Distributed Spatial Filtering by a Two-Hop Consensus-Type Algorithm

Shinsaku Izumi, *Member, IEEE*, Xin Xin, *Senior Member, IEEE*, and Taiga Yamasaki

Abstract—In this letter, we discuss distributed spatial filtering (DSF) on networked systems to obtain signal values with a desired spatial frequency characteristic from those assigned to nodes by a distributed algorithm. We present a two-hop consensus-type algorithm for DSF based on an existing one-hop algorithm. We prove that the range of the filter characteristics the presented algorithm can achieve is broader than that for the existing algorithm by deriving a necessary and sufficient condition for achieving DSF. Simulation results show that our filtering algorithm and a new filter characteristic it provides are effective in distributed anomaly detection by sensor networks.

Index Terms—Control of networks, distributed control, filtering, sensor networks.

Fig. 1. Example of DSF with a high-pass characteristic.

I. INTRODUCTION

IN the systems and control community, networked systems, constructed by connecting subsystems (or nodes), have been a major research topic. This is motivated by the fact that modern engineering applications, *e.g.,* swarm robots and sensor networks, can be categorized into networked systems.

Herein, we focus on *distributed spatial filtering (DSF)* on networked systems. DSF is to transform signal values given for nodes into ones with a desired spatial frequency characteristic in a distributed way. Fig. 1 shows an example for high-pass filtering, where x_i denotes the signal value given for node i and it is assumed that nodes are closer to each other as their indices are closer. The nodes transform the given signal values for reducing the amplitudes of the low-frequency components in a distributed way. The main advantage of DSF is that the spatial properties of signals to be processed can be used. For instance, the spatial frequency of temperature is generally low because it takes similar values at close locations. With this property, we can reduce the noise in temperature measurements from a sensor network through spatial low-pass filtering. Further,

some anomalies (*e.g.,* fires) increase or decrease temperature at specific locations and make the spatial frequency of the measurements high, and thus spatial high-pass filtering allows to detect the anomalies.

Izumi et al. [1] proposed a DSF method by focusing on the relation between a consensus algorithm and signal processing on graphs [2]. Then, in [3], a more sophisticated DSF method was proposed and verified through experiments with a real sensor network. However, the DSF method in [3] has the drawback that the achievable filter characteristics are limited to ones described as polynomials with non-zero *real* roots. This limitation is severe because the degree of the describing polynomial corresponds to the degree of freedom in choosing the filter characteristic, while high-degree polynomials generally have complex roots.

We thus aim to develop a DSF method to overcome this limitation. Our contributions are as follows. First, we present an extended version of the DSF method in [3]. By focusing on the fact that the filtering algorithm in [3] is a one-hop consensus-type algorithm, we extend the existing algorithm to the two-hop version where each node uses information on its two-hop neighbors, *i.e.,* the neighbors of its neighbors. We then prove that DSF is achieved using our algorithm even for the filter functions described as polynomials with complex roots. This overcomes the above limitation and makes the range of the achievable filter characteristics wider. Second, we demonstrate the effectiveness of our DSF method through its application to anomaly detection by sensor networks. We choose a filter function that cannot be handled by the existing algorithm [3]. We then show that our DSF method with this filter function achieves higher detection performance than that when using the existing algorithm.

Finally, we discuss related works. This letter is related to several topics: spatial frequencies for distributed systems [4],

signal processing over graphs [5] and its applications [6], and distributed control with multi-hop communication [7], [8]. In [4], Bode integrals based on spatial frequencies were introduced for distributed systems. Mollaebrahim and Beferull-Lozano [5] investigated the linear transformations of signals over graphs, and Yi et al. [6] considered average consensus with the filtering of signals over graphs. In [7], [8], coverage and consensus with multi-hop communication were discussed, respectively. However, the purposes of these works are different from ours.

Notation: We describe the real number field and the set of positive real numbers using $\mathbb{R}$ and $\mathbb{R}_+$, respectively. Let I be the identity matrix. For the numbers $x_1, x_2, \ldots, x_n \in \mathbb{R}$, we use $\mathrm{diag}(x_1, x_2, \ldots, x_n)$ to represent the diagonal matrix with $x_1, x_2, \ldots, x_n$ on its diagonal. We define $[x_i]_{i \in \mathbb{I}} :=$ $[x_{i_1} \ x_{i_2} \ \cdots \ x_{i_m}]^\top \in \mathbb{R}^m$ with $\mathbb{I} := \{i_1, i_2, \ldots, i_m\} \subseteq \{1, 2, \ldots, n\}$. The cardinality of the set $\mathbb{S}$ is denoted by $|\mathbb{S}|$. For an undirected graph, we represent its graph Laplacian by L; *i.e.*, L is defined as the difference between the degree matrix and adjacency matrix of the graph.

II. EXISTING RESULTS ON DSF

We first review the existing results [3] on DSF.

A. Preliminary: Spatial Filtering of Graph Signals

Consider an undirected graph with n vertices, which is described by $G = (\mathbb{V}, \mathbb{E})$ for the vertex set $\mathbb{V} := \{1, 2, \ldots, n\}$ and the edge set $\mathbb{E}$. We suppose that a signal value is assigned to each vertex of G, and describe the signal values for n vertices using $s \in \mathbb{R}^n$. Then, (G, s) is referred to as a graph signal, where the edges of G specify the connections between the signal values.

For the graph signal (G, s), its Fourier transform (*i.e.*, the graph Fourier transform) is given as follows. For the graph Laplacian L of G, we define its i-th smallest eigenvalue in the sense of the modulus by λ_i ($i \in \mathbb{V}$). Then, the graph Fourier transform $f \in \mathbb{R}^n$ is written as

$$f(\lambda_1, \lambda_2, \ldots, \lambda_n) := V^\top s \qquad (1)$$

for the orthogonal matrix $V \in \mathbb{R}^{n \times n}$ such that $V^\top L V = \Lambda$ with $\Lambda := \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Because L is a symmetric matrix by its definition, we can always find a V that satisfies $V^\top L V = \Lambda$ for a given L. The graph Fourier transform f provides an expression for (G, s) in spatial frequency domain. Specifically, f expresses the magnitude of the differences between signal values connected by the edges of G. In (1), λ_i ($i \in \mathbb{V}$) corresponds to the spatial frequency, and the component of λ_i is denoted by the i-th element of f, where it is noteworthy that $\lambda_1, \lambda_2, \ldots, \lambda_n$ are nonnegative real numbers because L is the graph Laplacian of the undirected graph G.

The graph Fourier transform (1) allows the spatial filtering of graph signals described by

$$\tilde{s} = V \mathrm{diag}(h(\lambda_1), h(\lambda_2), \ldots, h(\lambda_n))V^\top s, \qquad (2)$$

where $\tilde{s} \in \mathbb{R}^n$ represents the signal values of a filtered graph signal and $h : \mathbb{R}_+ \cup \{0\} \to \mathbb{R}$ is a function to characterize the filter. From (1) and the orthogonality of V, *i.e.*, $V^\top = V^{-1}$, (2)

implies that the spatial filtering is to filter a signal transformed into one in the spatial frequency domain by (1) using h and to perform the inverse transform of (1). In this way, we can obtain an $\tilde{s}$ such that the graph signal $(G, \tilde{s})$ has a spatial frequency characteristic specified by h.

B. Problem Formulation

Consider the networked system Σ having n nodes. Node i ($i \in \mathbb{V}$) is given as the discrete-time model

$$x_i(t + 1) = g([x_j(t)]_{j \in \mathbb{N}_i}, t), \qquad (3)$$

where $x_i(t) \in \mathbb{R}$ and $[x_j(t)]_{j \in \mathbb{N}_i} \in \mathbb{R}^{|\mathbb{N}_i|}$ are the state and the input, respectively, and $\mathbb{N}_i \subseteq \mathbb{V}$ is the index set of nodes of which node i has access to the information. The function $g : \mathbb{R}^{|\mathbb{N}_i|} \times \{0, 1, \ldots\} \to \mathbb{R}$ specifies the behavior of the node. Using g rather than g_i means that the behavior of all the nodes is determined by a common function, which allows to make Σ scalable.

We represent the network topology of the system Σ using the undirected graph G defined above, where the vertex set $\mathbb{V}$ and the edge set $\mathbb{E}$ express the indices of n nodes and the connections between them, respectively. In this case, the set $\mathbb{N}_i$ is given by $\mathbb{N}_i := \{i\} \cup \{j \in \mathbb{V} \mid (j, i) \in \mathbb{E}\}$. Moreover, let $x(t) \in \mathbb{R}^n$ denote the state of Σ, *i.e.*, $x(t) := [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]^\top$. Then, $(G, x(t))$ can be regarded as one of graph signals described in Section II-A by considering G and $x(t)$ as the graph and the signal values, respectively.

Based on this observation, our interest here is to construct a system Σ that works as a desired spatial filter for graph signals. To this end, we regard the initial state $x(0)$ and the final state $x(\infty)$ as the signal values of an original graph signal and those of the filtered one, respectively, and address the following problem.

Problem 1: For the networked system Σ, suppose that a filter function h and an initial state $x(0)$ are given. Find a function g (*i.e.*, a distributed algorithm executed by n nodes) to produce an $x(\infty)$ such that the graph signal $(G, x(\infty))$ has a spatial frequency characteristic specified by h.

C. Filtering Algorithm

The idea of [3] to solve Problem 1 is to design the function g so that the relation between $x(0)$ and $x(\infty)$ is equivalent to the spatial filtering (2) of graph signals when regarding $x(0)$ and $x(\infty)$ as s and $\tilde{s}$, respectively. Based on this,

$$g([x_j(t)]_{j \in \mathbb{N}_i}, t) := \ell_0(t)x_i(t) + \ell_1(t) \sum_{j \in \mathbb{N}_i} (x_j(t) - x_i(t)) \qquad (4)$$

was provided, where $\ell_0(t) \in \mathbb{R} \setminus \{0\}$ and $\ell_1(t) \in \mathbb{R}$ are time-varying gains that satisfy $\ell_0(t) = 1$ and $\ell_1(t) = 0$ when $t \geq m$ for a positive integer m. The provided g and (3) yield a consensus-type distributed algorithm. It should be noted that this algorithm converges in m timesteps due to $\ell_0(t) = 1$ and $\ell_1(t) = 0$ for $t \geq m$.

We suppose that the system Σ operates according to a distributed algorithm such as that given by (3) and (4). If for a filter function h, the relation between $x(0)$ and $x(\infty)$ is equivalent to (2) as described above, we say that Σ achieves DSF
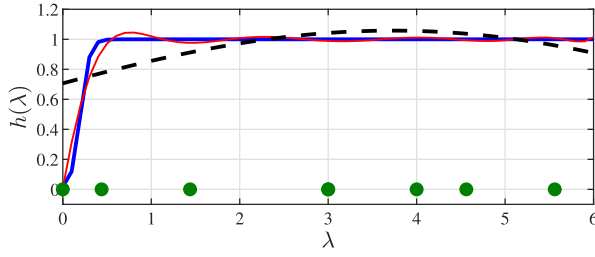
Fig. 2. Filter function $h(\lambda) := 1/(1 + e^{-20(\lambda-0.2)})$ (the thick blue line) and its polynomial approximations h_{10} (the thin red line) and h_{10}^r (the dashed black line).

of the graph signal $(G, x(0))$ for h. Then, the following result was obtained in [3].

Lemma 1: For the networked system Σ, suppose that a filter function h is given and let g be given by (4). Then, Σ achieves DSF of the graph signal $(G, x(0))$ for h if and only if h is a polynomial of the form

$$h(\lambda) := a_m \lambda^m + a_{m-1}\lambda^{m-1} + \cdots + a_1\lambda + a_0 \quad (5)$$

with a frequency variable λ, where $a_0, a_1, \ldots, a_m \in \mathbb{R}$ and the roots of the polynomial must be non-zero real numbers.

Lemma 1 indicates that DSF is achieved using (3) and (4) if and only if the filter function h is a real polynomial having non-zero real roots.

D. Problem to Be Considered

Although Lemma 1 gives a necessary and sufficient condition on h for achieving DSF, it is difficult to achieve some filter characteristics under this condition. An example is shown in Fig. 2. The thick blue line and the thin red line indicate $h(\lambda) := 1/(1 + e^{-20(\lambda-0.2)})$ and its polynomial approximation $h_{10}(\lambda) \approx 0.000149\lambda^{10} - 0.00453\lambda^9 + 0.0583\lambda^8 - 0.413\lambda^7 + 1.74\lambda^6 - 4.41\lambda^5 + 6.15\lambda^4 - 3.21\lambda^3 - 2.18\lambda^2 + 3.27\lambda + 0.0100[1] via a least square fitting, respectively (and the green circles are detailed in Section III-C). As demonstrated later, h_{10} describes a filter characteristic that is useful in an application to sensor networks. However, we can numerically confirm that h_{10} has complex roots and does not satisfy the condition in Lemma 1. Therefore, we seek the polynomial approximation with non-zero real roots by solving an optimization problem for fitting using the function "fminsearch" in MATLAB, and obtain $h_{10}^r(\lambda) \approx -(4.37 \times 10^{-13})\lambda^{10} + (2.60 \times 10^{-11})\lambda^9 + (1.16 \times 10^{-9})\lambda^8 - (5.38 \times 10^{-8})\lambda^7 - (2.81 \times 10^{-7})\lambda^6 + (2.07 \times 10^{-5})\lambda^5 + (3.64 \times 10^{-5})\lambda^4 - 0.00305\lambda^3 - 0.00790\lambda^2 + 0.160\lambda + 0.706[1]. However, the plot of h_{10}^r indicated by the dashed black line in Fig. 2 is quite different from that of the original h, compared with the case of h_{10}.

III. DSF BY TWO-HOP CONSENSUS-TYPE ALGORITHM

In this section, we overcome the aforementioned difficulty by extending the results in [3].

A. Proposed Algorithm

Consider the networked system Σ again. We suppose here that the model of node i ($i \in \mathbb{V}$) is given by

---

[1]The coefficients of the polynomials are rounded off for brevity.

$$\begin{cases} x_i(t+1) = g_1([x_j(t)]_{j\in\mathbb{N}_i}, [y_j(t)]_{j\in\mathbb{N}_i}, t), \\ y_i(t) = g_2([x_j(t)]_{j\in\mathbb{N}_i}, t), \end{cases} \quad (6)$$

where $[y_j(t)]_{j\in\mathbb{N}_i} \in \mathbb{R}^{|\mathbb{N}_i|}$ is the input, $y_i(t) \in \mathbb{R}$ is the output, and $g_1 : \mathbb{R}^{|\mathbb{N}_i|} \times \mathbb{R}^{|\mathbb{N}_i|} \times \{0, 1, \ldots\} \to \mathbb{R}$ and $g_2 : \mathbb{R}^{|\mathbb{N}_i|} \times \{0, 1, \ldots\} \to \mathbb{R}$ are functions to determine the node behavior. Under this setting, we extend the results in [3] by solving Problem 1 where the design parameter g is replaced by g_1 and g_2.

Our approach to the modified problem is to extend the existing filtering algorithm given by (3) and (4) to the two-hop version, *i.e.,* a consensus-type algorithm that uses the information on two-hop neighbors on the graph G in addition to that on the (one-hop) neighbors. Based on this, we propose

$$g_1([x_j(t)]_{j\in\mathbb{N}_i}, [y_j(t)]_{j\in\mathbb{N}_i}, t) := \ell_0(t)x_i(t) + \ell_1(t)$$
$$\times \sum_{j\in\mathbb{N}_i}(x_j(t) - x_i(t)) + \ell_2(t)\sum_{j\in\mathbb{N}_i}(y_j(t) - y_i(t)), \quad (7)$$

$$g_2([x_j(t)]_{j\in\mathbb{N}_i}, t) := \sum_{j\in\mathbb{N}_i}(x_j(t) - x_i(t)), \quad (8)$$

where $\ell_2(t) \in \mathbb{R}$ is a time-varying gain satisfying $\ell_2(t) = 0$ for $t \geq m$. In the proposed algorithm given by (6)–(8), the output $y_i(t)$, transmitted to the neighbors of node i, contains the state $x_j(t)$ ($j \in \mathbb{N}_i$), and thus the nodes can obtain the information on their two-hop neighbors.

B. Main Result

For the proposed algorithm given by (6)–(8), the following main result is obtained.

Theorem 1: For the networked system Σ where the node model (3) is replaced by (6), suppose that a filter function h is given and let g_1 and g_2 be given by (7) and (8), respectively. Then, Σ achieves DSF of the graph signal $(G, x(0))$ for h if and only if h is a real polynomial of the frequency variable λ with at most degree $2m$ and non-zero roots.

Proof: See the Appendix. ∎

Theorem 1 shows that DSF is achieved using the proposed algorithm given by (6)–(8) even if the filter function h is a polynomial with complex roots. This makes the range of the achievable filter characteristics wider. Here, the gains $\ell_0(t)$, $\ell_1(t)$, and $\ell_2(t)$ are chosen as, for instance, (12)–(14) or (12)–(14) and (18) (see the Appendix). Further, Lemma 1 ensures that the existing one-hop algorithm given by (3) and (4) cannot handle h other than polynomials with non-zero real roots. In this sense, Theorem 1 implies that the extension to the two-hop algorithm is a key to handling the case of the complex roots.

From the above discussion, our method to solve Problem 1 with the design parameters g_1 and g_2 is as follows: (i) approximate the given filter function h by a polynomial with non-zero roots; (ii) give g_1 and g_2 using (7), (8), (12)–(14), and (18).

An example to demonstrate our results is provided. Consider the system Σ with $n := 8$ and a network topology G (detailed later). The initial state $x(0)$ is given as $x(0) := [0.564 \ 0.567 \ 0.454 \ 0.325 \ 0.324 \ 0.237 \ 0.180 \ 0.177]^\top$. We choose h_{10} in Section II-D as the filter function h, for which
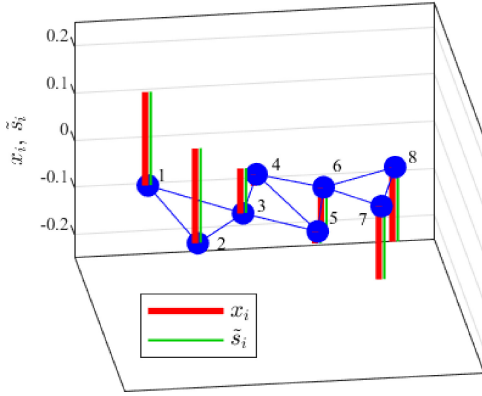
Fig. 3. Final state $x(5)$ given by the proposed algorithm and the collective signal value $\tilde{s}$ given by (2) for $s := x(0)$.

Fig. 4. Final states for the two filter functions h_{10} and h_{10}^r.

TABLE I
SUCCESS RATES [%] OF THE ANOMALY DETECTION FOR 1000 TRIALS

	Proposed algorithm (for h_{10})	[3] (for h_{10}^r)
$b := 10$	100	51.2
$b := 20$	100	0
$b := 30$	100	0

$m := 5$. Let us recall that h_{10} is a polynomial with complex roots and thus cannot be handled by the existing algorithm given by (3) and (4). From h_{10}, (6)–(8), and (12)–(14), the algorithm for the nodes is obtained, where we use the fact that the degree of h_{10} is even. The resulting final state $x(5)$, which is equal to $x(\infty)$ (see the Appendix), is depicted in Fig. 3, where the thick red lines represent the elements of $x(5)$ and the blue circles and lines represent the vertices and edges of G, respectively. Moreover, the thin green lines indicate the collective signal value $\tilde{s}$ given by (2) for $s := x(0)$, where $\tilde{s}_i$ is the i-th element of $\tilde{s}$. We see that $x(5)$ and $\tilde{s}$ are the same although the proposed algorithm is distributed but (2) is not. This demonstrates our results.

Remark 1: The degree of the polynomial approximation of the filter function h should be chosen according to the guideline given in [1, Remark 6]. Note here that we do not need to consider the constraint that the resulting polynomial has no complex roots, unlike the case of [1].

C. Application to Distributed Anomaly Detection by Sensor Networks

Next, we apply the proposed DSF method to distributed anomaly detection by sensor networks, *i.e.,* finding sensor nodes such that the measurements are quite different from those from other nodes in a distributed way. As explained in Section I, an example of anomaly is a fire, which causes the differences between temperature measurements.

Our idea to achieve the anomaly detection is to remove *only* the component of the zero frequency at which the signal values for all the nodes are the same. By removing the zero frequency component, we can extract the differences of the signal values from a typical one in the group of the nodes, which allows the anomaly detection. In the following, we show that the proposed algorithm can handle even a specialized filter function to remove only the specific frequency component, and demonstrate that the anomaly detection can be achieved by the proposed algorithm and the filter function.

Consider the example in Section III-B again. We represent the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_8$ of the graph Laplacian L of the graph G depicted in Fig. 3 by the green circles in Fig. 2. These eigenvalues correspond to the spatial frequencies of which the associated graph signal has the components, and
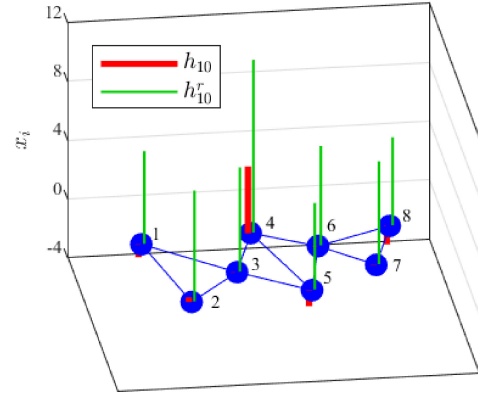
thus Fig. 2 indicates that the filter function h_{10} has the property of removing only the zero frequency component of graph signals with G. As seen above, the proposed algorithm given by (6)–(8) and (12)–(14) can handle h_{10}. Hence, we perform the distributed anomaly detection using the proposed algorithm for h_{10}.

For $x(0) := [9.39 \ 10.6 \ 10.2 \ 14.8 \ 9.25 \ 9.80 \ 9.94 \ 8.96]^\top$, we show the resulting final state $x(5)$ by the thick red lines in Fig. 4, where node 4 has to be detected. It turns out that the elements other than $x_4(5)$ are close to zero. Thus, by introducing a threshold, node 4 can be detected. The thin green lines in Fig. 4 represent the corresponding result when using the existing algorithm given by (3) and (4) for the filter function h_{10}^r in Section II-D. We see that the elements other than $x_4(5)$ are not sufficiently reduced compared with the result for h_{10}. This implies that the anomaly detection with the existing algorithm for h_{10}^r is difficult because the choice of the threshold heavily depends on the final states of nodes other than those at which anomaly occurs.

We evaluate the detection performance of the proposed algorithm for h_{10} in the following way. We suppose that anomaly occurs at a node j and j is randomly chosen from the set $\mathbb{V}$ with equal probability. The initial state $x_j(0)$ of node j is set as $x_j(0) := b + \epsilon + w$, where $b, \epsilon, w \in \mathbb{R}$ are a base value, measurement noise, and disturbances due to the anomaly, respectively. The noise ϵ is Gaussian noise with mean 0 and variance 0.5, and w takes ± 10 randomly with equal probability. For the other nodes, let $x_i(0) := b + \epsilon$ ($i \in \mathbb{V} \setminus \{j\}$). Then, we say that the anomaly detection is successful if only $|x_j(m)|$ is larger than the threshold $d \in \mathbb{R}_+$. Under this setting, the success rates for 1000 trials are summarized in Table I, where by considering the differences in the algorithms and the filter functions, we choose the different thresholds $d := 5$ and $d := 9$ for the proposed algorithm and the existing algorithm, respectively. Table I indicates that the detection performance of the

proposed algorithm is higher than that of the existing algorithm for all of $b := 10$, $b := 20$, and $b := 30$. The reason for this is explained as follows. As seen in Fig. 4, the existing algorithm for h_{10}^r cannot sufficiently reduce the states of the nodes other than that to be detected. As a result, the existing algorithm cannot handle the case where $|x_j(0)|$ is smaller than the other states. Therefore, the anomaly detection for $w = -10$ fails, demonstrated as the success rate 51.2 % for $b := 10$. In addition, the existing algorithm cannot handle the different values of b by the common threshold d because the remaining states depend on b. This leads to the differences in the success rates for $b := 20$ and $b := 30$.

Remark 2: From (6)–(8), (12)–(14), and (18), the proposed algorithm does not depend on the graph G. However, the resulting signal values depend on G because the eigenvalues of the graph Laplacian L, *i.e.,* the distribution of the frequency components of the associated graph signal, change depending on G.

IV. CONCLUSION

In this letter, we improved an existing DSF method for networked systems to overcome the difficulty of the limited achievable filter characteristics. In particular, we extended the existing one-hop filtering algorithm to the two-hop version. We then derived a necessary and sufficient condition on the filter function for achieving DSF. The advantage of the proposed algorithm is that the filter functions described as polynomials with complex roots can be handled, unlike the existing algorithm.

APPENDIX
PROOF OF THEOREM 1

We begin with the proof of the "only if" part. It is assumed that for a filter function h, DSF of the graph signal $(G, x(0))$ is achieved by the system Σ. From (6)–(8), the behavior of Σ is described as

$$x(t+1) = \ell_0(t)x(t) - \ell_1(t)Lx(t) - \ell_2(t)Ly(t)$$
$$= (\ell_0(t)I - \ell_1(t)L + \ell_2(t)L^2)x(t) \quad (9)$$

for $y(t) := [y_1(t) \ y_2(t) \ \cdots \ y_n(t)]^\top$. Hence, the relation between $x(0)$ and $x(m)$ is written as

$$x(m) = \left(\prod_{t=1}^{m}(\ell_0(m-t)I - \ell_1(m-t)L + \ell_2(m-t)L^2)\right)x(0)$$
$$= \left(\prod_{t=1}^{m}(\ell_0(m-t)VV^\top - \ell_1(m-t)V\Lambda V^\top\right.$$
$$\left. + \ell_2(m-t)V\Lambda V^\top V\Lambda V^\top)\right)x(0)$$
$$= V\left(\prod_{t=1}^{m}(\ell_0(m-t)I - \ell_1(m-t)\Lambda\right.$$
$$\left. + \ell_2(m-t)\Lambda^2)\right)V^\top x(0), \quad (10)$$

where the second equality is derived from $V^\top = V^{-1}$ and $V^\top LV = \Lambda$ and the last one follows because $\ell_0(t)$, $\ell_1(t)$, and $\ell_2(t)$ are scalars and $V^\top V = I$ holds. Noting that the matrix

$\ell_0(m-t)I - \ell_1(m-t)\Lambda + \ell_2(m-t)\Lambda^2$ in (10) is diagonal for every $t \in \{1, 2, \ldots, m\}$, we obtain

$$x(m) = V\text{diag}\left(\prod_{t=1}^{m}(\ell_0(m-t) - \ell_1(m-t)\lambda_1\right.$$
$$+ \ell_2(m-t)\lambda_1^2), \prod_{t=1}^{m}(\ell_0(m-t) - \ell_1(m-t)\lambda_2$$
$$+ \ell_2(m-t)\lambda_2^2), \ldots, \prod_{t=1}^{m}(\ell_0(m-t)$$
$$\left. - \ell_1(m-t)\lambda_n + \ell_2(m-t)\lambda_n^2)\right)V^\top x(0). \quad (11)$$

Besides, $x(m) = x(\infty)$ holds due to (6)–(8) and $\ell_0(t) = 1$, $\ell_1(t) = 0$, and $\ell_2(t) = 0$ for $t \geq m$. By substituting this for (11) and considering $x(0)$ and $x(\infty)$ as s and $\tilde{s}$ in (2), respectively, we can show $h(\lambda) = \prod_{t=1}^{m}(\ell_0(m-t) - \ell_1(m-t)\lambda + \ell_2(m-t)\lambda^2)$ from the assumption of achieving DSF by Σ. Because of $\ell_0(t) \in \mathbb{R} \setminus \{0\}$ and $\ell_1(t), \ell_2(t) \in \mathbb{R}$ for every $t \in \{1, 2, \ldots, m\}$, this h is a real polynomial of λ with at most degree $2m$ and non-zero roots. Therefore, the "only if" part is proven.

Next, we give the proof of the "if" part. It is assumed that a real polynomial of λ with at most degree $2m$ and non-zero roots is given as the filter function h. Because the complex roots of real polynomials come in complex conjugate pairs [9], we denote the complex roots of h by $\alpha_j \pm \beta_j\text{i}$ ($j = 1, 2, \ldots, m_c$), where $\text{i} := \sqrt{-1}$ and $m_c \in \mathbb{R}_+ \cup \{0\}$ is the number of the pairs. Further, we let $\alpha_j \in \mathbb{R} \setminus \{0\}$ ($j = m_c + 1, m_c + 2, \ldots, \mu$) be the real roots of h, where $\mu := m_c + m_r$ for the number m_r of the real roots. In the following, we assume that the degree of h is even. Then, m_r is even from the above property of the complex roots of real polynomials.

With the above notations and $\bar{\mu} := m_c + m_r/2$, we set

$$\ell_0(t) := \begin{cases} a_0 & \text{if } t = 0, \\ 1 & \text{otherwise,} \end{cases} \quad (12)$$

$$\ell_1(t) := \begin{cases} \dfrac{2a_0\alpha_1}{\alpha_1^2 + \beta_1^2} & \text{if } t = 0, \\[2mm] \dfrac{2\alpha_{t+1}}{\alpha_{t+1}^2 + \beta_{t+1}^2} & \text{if } t \in \{1, 2, \ldots, m_c - 1\}, \\[2mm] \dfrac{\alpha_{t+1} + \alpha_{t+m_r/2+1}}{\alpha_{t+1}\alpha_{t+m_r/2+1}} & \\[1mm] & \text{if } t \in \{m_c, m_c + 1, \ldots, \bar{\mu} - 1\}, \\[2mm] 0 & \text{otherwise,} \end{cases} \quad (13)$$

$$\ell_2(t) := \begin{cases} \dfrac{a_0}{\alpha_1^2 + \beta_1^2} & \text{if } t = 0, \\[2mm] \dfrac{1}{\alpha_{t+1}^2 + \beta_{t+1}^2} & \text{if } t \in \{1, 2, \ldots, m_c - 1\}, \\[2mm] \dfrac{1}{\alpha_{t+1}\alpha_{t+m_r/2+1}} & \\[1mm] & \text{if } t \in \{m_c, m_c + 1, \ldots, \bar{\mu} - 1\}, \\[2mm] 0 & \text{otherwise,} \end{cases} \quad (14)$$

where a_0 is the zero degree term of h, as shown in (5). From $m \geq \bar{\mu}$, these gains satisfy $\ell_0(t) = 1$, $\ell_1(t) = 0$, and $\ell_2(t) = 0$ for $t \geq m$, where the inequality on m is derived from the facts

that the degree of h is at most $2m$ and the number of the roots of h is $2m_c + m_r$. From (12)–(14), the following equality holds for every $i \in \mathbb{V}$:

$$\prod_{t=1}^{m}(\ell_0(m-t) - \ell_1(m-t)\lambda_i + \ell_2(m-t)\lambda_i^2)$$

$$= a_0 \left(\prod_{t=0}^{m_r/2-1} \left(1 - \frac{(\alpha_{\bar{\mu}-t} + \alpha_{\mu-t})\lambda_i - \lambda_i^2}{\alpha_{\bar{\mu}-t}\alpha_{\mu-t}} \right) \right)$$

$$\times \left(\prod_{t=0}^{m_c-2} \left(1 - \frac{2\alpha_{m_c-t}\lambda_i - \lambda_i^2}{\alpha_{m_c-t}^2 + \beta_{m_c-t}^2} \right) \right) \left(1 - \frac{2\alpha_1\lambda_i - \lambda_i^2}{\alpha_1^2 + \beta_1^2} \right). \quad (15)$$

By applying this equality and $x(m) = x(\infty)$ to (11) and by a discussion similar to the above, we see that if the right-hand side of (15) is equivalent to the given h for every $i \in \mathbb{V}$, the system Σ achieves DSF of the graph signal $(G, x(0))$ for h.

To show the equivalence of the right-hand side of (15) and h, we factorize h as $h(\lambda) = a_{m_a} \prod_{j=0}^{m_a-1}(\lambda - r_{m_a-j})$ for $m_a := 2m_c + m_r$. Noting that m_r is even as explained above, we obtain

$$a_{m_a} \prod_{j=0}^{m_a-1} (\lambda - r_{m_a-j})$$

$$= a_{m_a} \left(\prod_{j=0}^{m_c-1} (\lambda - \alpha_{m_c-j} - \beta_{m_c-j}\mathrm{i})(\lambda - \alpha_{m_c-j} \right.$$

$$+ \beta_{m_c-j}\mathrm{i}) \left. \right) \left(\prod_{j=0}^{m_r/2-1} (\lambda - \alpha_{\bar{\mu}-j})(\lambda - \alpha_{\mu-j}) \right)$$

$$= a_{m_a} \left(\prod_{j=0}^{m_c-1} (\lambda^2 - 2\alpha_{m_c-j}\lambda + \alpha_{m_c-j}^2 + \beta_{m_c-j}^2) \right)$$

$$\times \left(\prod_{j=0}^{m_r/2-1} (\lambda^2 - (\alpha_{\bar{\mu}-j} + \alpha_{\mu-j})\lambda + \alpha_{\bar{\mu}-j}\alpha_{\mu-j}) \right)$$

$$= a_{m_a} \left(\prod_{j=0}^{m_c-1} \left(\alpha_{m_c-j}^2 + \beta_{m_c-j}^2 \right) \right)$$

$$\times \left(\prod_{j=0}^{m_c-1} \left(\frac{\lambda^2 - 2\alpha_{m_c-j}\lambda}{\alpha_{m_c-j}^2 + \beta_{m_c-j}^2} + 1 \right) \right) \left(\prod_{j=0}^{m_r/2-1} \alpha_{\bar{\mu}-j}\alpha_{\mu-j} \right)$$

$$\times \left(\prod_{j=0}^{m_r/2-1} \left(\frac{\lambda^2 - (\alpha_{\bar{\mu}-j} + \alpha_{\mu-j})\lambda}{\alpha_{\bar{\mu}-j}\alpha_{\mu-j}} + 1 \right) \right). \quad (16)$$

In (16), we notice that $a_{m_a}(\prod_{j=0}^{m_c-1}(\alpha_{m_c-j}^2 + \beta_{m_c-j}^2))(\prod_{j=0}^{m_r/2-1} \alpha_{\bar{\mu}-j}\alpha_{\mu-j}) = a_0$ holds because the zero degree term of h is a_0. Hence, it follows that

$$h(\lambda) = a_0 \left(\prod_{j=0}^{m_c-1} \left(\frac{\lambda^2 - 2\alpha_{m_c-j}\lambda}{\alpha_{m_c-j}^2 + \beta_{m_c-j}^2} + 1 \right) \right)$$

$$\times \left(\prod_{j=0}^{m_r/2-1} \left(\frac{\lambda^2 - (\alpha_{\bar{\mu}-j} + \alpha_{\mu-j})\lambda}{\alpha_{\bar{\mu}-j}\alpha_{\mu-j}} + 1 \right) \right). \quad (17)$$

Replacing λ in (17) by λ_i ($i \in \mathbb{V}$) shows the equivalence of the right-hand side of (15) and h for every $i \in \mathbb{V}$, which proves the "if" part by the above discussion. Also in the case that the degree of h is odd, we can prove the statement in a similar way, where $\ell_1(t)$ in (13) for $t \geq m_c$ is replaced by

$$\ell_1(t) := \begin{cases} \dfrac{\alpha_{t+1} + \alpha_{t+(m_r-1)/2+1}}{\alpha_{t+1}\alpha_{t+(m_r-1)/2+1}} \\ \quad \text{if } t \in \{m_c, m_c + 1, \dots, m_c + (m_r - 1)/2 - 1\}, \\ \dfrac{1}{\alpha_\mu} \qquad \text{if } t = m_c + (m_r - 1)/2, \\ 0 \qquad \text{otherwise.} \end{cases} \quad (18)$$

This completes the proof.

ACKNOWLEDGMENT

REFERENCES

[1] S. Izumi, S. Azuma, and T. Sugie, "Analysis and design of multi-agent systems in spatial frequency domain: Application to distributed spatial filtering in sensor networks," *IEEE Access*, vol. 8, pp. 34909–34918, 2020.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[3] S. Izumi, R. Katayama, X. Xin, and T. Yamasaki, "Distributed spatial filtering over networked systems," *IEEE Contr. Syst. Lett.*, vol. 5, pp. 617–622, 2021.

[4] A. Sarlette, "Performance limitations for distributed systems via spatial-frequency Bode integrals," in *Proc. Eur. Control Conf.*, 2013, pp. 2274–2279.

[5] S. Mollaebrahim and B. Beferull-Lozano, "Distributed linear network operators based on successive graph shift matrices," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 9, pp. 315–328, 2023.

[6] J.-W. Yi, L. Chai, and J. Zhang, "Average consensus by graph filtering: New approach, explicit convergence rate, and optimal design," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 191–206, Jan. 2020.

[7] W. Gosrich et al., "Coverage control in multi-robot systems via graph neural networks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 8787–8793.

[8] L. Yuan and H. Ishii, "Event-triggered approximate Byzantine consensus with multi-hop communication," *IEEE Trans. Signal Process.*, vol. 71, pp. 1742–1754, 2023.

[9] B. Fine and G. Rosenberger, *The Fundamental Theorem of Algebra*. New York, NY, USA: Springer, 1997.