



**HAL**  
open science

## Distributed leader-follower formation control for multiple quadrotors with weighted topology

Zhicheng Hou, Isabelle Fantoni

► **To cite this version:**

Zhicheng Hou, Isabelle Fantoni. Distributed leader-follower formation control for multiple quadrotors with weighted topology. 10th IEEE System of Systems Engineering Conference (SoSE 2015), May 2015, San Antonio, TX, United States. pp.256 - 261, 10.1109/SYSOSE.2015.7151924 . hal-01180491

**HAL Id: hal-01180491**

**<https://hal.science/hal-01180491>**

Submitted on 27 Jul 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Distributed Leader-Follower Formation Control for Multiple Quadrotors with Weighted Topology

Zhicheng HOU and Isabelle FANTONI

**Abstract**—This paper addresses the problem of controlling a leader-follower formation of quadrotors (UAVs), which can be considered as a System of Systems. A distributed control scheme for the motion of the formation is proposed, ensuring consensus of the UAVs and collision avoidance. Each UAV has local and limited neighbors and uses weighted relative positions and velocities of its neighbors. In the simulation section, a comparison of using the formation controllers with weighted and unweighted topology is given. The results show that the proposed control strategy can keep the formation with some initial conditions, unlike the strategy with unweighted topology. The simulations also show that our proposed control strategy can be applied for both one leader and multiple leaders formation.

## I. INTRODUCTION

The Systems of Systems (SoSs) are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal [1]. A robotic multi-agent system can be considered as an application of Systems of Systems (SoS). Our objective is to control a formation of robotic flying robots (quadrotors) as a multi-agent system for object searching in a huge area. For multi-agent systems, consensus is one of the most studied problems, which means that all agents need to reach an agreement on certain quantities of interest (such as positions and velocities) under some control protocols [2]. The cooperative control of leader-follower (L-F) multi-agent systems has recently attracted the attention. Within the field of mobile robotics, L-F formations arise in applications ranging from searching, surveillance, inspection, and exploration [3]. According to [2] and [4], we can rephrase the L-F consensus problem as follows: for L-F multi-agent systems, all the followers follow the leader (leaders) and all of the agents can track a reference signal (e.g. desired moving trajectory), which is given to the leader (leaders). Additionally, all the agents maintain some inter-distances to avoid collisions. Such a L-F consensus has practical significance. For example, in the task of searching [5], it is more efficient to find the object in a large area by using multiple robots with L-F formation than using a single robot. The followers can be considered as the extended “eyes” of the leader.

Zhicheng HOU and Isabelle FANTONI are with Sorbonne universités, Université de technologie de Compiègne, CNRS, UMR 7253 Heudiasyc, 60200 Compiègne, France. zhicheng.hou/isabelle.fantoni@hds.utc.fr

This work was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program “Investments for the future” managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02) and the French project SEARCH funded by the Picardie Region Council.

The L-F approach has the advantage of simplicity, since the moving trajectory of the flock is clearly given to the leader (or leaders) [6]. Then, the followers follow the leader (or leaders) to keep the formation. In the “behavior-based” approach without leader, the agent in the flock usually has random behaviors to overcome local maxima or minima [7]. Although the behavior-based approach is usually considered as an approach more robust than L-F approach, the random behavior could possibly have blindness of some searching areas. Therefore, the L-F approach is efficient and simple for object searching in a huge area.

The L-F formation problem is treated in a large amount of papers, but in most of them, for example papers [4][8], the leader is treated as a special agent whose motion is independent of all the other agents. The formation of multiple UAVs (quadrotors for example) is considered by several papers. In [9], the generation of the formation trajectory is developed. In [10], each quadrotor can obtain the error of position of the group from the prescribed trajectory. In our work, A UAV, although the leader, senses its neighbors instead of all the UAVs in the flock. Each leader has interactions with neighboring UAVs (leaders or followers).

Motivated by this searching application, we propose a distributed control law for the L-F multiple UAVs system, using weighted relative positions and velocities to its neighbors, such that all the UAVs achieve L-F consensus and maintain some formations without collision. In the sequel, the relative position and velocity vectors between a UAV and its neighbors will be called RPVVs. The formation controller of each UAV is designed by using RPVVs and their corresponding weights. The weights are added to improve the robustness of the formation. This argument is illustrated by the simulation results, which are given by the simulator designed by our laboratory Heudiasyc. The simulator will be introduced in section IV. In general, a distributed control means that algorithm runs in each UAV instead of in a ground station [11]. In our proposed formation control strategy, the control algorithm runs in each UAV. Additionally, each UAV uses relative positions or velocities of its nearest neighbors. Therefore, our formation control strategy is distributed according to [11][12].

The outline of this paper is as follows. The mathematical modeling of the multi-UAV system is given in section II. The distributed control with weighted RPVVs is proposed in section III. A comparison of using weighted and unweighted RPVV is given in the simulation section IV. Finally, some concluding remarks and future works are stated in section V.

## II. MULTI-UAV SYSTEM MODELING

### A. System structure

A multi-UAV system with  $n$  autonomous UAVs is shown in Fig.1. In this system, each UAV has a distributed formation controller. To calculate the weight of the RPVV, each UAV is identified by a Priority Coefficient (abbreviated by ‘PrC’, which will be detailed in section III). The input for each UAV is the received data, which contain its neighbors’ positions, velocities and PrCs, and its own position and velocity. The output is the position, velocity and the PrC of itself. The data communication is carried out by WIFI.

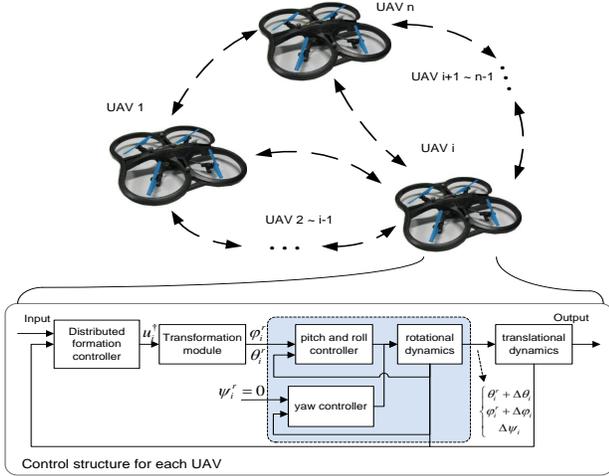


Fig. 1. Distributed control for the L-F formation. Since the algorithm is distributed, UAVs do not necessarily know all the others. The lines of dashes just show the possible communications. The objective of this paper is to design the ‘‘Distributed formation controller’’ module and the ‘‘Transformation module’’.

The dynamics of quadrotor can be divided into rotational and translational dynamics as shown in Fig.1. We assume that the rotational dynamics have been stabilized, and the attitude angles ( $\phi_i$ ,  $\theta_i$ , and  $\psi_i$ ) are able to track some smooth references  $\phi_i^r$ ,  $\theta_i^r$ , and  $\psi_i^r$ . Therefore, in this paper, we are not concerned by the tracking problem of the rotational dynamics, which is well studied in a large number of papers for example [15]. We assume that the reference yaw angle  $\psi_i^r = 0$  and that

$$\theta_i = \theta_i^r + \Delta\theta_i, \quad \phi_i = \phi_i^r + \Delta\phi_i, \quad \text{and} \quad \psi_i = \Delta\psi_i$$

where  $\Delta\theta_i$ ,  $\Delta\phi_i$  and  $\Delta\psi_i$  are the tracking errors of the angles, which are bounded and approximately equal to zero. In the sequel, we assume that the tracking errors are zero.

### B. Multiple UAVs dynamics

In our scenario, we are only interested in the dynamics of translation for attaining some formations of multiple UAVs. We assume that the altitudes of UAVs are stabilized and keep constant such that  $\ddot{Z}_i = 0$  and  $\dot{Z}_i = 0$ , so we are only concerned by the planar dynamics. We assume the UAVs have the same masses  $m$ , then, according to [13], the translational dynamics of UAV  $i$ ,  $i \in \mathcal{V}$  w.r.t a global frame (note that the yaw angle is zero) can be rewritten as follows

$$\begin{aligned} \ddot{X}_i &= g \tan \theta_i^r \\ \ddot{Y}_i &= -(g / \cos \theta_i^r) \cdot \tan \phi_i^r \end{aligned} \quad (1)$$

We refer to  $X_i$ ,  $Y_i$  as the position states in the global frame,  $\dot{X}_i$ ,  $\dot{Y}_i$  as the velocity states.

For a quadrotor, the translational movement is produced by using pitch and roll angles. Therefore, we have to find proper pitch and roll angles for realizing a formation. The control output  $u_i^r = [u_i^X, u_i^Y]^T$  for UAV  $i$  are related to the reference pitch and roll angles by  $u_i^X = g \tan \theta_i^r$  and  $u_i^Y = -g \frac{\tan \phi_i^r}{\cos \theta_i^r}$  such that the transformation module in Fig.1 satisfies the following two functions

$$\begin{aligned} \theta_i^r &= \arctan\left(\frac{u_i^X}{g}\right) \\ \phi_i^r &= \arctan\left(\frac{-u_i^Y \cos(\arctan(u_i^X/g))}{g}\right) \end{aligned} \quad (2)$$

We replace  $\theta_i^r$  and  $\phi_i^r$  in (1) by equation (2), then, we obtain the planar dynamics in the following decoupled form:  $\ddot{X}_i = u_i^X$  and  $\ddot{Y}_i = u_i^Y$ .

Since the dynamics along axes  $X$  and  $Y$  are decoupled, we only consider the dynamics along one axis ( $X$  for example) in the following analysis. The result on the other axis can be obtained with the same analysis.

The vector  $x_i = [X_i, \dot{X}_i]^T$  consists of the states of the decoupled dynamics on  $X$ -axis. Such a simplified model of dynamics can be represented by the following two cascade integrators equation

$$\dot{x}_i = Ax_i + Bu_i \quad (3)$$

where we abbreviate  $u_i^X$  by  $u_i$ . This abbreviation will be used throughout the rest of the paper. Matrices  $A$  and  $B$  satisfy

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

To design the controller for such a system, we represent the system in discrete-time. We denote by  $T$  the sample period. Then, we can obtain an approximate discrete-time model of (3) as follows

$$x_i((k+1)T) = A'x_i(kT) + B'u_i(kT)$$

where we suppose that the sample time  $T$  is sufficiently small such that  $A' \approx I_2 + AT$ ,  $B' \approx BT$ . Notation  $I_2$  represents the identity.

Now we are ready to present the model of cooperation of multiple UAVs. First, we define a full state as  $x = [x_1, \dots, x_n]^T$  and a full input vector as  $u = [u_1, \dots, u_n]^T$ , where  $n$  is the number of UAVs. We denote a constant matrix  $I_N \in \mathbb{R}^{N \times N}$  as a identity matrix of size  $N \times N$ . Then, the discrete-time overall model for the flock of UAVs can be represented as follows

$$x(k+1) = A'x(k) + B'u(k) \quad (4)$$

where  $A' = I_n \otimes A'$  and  $B' = I_n \otimes B'$ . The symbol ‘‘ $\otimes$ ’’ represents the Kronecker product. We omit the sampling time ‘‘ $T$ ’’ here and in the sequel as long as it will not cause any ambiguity. The design of the distributed formation controller ‘‘ $u$ ’’ will be given in the following section.

## III. DISTRIBUTED FORMATION CONTROLLER DESIGN

We denote the set  $\mathcal{V} = \{1, 2, \dots, n\}$  by the indices of UAV  $1, 2, \dots, n$ . The neighbor set is represented by  $\mathcal{N}_i = \{j \in \mathcal{V} : \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \leq d\}$ , where  $d$  is a positive scalar. The definition of neighbor set indicates that two UAVs are

neighbors, if their inter distance is smaller than  $d$ . (shown in Fig.3). The interaction topology of agents are usually represented by a graph [2][4][6]. Before designing the distributed formation controller, we calculate the weighted Relative Position and Velocity Vectors (RPVVs) in subsection A.

#### A. Weighted RPVVs

We denote a matrix  $\mathbf{1}_M \in \mathbb{R}^{1 \times M}$  of size  $1 \times M$  with all the entries equivalent to 1. The UAV  $i$  (leader or follower) makes a weighted measurement, which can be represented by the following weighted RPVV

$$y_i = \omega_{ii}^l(x_i - r(t)) + \sum_{j \in \mathcal{N}_i} \omega_{ij}^f(x_i - x_j) \quad (5)$$

where “ $x_i - x_j$ ” represents the RPVV of UAV  $i$  and  $j$ , “ $x_i - r(t)$ ” represents the RPVV of UAV  $i$  and the reference trajectory  $r(t)$ . We assume that the reference signal  $r(t) = [r_X(t), \dot{r}_X(t)]^T$  given to the leader (or leaders) is slowly changing such that we have  $\ddot{r}_X \approx 0$  and  $\ddot{r}_Y \approx 0$ . Scalars  $\omega_{ij}^f$  and  $\omega_{ii}^l$  are some weights. The weights  $\omega_{ii}^l = 1$  (or  $\omega_{ii}^l = 0$ ), if UAV  $i \in \mathcal{V}$  is a leader (or follower). The weights  $\omega_{ij}^f$  embody which neighbor  $j \in \mathcal{N}_i$  is more “believable” for UAV  $i$ . For example, among the neighbors of UAV  $i$ , there are a leader and several followers, thus, the leader is more believable and the leader’s weight within  $\omega_{ij}^f$  is greater.

For UAV  $i \in \mathcal{V}$ , the PrC is a scalar represented by  $p_i(k)$ . The calculation of  $p_i(k)$  is given by algorithm 1. According to algorithm 1, a UAV, which has a smaller PrC, is closer to the leader. The leader’s PrC is 1. The weights  $\omega_{ij}^f$ ,  $j \in \mathcal{N}_i$  are calculated according to the PrCs ( $p_j(k)$ ,  $j \in \mathcal{N}_i$ ) of the neighbors of UAV  $i$ . The detailed calculation of the weights and the weighted RPVV can be found in algorithm 2, and detailed for an example below.

---

#### Algorithm 1 Update PrC $p_i$ for UAV $i$

---

**Input:**  
PrCs of neighbors:  $p_j(k)$ ,  $j \in \mathcal{N}_i$ .

**Output:**  
Updated PrC of UAV  $i$ :  $p_i(k+1)$ .

```

1: for  $j = 1$ ;  $j \leq n$ ;  $j++$  do
2:   if UAV  $j$  is a neighbor of  $i$  then
3:      $per[j] = p_j(k)$ 
4:   else
5:      $per[j] = n$  //Store  $p_j(k)$ ,  $j \in \mathcal{N}_i$  into vector  $per[n]$ 
6:   end if
7: end for
8: if UAV  $i$  is a leader then
9:    $p_i(k+1) = 1$ 
10: else
11:    $p_i(k+1) = \sigma_n \{ \min_{j \in \mathcal{N}_i} \{ p_j(k) \} + 1 \}$ 
12: end if
13: return  $p_i(k+1)$  //UAV  $i$  transmits  $p_i(k+1)$  to others within its neighborhood.
```

---

As shown in Fig.1, the formation control algorithm runs in every UAV instead of running in a central UAV or ground station, such that the formation control strategy is distributed. Additionally, in this work, each UAV takes its own decision depending only on neighboring UAVs’ behavior instead of all the other UAVs. We give the flow chart of the program on a UAV in Fig.2.

---

#### Algorithm 2 Calculate weighted RPVVs for UAV $i$

---

**Input:**  
Positions  $(X_j, Y_j)$ , velocities  $(\dot{X}_j, \dot{Y}_j)$  and PrCs  $p_j(k)$  where  $j \in \mathcal{N}_i$

**Output:**  
Weighted RPVV  $y_i$  // shown in equation (5)

```

1: for  $j = 1$ ;  $j \leq n$ ;  $j++$  do
2:   if UAV  $j$  is a neighbor of  $i$  then
3:      $\omega_{ij}^f(k) = \frac{\frac{1}{p_j(k)}}{\sum_{j \in \mathcal{N}_i} \frac{1}{p_j(k)}}$ 
4:   else
5:      $\omega_{ij}^f(k) = 0$ 
6:   end if
7: end for
8: if UAV  $i$  is a leader then
9:    $\omega_{ii}^l(k) = 1$ 
10: else
11:    $\omega_{ii}^l(k) = 0$ 
12: end if
13: return  $y_i = \sum_{j \in \mathcal{N}_i} \omega_{ij}^f(x_i - x_j) + \omega_{ii}^l(x_i - r(t))$ 
```

---

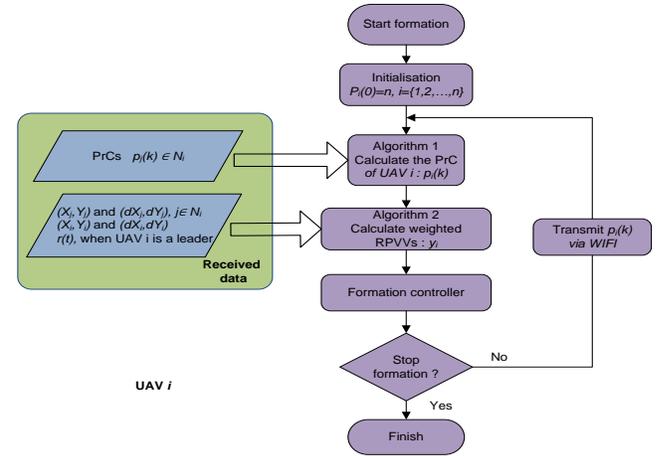


Fig. 2. The program flow chart of a UAV in the flock.

Now we give an example to explain how to calculate the weighted RPVVs by using algorithms 1 and 2.

*Example 1:* A multi-UAV system has four UAVs. There are one leader UAV 1 and three followers UAV 2, UAV 3, and UAV 4, which are shown in Fig.3.

In Fig.3.(a), the inter distance of leader UAV 1 and 2 is smaller than  $d$ , therefore, UAV 1 is a neighbor of UAV 2. Similarly, UAV 1 and UAV 3 are the neighbors of UAV 2. UAV 3 has two neighbors UAV 2 and 4. UAV 4 has only one neighbor UAV 3.

According to Fig.2.(a), all the PrCs are initialized by  $p_i(0) = 0$ ,  $i = 1, 2, 3, 4$ . According to algorithm 1, we have  $p_1(1) = 1$ ,  $p_2(1) = 4$ ,  $p_3(1) = 4$ ,  $p_4(1) = 4$ . After the second sampling time, we have  $p_1(2) = 1$ ,  $p_2(2) = 2$ ,  $p_3(2) = 4$ ,  $p_4(2) = 4$ . Then, after the third sampling time, we have  $p_1(3) = 1$ ,  $p_2(3) = 2$ ,  $p_3(3) = 3$ ,  $p_4(3) = 4$ . The PrCs will not change after the fourth sampling time except that the neighboring relationship changes or the leader changes. This paper does not deal with a formation with switching topology [14]. Nevertheless, the proposed strategy can be applied with switching topology (such as change of leader and followers).

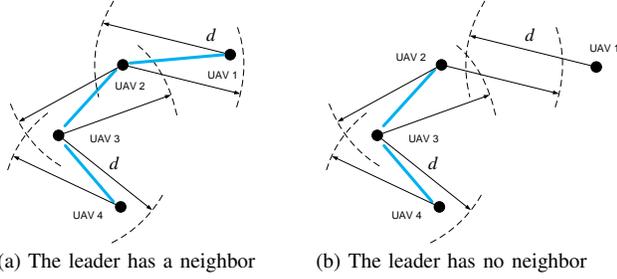


Fig. 3. Examples of possible topologies. The leader is UAV 1, and the followers are represented by UAV 2, UAV 3, and UAV 4. The limited communicating distance is represented by  $d$ .

It will be the purpose of an other future contribution. We concentrate here on the weighted topology.

We can write out the weighted RPVV for UAVs by algorithm 2:  $y_1 = x_2 + r$ ,  $y_2 = \frac{3}{4}x_1 + \frac{1}{4}x_3$ ,  $y_3 = \frac{2}{3}x_2 + \frac{1}{3}x_4$ , and  $y_4 = x_3$ .

Similarly, according to algorithm 1, we can obtain the PrCs will keep invariable after one sampling time, such that  $p_1(k) = 1$ ,  $p_2(k) = 4$ ,  $p_3(k) = 4$ ,  $p_4(k) = 4$ ,  $k = 2, 3, \dots$ . Then, we can also rewrite out the weighted RPVV as follows:  $y_1 = r$ ,  $y_2 = x_3$ ,  $y_3 = \frac{1}{2}x_2 + \frac{1}{2}x_4$ , and  $y_4 = x_3$ .

### B. Distance of security

**Definition 1:** The L-F consensus of system (4), is said to be achieved if, for each UAV  $i$ ,  $i \in \mathcal{V}$ , there is a distributed control law  $u_i(k)$ , such that the closed-loop system satisfies

$$\lim_{k \rightarrow \infty} \|x_i(k) - r(k) - d_{i0}\| = 0, \quad i = 1, \dots, n \quad (6)$$

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

The constant offset vector  $d_{i0}$  represents the desired inter distance of UAV  $i$  and the reference signal with respect to  $X$ -axis. After the L-F consensus has been achieved, we obtain that  $d_{i0} = [X_i - r_X, 0]^T$ . Similarly, for  $j \in \mathcal{N}_i$ , we also have  $d_{j0} = [X_j - r_X, 0]^T$  after L-F consensus has been achieved. If we define  $d_{i0} = [r_X - X_i, 0]^T$  and  $d_{ij} = [X_i - X_j, 0]^T$ , then, we have

$$d_{ij} = d_{i0} + d_{0j} = d_{i0} - d_{j0} \quad (7)$$

where  $d_{ij}$  is called the distance of security of UAV  $i$  and its neighbor UAV  $j$ .

We can imagine if we get rid of the term  $d_{i0}$  in definition 1 (both on  $X$ -axis and  $Y$ -axis) for all the UAVs, the inter-distance of UAVs will be equal to zero after the L-F consensus have been achieved, which practically means collisions of the UAVs.

According to definition 1, if we define an error vector as follows

$$e_i(k) = x_i(k) - r(k) - d_{i0} \quad (8)$$

then, we can write the error dynamics for UAV  $i$  as follows

$$e_i(k+1) = A'e_i(k) + B'u_i(k) \quad (9)$$

Note that  $\dot{r}_X \approx 0$  and  $\dot{r}_Y \approx 0$ , we have  $\dot{r} = Ar$ . Then,  $r(k+1) = A'r(k)$ .

For the overall system, the error dynamics is

$$e(k+1) = A'e(k) + B'u(k) \quad (10)$$

where  $u(k) = [u_1, u_2, \dots, u_n]^T$ . According to equation (8), (10) and definition 1, we can include that the L-F consensus will be achieved, if the overall error "e" converge to zero.

### C. Distributed formation control

The distributed controller design is given in this subsection to guaranty that the error "e" in equation (9) converges to zero.

We propose a distributed control law  $u_i(k)$  for UAV  $i$  (either a leader or a follower) as follows

$$u_i(k) = -K \left( y_i(k) - \sum_{j=1}^n \omega_{ij}^f d_{ij} - \omega_{ii}^l d_{i0} \right) \quad (11)$$

where  $K \in R^{1 \times 2}$  is the gain matrix. If UAV  $i$  is a leader, then,  $\omega_{ii}^l > 0$ . Otherwise,  $\omega_{ii}^l = 0$ . We can see that the proposed controller only needs the relative positions and velocities instead of the absolute positions and velocities.

Since  $\omega_{ij}^f = 0$ , when  $j \notin \mathcal{N}_i$ , we have the following result

$$\sum_{j \in \mathcal{N}_i} \omega_{ij}^f (x_i(k) - x_j(k)) = \sum_{j=1}^n \omega_{ij}^f (x_i(k) - x_j(k)) \quad (12)$$

If we replace  $y_i(k)$  in (11) by (5) and using (12), (8) and (7), we obtain that, for each instant "k"

$$\begin{aligned} u_i &= -K \left( \omega_{ii}^l (x_i - r - d_{i0}) + \sum_{j=1}^n \omega_{ij}^f (x_i - x_j - d_{ij}) \right) \\ &= -K \left( \omega_{ii}^l e_i + \sum_{j=1}^n \omega_{ij}^f (x_i - r - x_j + r - (d_{i0} + d_{0j})) \right) \\ &= -K \left( \omega_{ii}^l e_i + \sum_{j=1}^n \omega_{ij}^f (x_i - r - d_{i0} - (x_j - r - d_{j0})) \right) \\ &= -K \left( \omega_{ii}^l e_i + \sum_{j=1}^n \omega_{ij}^f (e_i - e_j) \right) \\ &= -K \left( \left( \sum_{j=1}^n \omega_{ij}^f + \omega_{ii}^l \right) e_i - \sum_{j=1}^n \omega_{ij}^f e_j \right) \end{aligned}$$

According to the foregoing equation, we can write out the overall control input u as follows

$$u = -\mathcal{K} \cdot \left( \begin{bmatrix} \sum_{j=1}^n \omega_{1j}^f + \omega_{11}^l & \dots & -\omega_{1n}^f \\ \vdots & \ddots & \vdots \\ -\omega_{n1}^f & \dots & \sum_{j=1}^n \omega_{nj}^f + \omega_{nn}^l \end{bmatrix} \otimes I_2 \right) \cdot e \quad (13)$$

where  $\mathcal{K} = I_n \otimes K$ .

We denote the matrix before the Kronecker product symbol in the foregoing equation by interaction matrix  $G$ , which is composed of the weights  $\omega_{ij}^f$  and  $\omega_{ii}^l$ . The matrix  $G$  describes the weighted topology of the multi-UAV system. We can rewrite equation (13) by

$$u(k) = -\mathcal{K} \cdot (G \otimes I_2) \cdot e(k) \quad (14)$$

Note that we use the Kronecker product to adapt the dimension of  $G$  and  $e$ .

**Lemma 1:** Matrix  $G$  is nonsingular, if (a) the flock of UAVs has at least one leader; (b) each follower connects to the leader (or leaders) directly or via other followers.

*Proof:* According to the weights updating equation on line 3 of algorithm 2, we can conclude that  $\sum_{j=1}^n \omega_{ij}^f = 1$ ,  $i \in \mathcal{V}$ .

According to the condition (a), the flock has at least one leader. Without loss of generality, we suppose that the UAV  $1 \sim m$  are leaders and UAV  $m+1 \sim n$  are followers.

For the followers  $m+1 \sim n$ , we denote the first nonzero members of each line is  $G(m+1, \alpha_{m+1})$ ,  $\dots$ ,  $G(n, \alpha_n)$ . We suppose that  $\alpha_{m+1} \leq \alpha_{m+2} \leq \dots \leq \alpha_n$ . According to condition (b), we conclude that  $\alpha_{m+1} < m+1$ ,  $\dots$ ,  $\alpha_n < n$ .

Then, the matrix  $G$  can be represented as the following general form

$$\begin{pmatrix} 1 + \omega_{11}^f & \dots & -\omega_{1m}^f & -\omega_{1(m+1)}^f & \dots & -\omega_{1n}^f \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -\omega_{m1}^f & \dots & 1 + \omega_{mm}^f & -\omega_{m(m+1)}^f & \dots & -\omega_{mn}^f \\ \hline 0 & \dots & -\omega_{(m+1)\alpha_{m+1}}^f & \dots & \dots & -\omega_{(m+1)n}^f \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & -\omega_{n\alpha_n}^f & \dots & 1 \end{pmatrix}$$

We denote the four sub-blocks in the foregoing matrix by  $G_{ul}$  (upper-left),  $G_{ur}$  (upper-right),  $G_{ll}$  (lower-left) and  $G_{lr}$  (lower-right).

We prove this lemma by contradiction. Firstly, we suppose that  $G$  is singular. Therefore, it has a zero eigenvalue associated with the eigenvector  $v = [v_u^T, v_l^T]^T$ , which satisfies

$$\begin{pmatrix} G_{ul} & G_{ur} \\ G_{ll} & G_{lr} \end{pmatrix} \begin{pmatrix} v_u \\ v_l \end{pmatrix} = 0$$

Without loss of generality, we denote the maximum absolute value of the member of the nonzero eigenvector  $v$  by  $v_{\max}$ , where ' $v_{\max}$ ' is the  $i$ -th member of  $v$ .

- If  $i \leq m$ , then  $v_{\max}$  in  $v_u$ . We have  $(1 + \omega_{ii}^f)v_{\max} = \sum_{j=1, j \neq i}^n \omega_{ij}^f v_j$ . Since  $\omega_{ii}^f > 0$  (UAV  $i \leq m$  are leaders and  $\omega_{ii}^f = 1$ ), we have  $|(1 + \omega_{ii}^f)v_{\max}| = (1 + \omega_{ii}^f)|v_{\max}| = \sum_{j=1, j \neq i}^n \omega_{ij}^f |v_j|$  and  $1 + \omega_{ii}^f \leq \sum_{j=1, j \neq i}^n \omega_{ij}^f \frac{|v_j|}{|v_{\max}|} \leq 1$ . This inequality contradicts  $1 + \omega_{ii}^f > 1$ . Therefore,  $v_{\max}$  should not be contained in  $v_u$  and the absolute value of the members of  $v_u$  should be smaller than  $v_{\max}$ .
- If  $i > m$ , we have  $|v_{\max}| = \sum_{j=1, j \neq i}^n \omega_{ij}^f |v_j|$  such that  $|v_j| = |v_{\max}|$ , for all  $j \in \mathcal{N}_i$ . Then, for example, if  $i = m + 1$ , there exists  $|v_j| = |v_{\max}|$ ,  $j < m + 1$ , which implies that a ' $v_{\max}$ ' exists in  $v_u$ . That contradicts the fact that " $v_{\max}$  should not be in  $v_u$ ". Therefore,  $|v_{\max}|$  is not the  $(m + 1)$ -th member of  $v$ . If we continue to take the example that  $i = m + 2, \dots, i = n$ , we will find that such an  $i$  exist unless there is  $G(i, \alpha_i) = 1$  ( $i = m + 1 \dots n - 1$ ) in the sub-block  $G_{lr}$ . It means that there are  $n - i + 1$  UAVs isolated, which contradicts condition (b).

Thus, if conditions (a) and (b) are satisfied, the eigenvector that renders the matrix  $G$  singular does not exist. Then, the matrix  $G$  is nonsingular. ■

In fact, according to lemma 1 and the Gershgorin circle theorem [16], we can obtain that the eigenvalues of matrix  $G$  satisfies  $0 < \lambda(G) \leq 3$ .

We replace "u" in equation (10) by equation (14), then we have  $e(k+1) = (\mathcal{A}' - \mathcal{B}'\mathcal{K} \cdot (G \otimes I_2)) \cdot e(k)$ . By using the mixed-product property of Kronecker product, we can rewrite the foregoing equation as follows  $e(k+1) = (I_n \otimes \mathcal{A}' - (G \otimes \mathcal{B}'\mathcal{K})) \cdot e(k)$ . According to matrices  $\mathcal{B}'$  and  $\mathcal{K}$ , we are able to find  $n$  elementary matrices  $S_1, \dots, S_n$ , which render  $G \otimes \mathcal{B}'\mathcal{K}$  as follows

$$(\Pi_{i=1}^n S_i) (G \otimes \mathcal{B}'\mathcal{K}) (\Pi_{i=1}^n S_i)^T = \begin{bmatrix} 0 & 0 \\ k_1 TG & k_2 TG \end{bmatrix}$$

We recall that " $T$ " is the sampling period. If we denote  $S = \Pi_{i=1}^n S_i$  and set  $\tilde{e} = Se$ , then, we have  $\tilde{e}(k+1) = \mathcal{A}_c \tilde{e}(k)$ ,

where  $\mathcal{A}_c$  satisfies

$$\mathcal{A}_c = \begin{bmatrix} I_n & TI_n \\ -k_1 TG & I_n - k_2 TG \end{bmatrix}$$

We denote vector  $[E_1^T, E_2^T]^T$  by a eigenvector of matrix  $\mathcal{A}_c$ , where  $E_1, E_2 \in R^n$ . Then, we have

$$\begin{cases} E_1 + TE_2 = \lambda E_1 \\ -k_1 TGE_1 + E_2 - k_2 TGE_2 = \lambda E_2 \end{cases}$$

where  $\lambda$  represents a eigenvalue of  $\mathcal{A}_c$ . Thus, we simplify the foregoing equation as follows

$$(-k_1 T^2 - k_2 T\lambda + k_2 T)GE_1 = (\lambda - 1)^2 E_1 \quad (15)$$

Equation (15) means that  $E_1$  is a eigenvector of matrix  $G$  with eigenvalue  $\lambda_i(G)$ . In this notation, we have  $(-k_1 T^2 - k_2 T\lambda + k_2 T)\lambda_i(G) = (\lambda - 1)^2$ . We rewrite this equation as follows

$$\lambda^2 + (k_2 T\lambda_i(G) - 2)\lambda + (k_1 T^2 - k_2 T)\lambda_i(G) + 1 = 0 \quad (16)$$

According to the definition of L-F consensus of multiple UAVs system (see definition 1), the origin of (10) should be asymptotically stable such that the eigenvalues of matrix  $\mathcal{A}_c$  should satisfy  $|\lambda| < 1$ . Therefore, the eigenvalues of matrix  $G$  should satisfy  $\lambda_i(G) \neq 0$ ,  $i = \{1, 2, \dots, n\}$ .

#### IV. SIMULATION

To illustrate the performance of the proposed distributed controller for L-F formation, we present here the results of simulation.

Heudiasyc laboratory has developed a PC-based simulator-experiment framework for controlling a quadrotor and also a flock of quadrotors. The programs (written in C++) running in the UAVs are the same, both in the simulator and in the embedded processors of real UAVs. This framework permits the simulation to reflect better the real-time experiment.

In this example of simulation, the number of UAV is  $n = 4$ , the maximum distance of sensing is  $d = 3m$ , and the distance of security is  $1.5m$ .

We present two simulations in Fig.4 with the same initial positions as follows: UAV 1, (0, 0); UAV 2, (-1.49, -2.6); UAV 3, (1.49, -2.6); and UAV 4, (0, -4). All the UAVs has zero initial velocities. The tasks of these two simulations are to track the same reference signals (shown in Fig.4). In Fig.4.(a), the controller (17), which is proposed in [6], is used for the purpose of comparison.

$$\begin{cases} u_i(k) = -K \left( \sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} (x_i - x_j - d_{ij}) \right), & \text{if } i \text{ is a follower} \\ u_i(k) = -K(x_i - r(t)), & \text{if } i \text{ is a leader} \end{cases} \quad (17)$$

We note that in (17), each agent has no special weights and they are undifferentiated. Since in controller (17), the RPVVs are used but without different weights, we call it "unweighted RPVVs" method. The motion of the leader only depends on the reference signal, as defined in [4]. The corresponding interaction matrix is represented by  $G_1$ .

In Fig.4.(b), our proposed controller in (11) is used. This controller is based on the weighted RPVVs. The weighted RPVVs introduced in section IV embodies which neighbor  $j \in \mathcal{N}_i$  is more "believable" for UAV  $i$ . A neighbor is more "believable", if it is nearer to the leader. In our strategy, the

leader has interaction with the followers. Then, according to example 1 and the structure of the interaction matrix, we obtain the  $G_2$  as follows (in fact, the interaction matrix  $G_2$  keeps constant after several times of updating).

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \quad G_2 = \begin{pmatrix} 2 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{6}{11} & 1 & -\frac{3}{11} & -\frac{2}{11} \\ -\frac{6}{11} & -\frac{3}{11} & 1 & -\frac{2}{11} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}$$

According to equation (16), the controller gain for each UAV is selected as  $k = [1.39, 1.2]$ , which ensures the spectral radius of the matrix  $\mathcal{A}_c$  smaller than 1. In this scenario, our task is to make all the UAVs achieve consensus around a destination point at (2,3) and keep some inter-distances to avoid collisions. In Fig.4, we give the comparison between the formation controllers with unweighted RPVVs and with weighted RPVVs. In Fig.4.(a) the UAVs fail to maintain the formation. The followers are outside of the leader's neighborhood a few seconds after the formation has begun. In Fig.4.(b), with the same initial positions and velocities, the followers always follow the leader. This simulation shows that the our control strategy with weighted RPVVs can better maintain the formation than unweighted RPVVs.

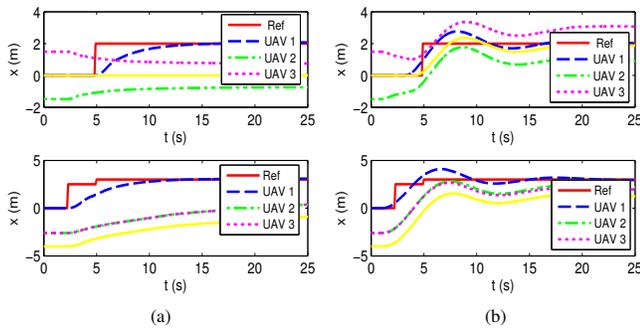


Fig. 4. Formations of four UAVs (UAV 1 is the leader and UAV 2, 3, 4 are followers). In (a), the unweighted RPVVs is used. The formation starts at  $t = 2.4s$ . After a very short time, UAV 2,3,4 are outside of the neighborhood of UAV 1. In (b), the weighted RPVVs is used. The UAVs keep connected and the followers follow the leader.

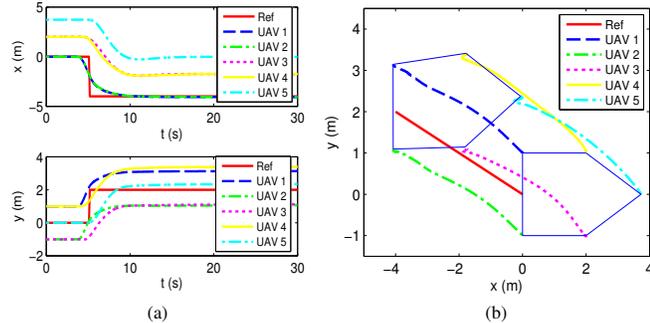


Fig. 5. Formation of 5 UAVs with 2 leaders and 3 followers. UAVs 1 and 2 are leaders. UAVs 3 ~ 5 are followers. The inter-distances of each UAV and its neighbors are 2m.

In Fig.5, a formation with 2 leaders and 3 followers is shown. The task of the flock is to move from the initial place to the point of destination. We note that in Fig.5.(a), the UAVs arrive the desired point  $(-4,2)$  after about 5 seconds. Fig.5.(b) shows that the UAVs keep some inter-distances to avoid collision.

A L-F multi-UAV system with multiple leaders can improve the robustness of the system, because if one leader does not work, the other leaders can also guide the followers. This problem will be presented in our future work.

## V. CONCLUSION AND FUTURE WORK

In this paper, a distributed formation control with weighted topology is proposed for the formation of L-F multi-UAV system, which can be considered as a System of Systems. The distributed control law of each UAV is built using weighted relative position and velocity vectors to the neighbors, which produces a weighted interaction matrix for the formation. The stability analysis of the L-F formation control has been proved. Simulations show the satisfactory performance of such a strategy. The future work is to implement our algorithm with a formation of real UAVs using their own sensors (like camera).

## VI. ACKNOWLEDGMENT

I would like to extend my gratitude to my colleague Guillaume Sanahuja who is engineer in Heudiasyc laboratory for his help on the simulator.

## REFERENCES

- [1] M. Jamshidi, "Systems of Systems Engineering: Principles and Applications," Taylor & Francis, 2008.
- [2] R. Olfati-Saber, J.A. Fax, R.M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol.95, no.1, pp.215-233, 2007.
- [3] D. Panagou, V. Kumar, "Maintaining visibility for leader-follower formations in obstacle environments," *IEEE International Conference on Robotics and Automation*, pp.1811-1816, Saint Paul, MN, 2012.
- [4] W. Ni, D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Systems & Control Letters*, Volume 59, Issues 3-4, pp.209-217, 2010.
- [5] A.K. Ray, P. Benavidez, L. Behera, M. Jamshidi, "Decentralized Motion Coordination for a Formation of Rovers," *IEEE Systems Journal*, vol.3, no.3, pp.369-381, 2009.
- [6] J.A. Fax, R.M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol.49, no.9, pp.1465-1476, 2004.
- [7] T. Balch, R.C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol.14, no.6, pp.926-939, 1998.
- [8] Hong, Y., Hu, J., Gao, L., "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol.42, iss.7, pp.1177-1182, 2006.
- [9] V. Roldao et al., "A leader-following trajectory generator with application to quadrotor formation flight," *Robotics and Autonomous Systems*, vol.62, iss.10, 2014.
- [10] A. Kushleyev, D. Mellinger, V. Kumar, "Towards A Swarm of Agile Micro Quadrotors," *Autonomous Robots*, 35(4):287-300, 2013.
- [11] B, A.S. Jaimes, M. Jamshidi, "Consensus-based and network control of UAVs," *5th International Conference on System of Systems Engineering (SoSE)*, Loughborough, 2010.
- [12] L. Bakule, "Decentralized control: An overview," *Annual Reviews in Control*, vol.32, no.1, pp: 87-98, 2008.
- [13] Z. Hou, I. Fantoni, A. Zavala-Rio, "Modeling and Decentralized Control for the Multiple UAVs Formation based on Lyapunov design and redesign," *IFAC Workshop RED-UAS*, Compiègne, 2013.
- [14] R. Olfati-Saber, R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol.49, no.9, pp.1520-1533, 2004.
- [15] F. Kendoul et al., "Real-time nonlinear embedded control for an autonomous quadrotor helicopter," *AIAA, Journal of Guidance, Control, and Dynamics*, vol.30, no.4, pp.1049-1061, July-August, 2007.
- [16] R.S. Varga, "Gersgorin and His Circles," New York: Springer, 2004.