Functions of the Laplacian matrix with application to distributed formation control

Fabio Morbidi

Functions of the Laplacian matrix with application to distributed formation control

Fabio Morbidi, *Senior Member, IEEE*

***Abstract*— In this paper, we study a class of matrix functions of the combinatorial Laplacian that preserve its structure, i.e. that define matrices which are positive semidefinite, and which have zero row-sum and non-positive off-diagonal entries. This formulation has the merit of presenting different incarnations of the Laplacian matrix appeared in the recent literature, in a unified framework. For the first time, we apply this family of Laplacian functions to consensus theory, and we show that they leave the agreement value unchanged and offer distinctive advantages in terms of performance and design flexibility. The theory is illustrated via worked examples and numerical experiments featuring four representative Laplacian functions in a shape-based distributed formation control strategy for single-integrator robots.**

***Index Terms*— Combinatorial Laplacian, Network systems, Consensus protocol, Matrix functions.**

I. Introduction

A. Related work

THE combinatorial Laplacian is ubiquitous in network science [1], and over the last decade we have witnessed the emergence of several variants to describe distributed dynamic processes. In particular, the recent advent of graph signal processing [2], [3], graph neural networks [4], [5], and the growing popularity of network systems [6], [7], has had a catalytic effect on the research in this field.

An interesting variant of combinatorial Laplacian is the *deformed Laplacian*, which has found applications in multi-agent systems theory [8], semi-supervised learning [9], and in the design of new centrality measures for undirected and directed networks [10]–[13]. Along the same lines as [8], the *parametric Laplacian* has been introduced in [14]. The *Bethe-Hessian matrix*, the reversal of the deformed Laplacian in the case of undirected graphs [11], has been used for spectral clustering [15] and community detection in sparse heterogeneous networks [16] (see also [17]). A different definition of deformed Laplacian, which encompasses several Laplacian-like matrices available in the literature (*connection*, *magnetic* [18], *signed* and *dilation*), was proposed in [19]. Notably, the dilation Laplacian has been shown to be useful for spectral ranking in directed graphs.

The p-*Laplacian*, a nonlinear generalization of the combinatorial Laplacian, has recently attracted the attention of the machine learning community. The p-Laplacian reduces to the standard Laplacian for $p = 2$, and it has been successfully applied to solve two-class [20] and multi-class [21] clustering problems.

The *exponential* of the negated Laplacian is common in the study of classical transport on lattices and networks, and it is often referred to as the "heat kernel" because of its interpretation as a diffusion process related to the heat equation [22]. It also appears in discrete-time agreement protocols [6].

Laplacian powers have been widely explored in the literature as well. In particular, the bi-Laplacian (or Laplacian squared) has been considered in [23] for image colorization. The bi-Laplacian corresponds to the discretization of the bi-harmonic PDE equation with Neumann boundary condition, which together with its numerical schemes is widely studied and applied in problems of linear elasticity [24, Ch. 11], data interpolation, and computer vision (image inpainting). For non-integer powers in the $(0,\, 1]$ interval, we obtain the so-called *fractional Laplacian*, which has been used to study random walks on networks, among other applications [22], [25]. In [26], the authors have proposed a generalization of the PageRank algorithm for semi-supervised learning based on a (non-necessary integer) power of the Laplacian matrix. Finally, it has been recently shown that graph filters can be represented by *matrix polynomials* of the combinatorial Laplacian [27], [28].

B. Original contributions, organization and notation

The original contribution of this paper is twofold. First of all, following [22], we study *matrix functions of the Laplacian* of an undirected graph, that preserve its main algebraic properties, i.e. that give rise to matrices which are positive semidefinite and which have zero row-sum and non-positive off-diagonal entries. These Laplacian functions promote the emergence of non-local network correlations: in fact, thanks to their ability to reorganize the information exchanges between the nodes of a graph, long-range interactions are enabled. This general formulation allows us to bring together a number of results scattered across different areas and shed new light on them. Second, we revisit the classical continuous-time consensus protocol by replacing the combinatorial Laplacian with our family of Laplacian functions. These functions leave the agreement value *unchanged* and offer some practical advantages. In fact, in many situations, by simply tuning a scalar parameter, the user can seamlessly modify the dynamic behavior of the multi-agent system, for example, to guarantee faster convergence towards consensus or to adapt to variable external conditions. This generalizes existing work in the literature, where integer powers of the Laplacian have been

The author is with the MIS Laboratory, Université de Picardie Jules Verne, 33 rue Saint-Leu, 80039 Amiens, France. Email: fabio.morbidi@u-picardie.fr

considered for fast consensus seeking (by introducing a "multi-hop relay" interpretation [29]), and for dynamic consensus over wireless sensor networks [30]. The price to pay for the increased design flexibility, is that the interaction graph associated with our Laplacian functions is generally more dense than that of the original Laplacian. A simple yet effective approximation of the Laplacian functions is thus proposed to ensure a distributed implementation of the new coordination protocols. The theoretical findings are illustrated via worked examples and validated via numerical simulations in which single-integrator robots run a consensus-based formation control algorithm.

The rest of this paper is organized as follows. In Sect. II, we briefly recall the definition of two matrix functions, and some basic notions of algebraic graph theory. In Sect. III, we study a family of functions of the combinatorial Laplacian that retain its special structure, and in Sect. IV we use them in a novel shape-based distributed formation control protocol, which is validated via extensive numerical experiments. Finally, in Sect. V, the main contributions of the paper are summarized and some possible avenues for future research are outlined.

Notation: Throughout this paper, $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$ denote the sets of positive and non-negative integers, respectively, $\mathbf{I}_n$ the $n \times n$ identity matrix, $\mathbf{0}_n$ the $n \times n$ matrix of zeros, $\mathbb{1}$ a column vector of n ones, and $\otimes$ the Kronecker product. For a complex number $z \in \mathbb{C}$, we use $\mathrm{Im}(z)$ to denote its imaginary part, and $\arg(z)$ its argument.

II. Background Material

In this section, we briefly recall the definitions of matrix logarithm and matrix pth root, and review some elementary notions of graph theory, which serve as the foundation of the next sections.

A. Matrix logarithm

A logarithm of $\mathbf{A} \in \mathbb{C}^{n\times n}$ is any matrix $\mathbf{X}$ such that $e^{\mathbf{X}} = \mathbf{A}$. Any nonsingular matrix $\mathbf{A}$ has infinitely many logarithms [31, Th. 1.27]. In this paper, we assume that $\mathbf{A} \in \mathbb{C}^{n\times n}$ has no eigenvalues on the closed negative real axis, and $\log(\cdot)$ always denotes the *principal logarithm*, which is the unique logarithm all of whose eigenvalues lie in the strip $\{z \in \mathbb{C} : -\pi < \mathrm{Im}(z) < \pi\}$ [31, Th. 1.31]. If $\mathbf{A}$ is real, then its principal logarithm is real.

B. Matrix pth root

Let $p \geq 2$ be an integer. Matrix $\mathbf{X}$ is a pth root of $\mathbf{A} \in \mathbb{C}^{n\times n}$ if $\mathbf{X}^p = \mathbf{A}$. We recall the following result on existence of pth roots [31, Th. 7.2].

Theorem 1 (Principal pth root) *Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ have no eigenvalues on the closed negative real axis. There is a unique pth root $\mathbf{X}$ of $\mathbf{A}$ all of whose eigenvalues lie in the sector $\{z \in \mathbb{C} : -\pi/p < \arg(z) < \pi/p\}$, and it is a primary matrix function[1] of $\mathbf{A}$. We refer to $\mathbf{X}$ as the principal pth root of $\mathbf{A}$ and write $\mathbf{X} = \mathbf{A}^{1/p}$. If $\mathbf{A}$ is real, then $\mathbf{A}^{1/p}$ is real.* ◇

[1]For more details on *primary* and *nonprimary matrix functions*, the reader is referred to [31, Sect. 1.4].

For $p = 2$, we have the following extension of Theorem 1 which allows $\mathbf{A}$ to be *singular* (see Problem 1.27 in [31]).

Proposition 1 (Square root of a singular matrix) *Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ have no eigenvalues on the closed negative real axis, except possibly for a semisimple[2] zero eigenvalue. There is a unique square root $\mathbf{X}$ of $\mathbf{A}$ that is a primary matrix function of $\mathbf{A}$ and whose nonzero eigenvalues lie in the open right half-plane. If $\mathbf{A}$ is real, then $\mathbf{X}$ is real.* ◇

There exist four main numerical algorithms for the computation of the matrix pth root: Schur, Newton, and Schur-Newton algorithms, and matrix sign method [31, Sect. 7]. The Schur algorithm assumes that $\mathbf{A} \in \mathbb{R}^{n\times n}$ is nonsingular, whereas the other methods assume that $\mathbf{A}$ has no eigenvalues on the closed negative real axis.

C. Graph theory

Let $\mathcal{G} = (V, E)$ be a connected graph, where $V = \{1, \ldots, n\}$ is the set of nodes and E is the set of edges. In this paper, all $\mathcal{G}$'s are simple, i.e. they are unweighted, undirected graphs containing no self-loops or multiple edges [32].

Definition 1 (Adjacency and Laplacian matrix)

- *The adjacency matrix $\mathbf{A}$ of graph $\mathcal{G}$ is an $n \times n$ matrix defined as $[\mathbf{A}]_{ij} = 1$ if $\{i, j\} \in E$ and $[\mathbf{A}]_{ij} = 0$ otherwise.*
- *The Laplacian matrix of graph $\mathcal{G}$ is an $n \times n$ symmetric positive semidefinite matrix defined as $\mathbf{L}(\mathcal{G}) = \mathbf{D} - \mathbf{A}$ where $\mathbf{D} = \mathrm{diag}(d_1, d_2, \ldots, d_n) = \mathrm{diag}(\mathbf{A}\mathbb{1})$ is the* degree matrix. ◇

It is easy to verify that $\mathbf{L}\mathbb{1} = \mathbf{0}$, i.e. the row-sums of $\mathbf{L}$ are equal to zero, and that the off-diagonal entries of $\mathbf{L}$ are non-positive. Since $\mathbf{L}$ is a symmetric matrix, we have $\mathbf{L}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, $i \in \{1, 2, \ldots, n\}$, where the eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of $\mathbf{L}$ are arranged in increasing order, i.e. $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$, and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ are the associated orthonormal eigenvectors, with $\mathbf{v}_1 = \mathbb{1}/\sqrt{n}$.

III. Functions of the Laplacian Matrix

In this section, we study matrix functions $f(\mathbf{L})$ of the combinatorial Laplacian $\mathbf{L}$. These functions define matrices which incorporate the network topology information and which promote non-local interactions between the nodes of graph $\mathcal{G}$. We will exploit them in Sect. IV, to design a new distributed formation control strategy for single-integrator robots. Our exposition here follows the outline of [22, Ch. 1]: however, several new examples and results are provided in Sect. III-A and Sect. III-B, and the notation has been modified to conform to that conventionally used in systems theory. For the sake of simplicity, we henceforth use $f_{ij}(\mathbf{L})$ to denote the (i, j) entry of matrix $f(\mathbf{L})$.

Given a well-defined function $f(x) : \mathbb{R} \to \mathbb{R}$, the matrix $f(\mathbf{L})$ can be obtained using the series expansion $f(x) = \sum_{\ell=0}^{\infty} c_\ell\, x^\ell$ where c_ℓ is the real coefficient of

[2]An eigenvalue is called *semisimple* if its algebraic multiplicity equals its geometric multiplicity.

the ℓth term, or using the spectral decomposition of $\mathbf{L}$. The latter option gives

$$f(\mathbf{L}) \;=\; \sum_{i=1}^{n} f(\lambda_i)\, \mathbf{v}_i \mathbf{v}_i^T. \tag{1}$$

Equation (1) shows that to find $f(\mathbf{L})$, one should calculate the spectrum $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of $\mathbf{L}$ and then compute $\{f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_n)\}$: the eigenvectors $f(\mathbf{L})$ remain the same as those of $\mathbf{L}$. From (1), we also notice that matrix $f(\mathbf{L})$ is symmetric by construction. For further details on the theory of functions of matrices, see [31] and [33, Ch. 9].

Note that while equation (1) allows to calculate general functions of $\mathbf{L}$, we are interested here in functions which *preserve* the special structure of the Laplacian matrix (as reported in Sect. II-C). In order to retain these desirable properties, the matrix $f(\mathbf{L})$ should satisfy the following three conditions:

- **Condition I**: $f(\mathbf{L})$ is positive semidefinite, i.e. the eigenvalues of $f(\mathbf{L})$ are restricted to be positive or zero.
- **Condition II**: the entries $f_{ij}(\mathbf{L})$ satisfy $\sum_{j=1}^{n} f_{ij}(\mathbf{L}) = 0$ for $i \in \{1, 2, \ldots, n\}$, or equivalently $f(\mathbf{L})\mathbb{1} = \mathbf{0}$, i.e. each row-sum is equal to zero.
- **Condition III**: the off-diagonal entries of $f(\mathbf{L})$ are non-positive and they are not allowed to be all simultaneously zero. Hence, owing to Condition II, the diagonal elements of $f(\mathbf{L})$ are all strictly positive.

Condition I is guaranteed if $f(x) \geq 0$ for $x \geq 0$. Then $f(\lambda_i) \geq 0$ for all i, and the eigenvalues of $f(\mathbf{L})$ can be arranged in increasing order as those of $\mathbf{L}$. On the other hand, by using equation (1), it is easy to verify that Condition II is fulfilled if $f(0) = 0$. However, these two conditions on the function f do not guarantee that the off-diagonal entries of $f(\mathbf{L})$ are non-positive, as required by Condition III. For example, if we take $\mathcal{G} = S_n$ (the star graph with n nodes), we have that

$$\mathbf{L}(\mathcal{G}) \;=\; \begin{bmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

The function $f(x) = x^2$ is non-negative for $x \geq 0$ and $f(0) = 0$. Thereby, $f(\mathbf{L}) = \mathbf{L}^2$, a.k.a. bi-Laplacian [23], satisfies Conditions I and II, but the structure imposed by Condition III is not preserved: in fact, $\mathbf{L}^2$ has strictly positive (unitary) off-diagonal entries. On the contrary, the function $f(x) = \log(x+1)$ is non-negative for $x \geq 0$, $f(0) = 0$, and $f(\mathbf{L}) = \log(\mathbf{L} + \mathbf{I}_n)$ has negative off-diagonal entries. Hence, $\log(\mathbf{L} + \mathbf{I}_n)$ is an admissible Laplacian function.

The following notion of *completely monotonic function* [34] plays an important role in the characterization of the admissible functions $f(\mathbf{L})$.

Definition 2 (Completely monotonic function) *A function $g(x)$ defined on $0 < x < \infty$ is said to be completely monotonic, if it possesses derivatives $g^{(m)}(x) = \frac{d^m}{dx^m} g(x)$ for all $m \in \mathbb{Z}_{\geq 0}$ and if $(-1)^m\, g^{(m)}(x) \geq 0$ for all $x > 0$.* $\diamond$

In [22, Sect. 1.4.2], the authors have proved that the functions $f(\mathbf{L})$ that satisfy the necessary Condition III, can be constructed via the scalar functions

$$f(x) \;=\; H(0) \;-\; H(-x), \quad 0 \leq x < \infty, \tag{2}$$

where $H(x)$ denotes a primitive of the auxiliary scalar function $\mathrm{h}(x)$, i.e. $\mathrm{h}(x) = \frac{d}{dx} H(x)$. The function $f(x)$ can be expressed in terms of a function $g(x) = \mathrm{h}(-x)$ defined on $0 \leq x < \infty$, which satisfies the following conditions:

$$\frac{d}{dx} f(x) \;=\; g(x) > 0, \quad 0 \leq x < \infty, \tag{3}$$

and

$$(-1)^m\, g^{(m)}(x) \geq 0, \quad 0 \leq x < \infty, \quad m \in \mathbb{Z}_{>0}. \tag{4}$$

According to Definition 2, a function $g(x)$ that fulfills conditions (3) and (4) is a completely monotonic function. By (3), $g(x)$ is strictly positive for $x \geq 0$, including, in particular, the entire spectral interval $0 \leq x \leq \lambda_n$ of the Laplacian $\mathbf{L}$, and the function $f(x)$ monotonically increases on its interval of definition $0 \leq x < \infty$. This monotonicity property implies that the algebraic multiplicity of the eigenvalues of $\mathbf{L}$ is maintained in $f(\mathbf{L})$ (in fact, $f(\lambda_i) > f(\lambda_j)$ for $\lambda_i > \lambda_j$). The function $f(x)$ is also given by the following integral

$$f(x) \;=\; \int_0^x g(y)\, dy, \quad 0 \leq x < \infty, \tag{5}$$

i.e. it is the primitive of $g(x)$ with $f(0) = 0$ and with $f(x) > 0$ for $x > 0$. In [22, Sect. 1.4.3], the authors have shown that relations (2)-(5) are indeed *sufficient* to generate scalar $\mathcal{C}^\infty$ functions $f(x)$ which define admissible Laplacian functions $f(\mathbf{L})$ satisfying Conditions I-III. Note that $f(x)$ in (5) has the general structure given in equation (2): hence, the admissible Laplacian functions can be represented as

$$f(\mathbf{L}) \;=\; H(0)\mathbf{I}_n - H(-\mathbf{L}) = \sum_{i=2}^{n} (H(0) - H(-\lambda_i))\, \mathbf{v}_i \mathbf{v}_i^T.$$

There is no term $\mathbf{v}_1 \mathbf{v}_1^T$ since $f(\lambda_1) = 0$ (recall equation (1)). The eigenvector of $f(\mathbf{L})$ associated with the zero eigenvalue is $\mathbb{1}/\sqrt{n}$, since $\mathbf{L}$ and $f(\mathbf{L})$ have the same set of eigenvectors.

A. Examples of $f(\mathbf{L})$

There exist several classes of completely monotonic functions which in conjunction with equation (5) allow to construct admissible Laplacian functions satisfying Conditions I-III. We focus here on six examples (half of which are not reported in [22]), that are of special interest.

1) The function $g(x) = (\lambda - x)^m$ for $0 \leq x < \lambda$ with $m \in \mathbb{Z}_{\geq 0}$ and $0 < \lambda_n < \lambda$, satisfies conditions (3) and (4). From equation (5), we obtain

$$\begin{aligned} f(x) &= \int_0^x (\lambda - y)^m\, dy \\ &= \frac{1}{m+1}\big(\lambda^{m+1} - (\lambda - x)^{m+1}\big). \end{aligned}$$

This yields the matrix function:

$$f(\mathbf{L}) = \frac{1}{m+1}\big(\lambda^{m+1}\,\mathbf{I}_n - (\lambda\mathbf{I}_n - \mathbf{L})^{m+1}\big) = \sum_{i=1}^{n} \frac{1}{m+1}\big(\lambda^{m+1} - (\lambda-\lambda_i)^{m+1}\big)\,\mathbf{v}_i\mathbf{v}_i^T.$$

Note that in the trivial case of $m = 0$, the function $g(x) = 1$, and $f(x)$ is the identity function, which yields $f(\mathbf{L}) = \mathbf{L}$.

2) Consider the function

$$g(x) = \frac{a}{(ax+b)^m}, \quad a > 0,\ b \geq 0,\ m \in \mathbb{Z}_{>0}.$$

Let us start with the case of $m = 1$. From equation (5), we obtain $f(x) = \log(ax+b)$ for $x \geq 0$. However, the constraint $f(0) = 0$, requires $f(x) = \log(ax+b) - \log b$. Hence, we have

$$f(\mathbf{L}) = \log(a\mathbf{L} + b\mathbf{I}_n) - \log(b\mathbf{I}_n).$$

Let us now consider the general case of $m \in \{2, 3, \ldots\}$. Following the same procedure as above, we obtain:

$$f(x) = -\frac{1}{(m-1)(ax+b)^{m-1}},$$

which we modify into

$$f(x) = \frac{1}{m-1}\left[-\frac{1}{(ax+b)^{m-1}} + \frac{1}{b^{m-1}}\right],$$

to guarantee that $f(0) = 0$. Hence, we end up with

$$f(\mathbf{L}) = \frac{1}{m-1}\left[-(a\mathbf{L} + b\mathbf{I}_n)^{1-m} + b^{1-m}\,\mathbf{I}_n\right].$$

3) The *exponential* $g(x) = ae^{-ax}$ with $a > 0$, is a completely monotonic function. The corresponding function that retains the Laplacian structure is $f(x) = 1 - e^{-ax}$ and thus $f(\mathbf{L}) = \mathbf{I}_n - e^{-a\mathbf{L}}$. Matrix $e^{-a\mathbf{L}}$ is doubly stochastic and it has appeared under different forms in consensus theory [6]. Functions like $e^{-a\mathbf{L}}$ and the regularized Laplacian $(\mathbf{I}_n + a\mathbf{L})^{-1}$, have been also used as kernels to compute similarities between the nodes of an undirected graph [35].

4) The function

$$g(x) = \frac{1}{b - ce^{-ax}}, \quad a > 0,\ b > c > 0,$$

defined on $0 < x < \infty$, is completely monotonic [34]. From (5), we obtain

$$f(x) = \frac{1}{ab}\big[\log(b - ce^{-ax}) + ax\big].$$

To guarantee $f(0) = 0$, we set

$$f(x) = \frac{1}{ab}\big[\log(b - ce^{-ax}) + ax - \log(b-c)\big],$$

yielding

$$f(\mathbf{L}) = \frac{1}{ab}\big[\log(b\mathbf{I}_n - ce^{-a\mathbf{L}}) + a\mathbf{L} - \log(b-c)\mathbf{I}_n\big].$$

5) The function $g(x) = (\beta+1)\,x^{\beta}$ with $\beta \leq 0$ is a completely monotonic function. From equation (5), we obtain $f(x) = x^{\beta+1}$. The condition $f(0) = 0$, requires that $-1 < \beta \leq 0$. Therefore, the function $f(x) = x^{\gamma}$ with $\gamma \in \mathbb{R}$ such that $0 < \gamma \leq 1$, preserves the Laplacian structure imposed by Conditions I-III. We thus obtain the so-called *fractional Laplacian*, $f(\mathbf{L}) = \mathbf{L}^{\gamma}$, which has been studied in the theory of random walks, and in diffusive and quantum transport on networks [22, Ch. 2]. Note that integer powers $\gamma \in \{2, 3, \ldots\}$, fail to satisfy Condition III.

6) Node-invariant graph filters are linear graph-signal operators of the form [27], [36]:

$$f(\mathbf{L}) = \sum_{j=0}^{m-1} b_j \mathbf{L}^j, \tag{6}$$

where $b_0, b_1, \ldots, b_{m-1}$ are real coefficients, i.e. graph filters are *polynomials* of the Laplacian $\mathbf{L}$ (or of any other $n \times n$ matrix, such as the adjacency matrix $\mathbf{A}$, whose sparsity pattern captures the local structure of graph $\mathcal{G}$). A graph filter can be equivalently defined as

$$f(\mathbf{L}) = a_0 \prod_{j=1}^{m-1} (\mathbf{L} - a_j\mathbf{I}_n),$$

where $a_0, a_1, \ldots, a_{m-1}$ are real coefficients, which also gives rise to a polynomial on $\mathbf{L}$ of degree $m-1$. From (6), we can see that Conditions I and II are fulfilled if $b_0 = 0$ and $b_j \geq 0$ for $j \in \{1, 2, \ldots, m-1\}$, but Condition III is not satisfied, in general. Graph filters based on Chebyshev and Cayley polynomials have also been recently introduced to process graph-structured data [5], [28].

B. General properties of $f(\mathbf{L})$

In the previous section, we have identified the family of functions that preserve the structure of the combinatorial Laplacian. We now briefly discuss some general properties of the matrix $f(\mathbf{L})$, which will be used in Sect. IV.

1) Generalized degree: By construction, the diagonal entries of $f(\mathbf{L})$ are positive, and analogously to the Laplacian $\mathbf{L}$, we can refer to $d_i = f_{ii}(\mathbf{L})$, $i \in \{1, 2, \ldots, n\}$, as the *generalized degree* associated with the function f [22, Sect. 1.5.1]. The average of the generalized degree is defined as

$$\frac{1}{n}\sum_{i=1}^{n} d_i = \frac{1}{n}\,\text{trace}(f(\mathbf{L})) = \frac{1}{n}\sum_{i=1}^{n} f(\lambda_i).$$

In the case of the fractional Laplacian, it is called the *average fractional degree*. Note that d_i does not only capture the local information (on its nearest neighbors) of node i, but also incorporates knowledge at the level of the whole network. This "non-locality" property is further explored below.

2) Laplacian functions for regular graphs: To gain some insight into the structure of the Laplacian functions $f(\mathbf{L})$, it is worth focusing on the special case of *regular graphs*. In a regular graph, each node has the same degree k, and the Laplacian takes the simple form $\mathbf{L} = k\mathbf{I}_n - \mathbf{A}$. By using the

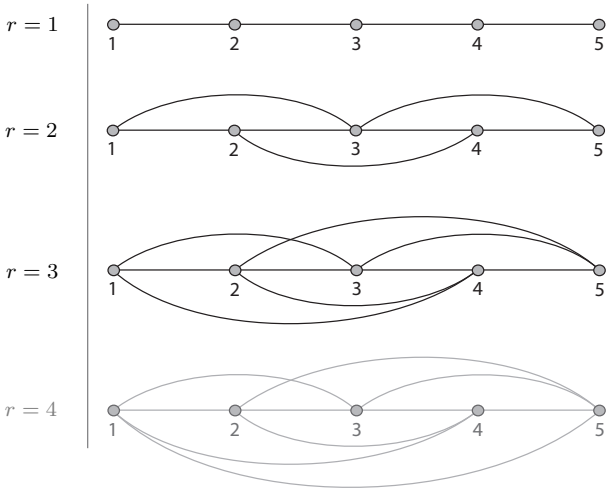


Fig. 1. Graphs $\mathcal{B}_5^r$ for $r \in \{1, 2, 3\}$. Note that $\mathcal{B}_5^4$ (light gray) is the complete graph with 5 nodes.

series expansion presented at the beginning of Sect. III, we can express $f(\mathbf{L})$ as follows [22, Sect. 1.5.2]:

$$f(\mathbf{L}) = \sum_{\ell=1}^{\infty} c_\ell (k\mathbf{I}_n - \mathbf{A})^\ell = \sum_{\ell=1}^{\infty} \sum_{h=0}^{\ell} c_\ell \binom{\ell}{h} k^{\ell-h} (-1)^h \mathbf{A}^h. \tag{7}$$

Equation (7) unveils a connection between $f(\mathbf{L})$ and the integer powers of the adjacency matrix $\mathbf{A}$. Recall that the (i, j) entry of $\mathbf{A}^h$ for $h \in \mathbb{Z}_{>0}$, is the number of all the possible paths connecting node i to node j with h edges, whereas the diagonal entry (i, i) of $\mathbf{A}^h$ is the number of closed paths with h edges which start and end at the same node i [32]. Therefore, equation (7) shows how the function f changes the local character of the Laplacian $\mathbf{L}$ and makes it a *long-range operator*: matrix $f(\mathbf{L})$ is thus well suited to define dynamical processes with non-local interactions on networks.

3) Exponential decay of functions of banded matrices: A *band matrix* is a sparse matrix whose nonzero entries are limited to a diagonal band, including the main diagonal and (zero or) more diagonals on either side. Given a matrix $\mathbf{B} \in \mathbb{C}^{n\times n}$, if all entries of $\mathbf{B}$ are zero outside a diagonally bordered band whose range is determined by $r_1, r_2 \in \mathbb{Z}_{\geq 0}$ (i.e. $[\mathbf{B}]_{ij} = 0$ if $j < i - r_1$ or $i + r_2 < j$), then r_1 and r_2 are called the *lower bandwidth* and *upper bandwidth* of $\mathbf{B}$, respectively. The *bandwidth* of $\mathbf{B}$ is the maximum of r_1 and r_2, i.e. it is the integer r such that $[\mathbf{B}]_{ij} = 0$ if $|i - j| > r$. For example, a band matrix with $r_1 = r_2 = 0$ ($r_1 = r_2 = 1$) is a diagonal (tridiagonal) matrix.

For the entries of functions of Hermitian band matrices, we have the following exponentially decaying bound (further decay results applying to general matrices, are available in [37]):

Theorem 2 (Benzi & Golub [38]) *Let $\mathbf{B} \in \mathbb{C}^{n\times n}$ be Hermitian and of bandwidth r, and let the function f be analytic in an ellipse containing the spectrum of $\mathbf{B}$. Then, $f(\mathbf{B})$ satisfies $|f_{ij}(\mathbf{B})| \leq M\varrho^{|i-j|}$, where M is a constant and $\varrho = \mu^{1/r}$, where $\mu \in (0, 1)$ depends only on f.* ◇

Theorem 2 shows that the entries of $f(\mathbf{B})$ are bounded in an exponential decay manner away from the diagonal (the rate of decay depending on f), with the bound decreasing as the bandwidth r decreases. Note that this does not necessarily mean that "decay to zero" is observed in practice [31, Sect. 14.2].

If we turn our attention again to the Laplacian functions $f(\mathbf{L})$, it is easy to verify that the following family of undirected graphs with n nodes, admits a Laplacian $\mathbf{L}$ of bandwidth $r \in \{1, 2, \ldots, n-2\}$: $\mathcal{B}_n^r = (V, E)$ where $\{i, j\} \in E$ if and only if $j = i + s$ with $i \in \{1, 2, \ldots, n-s\}$ and $s \in \{1, 2, \ldots, r\}$. Graph $\mathcal{B}_n^1$ coincides with P_n (the path graph with n nodes), whose Laplacian is a tridiagonal matrix. We excluded the case of $r = n - 1$, since it corresponds to the complete graph with n nodes, whose Laplacian has zero bandwidth (see the example in Fig. 1). Note that this family of graphs is not unique. In fact, the entries on the diagonal band of $\mathbf{L}(\mathcal{B}_n^r)$ are all different from zero, but the definition of band matrix does not preclude null values.

IV. Application of Laplacian functions to consensus-based formation control

In this section, we leverage the Laplacian functions $f(\mathbf{L})$ studied in Sect. III, to design a new class of continuous-time consensus protocols. In particular, we will focus on the following *generalization* of the shape-based formation control strategy presented in [6, Ch. 6.3]:

$$\dot{\mathbf{x}}(t) = (-f(\mathbf{L}) \otimes \mathbf{I}_2)(\mathbf{x}(t) - \boldsymbol{\xi}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{8}$$

where the state vector $\mathbf{x}(t) = [\mathrm{x}_1(t), \mathrm{y}_1(t), \ldots, \mathrm{x}_n(t), \mathrm{y}_n(t)]^T \in \mathbb{R}^{2n}$ contains the x-and y-coordinates of the positions of n single-integrator robots at time $t \geq 0$. This strategy allows to drive the n robots to a rotationally-invariant formation encoded through the formation graph $\mathcal{G}_\mathrm{F} = (V, E_\mathrm{F})$ and the associated constant vector of target locations $\boldsymbol{\xi} \in \mathbb{R}^{2n}$. For the sake of simplicity, in the following, we will assume that $\mathcal{G}_\mathrm{F}$ coincides with the interaction (or communication) graph $\mathcal{G} = (V, E)$ of the robots (thus, the inclusion $E_\mathrm{F} \subseteq E$ is trivially satisfied, cf. [6, Th. 6.12]).

Remark 1 (Invariance of the agreement value) *By virtue of the spectral properties of the admissible Laplacian functions discussed in Sect. III, it is easy to prove that the state vector $\mathbf{x}(t)$ of system (8) asymptotically converges, for any initial condition $\mathbf{x}_0 \in \mathbb{R}^{2n}$, to a constant vector whose value does not depend on the $f(\mathbf{L})$ chosen (in other words, the agreement value is f-invariant).*

Remark 2 (Distributed implementation of (8)) *While some admissible functions f conserve, at least in part, the sparsity pattern of the Laplacian $\mathbf{L}$, the emergence of long-range interactions between the nodes of the graph typically translates into dense matrix functions of the Laplacian (i.e. the weighted interaction graph associated with $f(\mathbf{L})$ tends to be fully connected). This means that to implement (8), the robots should adopt an all-to-all communication pattern, which is undesirable, in practice. However, in many instances (cf. Theorem 2), a large percentage of the off-diagonal entries of $f(\mathbf{L})$ is very close to zero, which correspond to network connections which bring negligible information to the nodes. This calls for an operator which most nearly transforms $f(\mathbf{L})$ into a new matrix*

$f^q(\mathbf{L})$ which is sparser than $f(\mathbf{L})$. A simple solution is to define $f^q(\mathbf{L})$ as follows:

$$f^q_{ij}(\mathbf{L}) = \begin{cases} 0 & \text{if } |f_{ij}(\mathbf{L})| < q,\ i \neq j, \\ f_{ij}(\mathbf{L}) & \text{otherwise}, \end{cases}$$

where q is a small positive constant (a threshold). This transformation does not break the symmetry of $f(\mathbf{L})$ and the off-diagonal entries $f^q(\mathbf{L})$ remain non-positive as required by Condition III. Moreover, the Gershgorin circle theorem ensures that $f^q(\mathbf{L})$, as $f(\mathbf{L})$, is positive semidefinite. In conclusion, while $f^q(\mathbf{L})$ still remains less sparse than $\mathbf{L}$ in general, it makes protocol (8) *amenable to a distributed implementation (see Sect. IV-B for more details).*

A. Case studies

For later use in Sect. IV-B, we recall here some elementary properties of four of the Laplacian functions examined in Sect. III-A.

1) *Logarithmic function*: Let $f(x) = \log(ax+1)$ with $a > 0$. Note that
$$\lim_{a \to 0^+} f(\mathbf{L}) = \log(a\mathbf{L} + \mathbf{I}_n) = \mathbf{0}_n.$$
Moreover, $f(\mathbf{L})$ grows unbounded, as $a \to \infty$.
2) *Exponential function*: Let $f(x) = 1 - e^{-ax}$ with $a > 0$. Matrix $f(\mathbf{L}) = \mathbf{I}_n - e^{-a\mathbf{L}}$ ranges between $\mathbf{0}_n$ (as $a \to 0^+$) and $\mathbf{I}_n - \frac{1}{n}\mathbb{1}\mathbb{1}^T$ (as $a \to \infty$). In fact
$$\lim_{a \to \infty} e^{-a\mathbf{L}} = \mathbf{I}_n - \mathbf{L}\mathbf{L}^{\#} = \frac{1}{n}\mathbb{1}\mathbb{1}^T,$$
where $\mathbf{L}^{\#}$ denotes the group generalized inverse of $\mathbf{L}$ [39, Prop. 11.8.2]. Finally, if we consider the Loewner ordering (i.e. the partial ordering "$\preceq$" defined by the convex cone of positive semidefinite matrices), the following property holds true for $f(\mathbf{L}) = \mathbf{I}_n - e^{-a\mathbf{L}}$:
$$\mathbf{0}_n \preceq f(\mathbf{L}) \preceq \mathbf{I}_n - \frac{1}{n}\mathbb{1}\mathbb{1}^T.$$

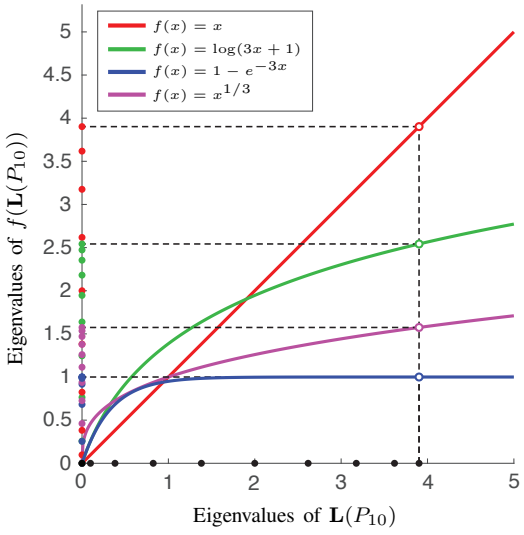


Fig. 2. Mapping of the spectrum of $\mathbf{L}(\boldsymbol{P_{10}})$ via four admissible functions $\boldsymbol{f(x)}$. The 10 eigenvalues of $\mathbf{L}(\boldsymbol{P_{10}})$ are represented as black dots on the horizontal axis and those of $\boldsymbol{f(\mathbf{L}(P_{10}))}$ as colored dots on the vertical axis.

3) *Quadratic function*: Let $g(x) = \lambda - x$ for $0 \leq x < \lambda$, which is positive over the spectrum of $\mathbf{L}$, if $\lambda > \lambda_n$ with $2-2\cos((n-1)\pi/n) \leq \lambda_n \leq n$. By choosing such a λ, from (5), we have that $f(x) = \frac{1}{2}[\lambda^2 - (\lambda - x)^2] = \frac{1}{2}x(2\lambda - x)$, which yields the quadratic matrix function $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + \lambda\mathbf{L}$.
4) *Fractional power*: Let $f(x) = x^\gamma$ with $0 < \gamma \leq 1$. Matrix $f(\mathbf{L}) = \mathbf{L}^\gamma$ ranges between $\mathbf{I}_n$ (as $\gamma \to 0^+$) and $\mathbf{L}$ (for $\gamma = 1$).

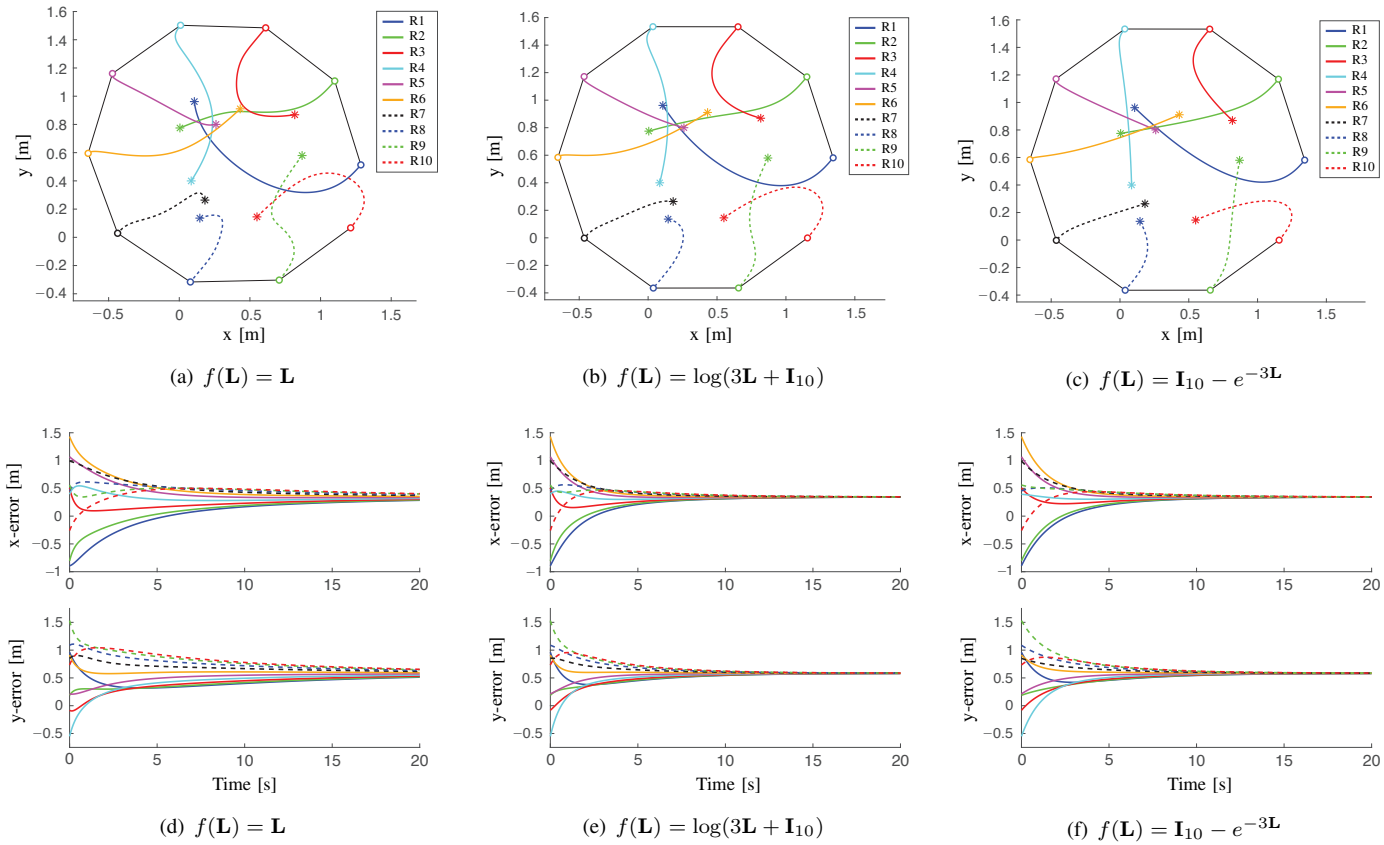
For the sake of illustration, Fig. 2 shows how the functions $f(x) = x$ (red), $f(x) = \log(3x+1)$ (green), $f(x) = 1 - e^{-3x}$ (blue), and $f(x) = x^{1/3}$ (magenta), map the 10 eigenvalues $2-2\cos(k\pi/10)$, $k \in \{0, 1, \ldots, 9\}$, of the Laplacian $\mathbf{L}$ of the path graph P_{10}. Note that the exponential function maps the spectrum of $\mathbf{L}(P_{10})$ into the $[0,\ 1)$ interval: in fact $1 - e^{-3\lambda_n} = 1 - 8 \times 10^{-6}$.

B. Numerical results

In our numerical experiments, we simulated system (8) for 10 robots, using different Laplacian functions, interaction graphs and vectors of target locations. We ran 20-second simulations using Matlab `ode45` solver with a variable step (max step size: 0.01 s).

We used Matlab built-in commands `logm` and `expm` for calculating the matrix logarithm and matrix exponential, respectively. The computation of the fractional Laplacian being more delicate, we opted for the `rootpm_real` routine of *The Matrix Function Toolbox* [40] (see also [31, Appendix D]). This routine computes the pth root of a real matrix via the real Schur form: similar results were obtained with `rootpm_sign` which computes the pth root via the matrix sign function. Both routines handle singular matrices ($\mathbf{L}$ has a zero eigenvalue), and return non-principal pth roots.

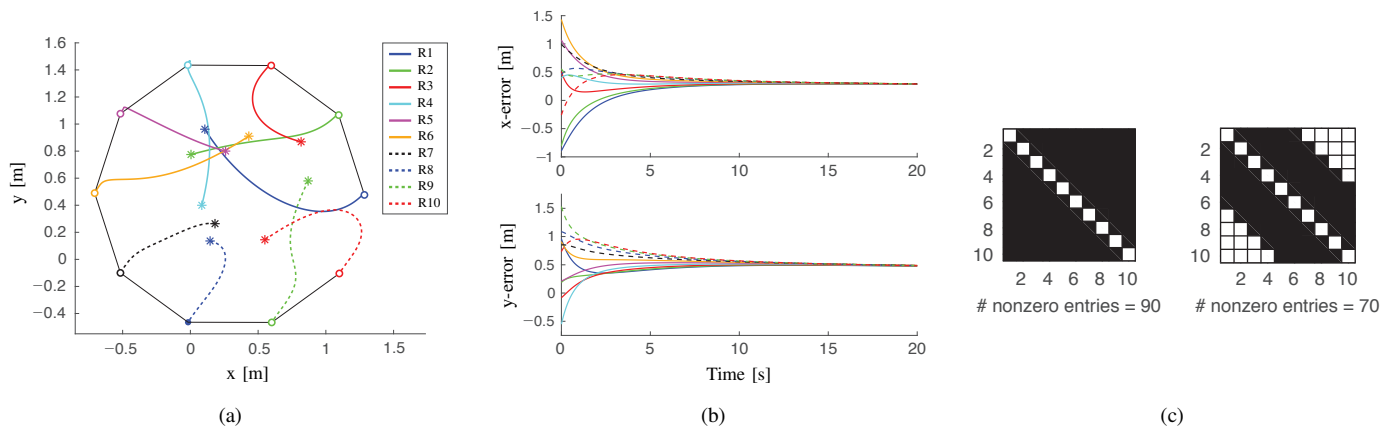
In our **first example**, the target formation is a regular decagon of unit radius centered at the origin, i.e. $\boldsymbol{\xi} = [\cos(0),\ \sin(0),\ \cos(2\pi/10),\ \sin(2\pi/10),\ \ldots,\ \cos(9\pi/10),\ \sin(9\pi/10)]^T \in \mathbb{R}^{20}$. The interaction graph is $\mathcal{G} = P_{10}$, and the Laplacian functions considered are $f(\mathbf{L}) = \mathbf{L}$, $f(\mathbf{L}) = \log(3\mathbf{L} + \mathbf{I}_{10})$ and $f(\mathbf{L}) = \mathbf{I}_{10} - e^{-3\mathbf{L}}$. The initial positions $\mathbf{x}_0$ of the 10 robots have been randomly generated by drawing the x- and y-coordinates from the standard uniform distribution on the open interval $(0,\ 1)$. The first row of Fig. 3 reports the trajectory of the robots for each of the three functions (the edges of the formation graph are solid black, and the initial positions of the robots are marked with stars). The second row of Fig. 3 shows the corresponding time-evolution of the formation error $\boldsymbol{e}(t) = \mathbf{x}(t) - \boldsymbol{\xi}$ (x-coordinates, top; y-coordinates, bottom). In the three cases, $\boldsymbol{e}(t) \to [\tau_{\mathrm{x}}, \tau_{\mathrm{y}}, \ldots, \tau_{\mathrm{x}}, \tau_{\mathrm{y}}]^T$ as $t \to \infty$, where $\tau_{\mathrm{x}} \simeq 0.3451$ and $\tau_{\mathrm{y}} \simeq 0.5841$ are constant offsets. The average of the generalized degree associated with the three Laplacian functions is 1.8, 1.54 and 0.7833, respectively (recall Sect. III-B.1), and their second smallest eigenvalues are $\lambda_2 = 2 - 2\cos(\pi/10) \simeq 0.0979$, $\log(3\lambda_2 + 1) \simeq 0.2575$ and $1 - e^{-3\lambda_2} \simeq 0.2545$, respectively. Therefore, the logarithmic and exponential functions accelerate convergence towards the desired formation, and the convergence speed can be adjusted



(a) $f(\mathbf{L}) = \mathbf{L}$ (b) $f(\mathbf{L}) = \log(3\mathbf{L} + \mathbf{I}_{10})$ (c) $f(\mathbf{L}) = \mathbf{I}_{10} - e^{-3\mathbf{L}}$

(d) $f(\mathbf{L}) = \mathbf{L}$ (e) $f(\mathbf{L}) = \log(3\mathbf{L} + \mathbf{I}_{10})$ (f) $f(\mathbf{L}) = \mathbf{I}_{10} - e^{-3\mathbf{L}}$

Fig. 3. *First example, $\boldsymbol{\xi}$ is the regular decagon and $\mathcal{G} = \boldsymbol{P_{10}}$*: (1st row) Trajectory of the 10 robots. The initial and final positions are marked with a star "$*$" and a hollow circle "o", respectively, and the edges of the formation graph are solid black; (2nd row) Time evolution of the formation error $\boldsymbol{e(t) = x(t) - \xi}$ of the 10 robots (for color coding, see the legends in the 1st row). (a),(d) $\boldsymbol{f(L) = L}$, (b),(e) $\boldsymbol{f(L) = \log(3L + I_{10})}$, and (c),(f) $\boldsymbol{f(L) = I_{10} - e^{-3L}}$.

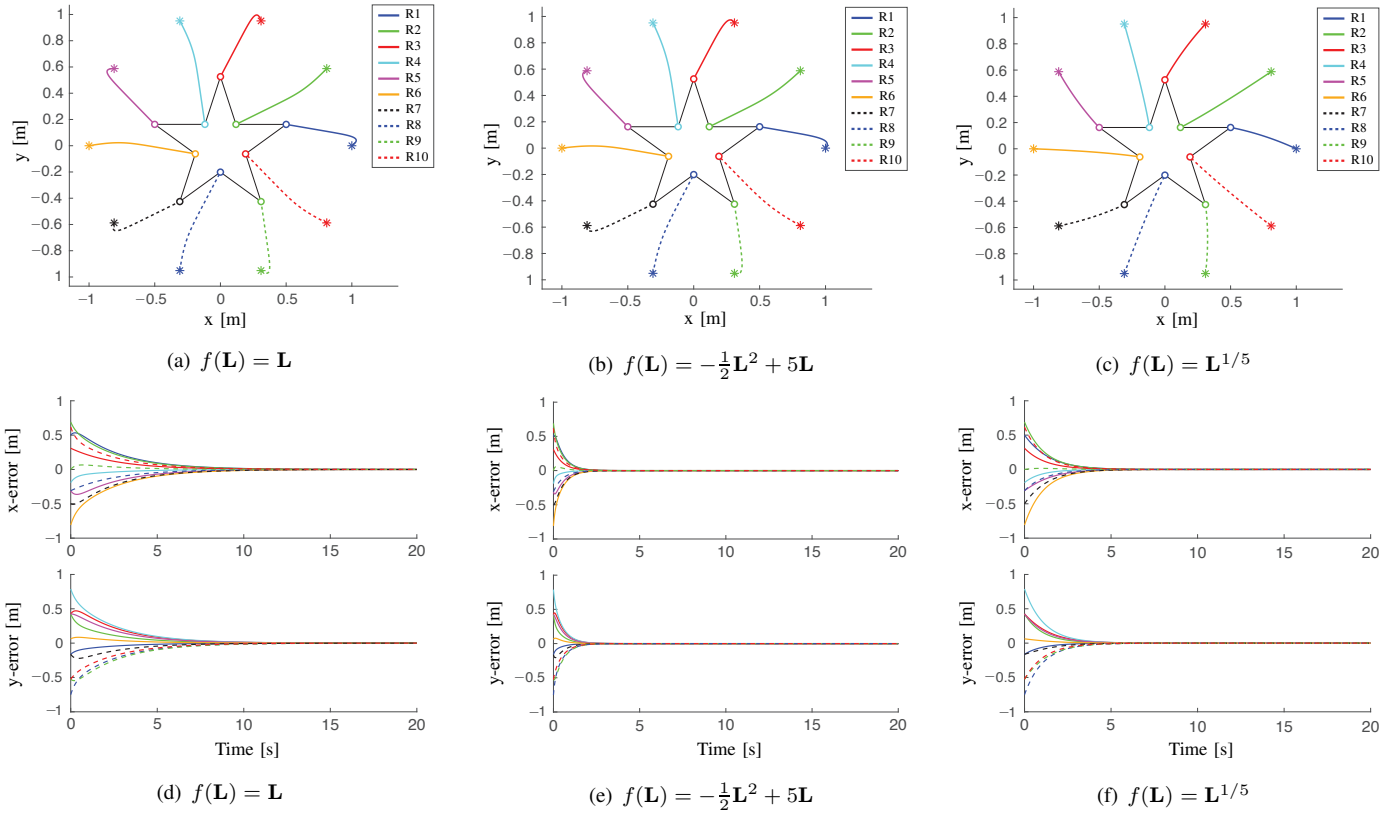


(a) (b) (c)

Fig. 4. *First example, $\boldsymbol{\xi}$ is the regular decagon and $\mathcal{G} = \boldsymbol{P_{10}}$*: (a) Trajectory of the 10 robots obtained by considering the approximation $\boldsymbol{f^q(L)}$ of $\boldsymbol{f(L) = \log(3L + I_{10})}$ with $\boldsymbol{q = 0.01}$; (b) Time evolution of the formation error $\boldsymbol{e(t) = x(t) - \xi}$ of the 10 robots; (c) Sparsity pattern (white squares, zero entries; black squares, nonzero entries) of the adjacency matrices associated with $\boldsymbol{f(L)}$ (left) and $\boldsymbol{f^q(L)}$ (right).

by tuning a single real parameter (the positive scalar a). However, both functions keep the agreement value unchanged.

To study the impact of the approximation introduced in Remark 2, in Fig. 4 we considered the same interaction graph, target formation, and initial conditions as in Figs. 3(b),(e), but we replaced $f(\mathbf{L}) = \log(3\mathbf{L} + \mathbf{I}_{10})$ with $f^q(\mathbf{L})$, where the threshold $q = 0.01$. Note that the exponential decay property of Theorem 2 holds for $f(\mathbf{L}) = \log(3\mathbf{L} + \mathbf{I}_{10})$ with $r = 1$. Figs. 4(a),(b) report the trajectory of the 10 robots and the time-evolution of the formation error $\boldsymbol{e}(t)$, respectively, and they qualitatively show that the proposed approximation has a negligible effect on the dynamic behavior of system (8). More specifically, at the end-time, $t = 20$ s, we have that $\|\boldsymbol{e}(t)\| \simeq 2.1453$ for $f(\mathbf{L})$ and $\|\boldsymbol{e}(t)\| \simeq 1.7851$



(a) $f(\mathbf{L}) = \mathbf{L}$ (b) $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + 5\mathbf{L}$ (c) $f(\mathbf{L}) = \mathbf{L}^{1/5}$

(d) $f(\mathbf{L}) = \mathbf{L}$ (e) $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + 5\mathbf{L}$ (f) $f(\mathbf{L}) = \mathbf{L}^{1/5}$

Fig. 5. *Second example, $\boldsymbol{\xi}$ is the pentagram and $\mathcal{G} = C_{10}$*: (1st row) Trajectory of the 10 robots. The initial and final positions are marked with a star "$*$" and a hollow circle "o", respectively, and the edges of the formation graph are solid black; (2nd row) Time evolution of the formation error $\boldsymbol{e}(t) = \mathbf{x}(t) - \boldsymbol{\xi}$ of the 10 robots (for color coding, see the legends in the 1st row). (a),(d) $f(\mathbf{L}) = \mathbf{L}$, (b),(e) $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + 5\mathbf{L}$, and (c),(f) $f(\mathbf{L}) = \mathbf{L}^{1/5}$.

for $f^q(\mathbf{L})$, where $\|\cdot\|$ denotes the Euclidean norm. However, the weighted interaction graph associated with $f^q(\mathbf{L})$ is sparser than that associated with $f(\mathbf{L})$. This is evident in Fig. 4(c), which shows the sparsity pattern of the adjacency matrices $\mathrm{diag}(f_{1,1}(\mathbf{L}), \ldots, f_{10,10}(\mathbf{L})) - f(\mathbf{L})$ (left) and $\mathrm{diag}(f^q_{1,1}(\mathbf{L}), \ldots, f^q_{10,10}(\mathbf{L})) - f^q(\mathbf{L})$ (right), where the zero entries are white and the nonzero entries are black.

In our **second example**, we chose the interaction graph $\mathcal{G} = C_{10}$, the cycle graph with 10 nodes, and the desired formation is the pentagram (or five-pointed star). Let

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

be the circumradius of the pentagram and the circumradius of its inner pentagon, respectively. Then, the target locations of the 10 robots can be expressed in polar coordinates as

$$\begin{bmatrix} [\boldsymbol{\xi}]_j \\ [\boldsymbol{\xi}]_{j+1} \end{bmatrix} = \begin{cases} \rho\big[\cos(\frac{j\pi}{10}),\ \sin(\frac{j\pi}{10})\big]^T & \text{if } j \in \{1,5,9,\ldots\}, \\ R\big[\cos(\frac{j\pi}{10}),\ \sin(\frac{j\pi}{10})\big]^T & \text{if } j \in \{3,7,11,\ldots\}, \end{cases}$$

where $j \in \{1,3,5,\ldots,19\}$. The Laplacian functions considered in this second example, are $f(\mathbf{L}) = \mathbf{L}$, $f(\mathbf{L}) = -\frac{1}{2}\mathbf{L}^2 + 5\mathbf{L}$ and $f(\mathbf{L}) = \mathbf{L}^{1/5}$. In the quadratic function, we set $\lambda = 5 > \lambda_{10} = 4$, to satisfy the condition discussed in Sect. IV-A. The vector of initial positions $\mathbf{x}_0$, is not random this time: the 10 robots are initially placed at the vertices of the same regular decagon of unit radius considered as target formation in the first example. As in Fig. 3, the first row of Fig. 5 reports the trajectory of the 10 robots for the three Laplacian functions, and the second row, the corresponding time-evolution of the formation error $\boldsymbol{e}(t) = \mathbf{x}(t) - \boldsymbol{\xi}$. Since the regular decagon is centered at the origin, we have that $\boldsymbol{e}(t) \to \mathbf{0}$ as $t \to \infty$, in all cases. Finally, note that the average fractional degree of $\mathbf{L}^{1/5}$ is 1.0115, and that the user can take advantage of the non-integer power γ to modulate the convergence speed of the 10 robots towards the desired formation. In fact, the second smallest eigenvalues of $\mathbf{L}$ and $\mathbf{L}^{1/5}$, are $\lambda_2 = 2 - 2\cos(\pi/5) \simeq 0.3820$ and $\lambda_2^{1/5} \simeq 0.8249$, respectively, which explains the faster convergence rate observed in Figs. 5(c),(f).

V. Conclusion and Future Work

In this paper, we have explored a general class of matrix functions of the combinatorial Laplacian, that retain its structural properties. This allowed us to present under a common framework, several variants of the Laplacian scattered across different domains. For the first time, this family of Laplacian functions has been utilized in a consensus-based formation control protocol, revealing some attractive features. In fact, the selected functions do not alter the agreement value, and offer greater design flexibility. For example, our numerical experiments show that in many situations, a single

scalar parameter is sufficient to adjust the convergence speed towards consensus without drastically increasing the number of communication exchanges between the agents.

This work provides fertile ground for further research and experimentation. One line of future research will be to relax Condition III and consider a broader class of admissible Laplacian functions. As far as the consensus protocol is concerned, we see potential for extension to weighted directed interaction graphs, and we plan to perform hardware experiments with mobile robots. Finally, in [22], the authors have shown a connection between the fractional Laplacian of a graph and the operators in fractional calculus. There might then exist a link between the agreement protocol driven by the fractional Laplacian, and the coordination algorithms for fractional-order systems studied in [41].

REFERENCES

[1] M. Newman. *Networks*. Oxford University Press, 2nd edition, 2018.

[2] D.I. Shuman, B. Ricaud, and P. Vandergheynst. Vertex-frequency analysis on graphs. *Appl. Comput. Harmon. Anal.*, 40(2):260–291, 2016.

[3] A. Ortega, P. Frossard, J. Kovačević, J.M.F Moura, and P. Vandergheynst. Graph Signal Processing: Overview, Challenges, and Applications. *Proc. IEEE*, 106(5):808–828, 2018.

[4] F. Gama, E. Isufi, G. Leus, and A. Ribeiro. Graphs, Convolutions, and Neural Networks: From graph filters to graph neural networks. *IEEE Signal Process. Mag.*, 37(6):128–138, 2020.

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S.Y. Philip. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021.

[6] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[7] F. Bullo. *Lectures on Network Systems*. CreateSpace, 1st edition, 2018. With contributions by J. Cortes, F. Dörfler, and S. Martínez.

[8] F. Morbidi. The deformed consensus protocol. *Automatica*, 49(10):3049–3055, 2013.

[9] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang. Deformed Graph Laplacian for Semisupervised Learning. *IEEE Trans. Neur. Net. Lear.*, 26(10):2261–2274, 2015.

[10] P. Grindrod, D.J. Higham, and V. Noferini. The Deformed Graph Laplacian and Its Applications to Network Centrality Analysis. *SIAM J. Matrix Anal. A.*, 39(1):310–341, 2018.

[11] F. Arrigo, P. Grindrod, D.J. Higham, and V. Noferini. On the exponential generating function for non-backtracking walks. *Linear Algebra Appl.*, 556:381–399, 2018.

[12] F. Arrigo, D.J. Higham, and V. Noferini. Non-backtracking alternating walks. *SIAM J. Appl. Math.*, 79(3):781–801, 2019.

[13] F. Arrigo, D.J. Higham, and V. Noferini. Beyond non-backtracking: non-cycling network centrality measures. *Proc. R. Soc. A*, 476:20190653, 2020.

[14] F. Morbidi. The Second-order Parametric Consensus Protocol. In *Proc. European Contr. Conf*, pages 202–207, 2014.

[15] A. Saade, F. Krzakala, and L. Zdeborová. Spectral Clustering of Graphs with the Bethe Hessian. In *Proc. 28th Conf. Neural Inf. Process. Syst.*, pages 406–414, 2014.

[16] L. Dall'Amico, R. Couillet, and N. Tremblay. Revisiting the Bethe-Hessian: Improved Community Detection in Sparse Heterogeneous Graphs. In *Proc. 33rd Conf. Neural Inf. Process. Syst.*, pages 4039–4049, 2019.

[17] M. Stoll. A Literature Survey of Matrix Methods for Data Science. *GAMM-Mitteilungen*, 43(3):e202000013, 2020.

[18] M. Fanuel, C.M. Alaíz, A. Fernández, and J.A.K. Suykens. Magnetic Eigenmaps for the visualization of directed networks. *Appl. Comput. Harmon. Anal.*, 44(1):189–199, 2018.

[19] M. Fanuel and J.A.K. Suykens. Deformed Laplacians and spectral ranking in directed networks. *Appl. Comput. Harmon. Anal.*, 47(2):397–422, 2019.

[20] T. Bühler and M. Hein. Spectral clustering based on the graph p-Laplacian. In *Proc. 26th Annual Int. Conf. Machine Learning*, pages 81–88, 2009.

[21] D. Luo, H. Huang, C. Ding, and F. Nie. On the eigenvectors of p-Laplacian. *Mach. Learn.*, 81(1):37–51, 2010.

[22] T. Michelitsch, A. Pérez Riascos, B. Collet, A. Nowakowski, and F. Nicolleau. *Fractional Dynamics on Networks and Lattices*. ISTE Ltd and John Wiley & Sons, 2019.

[23] F. Li and M.K. Ng. Image colorization by using graph bi-Laplacian. *Adv. Comput. Math.*, 45(3):1521–1549, 2019.

[24] A. Henrot. *Extremum Problems for Eigenvalues of Elliptic Operators*. Frontiers in Mathematics. Birkhäuser, 2006.

[25] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M.M. Meerschaert, M. Ainsworth, and G.E. Karniadakis. What is the fractional Laplacian? A comparative review with new results. *J. Comput. Phys.*, 404:109009, 2020.

[26] E. Bautista, P. Abry, and P. Gonçalves. $L^{\gamma}-$PageRank for Semi-Supervised Learning. *Appl. Network Sci.*, 4(1):Article n. 57, 2019.

[27] S. Segarra, A.G. Marques, and A. Ribeiro. Optimal Graph-Filter Design and Applications to Distributed Linear Network Operators. *IEEE Trans. Signal Process.*, 65(15):4117–4131, 2017.

[28] R. Levie, F. Monti, X. Bresson, and M.M. Bronstein. CayleyNets: Graph Convolutional Neural Networks With Complex Rational Spectral Filters. *IEEE Trans. Signal Process.*, 67(1):97–109, 2019.

[29] Z. Jin and R.M. Murray. Multi-Hop Relay Protocols for Fast Consensus Seeking. In *Proc. 45th IEEE Conf. Dec. Contr*, pages 1001–1006, 2006.

[30] S. Manfredi. Design of a multi-hop dynamic consensus algorithm over wireless sensor networks. *Control Eng. Pract.*, 21(4):381–394, 2013.

[31] N.J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

[32] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

[33] P. Lancaster and M. Tismenetsky. *The Theory of Matrices: with Applications*. Academy Press, 2nd edition, 1985.

[34] K.S. Miller and S.G. Samko. Completely monotonic functions. *Integral Transform Spec. Funct.*, 12(4):389–402, 2001.

[35] F. Fouss, M. Saerens, and M. Shimbo. *Algorithms and Models for Network Data and Link Analysis*. Cambridge University Press, 2016.

[36] S. Segarra, A.G. Marques, G. Mateos, and A. Ribeiro. Network Topology Inference from Spectral Templates. *IEEE Trans. Signal Inf. Process. Networks*, 3(3):467–483, 2017.

[37] M. Benzi and N. Razouk. Decay Bounds and $O(n)$ Algorithms for Approximating Functions of Sparse Matrices. *Electron. Trans. Numer. Anal.*, 28:16–39, 2007.

[38] M. Benzi and G.H. Golub. Bounds for the Entries of Matrix Functions with Applications to Preconditioning. *BIT Numer. Math.*, 39(3):417–438, 1999.

[39] D.S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, 2nd edition, 2009.

[40] N.J. Higham. The Matrix Function Toolbox, 2008. [web] `www.maths.manchester.ac.uk/~higham/mftoolbox`.

[41] Y. Cao, Y. Li, W. Ren, and Y. Chen. Distributed Coordination of Networked Fractional-Order Systems. *IEEE Trans. Syst. Man Cy. B*, 40(2):362–370, 2010.

Fabio Morbidi (Senior Member, IEEE) received the Ph.D. degree in control engineering and robotics from the University of Siena, Italy, in 2009. He was a visiting researcher at the University of California, Santa Barbara, USA, in 2008, and he held post-doctoral positions at Northwestern University, University of Texas at Arlington, USA, and at Inria Grenoble Rhône-Alpes, France.

Since 2014, he has been an Associate Professor of robotics with the Université de Picardie Jules Verne, France. He is a member of the MIS (Modeling, Information & Systems) laboratory, and the author or co-author of over seventy articles in the field of automatic control and robotics.

His main research interests include multi-agent and network systems, and robot vision with applications to autonomous vehicle localization and navigation.