

Enhancing Convergence Speed of Multi-Agent Formation Control via Laplacian Functions

Qilong Zhang, Binglin He, Yangyang Zhao, Song Liu, *Member, IEEE*, and Yang Wang, *Member, IEEE*

Abstract—Formation control is an essential research topic in multi-agent systems (MAS), while the convergence speed of formation is critically important for applications with real-time performance requirements, such as rescuing tasks. However, there is still a lack of effective methods for practically usable formation control with controllable convergence speed. This paper introduces a novel Laplacian function-based approach to enhance the convergence speed of MAS in formation control. By utilizing the Laplacian matrix of the communication graph, eigenvalues are mapped to desired positions, thereby improving the convergence speed of the formation process. Additionally, this approach enables estimation and manipulation of the convergence speed, offering flexibility and adaptability to meet application-specific requirements. The proposed scheme is experimentally validated through multiple quadrotors, demonstrating its effectiveness and practical feasibility. Experimental results show that the formation convergence speed can be well controlled by appropriately designing the Laplacian functions.

I. INTRODUCTION

The field of robotics and biomimetics has witnessed an increasing interest in multi-agent systems (MAS) due to their versatility and potential in accomplishing complex applications [1]–[3]. Formation control is a critical objective in MAS that aims to achieve desired spatial arrangements and patterns of a team of agents, as shown in Fig. 1. While there are well-established achievements in the literature for formation control [4]–[9], such as maintaining formation, optimizing resource allocation, enhancing collective decision-making, etc., an important research question remains open and unexplored: how quickly can the agents in the system reach and maintain the desired formation configuration?

Generally, three problems should be well considered in formation control. The first is the acquisition of *locomotion information*. Formation control typically involves a combination of global and local information [10]. Global information refers to knowledge about the desired formation shape and positions in global coordinates [11], while local information pertains to the agents' states, such as attitudes and velocities [4]. The second is the *realization condition* of the formation task. Different environments, such as indoor scenarios with limited space and restricted maneuverability [12], outdoor settings with varying weather conditions and open spaces [13], or obstacle-cluttered environments with complex surroundings [14], impose specific demands for successful

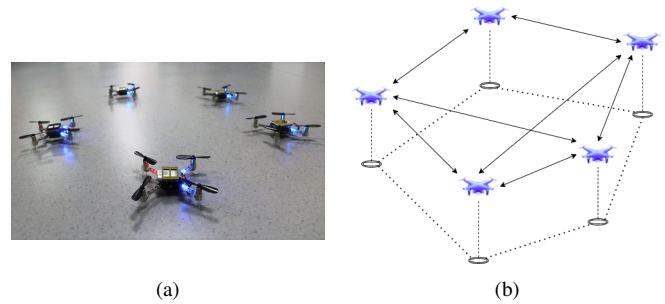


Fig. 1: A multi-agent system composed of five quadrotors (a) and the pentagon formation (b).

execution. Third, the *achievable performance* of avoidance strategies [15], fault tolerance mechanisms [16], time-varying formation adaptability [17], [18], communication delay [19], and disturbance rejection techniques [20] should also be taken into account.

Apart from the aforementioned problems, the convergence speed is also critically important, particularly for real-time applications. The convergence speed refers to how quickly the agents can reach and maintain the desired formation configuration. To enhance the convergence speed of formation control, two approaches can be considered. The first approach is *hardware-level improvements*, such as utilizing faster processors or advanced sensors. However, this approach inevitably increases cost or even technical constraints. The second approach is to *establish more interconnections* [21], allowing for more effective information exchange and coordination between agents. However, building more connections may introduce additional communication costs and increase computational complexity. Moreover, while theoretical studies have addressed the problem of improving the convergence speed in formation control using graph theory [22], [23], practical implementations are still lacking. This highlights the need to develop practical methods that can enhance the convergence speed in formation tasks.

In light of the great necessity for controllable convergence speed, our study focuses on the formation problem for a MAS under an undirected communication graph. We propose a novel control method based on the functions of the Laplacian matrix of the communication graph. By mapping the eigenvalues of the Laplacian matrix to desired positions, our aim is to achieve a faster convergence speed in the formation process. Additionally, this approach enables us to estimate and manipulate the convergence speed as desired, offering flexibility and adaptability to meet application-specific requirements. Finally, to investigate the practical convergence speed of the formation strategy in 3D space,

*This work was supported in part by the Yangfan Program of Shanghai, China, under Grant 21YF1429600. (Corresponding author: Yang Wang.)

The authors are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China {zhangql1, hebl2022, zhaoyy2, liusong, wangyang4}@shanghaitech.edu.cn

a group of quadrotors are chosen as representative agents for the experimental tests.

The rest of this paper is organized as follows. Sec. II provides an overview of graph theory concepts and introduces the agent model. In Sec. III, Laplacian functions are introduced, and the formation strategy designed based on these functions is presented. Sec. IV presents several Laplacian function design examples and evaluates their simulation performance. The experiment results, which implement the proposed approach using Crazyflie quadrotors, are discussed in Sec. V. Finally, Sec. VI concludes the study.

II. PRELIMINARIES

In this section, we present a concise introduction to fundamental concepts in graph theory and the single-integrator dynamics for the MAS considered in this paper, laying the foundation for understanding the subsequent sections.

A. Graph Theory

In relation to formation control, the MAS can be described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, 3, \dots, N\}$ denotes the set of vertices representing the agents, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges representing the connection between the agents. This paper focuses on an unweighted and undirected graph without self-loops or multiple edges.

The topology of the graph can be mathematically described by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$:

$$\mathbf{A} = \{a_{ij}\} : a_{ij} = \begin{cases} 1, & \text{if } (i, j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}, a_{ii} = 0$$

Here $a_{ij} = 1$ indicates a bidirectional connection between the i - and j -th agents, while $a_{ij} = 0$ indicates no connection.

Additionally, we define a symmetric positive semi-definite Laplacian matrix $\mathbf{L}$ of the graph $\mathcal{G}$ in the form:

$$\mathbf{L} = \{l_{ij}\} : l_{ij} = -a_{ij}, \text{ for } i \neq j, l_{ii} = \sum_{j=1, j \neq i}^N a_{ij}$$

It is worth mentioning that the Laplacian matrix $\mathbf{L}$ possesses a zero eigenvalue corresponding to the eigenvector $\mathbf{1}_N$, where $\mathbf{1}_N$ is a $N \times 1$ vector of ones. The off-diagonal elements of $\mathbf{L}$ are non-positive. The eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ of $\mathbf{L}$ are sorted in ascending order, with $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$, and the corresponding orthonormal eigenvectors are denoted as $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$, with $\mathbf{v}_1 = \frac{1}{\sqrt{N}}$.

B. Agent Model

Consider a group of N agents, the dynamics of each agent can be described using single-integrator equation:

$$\dot{q}_i(t) = u_i(t), \quad i \in \mathcal{V} \quad (1)$$

where $q_i(t) := [x_i(t), y_i(t)]^\top \in \mathbb{R}^2$ represents the global position of the i -th agent, and $u_i(t) \in \mathbb{R}^2$ represents the control input. To simplify notation, we define $\mathbf{q}(t) := [q_1^\top(t), q_2^\top(t), \dots, q_N^\top(t)]^\top \in \mathbb{R}^{2N}$ and $\mathbf{u}(t) := [u_1^\top(t), u_2^\top(t), \dots, u_N^\top(t)]^\top \in \mathbb{R}^{2N}$ as the aggregated position and input information of all agents, respectively.

III. LAPLACIAN FUNCTION-BASED FORMATION SCHEME

To enhance the convergence speed without increasing computational complexity and cost, we present an innovative approach to formation utilizing Laplacian functions in this section. These functions, denoted as $f(\mathbf{L})$, are designed as functions of the eigenvalues of the Laplacian matrix $\mathbf{L}$.

A. Laplacian Matrix Functions

To begin with, we introduce matrix functions $f(\mathbf{L})$ associated with the Laplacian matrix $\mathbf{L}$. These functions establish matrices that include network structure details and facilitate non-local interactions among agents. For simplicity, we use $f_{ij}(\mathbf{L})$ to refer to the (i, j) -th element of the matrix $f(\mathbf{L})$.

Consider a scalar function $f(x) : \mathbb{R} \rightarrow \mathbb{R}$, Laplacian functions $f(\mathbf{L})$ can be designed based on the spectral decomposition of the Laplacian matrix $\mathbf{L}$:

$$f(\mathbf{L}) = \sum_{i=1}^N f(\lambda_i) \mathbf{v}_i \mathbf{v}_i^\top \quad (2)$$

Equation (2) illustrates that the determination of $f(\mathbf{L})$ involves computing the spectrum $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ of $\mathbf{L}$, followed by the evaluation of $\{f(\lambda_1), f(\lambda_2), \dots, f(\lambda_N)\}$ accordingly. The eigenvectors of $f(\mathbf{L})$ remain unchanged from those of $\mathbf{L}$. Furthermore, it is noteworthy that the matrix $f(\mathbf{L})$ is symmetric in nature, as evident from equation (2).

It is important to mention that while equation (2) enables the calculation of general functions of $\mathbf{L}$, our focus here is on functions that maintain the specific structure of the Laplacian matrix, as discussed before. To maintain the desirable characteristics, three rules need to be satisfied by $f(\mathbf{L})$ [23].

Rule 1: $f(\mathbf{L})$ needs to be positive semi-definite, ensuring that the eigenvalues of $f(\mathbf{L})$ are all positive or zero.

Rule 2: Each element $f_{ij}(\mathbf{L})$ should meet $\sum_{j=1}^N f_{ij}(\mathbf{L}) = 0$ for all $i \in \mathcal{V}$, or equivalently $f(\mathbf{L})\mathbf{1}_N = \mathbf{0}_N$, i.e. each row sum of $f(\mathbf{L})$ should be equal to zero.

Rule 3: The off-diagonal elements $f_{ij}(\mathbf{L})$, $i \neq j$, are required to be non-positive and cannot all be zero simultaneously. Therefore, according to **Rule 2**, the diagonal entries of $f(\mathbf{L})$ should be strictly positive.

Further explanations on these design rules are provided. **Rule 1** holds true if the function $f(x)$ satisfies $f(x) \geq 0$ when $x \geq 0$, ensuring that the eigenvalues of $f(\mathbf{L})$ can be arranged in ascending order like those of $\mathbf{L}$. To verify **Rule 2**, by utilizing equation (2), we can check whether $f(0) = 0$. However, it is important to note that meeting the first two rules does not necessarily guarantee that the off-diagonal elements of $f(\mathbf{L})$ are non-positive, as specified by **Rule 3**.

B. Laplacian Function-based Formation Scheme

Assuming the three design rules are satisfied, we can use the Laplacian functions $f(\mathbf{L})$ to devise a novel category of continuous-time formation schemes:

$$\mathbf{u}(t) = (-f(\mathbf{L}) \otimes \mathbf{I}_2) \cdot (\mathbf{q}(t) - \mathbf{q}^*) \quad (3)$$

where $\mathbf{u}(t)$ and $\mathbf{q}(t)$ are defined in Sec. II-B. $\mathbf{q}^* \in \mathbb{R}^{2N}$ represents the desired formation position, which remains constant. This strategy drives N agents to a rotationally-invariant formation encoded through the formation graph $\mathcal{G}$. By utilizing the spectral properties of the Laplacian functions discussed in Sec. III-A, it has been proven that for any initial condition $\mathbf{q}(0) \in \mathbb{R}^{2N}$, the position vector $\mathbf{q}(t)$ asymptotically converges to the target position $\mathbf{q}^*$ if the formation graph $\mathcal{G}$ is connected [24, Thm. 6.12].

C. Extension to Directed Communication Graph

While this paper primarily focuses on undirected graphs, it is important to consider practical applications, especially in hardware experiments. In such cases, when two agents in an undirected graph are connected, they have access to each other's states, such as position and velocity. However, this can lead to increased communication loss and delay within the system, which is not conducive to practical applications.

To address this issue, we consider extending the interaction graph to a directed connected graph. This allows us to assess the fault-tolerance of the proposed formation scheme under less-than-ideal communication conditions. Detailed information will be provided in the subsequent Sec. IV-B.

IV. SIMULATION RESULTS

In this section, we will present various Laplacian functions that satisfy the three design rules and analyze their performance through numerical simulations.

A. Examples of Laplacian Functions

We consider several classes of completely monotonic functions that can be used to construct admissible Laplacian functions that meet the design rules. Here are four examples:

- 1) *Logarithmic function:* Denote $f(x) = \log(cx + 1)$ with $c > 0$. The corresponding Laplacian function is $f(\mathbf{L}) = \log(c\mathbf{L} + \mathbf{I}_N)$.
- 2) *Exponential function:* Denote $f(x) = 1 - e^{-cx}$ with $c > 0$. The corresponding Laplacian function is $f(\mathbf{L}) = \mathbf{I}_N - e^{-c\mathbf{L}}$.

- 3) *Quadratic function:* Denote $f(x) = \frac{1}{2}x(2c - x)$ with $c > \max_{i \in \mathcal{V}} \{\lambda_i\}$. The corresponding Laplacian function is $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + c\mathbf{L}$.
- 4) *Fractional power:* Denote $f(x) = x^\gamma$ with $0 < \gamma \leq 1$. The corresponding Laplacian function is $f(\mathbf{L}) = \mathbf{L}^\gamma$.

B. Performance Analysis

1) *Comparison of Laplacian functions:* To compare the effects of different Laplacian functions, we conduct simulations of system (1) for 10 agents using a selection of Laplacian functions introduced in Sec. IV-A. We run simulations for different time spans using the MATLAB `ode45` solver with a variable step size (maximum step size: 0.01 s).

In our simulation, we select a cycle graph with 10 nodes as the interaction graph, and the desired formation is a pentagram. The target locations of the 10 agents are determined based on the dimensions of the pentagram. We define R as the circumradius of the pentagram and ρ as the circumradius of its inner pentagon. The target locations are specified in polar coordinates as follows:

$$\begin{bmatrix} \mathbf{q}_k^* \\ \mathbf{q}_{k+1}^* \end{bmatrix} = \begin{cases} \rho [\cos(\frac{k\pi}{10}), \sin(\frac{k\pi}{10})]^\top, & \text{if } k \in \{1, 5, \dots\} \\ R [\cos(\frac{k\pi}{10}), \sin(\frac{k\pi}{10})]^\top, & \text{if } k \in \{3, 7, \dots\} \end{cases}$$

with

$$R = \sqrt{\frac{5 - \sqrt{5}}{10}}, \quad \rho = \sqrt{\frac{25 - 11\sqrt{5}}{10}}.$$

The Laplacian functions considered in the simulation are $f(\mathbf{L}) = \mathbf{L}$, $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + 5\mathbf{L}$, and $f(\mathbf{L}) = \mathbf{L}^{1/5}$. The initial positions $\mathbf{q}(0)$ of all agents are randomly generated from the open interval $(0, 1)$. Fig. 2 reports the trajectory of all agents for each Laplacian function, where the network topology edges are represented by **solid black lines**, and the initial/final positions are indicated by **stars/circles**. Fig. 3 displays the time evolution of the formation position error $\mathbf{e}(t) := \mathbf{q}(t) - \mathbf{q}^*$, which includes errors in the x -axis and y -axis. Both of these errors converge to zero. The second-smallest eigenvalues of the three Laplacian functions are $\lambda_2 = 2 - 2\cos(\pi/5) \simeq 0.3820$, $-\frac{1}{2}\lambda_2^2 + 5\lambda_2 \simeq 1.8369$ and

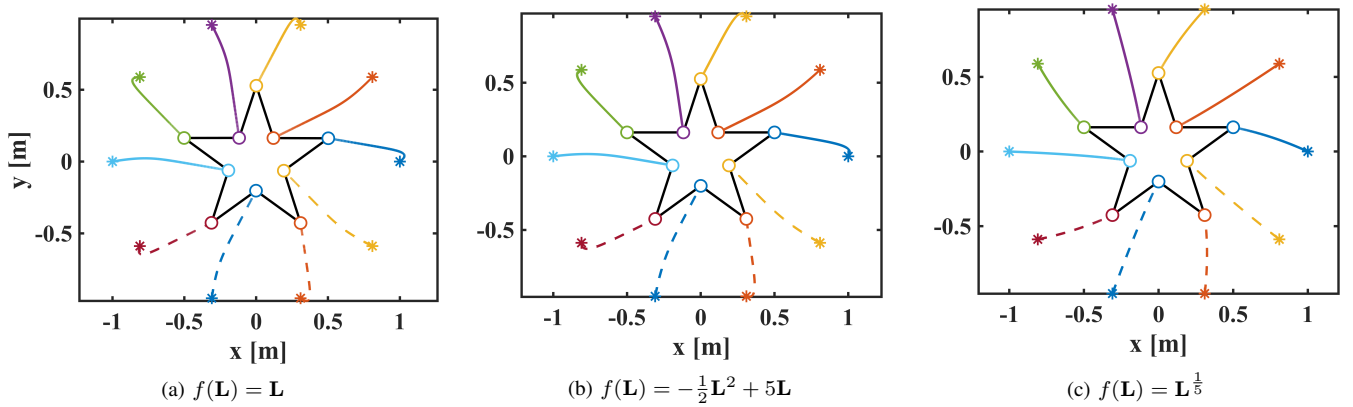


Fig. 2: The trajectory of the 10 agents for each Laplacian function. The edges of the network topology are represented by **solid black lines**, the initial/final positions are indicated by **stars/circles**, and the **colored lines** represent the trajectories produced by the agents. These figures imply that our formation control strategy can guarantee the formation to the desired positions.

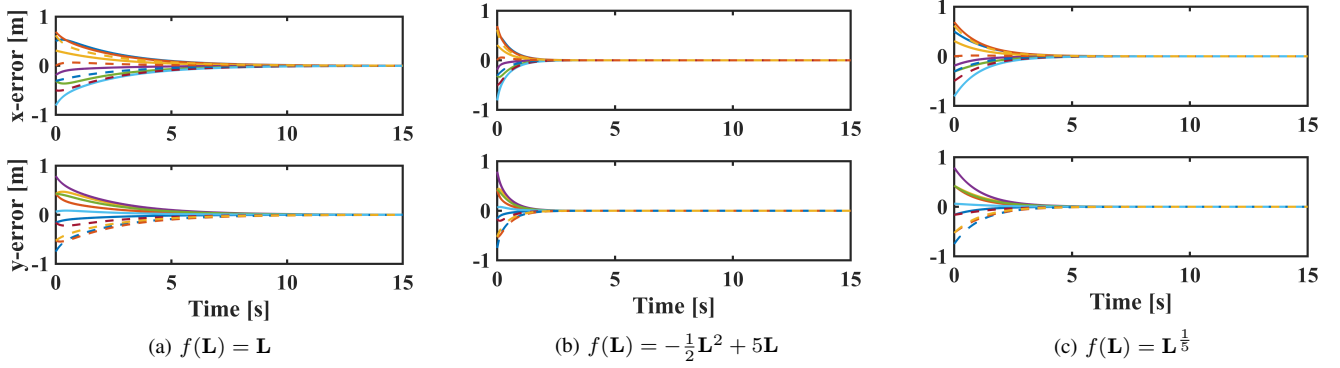


Fig. 3: The time evolution of the formation errors for each Laplacian function. The **colored lines** represent the formation errors of each agent, where Fig. 3b demonstrates the fastest convergence speed both in x - and y -directions.

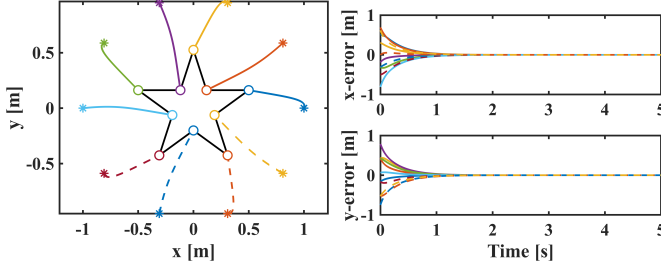


Fig. 4: The **combinatorial** Laplacian function $f(\mathbf{L}) = 6\mathbf{L} - \frac{1}{2}\mathbf{L}^2 + \mathbf{L}^{1/5}$ can achieve a much faster convergence speed.

$\lambda_2^{1/5} \simeq 0.8249$, respectively. It illustrates that the quadratic function and matrix power contribute to an accelerated convergence towards the desired formation, explaining the enhanced speed of convergence observed in Figs. 2b and 3b.

2) *Combinatorial Laplacian Function and Extension to Directed Graph:* However, the individual Laplacian functions presented above do not take into account the combination of various types of Laplacian functions. To address this limitation, we combine multiple Laplacian functions, expressed as $f(\mathbf{L}) = 6\mathbf{L} - \frac{1}{2}\mathbf{L}^2 + \mathbf{L}^{1/5}$, and proceed with the simulation again. As shown in Fig. 4, the convergence of the formation error is achieved in less than 5s, representing a significant enhancement in convergence speed.

We also investigate the applicability of our approach to directed graphs, where edges possess specific directions denoted by arrows between agents. For this purpose, we employ the same Laplacian function as previously mentioned. Fig. 5 shows that although the convergence of formation error in directed graphs is not as rapid as in undirected graphs, the agents are still able to achieve the desired formation. This verifies the efficacy and robustness of the designed formation strategy and the Laplacian matrix functions.

V. EXPERIMENTAL RESULTS

A. Experiment Setup

To assess the efficacy of the formation methodology in practical scenarios, we conduct experiments using Crazyflie nano-quadrators, operated with the Crazyswarm2 package [25]. The Crazyflies are equipped with the *Flow Deck*, which provides localization information by continuously measuring

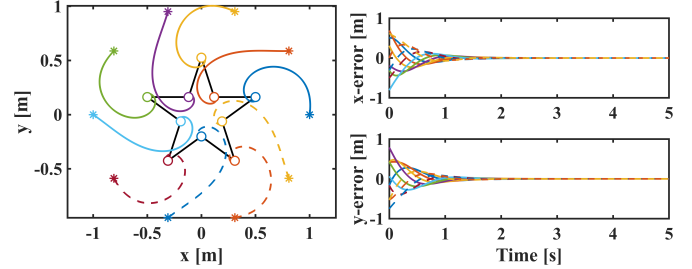


Fig. 5: In the case of a **directed graph**, our formation strategy remains effective, and the desired positions are still achieved.

the distance flown and estimating the overall displacement relative to the initial position. However, it is imperative to acknowledge that the reliance of the Flow Deck on relative displacement measurements may give rise to cumulative errors over time, leading to deviations between the reported position and the actual position of the Crazyflies.

During our experiments, we assign each Crazyflie a predetermined initial position and accumulate the measurements provided by the Flow Deck in relation to said initial position. This enables the conversion from relative positions to global positions. The interaction between the PC and the Crazyflies is facilitated through the CrazyRadio PA, a wireless communication module developed by Bitcraze Inc. The communication rate is set at 50 Hz, representing the highest achievable communication frequency while simultaneously managing all Crazyflies.

The overall experimental setup is illustrated in Fig. 6. The

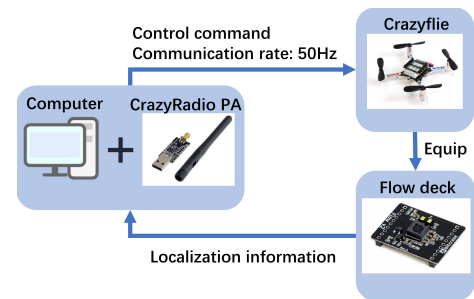


Fig. 6: **Crazyflies** equipped with **Flow Decks** are utilized as the agents to conduct the experiments. **CrazyRadio PA** communicates with the PC at a rate of **50 Hz**.

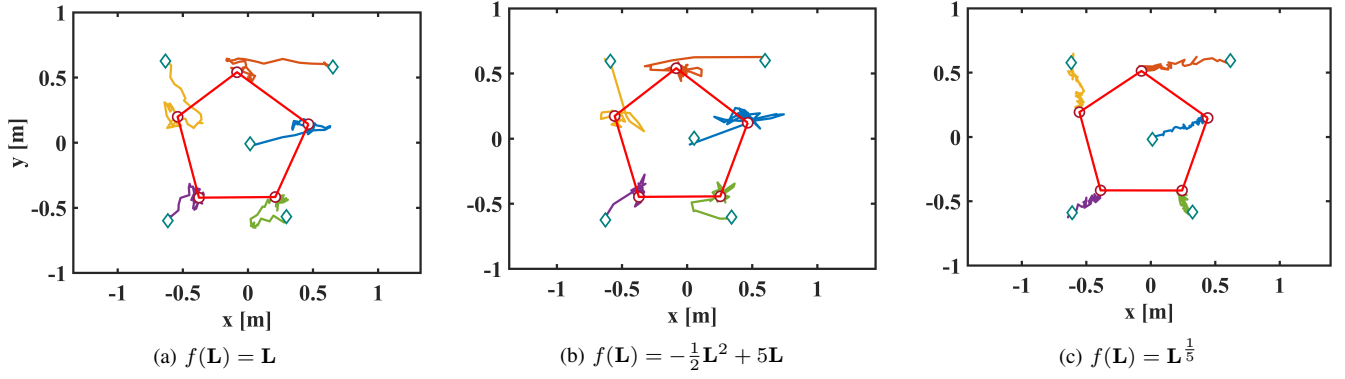


Fig. 7: Experiments: the trajectory of the 5 **Crazyflies** for each Laplacian function. The edges of the network topology are represented by **solid red** lines, the initial/final positions are indicated by **diamonds/circles**, and the **colored lines** represent the trajectories produced by the Crazyflies. We can conclude that our formation control strategy **remains effective** in practical scenarios.

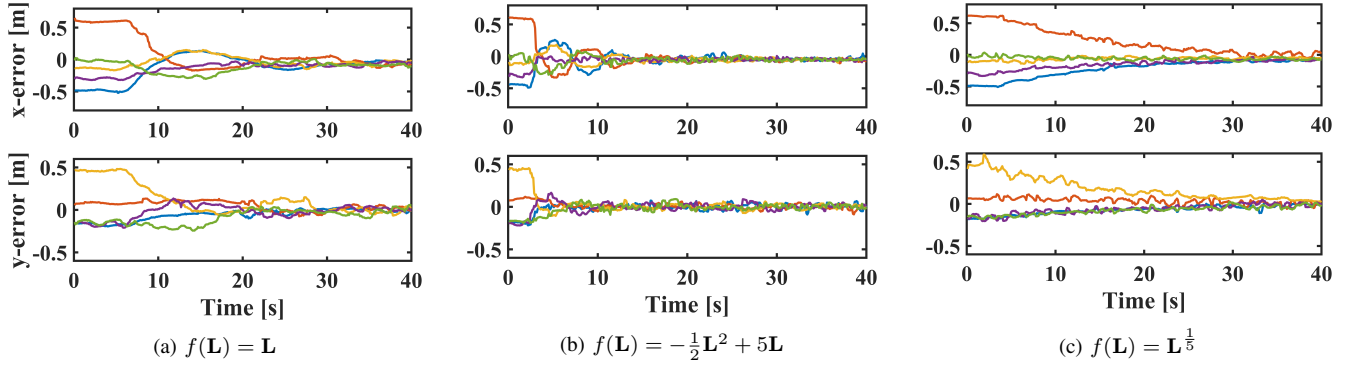


Fig. 8: Experiments: the time evolution of the formation errors. The **colored lines** represent the formation errors resulted by each Crazyfly, where Fig. 8b illustrates the fastest convergence speed in x - and y - directions.

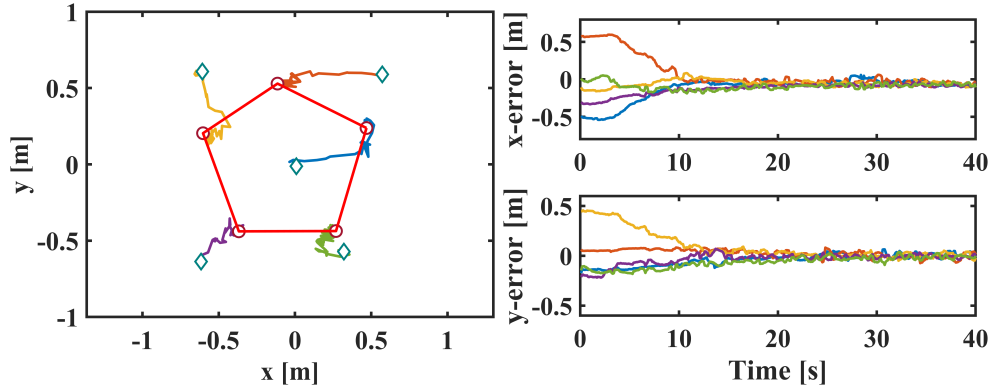


Fig. 9: Experiments: the **combinatorial** Laplacian function. Compared with the simulation results, it indicates that the **combinatorial** Laplacian function still enables the quadrotors to converge to the desired formation with **controllable speed**.

formation strategy (3) is computed on a computer equipped with an Intel Core i7 CPU operating at 1.8 GHz and 8 GB of RAM. This computer establishes a connection with each Crazyfly via the CrazyRadio PAs to transmit control commands. The Crazyflies, in turn, relay feedback regarding their localization information, which is measured by the Flow Deck, back to the computer. This information plays a crucial role in determining the subsequent control commands.

B. Comparative Experiments

To maintain consistency with the simulation setup employed in Sec. IV, we carry out comparative experiments

using the same set of four Laplacian functions: $f(\mathbf{L}) = \mathbf{L}$, $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + 5\mathbf{L}$, $f(\mathbf{L}) = \mathbf{L}^{1/5}$, and $f(\mathbf{L}) = (6\mathbf{L} - \frac{1}{2}\mathbf{L}^2 + \mathbf{L}^{1/5})/5$ (where the combinatorial Laplacian Function divides the output by 5 to prevent large oscillations). Moreover, in the experiment, we utilize 5 Crazyflies to achieve a regular pentagon configuration, specifically the shape formed by the five outer vertices of the expected pentagram mentioned in Sec. IV-B. It is important to note that due to the performance limitations of the Crazyfly, the actual flight process may deviate from the behavior observed in the simulation.

The experimental findings are depicted in Figs. 7-9. Fig. 7 showcases the trajectories followed by the Crazyflies under

the influence of the first three Laplacian functions, while Fig. 8 portrays the corresponding formation errors in the x -axis and y -axis. On the other hand, Fig. 9 presents the same information as previously described, under the control of the combinatorial Laplacian function. At the beginning of the experiments, all Crazyflies ascend to a height of 0.7 m before commencing their movement while continuously receiving control instructions from the PC.

In Fig. 7, it is evident that the desired formation can be achieved when each of the three functions is employed. The second-smallest eigenvalues, denoted as $\bar{\lambda}_2$, are $\bar{\lambda}_2 \simeq 1.3820$, $-\frac{1}{2}\bar{\lambda}_2^2 + 5\bar{\lambda}_2 \simeq 5.9550$, and $\bar{\lambda}_2^{1/5} \simeq 1.0668$, respectively, and are associated with the convergence speed of each formation process. Therefore, distinct differences in their convergence behavior become apparent, which is clearly demonstrated in Fig. 8. By virtue of the magnitude of the eigenvalues, the order of the convergence speed can be ranked as Fig. 8b, Fig. 8a, Fig. 8c, illustrating the varying convergence speed of the formations, consistent with the simulation results.

Fig. 9 displays the convergence behavior of the combinatorial Laplacian function $f(\mathbf{L}) = (6\mathbf{L} - \frac{1}{2}\mathbf{L}^2 + \mathbf{L}^{1/5})/5$, which possesses a second-smallest eigenvalue of $\bar{\lambda}_2 \simeq 1.6808$. Through the division of the combinatorial Laplacian function by a factor of 5, not only can excessive oscillations be prevented, but also the magnitude of the second-smallest eigenvalue can be effectively reduced. This observation highlights that the ability to manipulate the convergence speed of the formation process can be readily accomplished by adjusting a designated parameter in real-world scenarios.

VI. CONCLUSION

This study introduces an innovative approach to enhance the convergence speed of MAS in formation control through the utilization of Laplacian functions. By leveraging the Laplacian matrix of the communication graph, we map eigenvalues to desired positions, thereby improving the convergence speed during the formation process. The effectiveness and practical feasibility of the proposed strategy have been verified through both simulations and experiments with quadrotors. Further investigations could explore the performance of the approach in larger-scale formations, obstacle-cluttered and dynamic environments.

REFERENCES

- [1] J. Xie and C. C. Liu, "Multi-agent systems and their applications," *J. Int. Counc. Electr. Eng.*, vol. 7, no. 1, pp. 188–197, 2017.
- [2] L. Cruz-Piris, D. Rivera, S. Fernandez, and I. Marsa-Maestre, "Optimized sensor network and multi-agent decision support for smart traffic light management," *Sensors*, vol. 18, no. 2, p. 435, 2018.
- [3] Y. Wu, Y. Gao, and W. Song, "Multi-robot task assignment algorithm for medical service system," in *IEEE Int. Conf. Robot. Biomim. (ROBIO)*, pp. 1204–1209, 2022.
- [4] M. Aranda, G. Lopez Nicolas, C. Sagues, and M. M. Zavlanos, "Distributed formation stabilization using relative position measurements in local coordinates," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3925–3935, 2016.
- [5] M. H. Trinh, S. Zhao, Z. Sun, D. Zelazo, B. D. Anderson, and H. S. Ahn, "Bearing-based formation control of a group of agents with leader-first follower structure," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 598–613, 2018.
- [6] D. Zhang, Y. Tang, W. Zhang, and X. Wu, "Hierarchical design for position-based formation control of rotorcraft-like aerial vehicles," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 4, pp. 1789–1800, 2020.
- [7] F. Mehdifar, C. P. Bechlioulis, F. Hashemzadeh, and M. Baradarannia, "Prescribed performance distance-based formation control of multi-agent systems," *Automatica*, vol. 119, p. 109086, 2020.
- [8] Y. H. Choi and D. Kim, "Distance-based formation control with goal assignment for global asymptotic stability of multi-robot systems," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2020–2027, 2021.
- [9] I. Buckley and M. Egerstedt, "Infinitesimal shape-similarity for characterization and control of bearing-only multirobot formations," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1921–1935, 2021.
- [10] Q. Chen, G. Jia, J. Lin, Y. Lu, and P. Wang, "A formation control method combining the global and local informations," in *IEEE Int. Conf. Robot. Biomim. (ROBIO)*, pp. 1694–1699, 2018.
- [11] B. H. Lee and H. S. Ahn, "Distributed formation control via global orientation estimation," *Automatica*, vol. 73, pp. 125–129, 2016.
- [12] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2590–2603, 2020.
- [13] S. Guler, M. Abdelkader, and J. S. Shamma, "Peer-to-peer relative localization of aerial robots with ultrawideband sensors," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 5, pp. 1981–1996, 2021.
- [14] J. Guo, J. Qi, M. Wang, C. Wu, and G. Yang, "Collision-free distributed control for multiple quadrotors in cluttered environments with static and dynamic obstacles," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1501–1508, 2023.
- [15] J. Hu, H. Zhang, L. Liu, X. Zhu, C. Zhao, and Q. Pan, "Convergent multiagent formation control with collision avoidance," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1805–1818, 2020.
- [16] M. A. Kamel, X. Yu, and Y. Zhang, "Fault-tolerant cooperative control design of multiple wheeled mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 2, pp. 756–764, 2018.
- [17] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 340–348, 2015.
- [18] T. M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Persistently excited adaptive relative localization and time-varying formation of robot swarms," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 553–560, 2020.
- [19] A. Petrillo, A. Pesce, and S. Santini, "A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1134–1149, 2021.
- [20] D. Van Vu, M. H. Trinh, P. D. Nguyen, and H. S. Ahn, "Distance-based formation control with bounded disturbances," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 451–456, 2021.
- [21] X. He and Q. Wang, "Distributed finite-time leaderless consensus control for double-integrator multi-agent systems with external disturbances," *Appl. Math. Comput.*, vol. 295, pp. 65–76, 2017.
- [22] A. K. Schug, A. Eichler, and H. Werner, "A decentralized asymmetric weighting approach for improved convergence of multi-agent systems with undirected interaction," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 8317–8322, 2014.
- [23] F. Morbidi, "Functions of the laplacian matrix with application to distributed formation control," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 3, pp. 1459–1467, 2022.
- [24] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [25] W. Hönig and K. McGuire, "Crazyswarm2: A ros 2 testbed for aerial robot teams," <https://github.com/IMRCLab/crazyswarm2>.