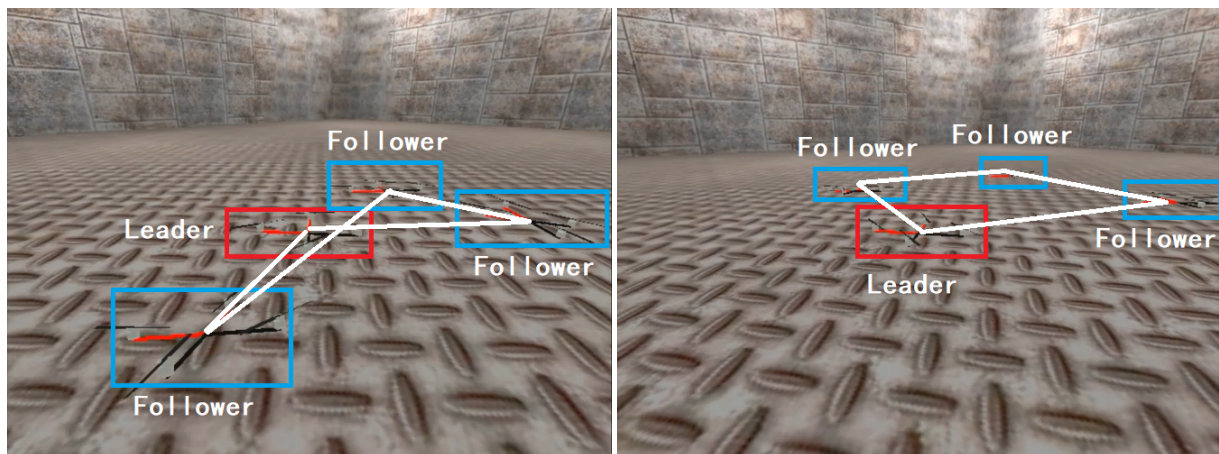


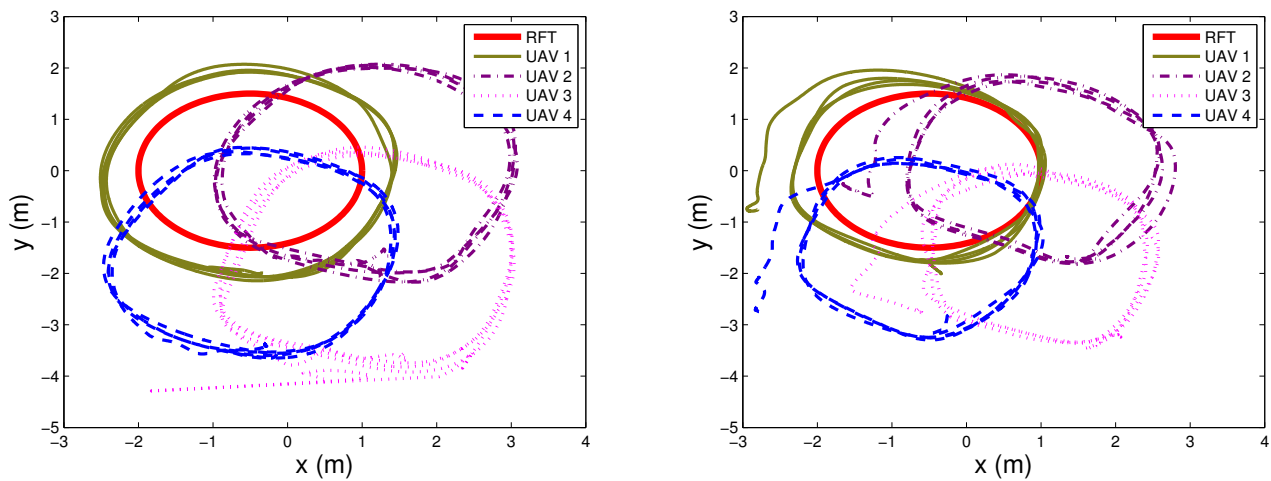
Figure 7. Altitudes of the quadrotors. **Left:** Formation control without high-order feedforwards. **Right:** Our proposed formation control.

Figure 8 shows the animation in the simulator. On the left of Figure 8, the formation control without high-order feedforwards is used, where the quadrotors fail to preserve the rectangle formation pattern after several second of flight. On the contrary, the nonlinear-flatness-based formation control can preserve the rectangle formation and track the RFT that is only given to the leader.

A real-time experiment is also given in Figure 9, where four quadrotors are used in the tests. The reference is a circular trajectory. Two tests are given for comparison. We can observe that by using our proposed controller, the formation trajectory is more precisely followed. Figure 10 shows a photo of flight in a real-time experiment.



**Figure 8.** Four quadrotors with rectangle formation pattern, tracking a circular trajectory, using formation control without high-order feedforwards (left), and the proposed formation control (right).



**Figure 9.** The real-time four quadrotors formation. Left: Formation control without high-order feedforwards; Right: our proposed method.



**Figure 10.** A snap of the real flight of four quadrotors' formation, tracking a circular trajectory while maintaining a rectangle pattern.

## 6. Conclusions

In this paper, a nonlinear-flatness-based formation control with feedforwards of estimated high-order guidance vector (neighbors' states, and RFT, if the quadrotor is a leader) derivatives is developed. The formation controller is decentralized, which is designed based on neighbors' states. In order to attain an aggressive formation, an extended finite-time observer of the guidance vector is investigated. In order to deal with the great initial deviation of the estimated states, the saturated feedforwards are developed. Then, the convergence of the formation tracking error is analyzed. Furthermore, the proposed formation control is illustrated by simulation and real-time experiment by means of MATLAB and the simulator-experiment platform, comparing with the formation control without high-order feedforwards. Both simulation and real-time experimental results show that the aggressive formation performance is improved by using our proposed formation controller.

One of the directions of the future work is to implement the proposed control with advanced nonlinear control methods, in order to deal with the effects of nonlinearities (uncertainties), to achieve asymptotic finite-time convergence, to control explicitly the formation tracking speed, etc.

**Author Contributions:** Conceptualization, methodology, and validation, Z.H.; writing—review and editing, G.Z.; visualization, W.Y.; project administration, W.W.; supervision, C.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Natural Science Foundation of Guangdong Province (grant number 2018A030310046), and China Postdoctoral Science Foundation funded project (grant number 2019M662848).

**Acknowledgments:** I would like to extend my gratitude to Isabelle Fantoni for her supervision on my thesis. I would like also thank Guillaume Sanahuja and other colleagues who are in Heudiasyc laboratory for their help on the simulator-experiment platform.

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

## Appendix A

### Appendix A.1. Proof of Proposition 3

**Proof of Proposition 3.** According to (5), we obtain that

$$F_{T_i} = \frac{(\ddot{Z}_i + g)m}{\cos \theta_i \cos \phi_i} \quad (\text{A1})$$

Replacing  $F_{T_i}$  by (A1) in the first two equations in (5), we obtain

$$\begin{aligned} \ddot{X}_i &= \left( \sin \psi_i \frac{\tan \phi_i}{\cos \theta_i} + \cos \psi_i \tan \theta_i \right) (\ddot{Z}_i + g) \\ \ddot{Y}_i &= \left( \sin \psi_i \tan \theta_i - \cos \psi_i \frac{\tan \phi_i}{\cos \theta_i} \right) (\ddot{Z}_i + g) \end{aligned} \quad (\text{A2})$$

We assume that  $\ddot{Z}_i \neq -g$ . Then, according to Equation (A2), we obtain that

$$\begin{aligned} \frac{\tan \phi_i}{\cos \theta_i} &= \frac{\ddot{X}_i \sin \psi_i - \ddot{Y}_i \cos \psi_i}{\ddot{Z}_i + g} \\ \tan \theta_i &= \frac{\ddot{X}_i \cos \psi_i + \ddot{Y}_i \sin \psi_i}{\ddot{Z}_i + g} \end{aligned}$$

Then, we have

$$\begin{aligned} \theta_i &= \arctan\left(\frac{\ddot{X}_i \cos \psi_i + \ddot{Y}_i \sin \psi_i}{\ddot{Z}_i + g}\right) \\ \phi_i &= \arctan\left(\frac{\ddot{X}_i \sin \psi_i - \ddot{Y}_i \cos \psi_i}{\ddot{Z}_i + g} \cos\left(\arctan\left(\frac{\ddot{X}_i \cos \psi_i + \ddot{Y}_i \sin \psi_i}{\ddot{Z}_i + g}\right)\right)\right) \end{aligned} \tag{A3}$$

Since  $F_i = [\mathcal{X}_i^T, \psi_i]^T$ , we denote  $F_{i1} = X_i, F_{i2} = Y_i, F_{i3} = Z_i$  and  $F_{i4} = \psi_i$ . Then, the state of the quadrotor  $\varkappa_i$  yields

$$\begin{aligned} X_i &= F_{i1}, Y_i = F_{i2}, Z_i = F_{i3}, \\ \dot{X}_i &= \dot{F}_{i1}, \dot{Y}_i = \dot{F}_{i2}, \dot{Z}_i = \dot{F}_{i3}, \\ \theta_i &= \arctan\left(\frac{\dot{F}_{i1} \cos F_{i4} + \dot{F}_{i2} \sin F_{i4}}{\dot{F}_{i3} + g}\right) \\ \phi_i &= \arctan\left(\frac{\dot{F}_{i1} \sin F_{i4} - \dot{F}_{i2} \cos F_{i4}}{\dot{F}_{i3} + g} \cos\left(\arctan\left(\frac{\dot{F}_{i1} \cos F_{i4} + \dot{F}_{i2} \sin F_{i4}}{\dot{F}_{i3} + g}\right)\right)\right) \\ \psi_i &= F_{i4} \\ \dot{\theta}_i &= \left(\arctan\left(\frac{\dot{F}_{i1} \cos F_{i4} + \dot{F}_{i2} \sin F_{i4}}{\dot{F}_{i3} + g}\right)\right)^{(1)} \\ \dot{\phi}_i &= \left(\arctan\left(\frac{\dot{F}_{i1} \sin F_{i4} - \dot{F}_{i2} \cos F_{i4}}{\dot{F}_{i3} + g} \cos\left(\arctan\left(\frac{\dot{F}_{i1} \cos F_{i4} + \dot{F}_{i2} \sin F_{i4}}{\dot{F}_{i3} + g}\right)\right)\right)\right)^{(1)} \\ \dot{\psi}_i &= \dot{F}_{i4} \end{aligned} \tag{A4}$$

Equation (A4) imply that the state  $\varkappa_i$  can be parameterized by variable  $F_i$  and its derivatives. According to (9), the output  $y_i$  is in terms of  $\mathcal{X}_i, \dot{\mathcal{X}}_i, \Theta_i$  and  $\dot{\Theta}_i$ , therefore  $y_i$  can also be parameterized by  $F_i$ . Additionally, according to (8), the control input  $u_i$  is in terms of  $Z_i, \dot{Z}_i, \Theta_i, \dot{\Theta}_i$  and  $\ddot{\Theta}_i$ . Therefore, the variable  $F_i$  is the flat output of system (8). □

#### Appendix A.2. Model Simplification

According to Equation (A2), we can obtain the following equations

$$\begin{aligned} \theta_i &= \arctan\left(\frac{\ddot{X}_i \cos \psi_i + \ddot{Y}_i \sin \psi_i}{\ddot{Z}_i + g}\right) \\ \phi_i &= \arctan\left(\frac{\ddot{X}_i \sin \psi_i - \ddot{Y}_i \cos \psi_i}{\ddot{Z}_i + g} \cos\left(\arctan\left(\frac{\ddot{X}_i \cos \psi_i + \ddot{Y}_i \sin \psi_i}{\ddot{Z}_i + g}\right)\right)\right) \end{aligned} \tag{A5}$$

Without loss of generality, we assume that the yaw angle is controlled at origin, and the altitude is stabilized at some constant value, then,

$$\theta_i = \arctan(\ddot{X}_i/g) \quad \text{and} \quad \phi_i = \arctan(-(\ddot{Y}_i/g) \cos \theta_i) \tag{A6}$$

Furthermore, considering Assumption 1, we obtain that

$$\begin{aligned} \ddot{X}_i &= \theta_i g + \Delta_{X_i} \\ \ddot{Y}_i &= -\phi_i g + \Delta_{Y_i} \end{aligned} \tag{A7}$$

where the nonlinearities  $\Delta_{X_i}$  and  $\Delta_{Y_i}$  are bounded since the absolute values of  $\theta_i$  and  $\phi_i$  are bounded within  $30^\circ$  by assumption 1. Then, we know that (A7) is Lyapunov stable, if the simplified system (16) is asymptotically stable.

## References

1. Liu, H.; Ma, T.; Lewis, F.L.; Wan, Y. Robust Formation Control for Multiple Quadrotors with Nonlinearities and Disturbances. *IEEE Trans. Cybern.* **2020**, *50*, 1362–1371. [[CrossRef](#)] [[PubMed](#)]
2. Villa, D.K.D.; Brandão, A.S.; Sarcinelli-Filho, M. Path-Following and Attitude Control of a Payload Using Multiple Quadrotors. In Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; pp. 535–540. [[CrossRef](#)]
3. Wang, Z.; Singh, S.; Pavone, M.; Schwager, M. Cooperative Object Transport in 3D with Multiple Quadrotors Using No Peer Communication. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1064–1071. [[CrossRef](#)]
4. Villa, D.K.D.; Brandão, A.S.; Sarcinelli-Filho, M. Rod-shaped payload transportation using multiple quadrotors. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 1036–1040. [[CrossRef](#)]
5. Kushleyev, A.; Kumar, V.; Mellinger, D. Towards a Swarm of Agile Micro Quadrotors. In Proceedings of the Robotics: Science and Systems, Vilamoura, Portugal, 7–12 October 2012.
6. Sreenath, K.; Kumar, V. Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots. In Proceedings of the Robotics: Science and Systems, Berlin, Germany, 24–28 June 2013.
7. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Trans. Ind. Inform.* **2013**, *9*, 427–438. [[CrossRef](#)]
8. Shao, X.; Tian, B.; Li, D. Neural Dynamic Surface Containment Control for Multiple Quadrotors. In Proceedings of the 2019 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 17–19 October 2019; pp. 373–378. [[CrossRef](#)]
9. Katt, C.; Castaneda, H. Containment Control Based on Adaptive Sliding Mode for a MAV Swarm System under perturbation. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 270–275. [[CrossRef](#)]
10. Meng, Z.; Ren, W.; You, Z. Distributed finite-time attitude containment control for multiple rigid bodies. *Automatica* **2010**, *46*, 2092–2099. [[CrossRef](#)]
11. Roldao, V.; Cunha, R.; Cabecinhas, D.; Silvestre, C.; Oliveira, P. A leader-following trajectory generator with application to quadrotor formation flight. *Robot. Auton. Syst.* **2014**, *62*, 1597–1609. [[CrossRef](#)]
12. Mercado, D.A.; Castro, R.; Lozano, R. Quadrotors flight formation control using a leader-follower approach. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 3858–3863. [[CrossRef](#)]
13. Hou, Z.; Fantoni, I. Interactive Leader Follower Consensus of Multiple Quadrotors Based on Composite Nonlinear Feedback Control. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1732–1743. [[CrossRef](#)]
14. Saber, R.; Murray, R. Flocking with obstacle avoidance: Cooperation with limited communication in mobile networks. In Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; Volume 2, pp. 2022–2028. [[CrossRef](#)]
15. Olfati-Saber, R.; Fax, J.; Murray, R. Consensus and Cooperation in Networked Multi-Agent Systems. *Proc. IEEE* **2007**, *95*, 215–233. [[CrossRef](#)]
16. Turpin, M.; Michael, N.; Kumar, V. Trajectory design and control for aggressive formation flight with quadrotors. *Auton. Robots* **2012**, *33*, 143–156. [[CrossRef](#)]
17. Zhao, D.; Krishnamurthi, J.; Gandhi, F.; Mishra, S. Differential flatness based trajectory generation for time-optimal helicopter shipboard landing. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 2243–2250. [[CrossRef](#)]
18. Son, C.Y.; Jang, D.; Seo, H.; Kim, T.; Lee, H.; Kim, H.J. Real-time Optimal Planning and Model Predictive Control of a Multi-rotor with a Suspended Load. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5665–5671. [[CrossRef](#)]
19. Zeng, J.; Kotaru, P.; Mueller, M.W.; Sreenath, K. Differential Flatness Based Path Planning with Direct Collocation on Hybrid Modes for a Quadrotor With a Cable-Suspended Payload. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3074–3081. [[CrossRef](#)]
20. Hou, Z.; Xu, J.; Zhang, G. Interaction Matrix Based Analysis and Asymptotic Cooperative Control of Multi-agent Systems. *Int. J. Control Autom. Syst.* **2020**, *18*, 1103–1115. [[CrossRef](#)]
21. Irofti, D. An anticipatory protocol to reach fast consensus in multi-agent systems. *Automatica* **2020**, *113*, 108776. [[CrossRef](#)]
22. Liu, H.; Tian, Y.; Lewis, F.L.; Wan, Y.; Valavanis, K.P. Robust formation tracking control for multiple quadrotors under aggressive maneuvers. *Automatica* **2019**, *105*, 179–185. [[CrossRef](#)]
23. Zhao, B.; Wang, D.; Shi, G.; Liu, D.; Li, Y. Decentralized Control for Large-Scale Nonlinear Systems with Unknown Mismatched Interconnections via Policy Iteration. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1725–1735. [[CrossRef](#)]
24. Bhat, S.P.; Bernstein, D.S. Finite-Time Stability of Continuous Autonomous Systems. *SIAM J. Control Optim.* **2000**, *38*, 751–766. [[CrossRef](#)]