Robotica: (2021) volume 39, pp. 2061–2078. © The Author(s), 2021. Published by Cambridge University Press.
doi:10.1017/S0263574721000126

Fast Convergence of Multi-quadrotor Cooperation Using Weighted-Neighbor-based Control

Zhicheng Hou†‡* and Gong Zhang†

†*Guangzhou Institute of Advanced Technology, Chinese Academy of Sciences, Beijing, China*
E-mail: gong.zhang@giat.ac.cn
‡*Sorbonne universités, Université de technologie de Compiègne, CNRS, UMR 7253 Heudiasyc, 60200 Compiègne, France*

(Accepted January 17, 2021. First published online: February 26, 2021)

SUMMARY
In this paper, a weighted-neighbor-based cooperation control of multi-quadrotor systems is investigated. A formation tracking problem is treated, where the reference formation trajectory (RFT) is not given *a priori*. The RFT is only available to some of the quadrotors (i.e. the leaders). In order to attain the fast convergence of the agents, we propose an algorithm to calculate the neighbors' weights in decentralized way. Then, the weights are used to compose the formation controller. Compared to the widely used average-neighbor-based control method, the proposed control protocol can increase the convergence speed of the cooperation error. Since the formation control is improved in topological scale, the utilization of the proposed algorithms can be extended on any multi-robot systems. We show the improvement of the proposed control protocol by theoretical proof, simulation, and real-time experiments.

KEYWORDS: Multi-robot systems; Weighted neighbors; Quadrotors; Fast converging.

1. Introduction

The consensus of high-order nonlinear multi-agent systems has attracted attention in the area of robotics. Specifically, research on multi-quadrotor systems increases progressively in application such as large object transportation.[1–3] It is a typical formation tracking problem, in which the robots keep a formation pattern while tracking some external given reference formation trajectory (RFT).

In practice, the robot usually has limited sensing/communicating range, such that the connectivity of the topology is probably broken in case of aggressive formation tracking. Therefore, research on increasing the converging speed of the cooperative error has practical significance. In general, to improve the converging speed of a multi-robot system, we can carry out with two scales, that is, individual scale and topological scale.

Individual scale: The optimal convergence rate of multi-agent systems is investigated, such as in refs. [4, 5]. The optimal tracking control of large-scale networked system is developed by using integral reinforcement learning method.[6] The fast converging problem of the consensus of networked high-order agents is investigated in ref. [4], where the authors show the possibility of augmenting the consensus speed by designing a linear control input of the leader. The author in ref. [5] reveals that the converging speed of a discrete multi-agent system cannot be improved by enlarging memory slot of agents in special cases. In ref. [7], the distributed Kalman filter is proposed for trajectory estimation. The authors in ref. [8] propose a nonlinear feedback control method, using which the aggressive trajectory tracking is achieved. In order to achieve aggressive trajectory tracking goal, the authors in the GRASP laboratory also propose a geometric control of quadrotor dynamics [16]. The

* Corresponding author. E-mail: zc.hou@giat.ac.cn

trajectory given to the leader is prescribed by a serial of polynomials. The leader is considered as a special agent, which computes its desired trajectory and broadcasts to all follower robots a message containing the polynomial coefficients and time intervals that fully specify the trajectory.

Topological scale: In some optimal consensus speed literature,[9–11] the iteration matrix "$W = I - \alpha L$," which is built based on Laplacian, is designed to achieve arbitrary consensus speed. In these papers, the neighbors of an agent are considered to be equally weighted, that is, the average-neighbor-based method is used. It then results in symmetric adjacency matrices in the case of undirected graphs. The average-neighbor-based method is sometimes related to average neighbors in literature.[12,13] For a multi-robot system, the robots are average if they are "a priori indistinguishable.".[14] The conception of anonymity is firstly introduced and implemented on quadrotors by the GRASP laboratory.[12,13] In ref. [15], the authors also developed a decentralized method to perform mutual localization for multiple mobile robots with average relative measurement. This type of formation has advantages of scalable agents' number and robustness with respect to agents' failure. The controller is also designed using the average of neighbors' states.

We propose this work aiming at attaining aggressive formation tracking in topological scale. Different from the existing work, we investigate the weighed-neighbor-based formation tracking control to improve the tracking performance w.r.t. the average-neighbor-based method. Nevertheless, we should note that the proposed method in this paper can also be combined with the methods investigated in the individual scale.

Although the terminology "weighted Laplacian" appears in some previous papers, we do not share the same idea. In the paper,[17] the authors proposed a "weighted Laplacian matrix" which is based on the incidence matrix and a diagonal matrix containing some weights scalars on the diagonal. This approach is different from ours. Firstly, the "weighted Laplacian matrix" is always symmetric. In our case, the Laplacian can be non-symmetric. In fact, if the weights matrix is changed, the gain of the cooperative controller is changed. But in our work, we change the weight while keeping the gain invariant. In ref. [18], the authors investigate the approach to assign the node and edge weights of graphs using convex optimization to impose bounds on their Laplacian spectra. The weighted Laplacian is constructed by left multiplying a positive-definite diagonal matrix on the traditional Laplacian. They are more focus on the effects of the highest eigenvalue of the Laplacian. In ref. [19], the bipartite consensus problem is considered, where the edge weights are matrix-valued. The objective is different from that of our paper.

We investigate the accelerating formation tracking problem of multiple quadrotors with two constraints. First, the RFT is not given a priory and not predictable. Second, the leader cannot inform any other agents the information of the RFT through any ways. The follower can only obtain the instantaneous states of its neighbors. Then, the contribution of the paper is threefold. A weighted-neighbors-based control algorithm is proposed. For each agent, a protocol of updating the weights of its neighbors is designed, such that the consequent adjacency matrix can be non-symmetric, although the graph is undirected. Furthermore, an interaction matrix is developed for the analysis of stability and converging speed rather than Laplacian. In addition, an iterative algorithm is given to online calculate the weights in a distributed way.

Comparing to the average-neighbor-based algorithm, we show the improvement of converging speed of the multi-quadrotor navigation by using our proposed weighted-neighbor-based control algorithm, through theoretical proof, simulation, and also real-time experiments.

The rest of the paper is organized as follows. Some preliminaries about the quadrotor dynamics and interaction matrix are given in Section 2. The weighted-neighbor-based control is proposed in Section 3. In Section 4, we discuss the analysis of stability and convergence rate of the multi-robot system. The main results proposed in this paper are validated by simulation and experiments, which is given in Section 5. A conclusion and perspective is given in Section 6.

2. Preliminaries

2.1. Quadrotor model

A quadrotor contains two pairs of counter-rotating rotors and propellers, located at the vertices of the crossed arms, as shown in Fig. 1. When a propeller rotates, an upward thrust and a torque parallel to the plane of the rotor are generated. The thrusts of the four rotors compose a total thrust

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

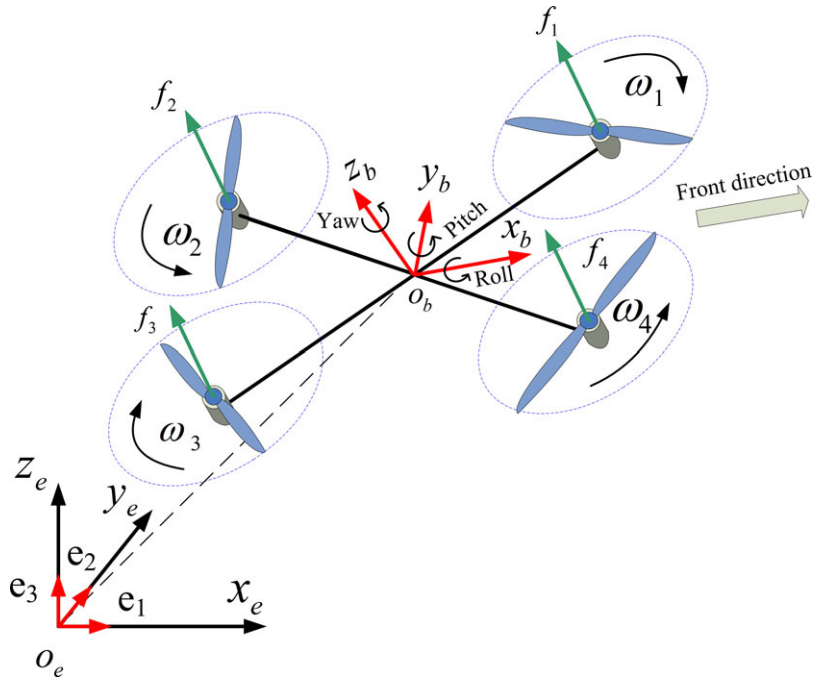

Fig. 1. Quadrotor schema. The inputs are four thrust forces generated by the four propellers. The attitude is represented by the Euler angles ϕ, θ, and ψ, giving the rotation matrix R.

$F_T = f_1 + f_2 + f_3 + f_4$. The torques of the four rotors compose a moment, which can generate the yaw movement. The different thrusts of the four rotors can generate the moments for pitch and roll movements. Then, the dynamics of a quadrotor is modeled as the motion of rigid body in 3-D space under a thrust force and three moments. As Euler angles representation is used, the state of quadrotors is represented in an inertial frame $o_e x_e y_e$ and a body-fixed frame $o_b x_b y_b$. We denote the unit directional vectors of the inertial reference frame by $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, while the unit directional vectors of the body-fixed frame by $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$.

Modeled as a rigid body, the quadrotor dynamics are represented as follows:

$$\begin{cases} m\ddot{\mathcal{X}}_i = -mg\mathbf{e}_3 + R_i F_{T_i}\mathbf{e}_3 \\ \dot{R}_i = R_i \boldsymbol{S}(\Omega_i) \\ J\dot{\Omega}_i + \boldsymbol{S}(\Omega_i)J\Omega_i = \tau_i \end{cases} \tag{1}$$

where $\mathcal{X}_i = [X_i, Y_i, Z_i]^T$ is the coordinates of the center of mass of a quadrotor in inertial frame, and F_{T_i} is the thrust force. The diagonal matrix J represents the moments of inertia in body frame. We denote the unit directional vectors of the inertial reference frame by $\mathbf{e}_1 = [1, 0, 0]^T$, $\mathbf{e}_2 = [0, 1, 0]^T$, $\mathbf{e}_3 = [0, 0, 1]^T$. The angular velocity of the quadrotor i in the body-fixed frame is represented by $\Omega_i = [\iota_i, \kappa_i, \varsigma_i]^T$. For avoiding the calculation of cross product, we introduce the function $\boldsymbol{S}(\cdot) : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$. It 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 $\boldsymbol{S}(\cdot)$ satisfies the property $\boldsymbol{S}(v_1) \cdot v_2 = v_1 \times v_2$.

Let us denote the torque vector as follows: $\tau_i = [\tau_{\phi_i}, \tau_{\theta_i}, \tau_{\psi_i}]^T \in \mathbb{R}^3$. Notations ϕ_i, θ_i, and ψ_i represent roll, pitch, and yaw angles. Then, according to (1), the translational dynamics is derived as follows:

$$\begin{aligned} \ddot{X}_i &= (\sin\psi_i \sin\phi_i + \cos\psi_i \cos\phi_i \sin\theta_i)\frac{F_{T_i}}{m} \\ \ddot{Y}_i &= (\cos\phi_i \sin\psi_i \sin\theta_i - \cos\psi_i \sin\phi_i)\frac{F_{T_i}}{m} \\ \ddot{Z}_i &= -g + (\cos\theta_i \cos\phi_i)\frac{F_{T_i}}{m} \end{aligned} \tag{2}$$

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

Let us define the rotation matrices R_x^T, R_y^T, and R_z^T with the three axes of the inertia frame, respectively. The rotation R_i from the body-fixed frame to the inertial frame is the sequence of roll-pitch-yaw (namely, $\phi - \theta - \psi$), thus $R_i = R_z(\psi_i)R_y(\theta_i)R_x(\phi_i)$. Then, we obtain the relation of body-fixed angular velocity and the derivative of the Euler angles as follows:

$$\begin{aligned}\Omega_i = \begin{bmatrix} \iota_i \\ \kappa_i \\ \varsigma_i \end{bmatrix} &= R_i^T \dot{\psi}_i \mathbf{e}_3 + \dot{\theta}_i R_x^T(\phi_i) R_y^T(\theta_i) \mathbf{e}_2 + \dot{\phi}_i R_x^T(\phi_i) \mathbf{e}_1 \\ &= \left[R_x^T(\phi_i)\mathbf{e}_1 \; R_x^T(\phi_i) R_y^T(\theta_i)\mathbf{e}_2 \; R_i^T \mathbf{e}_3 \right] \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} = T_i \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix}\end{aligned} \tag{3}$$

where $T_i = [R_x^T(\phi_i)\mathbf{e}_1 \; R_x^T(\phi_i)R_y^T(\theta_i)\mathbf{e}_2 \; R_i^T\mathbf{e}_3]$. The derivatives of (3) satisfy

$$\begin{bmatrix} \dot{\iota}_i \\ \dot{\kappa}_i \\ \dot{\varsigma}_i \end{bmatrix} = T_i \cdot \begin{bmatrix} \ddot{\phi}_i \\ \ddot{\theta}_i \\ \ddot{\psi}_i \end{bmatrix} + \bar{T}_i \cdot \begin{bmatrix} \dot{\phi}_i \dot{\theta}_i \\ \dot{\phi}_i \dot{\psi}_i \\ \dot{\theta}_i \dot{\psi}_i \end{bmatrix}$$

where $\bar{T}_i$ is as follows:

$$\bar{T}_i = \begin{bmatrix} 0 & 0 & -\cos\theta_i \\ -\sin\phi_i & \cos\theta_i \cos\phi_i & -\sin\theta_i \sin\phi_i \\ -\cos\phi_i & -\cos\theta_i \sin\phi_i & -\sin\theta_i \cos\phi_i \end{bmatrix} \tag{4}$$

Now if we denote $\tilde{T}_i = -T_i^{-1}\bar{T}_i$, the rotational dynamics yields

$$\begin{bmatrix} \ddot{\phi}_i \\ \ddot{\theta}_i \\ \ddot{\psi}_i \end{bmatrix} = (JT_i)^{-1}\tau_i + \tilde{T}_i \cdot \begin{bmatrix} \dot{\phi}_i \dot{\theta}_i \\ \dot{\phi}_i \dot{\psi}_i \\ \dot{\theta}_i \dot{\psi}_i \end{bmatrix} + (JT_i)^{-1} \boldsymbol{S}^T \left(T_i \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} \right) JT_i \cdot \begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} \tag{5}$$

Remark 1: It is obvious to see that when the roll (or pitch) angle equals to $\pi/2$, the quadrotor cannot maintain its altitude since the total thrust is perpendicular to the gravity. The gimbal lock problem is caused by kinematic singularities. As a result, we define the case of singularity as a failure of control. In other words, the pitch and roll angles should be controlled within $\pm\pi/2$.

2.2. Interaction matrix

In a multi-quadrotor system, the interaction topology of the quadrotors or agents can be represented using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the sets of vertices $\mathcal{V}$ and edges $\mathcal{E}$. The set of vertices $\mathcal{V} = \{1, 2, \ldots, n\}$ 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. A graph $\mathcal{G}$ is said to be undirected if $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$. A graph is simple if self-loops or repeated edges do not exist. We consider simple undirected graphs in this paper.

A path between two vertices i, j is a sequence of edges in a graph of the form (i, i_1), (i_1, i_2), …, (i_k, j). A graph $\mathcal{G}$ is connected if there is a path between any two vertices, otherwise it is disconnected. The adjacency matrix of $\mathcal{G}$ is denoted by $G^A = [\omega_{ij}^a] \in R^{n\times n}$, where ω_{ij}^a represents the entry on the ith row jth column of matrix G^A. Since the simple graph is considered, we have $\omega_{ii}^a = 0$. $\omega_{ij}^a > 0$ if $(i, j) \in \mathcal{E}$, otherwise $\omega_{ij}^a = 0$. The degree matrix of $\mathcal{G}$ is denoted by $G^D = \text{diag}\left\{ \sum_{j=1}^n \omega_{1j}^a, \ldots, \sum_{j=1}^n \omega_{nj}^a \right\}$.

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 $\omega_{ij}^a > 0$, then agent j is a neighbor of agent i. The number of neighbors of agent i is equal to $|\mathcal{N}_i|$.

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

We also define a role matrix $G^L = \text{diag}\{\omega_1^l, \ldots, \omega_n^l\}$ distinguishing the role of agents. If $\omega_i^l > 0$, then agent i is a leader. Otherwise, if $\omega_i^l = 0$, agent i is a follower, for $i \in \mathcal{V}$. Then, the leader set is defined as $\mathcal{V}_L = \{i \in \mathcal{V} : \omega_i^l > 0\}$. 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 $\mathcal{V} - \mathcal{V}_L$.

The interaction matrix G for L-F formation is defined as follows:

$$G = G^D - G^A + G^L \tag{6}$$

Let us note that the component $G^D - G^A$ is normally called the Laplacian in graph theory. Since we are here concerned by the L-F formation, we use the interaction matrix G to represent the interactions of agents. Obviously, if no leader exists in the group, namely leaderless formation structure, the matrix G^L will be equal to zero. In this case, the matrix G is equal to the Laplacian of the graph.

According to the properties of Laplacian, we conclude the interaction matrix G satisfies

- $G_{ij} \leq 0$, for $i, j \in \mathcal{V}$ and $i \neq j$.
- $\sum_{j=1}^{n} G_{ij} = \omega_i^l$, for $i \in \mathcal{V}$.

3. Weighted-Neighbor-based Formation Control

3.1. Dynamics decoupling

According to (2) and (5), the quadrotor is a coupled-nonlinear system. Since a quadrotor is underactuated, it is also a four-order system. The motion of translation of a quadrotor is decided by quadrotor's attitude, that is, its dynamics can be divided into rotational and translational dynamics. The quadrotor closed-loop dynamics perform therefore in two time scales, that is, the slow and fast time scales. Concerning the fast time scale, we can design a high-gain feedback control for the attitude (rotational dynamics).[8] The formation controller is designed considering only the reduced model, that is, the translational dynamics in slow time scale. The attitude angles satisfy the following equations:

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

where θ_i^d, ϕ_i^d, and ψ_i^d are desired angles, $\Delta\theta_i$, $\Delta\phi_i$, and $\Delta\psi_i$ are tracking errors of the angles, which can be assumed to be zero by designing a high-gain-based attitude controller according to singular perturbation theory, which is investigated in ref. [8]. Considering that the absolute value of the pitch and roll attitude angles in the multi-UAV system keep smaller than $\frac{\pi}{2}$, we can propose the thrust force as follows:

$$F_{T_i} = \frac{(u_i^Z + g)m}{\cos\theta_i \cos\phi_i}, \tag{7}$$

where u_i^Z is the altitude controller. According to the third line in Eq. (2), we can obtain that $\ddot{Z}_i = u_i^Z$, if F_{T_i} is selected by Eq. (7). Then the design of u_i^Z can be any kinds of controllers, such as PID. This design is also validated by real-time experiments in refs. [8] and [22]. Thus, the control of the quadrotor in z-axis is decoupled from that in x and y axes.

Let us denote $v_i = [u_i^X, u_i^Y, u_i^Z]^T$, and

$$v_i = (u_i^Z + g) \begin{bmatrix} \sin\psi_i^d & \cos\psi_i^d & 0 \\ -\cos\psi_i^d & \sin\psi_i^d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\tan\phi_i^d}{\cos\theta_i^d} - \phi_i^d \\ \tan\theta_i^d - \theta_i^d \\ \frac{1}{u_i^Z + g} u_i^Z \end{bmatrix} \tag{8}$$

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

Then, using (7) and (8), the translational model yields $\ddot{X}_i = u_i^X$, $\ddot{Y}_i = u_i^Y$, and $\ddot{Z}_i = u_i^Z$. According to the above analysis, we conclude that we can design the formation controller in x, y, and z axes separately. Thus, the controller design for the high-order multi-agent system transforms to the design of multiple second-order-agent system.

Thereafter, we only consider the dynamics on one axis (x-axis for example) in the following analysis. The controller on y and z axes can be obtained in a similar way. Let us denote $x_i = [X_i, \dot{X}_i]^T$, we can then rewrite dynamics of x_i in state space as follows: $\dot{x}_i = Ax_i + Bu_i$, 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 $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = [0, 1]^T$.

3.2. Formation measurement

In the cooperation of quadrotors, each UAV can obtain or sense the states (position and velocity) of its neighboring UAVs. It calculates relative positions and velocities $z_{ij} = x_i - x_j$, $i \in \mathcal{V}$, $j \in \mathcal{N}_i$. If a UAV is a leader, it can also obtain the relative position and velocity with respect to the navigation reference (NR) $z_{i0} = x_i - \mathcal{R}$, $i \in \mathcal{V}_L$, where we denote by $\mathcal{R} = [r, \dot{r}]^T$ the NR.

In real applications, the measurement can be realized by the following setup. Each agent can be equipped by a global positioning sensor, such as GPS in outdoor environment and UWB anchors in indoor environment; the GPS is not used when the role of the agent is a follower, for the sake of saving energy. The follower just uses other sensors to detect relative distance and orientation to its neighbors, for example, the ultra-sonic sensors installed on the robot.

3.2.1. Weighted-neighbor-based measurement. In general, the formation measurement can be written as follows:

$$z_i = \sum_{j=1}^{n} \omega_{ij}^a z_{ij} + \omega_i^l z_{i0} \tag{9}$$

where ω_{ij}^a and ω_i^l are some weights. The weight $\omega_{ij}^a \geq 0$ represents one UAV's knowledge of other UAVs. Specifically, if UAV j is a neighbor of i (e.g. in terms of sensing range), then ω_{ij}^a is nonzero. The weight $\omega_i^l \geq 0$ represents the ability of acquiring the RFT (by communication or vision). When ω_i^l is nonzero, then $i \in \mathcal{V}_L$.

The average-neighbor-based measurement is a special case of the weighted-neighbor-based measurement, where $\omega_{ij}^a = \omega_{ji}^a > 0$, when $i \in \mathcal{N}_i$. In the mostly used case, where $\{\omega_{ij}^a = \omega_{ji}^a = 1 | i \in \mathcal{N}_i\}$, and $\{i \in \mathcal{V}_L | \omega_i^l = 1\}$, the average-neighbor-based measurement can be rewritten as $z_i = \sum_{j \in \mathcal{N}_i} z_{ij} + \omega_i^l z_{i0}$. If a normalized Laplacian is taken into account,[21] the foregoing measurement becomes $z_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} z_{ij} + \omega_i^l z_{i0}$, where $\{\omega_{ij}^a = \frac{1}{|\mathcal{N}_i|} | j \in \mathcal{N}_i\}$.

In this work, we consider a more general case, where the neighbors' weights of an agent can be different, that is, $\omega_{ij}^a \neq \omega_{ji}^a$, $j \in \mathcal{N}_i$. Indeed, the condition $\{\omega_{ij}^a = \omega_{ji}^a | i \in \mathcal{N}_j\}$ can be removed, even in an undirected graph.

3.2.2. Overall error measurement. Let us define three vectors $\mathrm{g}_i^a = [\omega_{i1}^a, \ldots, \omega_{i(i-1)}^a, 0, \omega_{i(i+1)}^a \ldots, \omega_{in}^a]^T$, $\mathrm{g}_i^d = [0, \ldots, 0, \sum_{j=1}^n \omega_{ij}^a, 0, \ldots, 0]^T$, and $\mathrm{g}_i^l = [0, \ldots, 0, \omega_i^l, 0, \ldots, 0]^T$. The vectors g_i^d and g_i^l have zero entries except the i-th entry. The full state vector of n quadrotors is defined by $\mathrm{x} = \left[x_1^T, x_2^T, \ldots, x_n^T\right]^T$, then (9) can be rewritten as follows:

$$z_i = \left(\left(\mathrm{g}_i^d - \mathrm{g}_i^a + \mathrm{g}_i^l\right)^T \otimes I_2\right) \cdot \mathrm{x} - \left(\left(\mathrm{g}_i^l\right)^T \otimes I_2\right) \cdot (\mathbf{1}_n \otimes \mathcal{R}) \tag{10}$$

where $I_2 \in \mathbb{R}^{2\times 2}$ represents the identity matrix and $\mathbf{1}_n \in \mathbb{R}^n$ represents a vector with all the entries equivalent to 1. The symbol "$\otimes$" represents the Kronecker product.

According to the definitions of the members of matrices G^A and G^L in Section 2, we conclude that $G^A = \left[g_1^a, \ldots, g_n^a\right]^T$, $G^D = [g_1^d, \ldots, g_n^d]^T$ and $G^L = \left[g_1^l, \ldots, g_n^l\right]^T$.

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

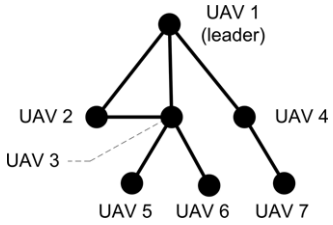

Fig. 2. The interaction topology of multiple quadrotors. The black nodes are vertices in the graph, which represents the interacting relations of UAVs.

The weighted error measurement of UAV i (either a leader or a follower) can be rewritten in (11)

$$z_i = \omega_i^l(x_i - r(t)) + \sum_{j\in\mathcal{N}_i} \omega_{ij}^a(x_i - x_j) \tag{11}$$

We assume that the RFT $\mathcal{R}$ given to the leader(s) is slowly changing such that we have $\ddot{r} = 0$.

3.3. Weights assignment

In order to introduce the assignments of the weights ω_{ij}^a, we firstly introduce the notion of "distance of vertices" (DOV) in graph theory. The distance between two vertices is the length of a shortest path between them. For example, in Fig. 2, a topology of seven vertices is shown. Between vertices 1 and 5, the distance is equal to 2, because the shortest path is $1-3-5$. The distance of a vertex to itself is zero.

Let us denote by "$\mathbf{d}_i$" the DOV of an UAV to a leader UAV. Then, the weights ω_{ij}^a in (11) are given according to $\mathbf{d}_j$. The smaller the distance $\mathbf{d}_j$ is, the bigger the weight is assigned. Obviously, $\mathbf{d}_j$ is a non-negative integer for each vertex in a graph.

When an agent (quadrotor) has multiple neighbors, the importance of them is differentiated. Intuitively, the neighbor who has smaller DOV to the leader should have higher weights. In the example of Fig. 2, vertex 3 (which stands for quadrotor 3) has four neighbors, that is, $\mathcal{N}_3 = \{1, 2, 5, 6\}$. Among them, vertex 1's DOV is zero, since it is the leader. Therefore, its weight should be the highest in $\mathcal{N}_3$. Similarly, according to the calculation of the DOVs of the $\mathcal{N}_3$, we conclude that the design of the weighted-neighbor-based control should satisfy $\omega_{31}^a > \omega_{32}^a > \omega_{35}^a = \omega_{36}^a$.

In order to calculate the weights ω_{ij}^a, we first define a Priority Coefficient (abbreviated by "PrC") for each UAV, which is a positive scalar represented by p_i. The PrC is given by $p_i = \mathbf{d}_i + 1$. The leader's PrC is 1. The PrCs of the followers that are the neighbors of the leader are equal to 2, etc.

The PrCs are calculated online and on-board quadrotor processor. We use $p_i(k)$ to represent the PrC of UAV i at the k-th sampling instant. 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 weights ω_{ij}^a, $j \in \mathcal{N}_i$ are calculated according to the PrCs $(p_j(k), j \in \mathcal{N}_i)$ of the neighbors of UAV i.

In this paper, we principally investigate the assignment of ω_{ij}^a. We simply give the assignment weights $\omega_i^l \in \{1, 0\}$ to specify the role of a quadrotor, which is either a leader or a follower.

The formation control algorithm runs in every UAV instead of running in a central UAV or ground station. Furthermore, each UAV takes its own decision depending only on neighboring UAVs' positions and velocities (and the RFT, if it is a leader) instead of all the other UAVs. In addition, the PrCs are diffused by using communication (e.g. through WiFi); therefore, the formation control strategy is distributed.

An example of the calculation of weighted-neighbor-based error measurement (defined in (11)) by using Algorithms 1 and 2 is given in ref. [20].

3.4. Formation control

Let us denote by $d_{ij} = [d_{ij}^X, \dot{d}_{ij}^X]^T$ the desired spatial distance of UAV i and j (if they are neighbors) on x_e-axis. To introduce the proposed weighted formation control strategy, we suppose that the rigid formation is taken into account, such that the second member of the vector d_{ij} is zero.

Let us define the tracking error vector as follows:

$$e_i = x_i - r - d_{i0} \tag{12}$$

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

Algorithm 1 Update PrC p_i for UAV i

Require:
PrCs of neighbors: $p_j(k), j \in \mathcal{N}_i$. // k is the current iteration step
Ensure:
Updated PrC of UAV i: $p_i(k+1)$.
for $j = 1; j <= n; j++$ **do**
  if UAV j is a neighbor of i **then**
    $per[j] = p_j(k)$
  else
    $per[j] = n$ *//Store $p_j(k)$, $j \in \mathcal{N}_i$ into vector $per[n]$*
  end if
end for
if UAV i is a leader **then**
  $p_i(k+1) = 1$
else
  $p_i(k+1) = \sigma_n(\min\{per[j]\} + 1)$ // where $\sigma_n(\cdot) = \mathrm{sgn}(\cdot)\min\{n, |\cdot|\}$
end if
return $p_i(k+1)$ *//UAV i transmits $p_i(k+1)$ to others within its neighborhood.*

Algorithm 2 Calculate weighted error measurement for UAV i

Require:
Positions (X_j, Y_j), velocities $(\dot{X}_j, \dot{Y}_j)$ and PrCs $p_j(k)$
where $j \in \mathcal{N}_i$
Ensure:
Weighted error measurement z_i *// shown in equation (11)*
for $j = 1; j <= n; j++$ **do**
  if UAV j is a neighbor of i **then**
    $\omega_{ij}^a(k) = \frac{\frac{1}{p_j(k)}}{\sum_{j\in\mathcal{N}_i}\frac{1}{p_j(k)}}$
  else
    $\omega_{ij}^a(k) = 0$
  end if
end for
if UAV i is a leader **then**
  $\omega_i^l(k) = 1$
else
  $\omega_i^l(k) = 0$
end if
return $z_i = \sum_{j\in\mathcal{N}_i}\omega_{ij}^a(x_i - x_j) + \omega_i^l(x_i - r(t))$

where $d_{i0} = [d_{i0}^X, 0]^T$ is a constant offset vector. Then, we can write the error dynamics for UAV i (on x_e-axis) as $\dot{e}_i = Ae_i + Bu_i$. Note that $\ddot{r} \approx 0$, the RFT then satisfies $\dot{\mathcal{R}} = A\mathcal{R}$.

For the overall system, the error dynamics is

$$\dot{e} = \mathcal{A}e + \mathcal{B}u \tag{13}$$

where $u(k) = [u_1, u_2, \cdots, u_n]^T$. According to Eqs. (12), (13), we can include that the L-F consensus will be achieved, if the overall error "e" converges to zero. We propose a distributed control law u_i for UAV i (either a leader or a follower) as follows:

$$u_i = -K\left(z_i - \sum_{j=1}^n \omega_{ij}^a d_{ij} - \omega_i^l d_{i0}\right) \tag{14}$$

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

where $K \in \mathbb{R}^{1\times 2}$ is a gain matrix. The weights ω_{ij}^a and ω_i^l are calculated by using Algorithms 1 and 2. When $\omega_{ij}^a = 0$, we have $j \notin \mathcal{N}_i$. We can rewrite

$$\sum_{j\in\mathcal{N}_i} \omega_{ij}^a (x_i - x_j) = \sum_{j=1}^n \omega_{ij}^a (x_i - x_j) \tag{15}$$

We note that $x_i - x_j - d_{ij} = x_i - \mathcal{R} - x_j + \mathcal{R} - (d_{i0} + d_{0j})$. If we replace z_i in (14) by (11) and using (15) and (12), then we obtain $u_i = -K\left((\sum_{j=1}^n \omega_{ij}^a + \omega_i^l)e_i - \sum_{j=1}^n \omega_{ij}^a e_j\right)$. According to u_i, we can write the controller outputs "u" of all the UAVs in matrix form as follows:

$$u = -\mathcal{K}\cdot\left(\begin{bmatrix} \sum_{j=1}^n \omega_{1j}^a + \omega_{11}^l & \cdots & -\omega_{1n}^a \\ \vdots & \ddots & \vdots \\ -\omega_{n1}^a & \cdots & \sum_{j=1}^n \omega_{nj}^a + \omega_{nn}^l \end{bmatrix} \otimes I_2\right)\cdot e \tag{16}$$

where $\mathcal{K} = I_n \otimes K$. We notice that the matrix before the Kronecker product symbol in the foregoing Eq. (16) is the interaction matrix G. Then, (16) is rewritten as

$$u = -\mathcal{K}\cdot(G \otimes I_2)\cdot e \tag{17}$$

4. Stability Analysis

Since the objective of the formation control is to force the tracking error vector (defined in Eq. (12), $i = \{1, 2, \ldots, n\}$) converge to zero, the stability of the system (13) is analyzed when the formation controller (14) is used.

4.1. Convergence condition

It is important to note that the interaction matrix G can be not symmetric since different weights are used in the formation controller. In order to show the stability of system (13), we need to show that the matrix G is invertible. Therefore, we introduce the following lemma.

Lemma 1. *Let $\mathcal{G} = \bigcup_{1\le j\le|\mathcal{V}|} \mathcal{G}_j$ be an undirected simple graph, where $\mathcal{G}_j$ represents connected subgraphs of $\mathcal{G}$. For any two subgraphs $\mathcal{G}_{j_a}$ and $\mathcal{G}_{j_b}$, their node sets satisfy $\mathcal{V}_{j_a} \cap \mathcal{V}_{j_b} = \Phi$. Then the interconnection matrix G in Eq. (6) has only positive eigenvalues, if $\mathcal{V}_L^j \neq \Phi$.*

The proof is similar to Lemma 1 in ref. [20].

In Lemma 1, we show the cases when the interaction matrix G is positive definite. We replace "u" in Eq. (13) by Eq. (17), then we have $\dot{e} = (\mathcal{A} - \mathcal{B}\mathcal{K}\cdot(G \otimes I_2))\cdot e$. By using the mixed-product property of Kronecker product, we can rewrite the foregoing equation as follows:

$$\dot{e} = (I_n \otimes A - (G \otimes BK))\cdot e \tag{18}$$

Proposition 1. *The origin of (18) is asymptotically stable, if $\lambda_{\min}(G) > 0$.*

Proof. According to matrices B and K, we are able to find n elementary matrices $S_1, \ldots, S_n$, which render $G \otimes BK$ as follows:

$$\left(\Pi_{i=1}^n S_i\right)(G \otimes BK)\left(\Pi_{i=1}^n S_i\right)^T = \begin{bmatrix} 0 & 0 \\ k_1 G & k_2 G \end{bmatrix}$$

If we denote $S = \Pi_{i=1}^n S_i$ and set $\tilde{e} = Se$, then we have $\dot{\tilde{e}} = \mathcal{A}_c\tilde{e}$, where $\mathcal{A}_c$ satisfies

$$\mathcal{A}_c = \begin{bmatrix} 0 & I_n \\ -k_1 G & -k_2 G \end{bmatrix} \tag{19}$$

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

$$\begin{cases} E_2 = \lambda E_1 \\ -k_1 G E_1 - k_2 G E_2 = \lambda E_2 \end{cases}$$

where λ represents an eigenvalue of $\mathcal{A}_c$. Thus, we simplify the foregoing equation as $(-k_1 - k_2\lambda)GE_1 = \lambda^2 E_1$, which means that E_1 is an eigenvector of matrix G with eigenvalue $\lambda_i(G)$. In this

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

notation, we have $(-k_1 - k_2\lambda)\lambda_i(G) = \lambda^2$. We rewriting this equation as $\lambda^2 + k_2\lambda_i(G)\lambda + k_1\lambda_i(G) = 0$. Therefore, if $\lambda_{\min}(G) > 0$, then

$$Re(\lambda) \leq Re\left(\frac{-k_2\lambda_{\min}(G) + \sqrt{k_2^2\lambda_{\min}^2(G) - 4k_1\lambda_{\min}(G)}}{2}\right) < 0 \tag{20}$$

Thus $\mathcal{A}_c$ is Hurwitz, the origin of (18) is asymptotically stable. □

According to inequality (20), we have the following corollary.

Corollary 1. *Given some constant gains k_1 and k_2, the converging speed of e in (18) is proportional to $\lambda_{\min}(G)$.*

According to (20), we observe that λ is proportional to $\lambda_{\min}(G)$; therefore, we have the result of Corollary 1.

4.2. Convergence speed

According to Corollary 1, we known that the minimum eigenvalue of the interaction matrix can decide the converging speed of the system (18). In the sequel, we investigate how the weighted-neighbor-based control increases the converging speed compared to the average-neighbor-based control.

We define $W_1 = \text{diag}\{\sum_{j\in\mathcal{N}_1} p_j^{-1}(k), \sum_{j\in\mathcal{N}_2} p_j^{-1}(k), \ldots, \sum_{j\in\mathcal{N}_n} p_j^{-1}(k)\}$ and $W_2 = \text{diag}\{p_1(k), p_2(k), \ldots, p_n(k)\}$ as two weights matrices, where the scalar $p_i(k)$ represents the PrC of UAV $i \in \mathcal{V}$. According to Algorithm 2, the weighted interaction matrix is obtained as follows:

$$G = G^L + I_n - W_1^{-1} \cdot G^A \cdot W_2^{-1} \tag{21}$$

In general, an interaction matrix G of a weighted topology is not symmetric, although the graph is undirected. However, we can prove that the eigenvalues of G are always real, which is shown in the following lemma.

Lemma 2. *The eigenvalues of the interaction matrix G are real.*

Proof. We can observe that matrices W_1 and W_2 are invertible and positive-definite. We set $T = W_1^{-\frac{1}{2}} W_2^{\frac{1}{2}}$ and carry out similarity transformation $T^{-1}GT$, which yields

$$T^{-1}GT = \left(W_1^{-\frac{1}{2}} W_2^{\frac{1}{2}}\right)^{-1} G \left(W_1^{-\frac{1}{2}} W_2^{\frac{1}{2}}\right) = G^L + I_n - W_2^{-\frac{1}{2}} W_1^{-\frac{1}{2}} G^A W_1^{-\frac{1}{2}} W_2^{-\frac{1}{2}}$$

We recall that matrices G^L, G^D, W_1, and W_2 are diagonal, and G^A is symmetric. We obtain that $T^{-1}GT$ is symmetric, whose eigenvalues are real. Therefore, the interaction matrix G is similar to a symmetric matrix, such that its eigenvalues are real. □

The distribution of the eigenvalues of the interaction matrix on the real axis is investigated by the following corollary.

Corollary 2. *If the weighted interaction matrix satisfies (21), then $\lambda(G) \in \mathbb{R}$ and $0 < \lambda(G) \leq 3$.*

The proof is in Appendix A.

Let us denote by G' the interaction matrix of multiple quadrotors with average formation control. G' then yields

$$G' = G^L + I_n - (G^D)^{-1} \cdot G^A \tag{22}$$

Proposition 2. *In the weighted and average neighbor-based formation control strategies, the corresponding interaction matrices are, respectively, (21) and (22). Then, their smallest eigenvalues of their interaction matrices yield*

$$\lambda_{\min}(G) \geq \lambda_{\min}(G')$$

Proof. Without loss of generality, we suppose that the UAVs have only one leader. We also assume that the vertices $1, 2, \ldots, n$ (which correspond to UAVs $1, 2, \ldots, n$) are sorted in increasing order of distances in the graph. Then, we conclude that in each row of the weighted interaction matrix G,

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

the magnitude of the non-diagonal and nonzero entries are decreasing. On the contrary to G, in the unweighted interaction matrix G', the non-diagonal entries in a row are equal. For both matrices G and G', we found that their Gershgorin circles are the same as shown in Fig. A.1.

In order to illustrate the distribution of the smallest eigenvalues of matrices G and G', we carry out a similar transformation on the interaction matrices G and G'. The idea is to increase the radius of the Gershgorin circle, whose center is $(1+\omega_i^l, 0)$, while at the same time, to reduce the radius of the Gershgorin circle whose center is $(1, 0)$. Then, the transformation matrix T is the product of n diagonal matrices

$$T = T_n \times \cdots \times T_1 = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & v_n \end{pmatrix} \times \cdots \times \begin{pmatrix} v_1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

where $v_1 > v_2 > \cdots > v_n = 1$ are positive scalars. Additionally, $v_i - v_{i+1} > v_{i+1} - v_{i+2}$, where $i = \{1, 2, \ldots, n-2\}$. We note that according to Lemma 1, in each row of both matrices G and G', at least one nonzero entry exists before the diagonal entry in that row. Then, the scalars v_i are selected with the following constraint:

$$\sum_{j=1, j\neq i}^{n} \frac{1}{|\mathcal{N}_i|} v_i v_j^{-1} < 1 \tag{23}$$

We now apply the similar transformations TGT^{-1} and $TG'T^{-1}$ in (24) and (25), to see the difference on their Gershgorin circles.

$$TGT^{-1} = \begin{pmatrix} 1+\omega_1^l & -\omega_{12}^a v_1 v_2^{-1} & & \cdots & -\omega_{1n}^a v_1 v_n^{-1} \\ -\omega_{21}^a v_2 v_1^{-1} & 1 & -\omega_{23}^a v_2 v_3^{-1} & \cdots & -\omega_{2n}^a v_2 v_n^{-1} \\ -\omega_{31}^a v_3 v_1^{-1} & -\omega_{32}^a v_3 v_2^{-1} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\omega_{n-1n}^a v_{n-1} v_n^{-1} \\ -\omega_{n1}^a v_n v_1^{-1} & \cdots & & -\omega_{nn-1}^a v_n v_{n-1}^{-1} & 1 \end{pmatrix} \tag{24}$$

$$TG'T^{-1} = \begin{pmatrix} 1+\omega_1^l & -\frac{1}{|\mathcal{N}_1|} v_1 v_2^{-1} & & \cdots & -\frac{1}{|\mathcal{N}_1|} v_1 v_n^{-1} \\ -\frac{1}{|\mathcal{N}_2|} v_2 v_1^{-1} & 1 & -\frac{1}{|\mathcal{N}_2|} v_2 v_3^{-1} & \cdots & -\frac{1}{|\mathcal{N}_2|} v_2 v_n^{-1} \\ -\frac{1}{|\mathcal{N}_3|} v_3 v_1^{-1} & -\frac{1}{|\mathcal{N}_3|} v_3 v_2^{-1} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\frac{1}{|\mathcal{N}_{n-1}|} \\ -\frac{1}{|\mathcal{N}_n|} v_n v_1^{-1} & \cdots & & -\frac{1}{|\mathcal{N}_n|} v_n v_{n-1}^{-1} & 1 \end{pmatrix} \tag{25}$$

According to Algorithm 2, we know that

- In the first rows of TGT^{-1} and $TG'T^{-1}$, the sums of the magnitudes of the non-diagonal entries of TGT^{-1} and $TG'T^{-1}$ are equal.
- In the rows $i = \{2, \ldots, n\}$, since ω_{ij}^a is decreasing and $v_i v_j$ are increasing, we have $\sum_{j=1, j\neq i}^{n} \omega_{ij} v_i v_j^{-1} \leq \sum_{j=1, j\neq i}^{n} \frac{1}{|\mathcal{N}_i|} v_i v_j^{-1}$
- According to (23), the radius of the Gershgorin circles whose center is (1,0) is smaller than 1.

The eigenvalues of a matrix are continuously changing when the matrix's entries vary continuously. It is obvious that the continuous change of the entries will lead to a continuous change of the distribution of the Gershgorin circle. Therefore, the Gershgorin circles of matrix TGT^{-1} with the center point $(1, 0)$ have smaller radius than that of matrix $TG'T^{-1}$, which implies that the distribution of the eigenvalues of the matrix TGT^{-1} is further from the origin than the matrix $TG'T^{-1}$. According to the principle of continuity, we conclude that $\lambda_{\min}(G) > \lambda_{\min}(G')$. □

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

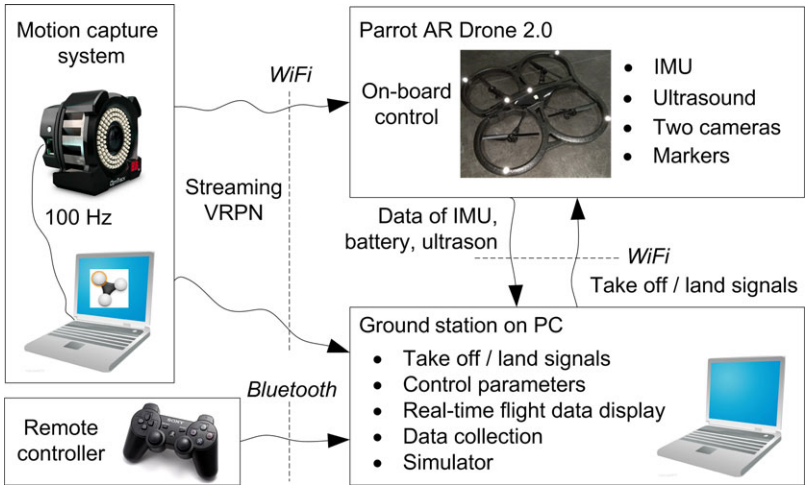

Fig. 3. The experimental setup of the real-time experiments.

According to (1), we conclude that for a multiple quadrotors system, if the weighed-neighbor-based formation controller is used, the convergence speed of the tracking error (12) is greater than using the average-neighbor-based formation controller. Furthermore, since the topology is disconnected when the interaction matrix is singular, a formation, whose minimum eigenvalue of its interaction matrix is greater, is less likely to become disconnected.

5. Simulation and Experiments

The *Heudiasyc* laboratory has developed a PC-based simulator-experiment framework for controlling a quadrotor and also a formation of quadrotors.[8] Within this framework, control algorithms are implemented on UAVs rather than on a PC. There does not exist a central controller that sends control signals to UAVs.

In the simulator, the complete UAV dynamics are implemented. The flight of the UAVs is animated using a high performance real-time 3D engine Irrlicht.[1] The experimental setup is shown in Fig. 3. In the experiments, the motion capture system OptiTrack[2] is used to localize the UAVs in the formation.

The Parrot AR.Drone 2 quadrotors[3] are used for real-time experiments. We have completely changed the software provided by AR.Drone 2 quadrotors. The on-board Linux system has been deleted and replaced by Poky 12.0[4] system. Some techniques in the Paparazzi UAV[5] Project, such as communication protocol, are applied here in order to implement our own algorithms on the quadrotors. We have then used the available sensors and the materials of Parrot quadrotors with our proper low-level and high-level controllers.

5.1. Simulation results

5.1.1. Comparison of average/weighted-neighbor-based formations. The average-neighbor-based controller (26), proposed in ref. [21], is used for purpose of comparison.

$$u_i(k) = \begin{cases} -K(x_i - r(t)), & i \in \mathcal{V}_L \\ -K\left(\sum_{j\in\mathcal{N}_i} \frac{1}{|\mathcal{N}_i|}(x_i - x_j - d_{ij})\right), & i \in \mathcal{V} - \mathcal{V}_L \end{cases} \tag{26}$$

[1] http://irrlicht.sourceforge.net/.

[2] http://optitrack.com/.

[3] https://www.parrot.com/global/drones/parrot-ardrone-20-elite-edition.

[4] https://www.yoctoproject.org/software-item/poky/.

[5] https://wiki.paparazziuav.org/wiki/Main_Page.

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

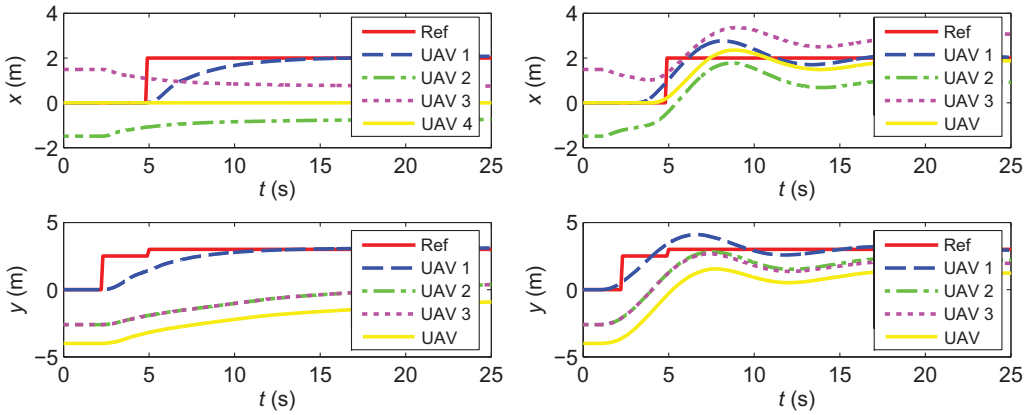

Fig. 4. Formations of four UAVs (UAV 1 is the leader and UAV 2, 3, 4 are followers) with average (left) and weighted (right) neighbor-based formation control.

where $K = \text{diag}\{1.39, 1.2\}$. We note that in (26), the neighbors of one UAV have no special weights and they are undifferentiated. Furthermore, the motion of the leader only depends on the reference signal, as defined in the work.[23]

In our proposed controller (14), we give the same gain matrix K as in (26). Then, we proceed the following two cooperation scenarios.

Scenario 1 Aggregation of four UAVs

In this test, each UAV can detect neighbors within an area of $d = 3$ m around itself, and the desired inter-distance is 1.5 m. The initial positions are as follows: UAV 1, (0, 0, 1); UAV 2, (−1.49, −2.6, 1); UAV 3, (1.49, −2.6, 1); and UAV 4, (0, −4, 1). All the UAVs have zero initial velocities. The goal of cooperation is to aggregate all the UAVs around a destination point at (2, 3, 1) and to keep a rectangular formation pattern.

Figure 4 shows the comparison of the average and weighted-neighbor-based cooperation controllers, whose corresponding interaction matrices are G_1 and G_2 in (27).

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \quad G_2 = \begin{pmatrix} 2 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{6}{11} & 1 & -\frac{3}{11} & -\frac{2}{11} \\ -\frac{6}{11} & -\frac{3}{11} & 1 & -\frac{2}{11} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \tag{27}$$

In Fig. 4 (left), we observe that the aggregation is failed, since the followers are outside of the leader's neighborhood few seconds after the formation has begun. However, in Fig. 4 (right), under the same initial conditions, the UAVs can finally achieve the goal of aggregation. This simulation shows that our control strategy with weighted neighbors has better robustness in connectivity preserving than the strategy with average neighbors, in terms of topology maintenance.

Scenario 2 Platooning of four UAVs

In this test, a platooning of four UAVs is carried out, the RFT is a stationary point at (1, 0, 1). The initial positions of the four UAVs are (0, 0, 1), (−1.5, 0, 1), (−3, 0, 1), and (−4.5, 0, 1). Figure 5 shows the planar translational motion curves. We observe that the convergence speed by using the weighted-neighbor-based formation controller (in Fig. 5 (right)) is faster than the average-neighbor-based formation controller (in Fig. 5 (left)). On the other hand, the improvement appears in Fig. 6 describing the inter-distance of UAVs. Indeed, we observe that one peak is smaller by using our proposed method and the convergence to the desired inter-distance is also faster.

In Fig. 6 (left), the maximum inter-distance of UAVs 1 and 2 is up to 1.8 m, where its desired value is 1.5 m. While in Fig. 6 (right), the inter-distance d_{12} is less than 1.75 m. Although the maximum deviation of d_{23} and d_{34} is getting larger, they are always smaller than 1.75 m. Consider the limited field of detection, we can observe that the proposed control method can reduce the tendency of disconnection of the topology, which can be concluded as another improvement on the average-neighbor-based control.

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

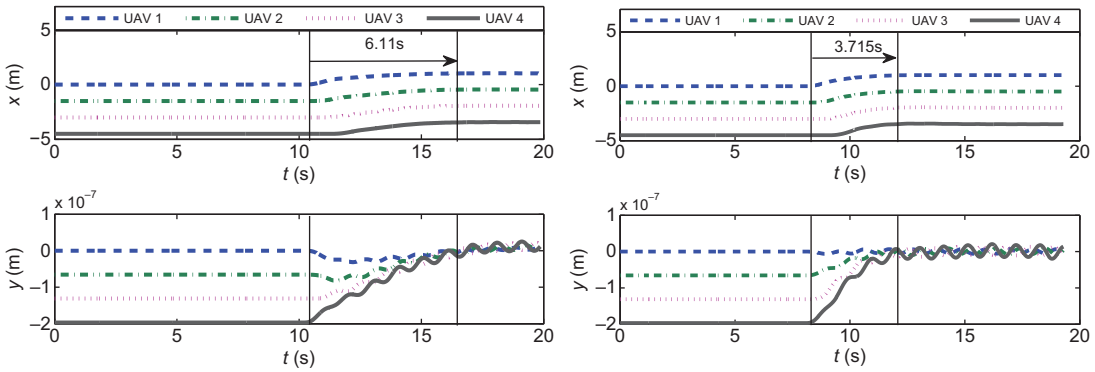

Fig. 5. *X* and *Y* curves of a platooning of four UAVs with average (left) and weighted (right) neighbor-based formation control.

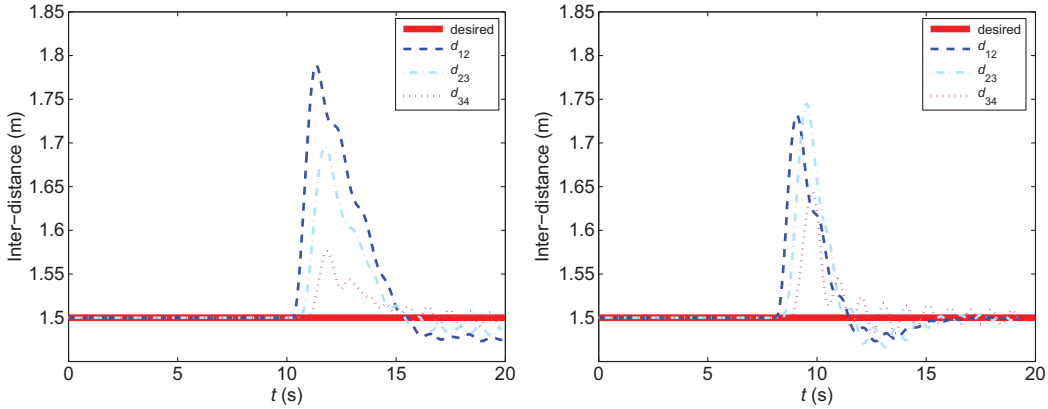

Fig. 6. Inter-distances of the four UAVs with average (left) and weighted (right) neighbor-based formation control.

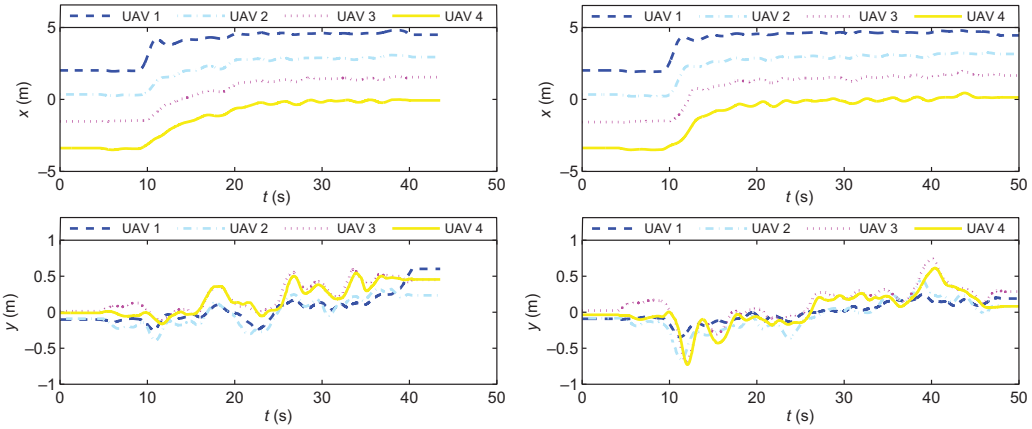

Fig. 7. X and Y curves of the platooning of four quadrotors. Average (left) and weighted (right) neighbor-based formation control.

5.2. *Experimental results*

Scenario 1 Platooning of four UAVs

A real-time platooning of four UAVs is carried out in experiments. Figures 7 and 8 respectively show the translational curves and the magnitudes of the inter-distance of the quadrotors.

In Fig. 7 (right), we can observe that the UAVs 2, 3, and 4 have faster response to the UAV 1 (which is the leader) than that in Fig. 7 (left). Indeed, the proposed weighted-neighbor-based cooperation control can reduce the inertia of the multi-agent system than that using average-neighbor-based

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

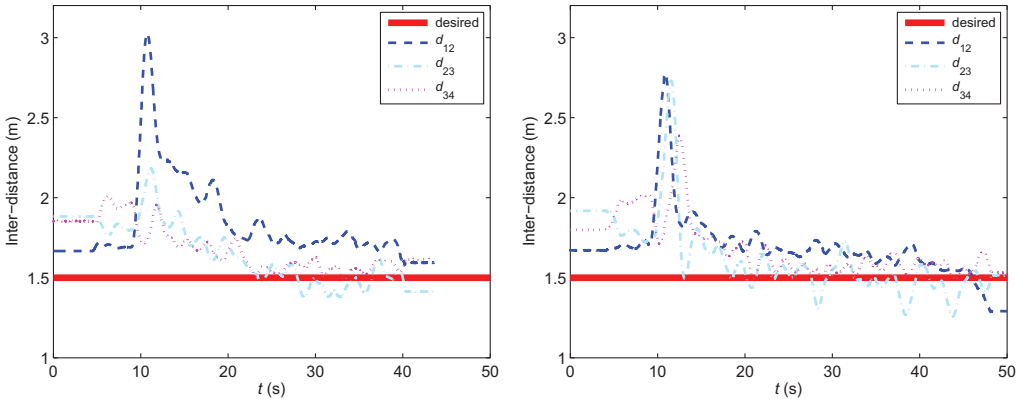

Fig. 8. Inter-distances of the quadrotors. Average (left) and weighted (right) neighbor-based formation control

Fig. 9. The snapshots of the quadrotors in formation of rectangle while tracking a trajectory "8" given to the leader, which is switchable.

control. Similar results can be drawn from observing Fig. 8. The maximum peak of the inter-distance deviation in Fig. 8 (right) is smaller than that in Fig. 8 (left).

In real-time experiments shown in Figs. 7 and 8, the data of quadrotors before taking off and after landing are also recorded. We should note that the cooperation control does not work during the taking off and landing period. Therefore, on the left side of Figs. 7 and 8, the quadrotors start to take off after around 5 s and land after 38 s. On the right side of Figs. 7 and 8, the quadrotors start to take off after around 5s and land after 48 s.

Scenario 2 Formation with dynamical weights

In this scenario, we implement a formation of four quadrotors tracking a trajectory of the "8" pattern. The leader changes between UAV 1 and UAV 2. The UAV 1 is initially assigned as a leader. In this case, according to Algorithms 1 and 2, the interaction matrix dynamically changes between G_1 and G_2.

$$G_1 = \begin{pmatrix} 2 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{3}{4} & 1 & -\frac{1}{4} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{3}{4} & 0 & -\frac{1}{4} & 1 \end{pmatrix} G_2 = \begin{pmatrix} 1 & -\frac{3}{4} & 0 & -\frac{1}{4} \\ -\frac{1}{2} & 2 & -\frac{1}{2} & 0 \\ 0 & -\frac{3}{4} & 1 & -\frac{1}{4} \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 1 \end{pmatrix}$$

The snapshots of the real-time experiment are given in Fig. 9. The interaction topology of the four quadrotors is shown in Fig. 10. The corresponding video is attached with this paper and is available on https://www.youtube.com/watch?v=KNzlogp4pgofeature=youtu.be The UAVs attain the center of the "8" trajectory. Each time the change of leader shows the real-time feasibility of such a dynamical weights calculation of the multi-quadrotor system.

In the video, the latency of the followers w.r.t. the leader is observed. Among the quadrotors, only the leader can obtain the referenced formation trajectory. Some of the followers (1 and 2) are

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

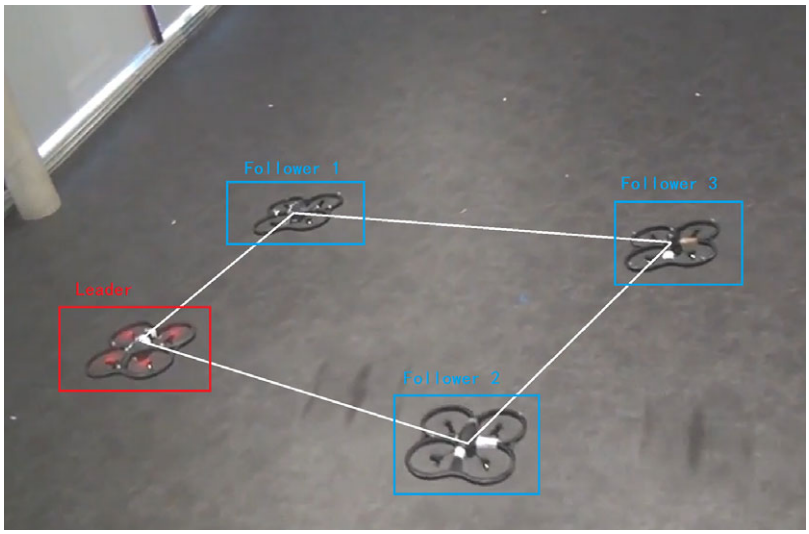

Fig. 10. Four quadrotors formation with one leader and three followers. The white lines represent the edges of interaction. The red/blue rectangles represent the leader/followers.

connected with the leader and follower 3 is not directly connected with the leader. The motion of follower 3 is based on followers 1 and 2. Since each quadrotor is a system with inertia, the motion of the leader will not cause a simultaneous motion of followers 1 and 2. The motion of followers 1 and 2 will not cause a simultaneous motion of follower 3. Therefore, we observe the latency of followers 1 and 2 w.r.t. the leader, and latency of follower 3 w.r.t. followers 1 and 2. Therefore, the difference of follower 3 w.r.t. the leader is more significant. This phenomenon of latency will become obvious as the increasing of the number of agents. Nevertheless, the proposed formation tracking control can reduce the latency which is shown in Figs. 5 and 7 for instance.

6. Conclusion

In this paper, a distributed weighted-neighbor-based control has been proposed for multi-robot systems, especially for multi-quadrotor systems. An algorithm is given to online calculate the weights. The stability analysis is proceeded in terms of interaction matrix, which can be non-symmetric in our case. We prove that the weighted-neighbor-based control can improve the converging speed of the error state of the quadrotors, compared to the average-neighbor-based control. These theoretical results are validated by simulation and real-time experiments, which show faster convergence rates of UAVs cooperation error and its capacity of dynamically updating the weights, thus to change the role of UAVs.

Some interesting work such as proving stabilities with switching topologies, implementing the proposed control in an environment with obstacles, and combining algorithms for collision avoidance of quadrotors can be proceeded in the future.

Acknowledgements

The special gratitude is given to the Heudiasyc laboratory, where this work was carried out and funded in the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02) and the ROBOTEX Equipment of Excellence (Reference ANR-10- EQPX-44). Now, some successive studies are performed in the CHRS (Complex Heterogeneous Systems) Laboratory, Guangzhou Institute of Advanced Technology, Chinese Academy of Science. These works are supported by the projects "Natural Science Foundation of Guangdong Province (2018A030310046)" and "China Postdoctoral Science Foundation funded project (2019M662848)". I would like also to extend my gratitude to Dr. Isabelle Fantoni for her direction on this work. I would like also thank Dr. Guillaume Sanahuja and other colleagues who are in Heudiasyc laboratory for their help on the simulator-experiment platform.

Conflicts of Interest

The authors declare none competing interests.

References

1. D. K. D. Villa, A. S. BrandÃčo and M. Sarcinelli-Filho, "Path-Following and Attitude Control of a Payload Using Multiple Quadrotors," *2019 19th International Conference on Advanced Robotics (ICAR)*, Belo Horizonte, Brazil (2019) pp.535–540.
2. Z. Wang, S. Singh, M. Pavone and M. Schwager, "Cooperative Object Transport in 3D with Multiple Quadrotors Using No Peer Communication," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD (2018) pp.1064–1071.
3. N. Michael, J. Fink and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Auton. Robots* **30**(1), 73–86 (2011).
4. G. Parlangeli and M. E. Valcher, "Accelerating consensus in high-order leader-follower networks," *IEEE Control Syst. Lett.* **2**(3), 381–386 (2018).
5. A. Sarlette, "Adding a single state memory optimally accelerates symmetric linear maps," *IEEE Trans. Autom. Control* **61**(11), 3533–3538 (2016).
6. C. Liu, H. Zhang, G. Xiao and S. Sun, "Integral reinforcement learning based decentralized optimal tracking control of unknown nonlinear large-scale interconnected systems with constrained-input," *Neurocomputing* **23**, 1–11 (2019).
7. Q. Liu, Z. Wang, X. He and D. H. Zhou, "On Kalman-consensus filtering with random link failures over sensor networks," *IEEE Trans. Autom. Control* **63**(8), 2701–2708 (2018).
8. Z. Hou and I. Fantoni, "Interactive leader-follower consensus of multiple quadrotors based on composite nonlinear feedback control," *IEEE Trans. Control Syst. Technol.* **26**(5), 1732–1743 (2018).
9. S. Kruzick and J. M. F. Moura, "Optimal filter design for signal processing on random graphs: Accelerated consensus," *IEEE Trans. Signal Process.* **66**(5), 1258–1272 (2018).
10. L. Xiao and S. Boyd, "Fast Linear Iterations for Distributed Averaging," *42nd IEEE International Conference on Decision and Control*, Maui, HI (2003) pp. 4997–5002.
11. X. Duan, J. He, P. Cheng and J. Chen, "Exploiting a mobile node for fast discrete time average consensus," *IEEE Trans. Control Syst. Technol.* **24**(6), 1993–2001 (2016).
12. M. Turpin, N. Michael and V. Kumar, "CAPT: Concurrent assignment and planning of trajectories for multiple robots," *Int. J. Rob. Res.* **33**(1), 98–112 (2014).
13. B. Grocholsky, V. Kumar and H. Durrant-Whyte, "Anonymous cooperation in robotic sensor networks," *J. Biomech.* **27**(6), 827–833 (2004).
14. G. Prencipe and N. Santoro, "Distributed Algorithms for Autonomous Mobile Robots," *Fourth IFIP International Conference on Theoretical Computer Science-TCS* (2006) pp. 47–62.
15. A. Franchi, G. Oriolo and P. Stegagno, "Mutual localization in multi-robot systems using anonymous relative measurements," *Int. J. Rob. Res.* **11**(32), 1102–1122 (2013).
16. M. Turpin, N. Michael and V. Kumar. "Trajectory design and control for aggressive formation flight with quadrotors," *Auton. Robots* **33**, 143–156 (2012).
17. R. Falconi, S. Gowal and A. Martinoli, "Graph Based Distributed Control of Non-Holonomic Vehicles Endowed with Local Positioning Information Engaged in Escorting Missions," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK (2010) pp. 3207–3214.
18. S. Y. Shafi, M. Arcak and L. El Ghaoui, "Graph weight allocation to meet Laplacian spectral constraints," *IEEE Trans. Autom. Control* **57**(7), 1872–1877 (2012).
19. L. Pan, H. Shao, M. Mesbahi, Y. Xi and D. Li, "Bipartite consensus on matrix-valued weighted networks," *IEEE Trans. Circ. Syst. II Exp. Briefs* **66**(8), 1441–1445 (2019).
20. Z. Hou and I. Fantoni, "Distributed Leader-Follower Formation Control for Multiple Quadrotors with Weighted Topology," *2015 10th System of Systems Engineering Conference (SoSE)*, San Antonio, TX (2015) pp. 256–261.
21. J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control* **49**(9), 1465–1476 (2004).
22. Z. Hou and I. Fantoni, "Composite Nonlinear Feedback-Based Bounded Formation Control of Multi-Quadrotor Systems," *2016 European Control Conference (ECC)*, Aalborg (2016) pp. 1538–1543.
23. W. Ni and D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Syst. Control Lett.* **59**(3–4), 209–217 (2010).

Appendix

A. Proof of Corollary 2

Proof. We prove this corollary using Gershgorin circle theorem. The Gershgorin circle theorem can be used to bound the spectrum of a square matrix. According to lemma 1, lemma 2, and the

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press

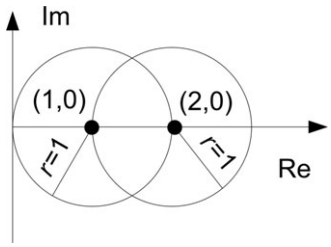

Fig. A.1. Gershgorin circle

Gershgorin circle theorem, we can draw the Gershgorin circle as in Fig.A.1. Since the eigenvalues are always within the Gershgorin circles, then we can obtain that the eigenvalues of matrix G satisfy $0 < \lambda(G) \leq 3$. □

https://doi.org/10.1017/S0263574721000126 Published online by Cambridge University Press