Nonsubsampled Graph Filter Banks: Theory and Distributed Algorithms

Junzheng Jiang, *Member, IEEE*, Cheng Cheng, and Qiyu Sun

***Abstract*—In this paper, we consider nonsubsampled graph filter banks (NSGFBs) to process data on a sparse graph. The analysis filter banks of NSGFBs have small bandwidth, pass/block the normalized constant signal, and have stability on ℓ^2. Given an analysis filter bank with small bandwidth, we introduce algebraic and optimization methods to construct well-localized synthesis filter banks such that the corresponding NSGFBs provide a perfect signal reconstruction in the noiseless setting. We also prove that the proposed NSGFBs can control the resonance effect in the presence of bounded noise and they can limit the influence of shot noise primarily to a small neighborhood near its location on the graph. We later introduce an iterative algorithm to implement the proposed NSGFBs in a distributed manner, and develop an NSGFB-based denoising technique which is demonstrated to have satisfactory performance on noise suppression.**

***Index Terms*—Graph signal processing, spatially distributed network, graph filter bank, nonsubsampled, distributed algorithm.**

I. Introduction

SPATIALLY distributed networks (SDNs) have an agent at each location equipped with some data processing and communication subsystems, and they have a fusion center with limited computing capacity or do not have a fusion center at all. SDNs have been widely used in (wireless) sensor networks, smart grids and many real world applications [1]–[7]. Our representative SDNs are distributed over a spatial domain with agents communicating with each other via signal broadcasting within a finite range. Data collected by an SDN resides naturally on vertices of a graph. Graph signal processing provides an innovative framework to process data on graphs. Many concepts in classical signal processing, such as the Fourier transform, wavelet transform and filter banks, have been extended to graph settings in recent years. However there are still lots of fundamental problems unexplored or not completely answered [8]–[13].

Manuscript received October 24, 2018; revised March 7, 2019 and May 6, 2019; accepted May 20, 2019. Date of publication June 11, 2019; date of current version June 28, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yuichi Tanaka. This work was supported in part by the National Natural Science Foundation of China under Grant 61761011, in part by SAMSI under the National Science Foundation Grant DMS-1638521, and in part by the National Science Foundation under Grant DMS-1412413 and under Grant 1816313). *(Corresponding author: Cheng Cheng.)*

J. Jiang is with the School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: jzjiang@guet.edu.cn).

C. Cheng is with the Department of Mathematics, Duke University, and Statistical and Applied Mathematical Sciences Institute, Durham, NC 27708 USA (e-mail: cheng87@math.duke.edu).

Q. Sun is with the Department of Mathematics, University of Central Florida, Orlando, FL 32816 USA (e-mail: qiyu.sun@ucf.edu).

Digital Object Identifier 10.1109/TSP.2019.2922160

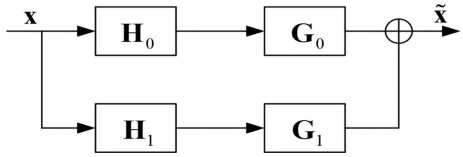


Fig. 1. Block diagram of an NSGFB with analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$, where $\mathbf{x}$ is the input of the NSGFB and $\tilde{\mathbf{x}}$ is its output.

The wavelet transform is one of the most prominent techniques to process signals in regular domains [14]–[16]. During the past decades, graph wavelet transforms have been introduced and some of them are designed to use the eigenvalue and eigenspace information of the graph Laplacian matrix [17]–[21]. Graph wavelet transform is under the same theoretical structure as graph filter banks, and the corresponding wavelet filter banks carry downsampling and upsampling procedures [9], [10], [22]–[28]. Several forms of the downsampling and upsampling have been defined by the partitioned graph coloring in [22], the maximum spanning tree structure of the graph in [26], and the SVD decomposition of the graph Laplacian matrix in [27]. A proper definition of the downsampling and upsampling procedure is not obvious especially when the residing graph is of large order and it has complicated topological structure. This motivates us to consider a *nonsubsampled graph filter bank* (NSGFB) with downsampling and upsampling procedures circumvented; see Figure 1 for its block diagram.

An NSGFB contains an analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and a synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$. An important concept for an NSGFB is the *perfect reconstruction* condition so that the output $\tilde{\mathbf{x}}$ in Figure 1 is always the same as the input $\mathbf{x}$. Due to the nonsubsampled structure in an NSGFB, the perfect reconstruction condition can be easily characterized by

$$\mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1 = \mathbf{I}, \tag{1}$$

where $\mathbf{I}$ is the identity matrix of appropriate size. An equivalent statement to the above matrix equation is that columns of $\mathbf{G}_0$ and $\mathbf{G}_1$ form a dual frame to columns of $\mathbf{H}_0^T$ and $\mathbf{H}_1^T$. The frame and wavelet approach to NSGFBs on graphs has been discussed in [21], [29], [30], where the analysis/synthesis filter banks are polynomials of a graph shift matrix or they are well approximated by Chebyshev polynomials of a graph shift matrix. In this paper, we work on graphs with complicated topological structures and we consider designing NSGFBs with some desired features, such as spectral decomposition property of the analysis procedure, robustness of the analysis/synthesis procedure against bounded input noises and bounded subband processing

errors, and distributed implementation of the analysis/synthesis procedure.

The analysis procedure in an NSGFB on a graph $\mathcal{G} := (V, E)$ decomposes an input graph signal $\mathbf{x}$ into two subband signals $\mathbf{z}_0 = \mathbf{H}_0\mathbf{x}$ and $\mathbf{z}_1 = \mathbf{H}_1\mathbf{x}$. To apply an NSGFB to some real world applications, such as noise suppression and abnormal phenomenon detection, two subband signals $\mathbf{z}_0$ and $\mathbf{z}_1$ should carry different frequency information of the input signal $\mathbf{x}$ [22], [23], [25], [31]. In this paper, analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ are designed to pass/block the normalized constant signal $\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1}$,

$$\mathbf{H}_0\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1} = \mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1} \text{ and } \mathbf{H}_1\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1} = \mathbf{0}, \tag{2}$$

where $\mathbf{D}_{\mathcal{G}} = (D_{\mathcal{G}}(i,j))_{i,j\in V}$ is the degree matrix of the graph $\mathcal{G}$. Our representative analysis filter banks are *spline filter banks* $(\mathbf{H}_{0,n}^{\text{spln}}, \mathbf{H}_{1,n}^{\text{spln}})$ of order $n \geq 1$,

$$\mathbf{H}_{0,n}^{\text{spln}} = \left(\mathbf{I} - \frac{1}{2}\mathbf{L}_{\mathcal{G}}^{\text{sym}}\right)^n \text{ and } \mathbf{H}_{1,n}^{\text{spln}} = \left(\frac{1}{2}\mathbf{L}_{\mathcal{G}}^{\text{sym}}\right)^n, \tag{3}$$

and the *node-variant filter bank* $(\mathbf{H}_0^{\text{nv}}, \mathbf{H}_1^{\text{nv}})$ given by

$$\mathbf{H}_0^{\text{nv}} = \mathbf{I} - \alpha^{-1}\mathbf{D}_{\mathcal{G}}\mathbf{L}_{\mathcal{G}}^{\text{sym}},\ \mathbf{H}_1^{\text{nv}} = \alpha^{-1}\mathbf{D}_{\mathcal{G}}\mathbf{L}_{\mathcal{G}}^{\text{sym}}, \tag{4}$$

where $\alpha = \max_{1\leq i\leq N} D_{\mathcal{G}}(i,i)$ is a degree normalization constant, $\mathbf{A}_{\mathcal{G}}$ and $\mathbf{L}_{\mathcal{G}}^{\text{sym}} = \mathbf{D}_{\mathcal{G}}^{-1/2}(\mathbf{D}_{\mathcal{G}} - \mathbf{A}_{\mathcal{G}})\mathbf{D}_{\mathcal{G}}^{-1/2}$ are the adjacency matrix and symmetric normalized Laplacian matrix of the graph $\mathcal{G}$ respectively. The spline filter banks were introduced in [32] as graph-spline wavelet transform for circulant graph setting, while the node-variant filter bank has been used in [40].

In practical applications, the input $\mathbf{x}$ is the original signal $\mathbf{x}_o$ corrupted by an additive noise $\boldsymbol{\epsilon}$. For an SDN, the agent at each vertex operates almost independently and the noise that arises at each vertex is usually contained in some range [7]. We then believe that a reasonable fidelity measure to assess the robustness of NSGFBs is the bounded difference $\|\tilde{\mathbf{x}} - \mathbf{x}_o\|_\infty$ between the original signal $\mathbf{x}_o$ and the output signal $\tilde{\mathbf{x}}$ of the NSGFB in the noise environment, instead of the standard least squares error $\|\tilde{\mathbf{x}} - \mathbf{x}_o\|_2$, where for $1 \leq p \leq \infty$, ℓ^p is the space of all p-summable sequences with norm $\|\cdot\|_p$ [7], [33], [34]. In this paper, NSGFBs are designed to be robust against bounded input noises and bounded subband processing errors.

Filter banks can be implemented either in a centralized system or a distributed system. In a centralized system, the fusion center receives data from agents at vertices, performs designed data processing and sends the processed data back to agents. Most filter banks on graphs are designed for centralized processing, however for the implementation of filter banks on an SDN of large size, a centralized system may suffer from high computational burden and call for significant efforts to create a data exchange network. For signal processing on an SDN or a graph of large order, a distributed system provides an indispensable tool. The distributed system has been used for signal sampling and reconstruction on an SDN, graph signal inpainting and economic dispatch in power networks. The reader may refer to [7], [35], [36] and references therein on distributed implementations of signal processing on graphs. In this paper, we design well-localized NSGFBs with analysis filter banks having small bandwidths and synthesis filter banks having either small bandwidths or exponentially off-diagonal decay, so that the corresponding analysis and synthesis procedure is applicable in spatially distributed systems with each agent equipped with limited data processing and communication abilities.

The objective of this paper is to study well-localized NSGFBs on a distributed system from designs to algorithms, and then to noise suppression.

A. Main Contributions

Given a stable analysis filter bank, the existence of synthesis filter banks is theoretically guaranteed so that the corresponding NSGFB satisfies the perfect reconstruction condition (1) [15], [16]. It is highly nontrivial to design well-localized synthesis filter banks that can be fulfilled in a distributed system due to the complexity of graphs to describe SDNs of large size, cf. [21]–[24] and [28]. The first contribution of this paper is that two novel methods are proposed to construct well-localized synthesis filter banks such that the corresponding NSGFB provides a perfect reconstruction in a noiseless environment, see Sections V and VI. In the first approach, the analysis filter banks are polynomials of the symmetric normalized Laplacian matrix $\mathbf{L}_{\mathcal{G}}^{\text{sym}}$, and the synthesis filter banks are designed by solving a Bezout identity for polynomials, cf. [15], [16], [37]. Their bandwidth can be no larger than the bandwidth of the analysis filter banks. In the second approach, analysis filter banks are not necessarily to be polynomials of the symmetric normalized Laplacian matrix $\mathbf{L}_{\mathcal{G}}^{\text{sym}}$, and the synthesis filter banks are the solution of some constrained optimization problems. The synthesis filter bank is proved to have an exponential off-diagonal decay, and hence the output of the corresponding NSGFB suffers primarily in a small neighborhood of vertices where agents lose data processing ability and/or communication capability.

The robustness of an NSGFB is of paramount importance. The second contribution of this paper is that for the NSGFB with analysis filter bank having small bandwidth and synthesis filter banks obtained from our approaches, we establish a quantitative estimate on the bounded difference $\|\tilde{\mathbf{x}} - \mathbf{x}_o\|_\infty$, which is *independent* on the order of the graph. This indicates that the proposed NSGFB can control the resonance effect in the presence of bounded additive noises, see Propositions V.2 and VI.4.

For NSGFB on SDNs, data processing should be implemented in a distributed manner. It is observed that synthesis filter banks in the second approach may not have the small bandwidth and they may not be well approximated by Chebyshev polynomials of the symmetric normalized Laplacian matrix. This leads to the third contribution of this paper, an iterative algorithm in Section VII to implement the synthesis procedure in a distributed manner. The proposed algorithm includes a graph decomposition of overlapping regions, local synthesis procedure in each region, and appropriate adjustment over overlapped vertices in each iteration. The approach is motivated by the observations that solutions of some global optimization problem can be well approximated by a weighted sum of solutions to local optimization problems [7] and local synthesis procedures have boundary effects, see Figure 4. The novelty of the proposed iterative algorithm is that we decompose the network into a family of overlapping neighboring subnetworks of appropriate sizes (6), which ensures every agent in the network is located in the region of some neighboring subnetworks that has negligible boundary effects of the local synthesis procedure. Moreover, the proposed iterative algorithm is scalable and implementable in SDNs without a fusion center, and it has linear complexity.

As an application of NSGFBs, we develop a distributed denoising technique that has satisfactory performance on noise suppression, which is the fourth contribution of this paper.

B. Organization

In Section II, we briefly review some fundamental concepts of graphs and introduce an overlapping graph decomposition (6). In Section III, we introduce the concept and properties of graph filters on $\ell^p, 1 \le p \le \infty$. In Section IV, we discuss analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ having small bandwidth and ℓ^2-stability and passing/blocking the normalized constant signal. In Section V, we propose an algebraic design of synthesis filter banks $(\mathbf{G}_0, \mathbf{G}_1)$ when analysis filters $\mathbf{H}_0$ and $\mathbf{H}_1$ are polynomials of the symmetric normalized Laplacian on the graph. In Section VI, we construct synthesis filter banks $(\mathbf{G}_0, \mathbf{G}_1)$ by solving some constrained optimization problem. In Section VII, we propose an exponentially convergent iterative algorithm to implement the synthesis procedure. In Section VIII, we introduce a distributed denoising technique associated with spline/node-variant NSGFBs and demonstrate its performance for signal denoising on graphs of large order. All proofs are collected in the appendices.

C. Notation

We use the common convention of representing matrices and vectors with boldface letters and scalars with normal letters. For a matrix $\mathbf{A}$, denote its transpose, pseudo-inverse, trace, Frobenius norm and operator norm on $\ell^p, 1 \le p \le \infty$, by $\mathbf{A}^T, \mathbf{A}^\dagger, \mathrm{tr}(\mathbf{A}), \|\mathbf{A}\|_F$ and $\|\mathbf{A}\|_{\mathcal{B}_p}$ respectively. For a graph $\mathcal{G} := (V, E)$, define the geodesic distance $\rho(i, j)$ between vertices i and $j \in V$ in a connected component of the graph $\mathcal{G}$ by the number of edges in a shortest path connecting them, and set the distance $\rho(i, j) = \infty$ if vertices i and $j \in V$ belong to different connected components. For a scalar t, let $\mathbf{t}$ be the vector of appropriate size with all entries taking value t. For a set F, denote its cardinality and indicator function by $\mu(F)$ and χ_F respectively, where $\chi_F(i) = 1$ if $i \in F$ and zero otherwise. For a polynomial P, we denote its degree by $\deg(P)$.

II. Preliminaries on Graphs

Let $\mathcal{G} := (V, E)$ be a graph, where $V = \{1, 2, \ldots, N\}$ is the set of vertices and E is the set of edges. Throughout this paper, we presume that the graph $\mathcal{G}$ has the following global features [7], [8]:

- The graph $\mathcal{G}$ is *simple*, i.e., it is undirected and unweighted, and it does not contain self-loops and multiple edges.
- The graph $\mathcal{G}$ has *polynomial growth*, i.e., numbers of r-hop neighbors of any vertex are dominated by a polynomial about $r \ge 0$.

A quantitative description of the second feature is that the counting measure μ on the graph $\mathcal{G}$ satisfies

$$\mu(B(i, r)) \le D_1(\mathcal{G})(r + 1)^d \text{ for all } i \in V \text{ and } r \ge 0, \quad (5)$$

where $B(i, r) := \{j \in V : \ \rho(i, j) \le r\}$ contains all r-hop neighbors of a vertex $i \in \mathcal{G}$. The minimal positive constants d and $D_1(\mathcal{G})$ in (5) are known as *Beurling dimension* and *density* of the graph $\mathcal{G}$ respectively. We remark that a graph $\mathcal{G}$ with the above two global features is sparse and its Beurling dimension is not necessarily to be a positive integer [7].

For a graph $\mathcal{G} = (V, E)$, define the *r-neighboring subgraph* of $i \in V$ by $\mathcal{G}_{i,r} := (B(i, r), E(i, r))$, where $E(i, r)$ contains all edges of the graph $\mathcal{G}$ with endpoints in $B(i, r)$. Then for $r \ge 1$, we can decompose the graph $\mathcal{G}$ into a family of overlapping subgraphs $\mathcal{G}_{i,r}, i \in V$, of diameters at most $2r$,

$$\mathcal{G} = \cup_{i \in V} \mathcal{G}_{i,r}. \quad (6)$$

The decomposition (6) plays a crucial role in the proposed distributed algorithm for an NSGFB. The selection of the radius parameter r in (6) depends on Beurling dimension d and density $D_1(\mathcal{G})$ of the graph $\mathcal{G}$; see Theorem VII.2. Accordingly, we expect that the Beurling dimension d and density $D_1(\mathcal{G})$ of the graph $\mathcal{G}$ are much smaller than (or even independent on) the order of the graph. Shown in Figure 2 are two representative graphs that are reproduced by the GSPToolbox [38].

- The Minnesota traffic graph has two connected components and it has 2642 vertices and 3303 edges, where each vertex represents a spatial location in the state of Minnesota equipped with a traffic monitoring sensor and each edge denotes a direct communication link between monitoring sensors [22], [23].
- The random geometric graph RGG_N is connected and it has N vertices randomly deployed in the region $[0, 1]^2$ with an edge existing between two vertices if their physical distance is not larger than $\sqrt{2}N^{-1/2}$ [27], [39].

III. Graph Signal and Filtering

A *signal* $\mathbf{x}$ residing on the graph $\mathcal{G}$ is a column vector $(x(i))_{i \in V}$, where $x(i)$ refers to the signal value at vertex $i \in V$. In SDNs and many real world applications, it is natural to assume that signals collected in the dataset reside in some sequence space $\ell^p, 1 \le p \le \infty$ [7], [33], [34].

A *filter* $\mathbf{A}$ on the graph $\mathcal{G}$ is a linear transformation from one signal $\mathbf{x}$ on $\mathcal{G}$ to another signal $\mathbf{y} = \mathbf{A}\mathbf{x}$ on $\mathcal{G}$, which is usually represented by a matrix $\mathbf{A} = (a(i, j))_{i,j \in V}$. Graph filters are widely designed to be either polynomials of a graph shift matrix, or well approximated by Chebyshev polynomials of a graph shift matrix [10], [12], [13], [21]–[24], [27], [32], [36], [40]. A favorite example of graph shift matrices is the symmetric normalized Laplacian matrix $\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}$ whose eigenvalues are contained in the interval $[0, 2]$ [10], [22]–[24], [27], [32]. Exemplary filters of this paper are

$$\mathbf{A} = P(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}) = p_0\mathbf{I} + \sum_{l=1}^{L} p_l(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}})^l, \quad (7)$$

where $P(t) = \sum_{l=0}^{L} p_l t^l$ for some coefficients $p_l, 0 \le l \le L$.

To develop a distributed algorithm for an NSGFB, analysis filters will be designed to have small bandwidths.

Definition III.1: The bandwidth $\sigma := \sigma(\mathbf{A})$ of a graph filter $\mathbf{A} = (a(i, j))_{i,j \in V}$ is the smallest nonnegative integer such that $a(i, j) = 0$ hold for all $i, j \in V$ with $\rho(i, j) > \sigma$. For a filter pair $(\mathbf{A}, \mathbf{B})$, we define its bandwidth $\sigma := \sigma(\mathbf{A}, \mathbf{B})$ by $\max(\sigma(\mathbf{A}), \sigma(\mathbf{B}))$.

For a nonzero graph filter $\mathbf{A}$, the bandwidth $\sigma(\mathbf{A})$ in the above definition is the same as its another common definition, which is the maximal integer $\tilde{\sigma}$ such that $a(i, j) \ne 0$ for some $i, j \in V$ with $\rho(i, j) = \tilde{\sigma}$. An advantage of a filter $\mathbf{A} = (a(i, j))_{i,j \in V}$ with small bandwidth $\sigma \ge 0$ is that the corresponding filtering procedure can be implemented in a distributed manner. Given an input signal $\mathbf{x} = (x(i))_{i \in V}$, the signal value $z(k)$ of the output signal $\mathbf{z} = (z(k))_{k \in V} = \mathbf{A}\mathbf{x}$ at the vertex k is a "weighted" sum of signal values of the input $\mathbf{x}$ in its σ-neighborhood,

$$z(k) = \sum_{\rho(i,k) \le \sigma} a(k, i)x(i), \ \ k \in V. \quad (8)$$

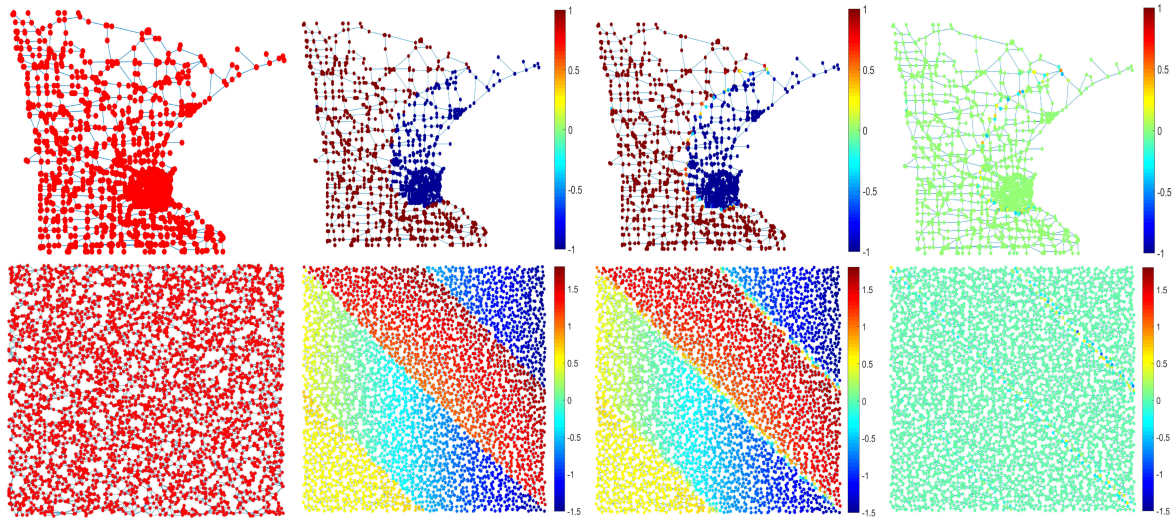


Fig. 2. Plotted on the first row from left to right are the Minnesota traffic graph, the original blockwise constant signal $\mathbf{x}$ that has only two values ± 1 [22], [23], the lowpass subband signal $\mathbf{H}_{0,2}^{\mathrm{spln}}\mathbf{x}$ and the highpass subband signal $\mathbf{H}_{1,2}^{\mathrm{spln}}\mathbf{x}$. On the second row from left to right are a random geometric graph with $N = 4096$, the original blockwise polynomial signal $\mathbf{x}$ consisting of four strips and imposing the polynomial $0.5 - 2c_x$ on the first and third diagonal strips and $0.5 + c_x^2 + c_y^2$ on the second and fourth strips respectively, where (c_x, c_y) are the coordinates of vertices [27], the lowpass subband signal $\mathbf{H}_{0,2}^{\mathrm{spln}}\mathbf{x}$ and the highpass subband signal $\mathbf{H}_{1,2}^{\mathrm{spln}}\mathbf{x}$.

The above distributed algorithm requires that each vertex k shares signal values with its σ-hop neighbors, memorizes entries $a(k,i), i \in B(k,\sigma)$, of the k-th row of the filter $\mathbf{A}$, and has computational capability to perform additions and multiplications in (8). From the above argument, we see that a graph filter of the form (7) has bandwidth $\sigma \le \deg(P)$ and the corresponding filtering procedure can be implemented in a distributed manner.

A filter $\mathbf{A}$ is expected to map a signal with finite energy to another signal with finite energy and a bounded signal to another bounded signal. A quantitative description of the above filtering procedure is

$$\|\mathbf{A}\mathbf{x}\|_p \le C\|\mathbf{x}\|_p \text{ for all } \mathbf{x} \in \ell^p, \tag{9}$$

where $1 \le p \le \infty$ and C is a positive constant.

Definition III.2: We say that $\mathbf{A}$ is a *graph filter* on ℓ^p if (9) is satisfied, and we call the minimal constant C for (9) to hold, denoted by $\|\mathbf{A}\|_{\mathcal{B}_p}$, the *filter bound* on ℓ^p.

Let $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_N \le 2$ be eigenvalues of the symmetric normalized Laplacian matrix $\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}$, and write

$$\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}} = \mathbf{U}^T\mathbf{\Lambda}\mathbf{U}, \tag{10}$$

where $\mathbf{U}^T = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ is an orthogonal matrix and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix. Then the matrix $\mathbf{A}$ in (7) satisfies

$$\mathbf{A} = \mathbf{U}^T P(\mathbf{\Lambda})\mathbf{U}, \tag{11}$$

and hence it is a symmetric matrix with eigenvalues $P(\lambda_1), \ldots, P(\lambda_N)$. Thus the filter bound $\|\mathbf{A}\|_{\mathcal{B}_2}$ of the graph filter $\mathbf{A}$ on ℓ^2 can be evaluated explicitly,

$$\|\mathbf{A}\|_{\mathcal{B}_2} = \sup_{1\le n\le N} |P(\lambda_n)| \le \sup_{0\le t\le 2} |P(t)|. \tag{12}$$

To estimate $\|\mathbf{A}\|_{\mathcal{B}_p}, p \ne 2$, of a graph filter $\mathbf{A} = (a(i,j))_{i,j\in V}$, we define the *bound* of $\mathbf{A}$ by

$$\|\mathbf{A}\|_\infty = \sup_{i,j\in V} |a(i,j)|, \tag{13}$$

and the *Schur norm* of $\mathbf{A}$ by

$$\|\mathbf{A}\|_{\mathcal{S}} = \max\left(\sup_{i\in V}\sum_{j\in V}|a(i,j)|,\ \sup_{j\in V}\sum_{i\in V}|a(i,j)|\right).$$

Following the argument used in [7], we can show that a bounded filter with small bandwidth is a graph filter on $\ell^p, 1 \le p \le \infty$, with filter bound dominated by some constant independent of the order of the graph $\mathcal{G}$.

Proposition III.3: Let $1 \le p \le \infty$ and $\mathbf{A}$ be a bounded graph filter with bandwidth σ. Then

$$\|\mathbf{A}\|_\infty \le \|\mathbf{A}\|_{\mathcal{B}_p} \le \|\mathbf{A}\|_{\mathcal{S}} \le D_1(\mathcal{G})(\sigma+1)^d\|\mathbf{A}\|_\infty, \tag{14}$$

where d and $D_1(\mathcal{G})$ are the Beurling dimension and density of the graph $\mathcal{G}$ respectively.

Our representative spline filters $\mathbf{H}_{0,n}^{\mathrm{spln}}$ and $\mathbf{H}_{1,n}^{\mathrm{spln}}, n \ge 1$, in (3) have bandwidth $\sigma \le n$, and they are graph filters on $\ell^p, 1 \le p \le \infty$, with filter bounds dominated by some constants depending only on the Beurling dimension and density of the graph,

$$\|\mathbf{H}_{l,n}^{\mathrm{spln}}\|_{\mathcal{B}_p} \le (\|\mathbf{H}_{l,1}^{\mathrm{spln}}\|_{\mathcal{B}_p})^n \le \begin{cases} 1 & \text{if } p = 2 \\ (2^{d-1}D_1(\mathcal{G})^n & \text{if } p \ne 2, \end{cases} \tag{15}$$

where the last inequality follows from (3), (12) and (14).

IV. Analysis Filter Banks

The design of graph analysis filter banks is crucial for competent performances of NSGFBs. There are substantial methods for the design of graph analysis filter banks in literature [22]–[25], [31]. In this section, we design analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ of NSGFBs to have small bandwidth, to pass/block the normalized constant signal, and to have stability on ℓ^2. Also in Theorem IV.4 of this section, we show that analysis filter banks in our design have stability on ℓ^p for all $1 \le p \le \infty$, with an

estimate on their lower and upper ℓ^p-stability bounds independent of the order of the graph.

A. Bandwidths

In this paper, we design analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ of an NSGFB to have small bandwidth $\sigma(\mathbf{H}_0, \mathbf{H}_1)$. Our exemplary analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ are polynomials of the symmetric Laplacian $\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}$ on $\mathcal{G}$, i.e.,

$$\mathbf{H}_0 = P_0(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}) \text{ and } \mathbf{H}_1 = P_1(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}) \tag{16}$$

for some polynomials P_0 and P_1 with degrees L_0 and L_1 respectively, which have bandwidth $\sigma(\mathbf{H}_0, \mathbf{H}_1) \le \max(L_0, L_1)$. Consequently, spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}}), n \ge 1$ have bandwidths $\sigma(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}}) \le n$, and the node-variant filter bank $(\mathbf{H}_0^{\mathrm{nv}}, \mathbf{H}_1^{\mathrm{nv}})$ has bandwidth one.

The analysis procedure associated with our analysis filter banks has an input $\mathbf{x} = (x(i))_{i\in V}$ and two outputs

$$\mathbf{z}_0 = \mathbf{H}_0\mathbf{x} \text{ and } \mathbf{z}_1 = \mathbf{H}_1\mathbf{x}. \tag{17}$$

The signal values of the outputs $\mathbf{z}_0 = (z_0(i))_{i\in V}$ and $\mathbf{z}_1 = (z_1(i))_{i\in V}$ at each vertex $k \in V$ are "weighted" sums of values of the input $\mathbf{x}$ in $\sigma(\mathbf{H}_0, \mathbf{H}_1)$-neighborhood of k by (8),

$$z_l(k) = \sum_{\rho(i,k)\le\sigma} h_l(k,i)x(i), \quad k \in V, \tag{18}$$

where $\sigma = \sigma(\mathbf{H}_0, \mathbf{H}_1)$ and $\mathbf{H}_l = (h_l(i,j))_{i,j\in V}, l = 0, 1$. The above distributed algorithm requires $O(\sigma^d)$ manipulations and additions for each agent, where d is the Beurling dimension of the graph $\mathcal{G}$. Therefore, for our analysis procedure, data processing cost and computational burden of each agent is independent on the graph order N, and the total computational cost for the whole graph is $O(\sigma^d N)$.

B. Spectral Decomposition

To apply an NSGFB to some real world applications, such as noise suppression and abnormal phenomenon detection, its analysis filter bank should constitute certain spectral decomposition. So throughout the paper, we design normal analysis filter banks to pass/block the normalized constant signal $\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1}$.

Definition IV.1: A graph filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ is said to be *normal* if the filter $\mathbf{H}_0$ passes the normalized constant signal $\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1}$, and the filter $\mathbf{H}_1$ blocks the normalized constant signal $\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1}$, i.e., (2) holds.

A normal analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ has bandwidth $\sigma(\mathbf{H}_0, \mathbf{H}_1) \ge 1$, except the trivial case that $\mathbf{H}_0$ is the identity matrix and $\mathbf{H}_1$ is the zero matrix, and $\sigma = 0$. So throughout the paper, we *only* consider analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ having positive bandwidth

$$\sigma := \sigma(\mathbf{H}_0, \mathbf{H}_1) \ge 1. \tag{19}$$

The analysis filter bank decomposes the input signal on a graph into two components carrying certain frequency information. However the frequency partition of an analysis filter bank on an arbitrary graph $\mathcal{G}$ is not as obvious as that in classical setting. For the case that filters $\mathbf{H}_0$ and $\mathbf{H}_1$ are of the form (16) for some polynomials P_0 and P_1, one may verify that they constitute normal filter bank if and only if

$$P_0(0) = 1 \text{ and } P_1(0) = 0. \tag{20}$$

The spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}}), n \ge 1$, in (3) are of the form (16) with $P_0(t) = (1 - t/2)^n$ and $P_1(t) = (t/2)^n$, and they are normal filter banks by (20), i.e.,

$$\mathbf{H}_{0,n}^{\mathrm{spln}}\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1} = \mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1} \text{ and } \mathbf{H}_{1,n}^{\mathrm{spln}}\mathbf{D}_{\mathcal{G}}^{1/2}\mathbf{1} = \mathbf{0}. \tag{21}$$

Shown in Figure 2 is local smoothing/blocking phenomenon of the spline filter bank $(\mathbf{H}_{0,2}^{\mathrm{spln}}, \mathbf{H}_{1,2}^{\mathrm{spln}})$ to a blockwise constant signal on the Minnesota traffic graph and a blockwise smooth signal on the random geometric graph RGG_{4096} in Figure 2. It is observed that the lowpass filtered signal is very close to the original signal except near the boundary between different blocks, and that the highpass filtered signal essentially vanishes except around the region where the original signal exhibits sharp local variation.

C. Stability

Robustness is a fundamental requirement in the context of filter bank to control the signal dynamic range and to regulate the input noise. For the robustness of an NSGFB on $\ell^p, 1 \le p \le \infty$, we introduce the stability of a graph filter pair on ℓ^p.

Definition IV.2: We say that $(\mathbf{H}_0, \mathbf{H}_1)$ has *ℓ^p-stability* if there are two positive constants C_p and D_p such that

$$C_p\|\mathbf{x}\|_p \le \left(\|\mathbf{H}_0\mathbf{x}\|_p^p + \|\mathbf{H}_1\mathbf{x}\|_p^p\right)^{1/p} \le D_p\|\mathbf{x}\|_p \tag{22}$$

hold for all $\mathbf{x} \in \ell^p$ if $1 \le p < \infty$, and

$$C_\infty\|\mathbf{x}\|_\infty \le \max\left(\|\mathbf{H}_0\mathbf{x}\|_\infty, \|\mathbf{H}_1\mathbf{x}\|_\infty\right) \le D_\infty\|\mathbf{x}\|_\infty \tag{23}$$

hold for all $\mathbf{x} \in \ell^\infty$ if $p = \infty$. The largest constant C_p and the smallest constant D_p for (22) and (23) to hold are known as *lower and upper stability bounds* of $(\mathbf{H}_0, \mathbf{H}_1)$ on ℓ^p respectively.

Given an NSGFB with the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$ such that the perfect reconstruction condition (1) holds, the input signal $\mathbf{x}$ and the output signals $\mathbf{z}_0 = \mathbf{H}_0\mathbf{x}$ and $\mathbf{z}_1 = \mathbf{H}_1\mathbf{x}$ of the analysis procedure have comparable energy, that is

$$\begin{aligned}&(\|\mathbf{G}_0\|_{\mathcal{B}_2}^2 + \|\mathbf{G}_1\|_{\mathcal{B}_2}^2)^{-1}\|\mathbf{x}\|_2^2 \le \|\mathbf{H}_0\mathbf{x}\|_2^2 + \|\mathbf{H}_1\mathbf{x}\|_2^2\\ &\le (\|\mathbf{H}_0\|_{\mathcal{B}_2}^2 + \|\mathbf{H}_1\|_{\mathcal{B}_2}^2)\|\mathbf{x}\|_2^2\end{aligned} \tag{24}$$

hold for all $\mathbf{x} \in \ell^2$, where the second inequality follows directly from Definition III.2 and the first inequality holds as for all $\mathbf{x} \in \ell^2$,

$$\begin{aligned}\|\mathbf{x}\|_2 &= \|(\mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1)\mathbf{x}\|_2\\ &\le \|\mathbf{G}_0\|_{\mathcal{B}_2}\|\mathbf{H}_0\mathbf{x}\|_2 + \|\mathbf{G}_1\|_{\mathcal{B}_2}\|\mathbf{H}_1\mathbf{x}\|_2\\ &\le \left(\|\mathbf{G}_0\|_{\mathcal{B}_2}^2 + \|\mathbf{G}_1\|_{\mathcal{B}_2}^2\right)^{1/2}\left(\|\mathbf{H}_0\mathbf{x}\|_2^2 + \|\mathbf{H}_1\mathbf{x}\|_2^2\right)^{1/2}\end{aligned}$$

by (1) and Definition III.2. So throughout the paper, we require that analysis filter banks $(\mathbf{H}_0, \mathbf{H}_1)$ have ℓ^2-stability.

For any $\mathbf{x} \in \ell^2$, direct calculation yields

$$\|\mathbf{H}_0\mathbf{x}\|_2^2 + \|\mathbf{H}_1\mathbf{x}\|_2^2 = \mathbf{x}^T\left(\mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1\right)\mathbf{x}. \tag{25}$$

This leads to the following characterization on the ℓ^2-stability.

Proposition IV.3: The analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ has ℓ^2-stability if and only if $\mathbf{H} := \mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$ is positive definite. Moreover, the lower and upper stability bounds C_2 and D_2 are given by

$$C_2^2 = (\|\mathbf{H}^{-1}\|_{\mathcal{B}_2})^{-1} \text{ and } D_2^2 = \|\mathbf{H}\|_{\mathcal{B}_2}. \tag{26}$$

For graph filters $\mathbf{H}_0$ and $\mathbf{H}_1$ of the form (16), $\mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$ is a positive definite matrix with eigenvalues

TABLE I
THE LOWER AND UPPER STABILITY BOUNDS FOR SPLINE ANALYSIS FILTER BANK OF ORDER TWO

Graph	Minnesota	RGG_{4096}	RGG_{300000}	SST
C_1	0.9108	0.7283	0.7158	0.5938
C_2	0.6483	0.5193	0.5076	0.4302
C_4	0.5463	0.4332	0.4283	0.3554
C_∞	0.6736	0.5663	0.5800	0.5100
D_1	0.9837	0.7747	0.7214	0.8952
D_2	0.6943	0.5508	0.5113	0.6245
D_4	0.5976	0.4760	0.4326	0.5944
D_∞	1.3950	1.0808	0.9709	1.0805

$(P_0(\lambda_n))^2 + (P_1(\lambda_n))^2, 1 \le n \le N$. Hence the corresponding lower and upper stability bounds C_2 and D_2 are evaluated by

$$C_2 = \inf_{1\le n\le N} \sqrt{(P_0(\lambda_n))^2 + (P_1(\lambda_n))^2} \le \sup_{1\le n\le N} \sqrt{(P_0(\lambda_n))^2 + (P_1(\lambda_n))^2} = D_2. \quad (27)$$

Taking polynomials $P_0(t) = (1 - t/2)^n$ and $P_1(t) = (t/2)^n$ in (27), and recalling that the eigenvalues of $\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}$ lie in the interval $[0, 2]$, we obtain that spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ of order $n \ge 1$ has ℓ^2-stability with lower bound $2^{-n+1/2}$ and upper bound 1.

Filters in a stable filter bank on ℓ^p are graph filters on $\ell^p, 1 \le p \le \infty$. In the following theorem, we show that analysis filter banks are stable on $\ell^p, 1 \le p \le \infty$, with quantitative estimates on their lower and upper stability bounds by some constants *independent* of the order of the graph.

Theorem IV.4: Let filters $\mathbf{H}_0$ and $\mathbf{H}_1$ have bandwidth $\sigma \ge 1$, set $\mathbf{H} := \mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$ and denote the condition number of the matrix $\mathbf{H}$ by

$$\kappa = \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}\|_{\mathcal{B}_2}. \quad (28)$$

If $(\mathbf{H}_0, \mathbf{H}_1)$ has ℓ^2-stability, then it has ℓ^p-stability for all $1 \le p \le \infty$. Moreover, we have the following estimates for its lower and upper stability bounds C_p and D_p:

$$C_p \ge \frac{\|\mathbf{H}\|_{\mathcal{B}_2}^{1/2}}{d!2^{d+1}(D_1(\mathcal{G}))^2(\sigma+1)^{2d}\kappa^{d+2}} \quad (29)$$

and

$$D_p \le 2D_1(\mathcal{G})(\sigma+1)^d\|\mathbf{H}\|_{\mathcal{B}_2}^{1/2}, \quad (30)$$

where d and $D_1(\mathcal{G})$ are the Beurling dimension and density of the graph $\mathcal{G}$ respectively.

The lower bound estimate for C_p in (29) and the upper bound estimate D_p in (30) are quite conservative. Shown in Table I are numerical results of the lower and upper stability bounds C_p and $D_p, p = 1, 2, 4, \infty$ for the spline analysis filter bank $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ of order $n = 2$, where Minnesota and $\mathrm{RGG}_{\mathrm{N}}$ stand for the Minnesota traffic graph and random geometric graph of order N in Figure 2 respectively, and SST is the graph used to describe sea surface temperature of 100 observation stations in Figure 8. Our numerical results confirm that the estimates of lower and upper stability bounds C_p and D_p in (29) and (30) are comparable with different p, cf. [41], [42] for historical remarks and recent advances on stability bound estimates for matrices with certain off-diagonal decay.

V. SYNTHESIS FILTER BANKS AND BEZOUT IDENTITY

Let $(\mathbf{H}_0, \mathbf{H}_1)$ be a normal graph filter bank with ℓ^2-stability. In this section, we propose an algebraic method to construct graph filters $\mathbf{G}_0$ and $\mathbf{G}_1$ so that the NSGFB with the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$ satisfies the perfect reconstruction condition (1) and the bandwidth of synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$ is no larger than the bandwidth of the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$.

Theorem V.1: Let $(\mathbf{H}_0, \mathbf{H}_1)$ be a normal filter bank with ℓ^2-stability. Assume that $(\mathbf{H}_0, \mathbf{H}_1)$ is of the form (16) for some polynomials P_0 and P_1. If polynomials Q_0 and Q_1 satisfy

$$P_0(z)Q_0(z) + P_1(z)Q_1(z) = 1,\ z \in \mathbb{C}, \quad (31)$$

then the NSGFB with the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$ satisfies the perfect reconstruction condition (1), where

$$\mathbf{G}_0 = Q_0(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}) \text{ and } \mathbf{G}_1 = Q_1(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}). \quad (32)$$

Proof: The conclusion follows immediately from (16), (31), and (32). ■

The approach of constructing synthesis filter banks via solving the Bezout identity (31) provides a tool to design them without a priori knowledge of global topological structure of the residing graph and then it simplifies their designs for perfect signal reconstruction. It is well known that the Bezout identity (31) is solvable if and only if polynomials P_0 and P_1 have no common root [37]. Recall from (20) that $P_0(0) = 1$ and $P_1(0) = 0$. Then we can find a *unique* solution pair (Q_0^B, Q_1^B) to the Bezout identity (31) such that $Q_0^B(0) = 1, Q_1^B(0) = 0$ and the degree of Q_0^B (resp. Q_1^B) is no larger than the degree of P_1 (resp. P_0). Using the above polynomial pair, we define a filter bank $(\mathbf{G}_0^B, \mathbf{G}_1^B)$ by

$$\mathbf{G}_0^B = Q_0^B(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}) \text{ and } \mathbf{G}_1^B = Q_1^B(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}). \quad (33)$$

The above filter bank $(\mathbf{G}_0^B, \mathbf{G}_1^B)$ has bandwidth no larger than bandwidth of the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$, and it together with the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ forms an NSGFB satisfying the perfect reconstruction condition (1) by Theorem V.1. So we call the filter bank $(\mathbf{G}_0^B, \mathbf{G}_1^B)$ in (33) as the *synthesis filter bank constructed from Bezout approach*, and use the superscript B to distinguish from arbitrary synthesis filter banks constructed in Theorem V.1.

Following (33), we define *synthesis spline filters* $\mathbf{G}_{0,n}^{\mathrm{B,spln}}$ and $\mathbf{G}_{1,n}^{\mathrm{B,spln}}$ of order $n \ge 1$ by

$$\mathbf{G}_{l,n}^{\mathrm{B,spln}} = Q_{l,n}^{\mathrm{B,spln}}(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}),\ l = 0, 1, \quad (34)$$

where $Q_{0,n}^{\mathrm{B,spln}}(t) = Q_n(t/2) + Q_n(1)(t/2)^n$, $Q_{1,n}^{\mathrm{B,spln}}(t) = Q_n(1 - t/2) - Q_n(1)(1 - t/2)^n$, and

$$Q_n(t) = \sum_{l=0}^{n-1} \binom{2n-1}{l} (1-t)^{n-1-l}t^l.$$

The filter bank $(\mathbf{G}_{0,n}^{\mathrm{B,spln}}, \mathbf{G}_{1,n}^{\mathrm{B,spln}})$ is the synthesis filter bank constructed from Bezout approach, since $Q_{0,n}^{\mathrm{B,spln}}$ and $Q_{1,n}^{\mathrm{B,spln}}$ are polynomials of degree n which satisfies $Q_{0,n}^{\mathrm{B,spln}}(0) = 1$, $Q_{1,n}^{\mathrm{B,spln}}(0) = 0$ and $(1 - t/2)^n Q_{0,n}^{\mathrm{B,spln}}(t) + (t/2)^n Q_{1,n}^{\mathrm{B,spln}}(t) = (1 - t/2)^n Q_n(t/2) + (t/2)^n Q_n(1 - t/2) = 1$ by (3) and [14, Proposition 6.1.2]. Therefore the NSGFB with the analysis spline filter bank $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ and synthesis spline filter bank $(\mathbf{G}_{0,n}^{\mathrm{B,spln}}, \mathbf{G}_{1,n}^{\mathrm{B,spln}})$ satisfies the perfect reconstruction condition (1).

Write $\mathbf{G}_l = (g_l(i,j))_{i,j\in V}, l = 0, 1$. As the synthesis filters $\mathbf{G}_0$ and $\mathbf{G}_1$ have finite bandwidth $\tilde{\sigma} \leq \max(\deg(Q_0), \deg(Q_1))$, the synthesis procedure can be implemented in a distributed manner,

$$\tilde{x}(k) = \sum_{\rho(j,k)\leq\tilde{\sigma}} (g_0(k,j)\tilde{z}_0(j) + g_1(k,j)\tilde{z}_1(j)),\ k \in V, \quad (35)$$

where $\tilde{\mathbf{x}} = (\tilde{x}(i))_{i\in V}$ is the reconstructed signal and $\Psi_l(\mathbf{z}_l) = (\tilde{z}_l(i))_{i\in V}, l = 0, 1$, are outputs of subband processing. Hence values of the reconstructed signals $\tilde{\mathbf{x}}$ at each vertex $k \in V$ are "weighted" sums of values of the subband processed outputs $\Psi_0(\mathbf{z}_0)$ and $\Psi_1(\mathbf{z}_1)$ in a $\tilde{\sigma}$-neighborhood of $k \in V$, cf. (18) for distributed implementation of the analysis procedure. Therefore for the synthesis procedure, data processing cost and computational burden for each agent is about $O(\tilde{\sigma}^d)$ manipulations and additions (independent on N) and the total computational cost for the whole graph is $O(\tilde{\sigma}^d N)$, where d and N are the Beurling dimension and the order of the graph $\mathcal{G}$ respectively.

In real world applications of an NSGFB such as the proposed distributed denoising method in Section VIII, the subband signals $\mathbf{z}_0$ and $\mathbf{z}_1$ in (17) are processed via some (non)linear procedure, such as hard/soft thresholding and quantization. In this case, the reconstructed signal

$$\tilde{\mathbf{x}} = \mathbf{G}_0\Psi_0(\mathbf{z}_0) + \mathbf{G}_1\Psi_1(\mathbf{z}_1) \quad (36)$$

is not necessarily the same as the input signal $\mathbf{x}$, where Ψ_0, Ψ_1 are subband processing operators. Next, we show that the reconstruction error is mainly dominated by the error caused by the subband processing.

Proposition V.2: Let $\mathcal{G}$, d, $D_1(\mathcal{G})$, $(\mathbf{H}_0, \mathbf{H}_1)$ and $(\mathbf{G}_0, \mathbf{G}_1)$ be as in Theorem V.1. Assume that the error caused by the subband processing Ψ_l on subband signals $\mathbf{z}_l = \mathbf{H}_l\mathbf{x}, l = 0, 1$, is dominated by ϵ for any input signal $\mathbf{x} \in \ell^p$, i.e.,

$$\|\mathbf{z}_l - \Psi_l(\mathbf{z}_l)\|_p \leq \epsilon,\ l = 0, 1, \quad (37)$$

where $\epsilon \geq 0$ and $1 \leq p \leq \infty$. For the input signal $\mathbf{x} \in \ell^p$, the reconstructed signal $\tilde{\mathbf{x}}$ in (36) via the corresponding NSGFB belongs to ℓ^p and satisfies

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_p \leq D_1(\mathcal{G})(\tilde{\sigma} + 1)^d(\|\mathbf{G}_0\|_\infty + \|\mathbf{G}_1\|_\infty)\epsilon, \quad (38)$$

where $\tilde{\sigma}$ is the bandwidth of the synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$.

Our representative subband processing procedures Ψ are hard/soft thresholding and uniform quantization. For those cases, the subband processing Ψ is of the form $\Psi(\mathbf{z}) = (\psi(z(i)))_{i\in V}$ for $\mathbf{z} = (z(i))_{i\in V}$. Thus the subband processing can be implemented in a distributed manner and the resulted error is bounded (i.e., (37) holds for $p = \infty$) by the thresholding and quantization level; see Table VI in Section VIII. This together with (18) and (35) implies that the NSGFB with analysis/synthesis filter banks in Theorem V.1 can be implemented in a distributed manner too, provided that the subband processing can be.

VI. Synthesis Filter Bank and Optimization

Let $(\mathbf{H}_0, \mathbf{H}_1)$ be a normal graph filter bank with ℓ^2-stability. For the distributed implementation of synthesis procedure on SDN or a graph of large order, (approximately) sparse and localized synthesis filter banks are favorable. In this section, we propose solving the minimization problem

$$\min_{\mathbf{G}_0,\mathbf{G}_1} \|\mathbf{G}_0\|_F^2 + \|\mathbf{G}_1\|_F^2 \text{ subject to } \mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1 = \mathbf{I} \quad (39)$$

to construct approximately sparse synthesis filter banks. The minimization problem (39) is a convex relaxation of the conventional sparse minimization problem:

$$\min_{\mathbf{G}_0,\mathbf{G}_1} \|\mathbf{G}_0\|_0 + \|\mathbf{G}_1\|_0 \text{ subject to } \mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1 = \mathbf{I}, \quad (40)$$

where $\|\mathbf{A}\|_0$ is the number of nonzero entries in a matrix $\mathbf{A}$. Although the unique solution of the minimization problem (39) is not the sparsest synthesis filter banks in the conventional sparse minimization problem (40), we observe that it has a closed form (42), and it is approximately sparse and localized by Theorem VI.1.

Define the Lagrange function $\mathcal{L}$ of the constrained optimization problem (39) by

$$\mathcal{L}(\mathbf{G_0}, \mathbf{G_1}, \mathbf{\Theta}) = \|\mathbf{G}_0\|_F^2 + \|\mathbf{G}_1\|_F^2 - \operatorname{tr}\left((\mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1 - \mathbf{I})\mathbf{\Theta}^T\right),$$

where $\mathbf{\Theta}$ is the matrix-valued Lagrange multiplier. By direct calculation, we have

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial\mathbf{G}_0} = 2\mathbf{G}_0 - \mathbf{\Theta}\mathbf{H}_0^T \\ \dfrac{\partial\mathcal{L}}{\partial\mathbf{G}_1} = 2\mathbf{G}_1 - \mathbf{\Theta}\mathbf{H}_1^T \\ \dfrac{\partial\mathcal{L}}{\partial\mathbf{\Theta}} = \mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1 - \mathbf{I}. \end{cases} \quad (41)$$

Set $\mathbf{H} = \mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$. Solving

$$\frac{\partial\mathcal{L}}{\partial\mathbf{G}_0} = \frac{\partial\mathcal{L}}{\partial\mathbf{G}_1} = \frac{\partial\mathcal{L}}{\partial\mathbf{\Theta}} = \mathbf{0}$$

leads to the unique solution of the constrained optimization problem (39),

$$\mathbf{G}_0^L = \mathbf{H}^{-1}\mathbf{H}_0^T \text{ and } \mathbf{G}_1^L = \mathbf{H}^{-1}\mathbf{H}_1^T. \quad (42)$$

The synthesis filter bank $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ in (42) satisfies (1), and the filter $\mathbf{G}_0^L$ passes the normalized constant signal $\mathbf{D}_\mathcal{G}^{1/2}\mathbf{1}$, as

$$\mathbf{G}_0^L\mathbf{D}_\mathcal{G}^{1/2}\mathbf{1} = \mathbf{H}^{-1}\mathbf{H}_0^T\mathbf{D}_\mathcal{G}^{1/2}\mathbf{1} = \mathbf{H}^{-1}(\mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1)\mathbf{D}_\mathcal{G}^{1/2}\mathbf{1} = \mathbf{D}_\mathcal{G}^{1/2}\mathbf{1}$$

by (2) and (42). We call the filter bank $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ as the *synthesis filter bank constructed from optimization approach* and use the superscript L to distinguish from other synthesis filter banks satisfying (1).

For the case that $\mathbf{H}$ is a diagonal matrix, the synthesis filter bank $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ in (42) has the same bandwidth as the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$, and its entries satisfy

$$|g_l^L(i,j)| \leq \begin{cases} \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}_l\|_\infty & \text{if } \rho(i,j) \leq \sigma \\ 0 & \text{otherwise,} \end{cases} \quad (43)$$

where $\mathbf{G}_l^L := (g_l^L(i,j))_{i,j\in V}, l = 0, 1$.

Let κ be the condition number of the matrix $\mathbf{H}$ in (28). It is well known that $\kappa > 1$ when $\mathbf{H}$ is not a diagonal matrix. For $\kappa > 1$, the synthesis filter bank $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ in (19) does not necessarily have a small bandwidth, however it always has an exponential off-diagonal decay, and hence it is approximately sparse, see [43] for sparsity measurements of a matrix.

Theorem VI.1: Let $(\mathbf{H}_0, \mathbf{H}_1)$ be a normal graph filter bank with ℓ^2-stability, and $\mathbf{G}_l^L := (g_l^L(i,j))_{i,j\in V}, l = 0, 1$, be as in

(19). Assume that $\kappa > 1$, then

$$|g_l^L(i,j)| \le D_1(\mathcal{G})(\sigma+1)^d(1-1/\kappa)^{-1/2} \times \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}_l\|_\infty \exp\left(-\frac{\theta}{2\sigma}\rho(i,j)\right) \quad (44)$$

hold for all $i,j \in V$ and $l = 0,1$, where $\theta = \ln(\kappa/(\kappa-1))$, $\sigma \ge 1$ is the bandwidth of the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$, and d and $D_1(\mathcal{G})$ are the Beurling dimension and density of the graph $\mathcal{G}$ respectively.

Remark VI.2: Agents in an SDN may lose data processing ability and/or communication capability. In that case, outputs of the analysis procedure of an NSGFB can be considered as being corrupted by shot noises. The exponential off-diagonal decay property in Theorem VI.1 implies that the reconstructed signal suffers mainly in their neighborhood of limited size.

Remark VI.3: By the exponential off-diagonal decay property in Theorem VI.1, the synthesis filters $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ are filters on $\ell^p, 1 \le p \le \infty$,

$$\|\mathbf{G}_l^L\|_{\mathcal{B}_p} \le d!2^d(D_1(\mathcal{G}))^2(\sigma+1)^{2d}\kappa^{d+1}(1-1/\kappa)^{-1/2} \times \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}_l\|_\infty,\ l=0,1. \quad (45)$$

The above conclusion with $p = \infty$ indicates that the NSGFB does not have a resonance effect.

Applying similar arguments used in the proof of Proposition V.2, we have the following result.

Proposition VI.4: Let $\mathcal{G}, (\mathbf{H}_0, \mathbf{H}_1), (\mathbf{G}_0^L, \mathbf{G}_1^L)$ be as in Theorem VI.1, and let $p, \Psi_0, \Psi_1, \epsilon$ be as in Proposition V.2. Assume that the input signal $\mathbf{x}$ of the corresponding NSGFB belongs to ℓ^p, then the reconstructed signal $\tilde{\mathbf{x}} = \mathbf{G}_0^L\Psi_0(\mathbf{H}_0\mathbf{x}) + \mathbf{G}_1^L\Psi_1(\mathbf{H}_1\mathbf{x})$ via the NSGFB belongs to ℓ^p and

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_p \le d!2^d(D_1(\mathcal{G}))^2(\sigma+1)^{2d}\kappa^{d+1}\|\mathbf{H}^{-1}\|_{\mathcal{B}_2} \times (1-1/\kappa)^{-1/2}\left(\|\mathbf{H}_0\|_\infty + \|\mathbf{H}_1\|_\infty\right)\epsilon. \quad (46)$$

Solving the constrained optimization program (39) associated with the analysis spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$, we have the synthesis spline filter banks $(\mathbf{G}_{0,n}^{\mathrm{L,spln}}, \mathbf{G}_{1,n}^{\mathrm{L,spln}}), n \ge 1$, where

$$\mathbf{G}_{l,n}^{\mathrm{L,spln}} = \left((\mathbf{H}_{0,n}^{\mathrm{spln}})^2 + (\mathbf{H}_{1,n}^{\mathrm{spln}})^2\right)^{-1}\mathbf{H}_{l,n}^{\mathrm{spln}},\ l=0,1. \quad (47)$$

The synthesis spline filters $\mathbf{G}_{0,n}^{\mathrm{L,spln}}$ and $\mathbf{G}_{1,n}^{\mathrm{L,spln}}, n \ge 1$, have full bandwidth, however they have exponential off-diagonal decay by (44) and Theorem VI.1,

$$|g_{l,n}^{\mathrm{L,spln}}(i,j)| \le 2^{3n-3/2}(2^{2n-1}-1)^{-1/2}(n+1)^d D_1(\mathcal{G}) \times \exp\left(-\frac{\ln(2^{2n-1}/(2^{2n-1}-1))}{2n}\rho(i,j)\right),\ \ i,j \in V,$$

where $\mathbf{G}_{l,n}^{\mathrm{L,spln}} = (g_{l,n}^{\mathrm{L,spln}}(i,j))_{i,j\in V}, l = 0,1$.

As $\mathbf{U}^T\mathbf{x}$ is thought as a graph Fourier transform of the signal $\mathbf{x}$, we may use the diagonal vector $P(\boldsymbol{\lambda})$ of $P(\boldsymbol{\Lambda})$ to describe frequency response of the filter $\mathbf{A} = P(\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}) = \mathbf{U}^T P(\boldsymbol{\Lambda})\mathbf{U}$ in (7) where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$ is composed of eigenvalues $0 \le \lambda_1 \le \cdots \le \lambda_N \le 2$ of the symmetric normalized Laplacian $\mathbf{L}_{\mathcal{G}}^{\mathrm{sym}}$. Shown in Figure 3 are frequency responses of the analysis spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ of order n, the synthesis spline filter banks $(\mathbf{G}_{0,n}^{\mathrm{B,spln}}, \mathbf{G}_{1,n}^{\mathrm{B,spln}})$ in (34), and the synthesis spline filter banks $(\mathbf{G}_{0,n}^{\mathrm{L,spln}}, \mathbf{G}_{1,n}^{\mathrm{L,spln}})$ just constructed, where $n = 1, 2$. It is observed that the frequency responses of analysis spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ and synthesis spline filter banks $(\mathbf{G}_{0,n}^{\mathrm{L,spln}}, \mathbf{G}_{1,n}^{\mathrm{L,spln}})$ have certain complementary property, while the synthesis spline filter banks $(\mathbf{G}_{0,n}^{\mathrm{B,spln}}, \mathbf{G}_{1,n}^{\mathrm{B,spln}})$ constructed via solving a Bezout identity do not.

VII. Iterative Distributed Algorithm for Synthesis Procedure

For the NSGFB with synthesis filter banks in Theorem V.1, the implementation of the corresponding synthesis procedure in a distributed manner has been discussed in (35). For the NSGFB with the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and synthesis filter bank $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ constructed from optimization approach, the output $\tilde{\mathbf{x}}$ of the synthesis procedure is

$$\tilde{\mathbf{x}} = \mathbf{G}_0^L\tilde{\mathbf{z}}_0 + \mathbf{G}_1^L\tilde{\mathbf{z}}_1, \quad (48)$$

where $\tilde{\mathbf{z}}_0$ and $\tilde{\mathbf{z}}_1$ be outputs of subband processing. As filters $\mathbf{G}_0^L$ and $\mathbf{G}_1^L$ may have full bandwidth, it is infeasible to evaluate $\mathbf{G}_0^L\tilde{\mathbf{z}}_0$ and $\mathbf{G}_1^L\tilde{\mathbf{z}}_1$ directly in a distributed manner. In this section, we circumvent directly evaluating the synthesis filter bank $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ by proposing an iterative distributed algorithm to implement the synthesis procedure (48), which has linear complexity and is implementable in SDNs without a fusion center.

The proposed iterative distributed algorithm is based on two pivotal observations. The first observation is that the output signal $\tilde{\mathbf{x}}$ in (48) is the unique solution of the following global least squares problem:

$$\min_{\mathbf{x}}\ \|\mathbf{H}_0\mathbf{x} - \tilde{\mathbf{z}}_0\|_2^2 + \|\mathbf{H}_1\mathbf{x} - \tilde{\mathbf{z}}_1\|_2^2. \quad (49)$$

To solve the global optimization problem (49) in a distributed manner, we introduce a family of least squares problems in the $(2r)$-neighborhood of vertices $k \in V$:

$$\min_{\mathbf{x}} \|\mathbf{H}_0\chi_k^{2r}\mathbf{x} - \tilde{\mathbf{z}}_0\|_2^2 + \|\mathbf{H}_1\chi_k^{2r}\mathbf{x} - \tilde{\mathbf{z}}_1\|_2^2, \quad (50)$$

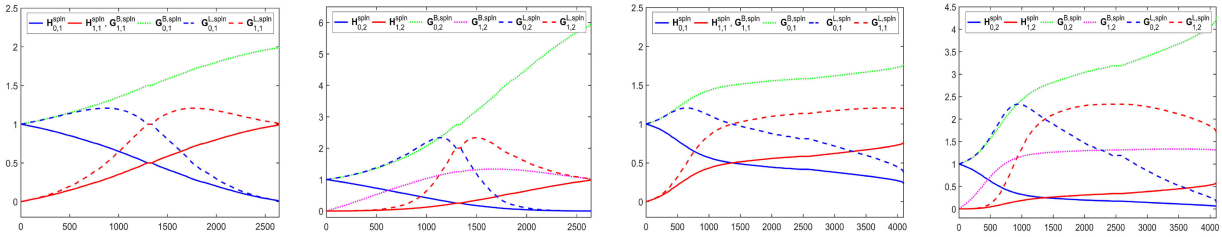
where for any $k \in V$ and $s \ge 0$, χ_k^s is the diagonal matrix

$$\chi_k^s :\ (x(i))_{i\in V} \longmapsto (\chi_{B(k,s)}(i)x(i))_{i\in V}, \quad (51)$$

whose diagonal entries take values of the indicator function $\chi_{B(k,s)}$ on the ball $B(k,s)$ centered at node k with radius s. For any $k \in V$, we differentiate (50) with respect to $\chi_k^{2r}\mathbf{x}$ to obtain a solution of the local least squares problem (50),

$$\mathbf{v}_{k,r} = \chi_k^{2r}(\chi_k^{2r}\mathbf{H}\chi_k^{2r})^\dagger\chi_k^{2r}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0 + \mathbf{H}_1^T\tilde{\mathbf{z}}_1), \quad (52)$$

where $\mathbf{H} = \mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$. The second crucial observation is that the above solution $\mathbf{v}_{k,r}$ provides a local good approximation to the solution $\tilde{\mathbf{x}}$ of the least squares problem (49) in a r-neighborhood of the vertex $k \in V$, but not in the whole $2r$-neighbor regions due to boundary effects, see Figure 4 for a numerical demonstration. Therefore we can patch $\mathbf{v}_{k,r}, k \in V$,



(a) (b) (c) (d)

Fig. 3. Plotted in (a) and (b) are the frequency responses of analysis/synthesis spline filters of order $n = 1, 2$ on the Minnesota traffic graph. Plotted in (c) and (d) are the frequency responses of analysis/synthesis spline filters of order $n = 1, 2$ on the random geometric graph RGG_{4096} in Figure 2.

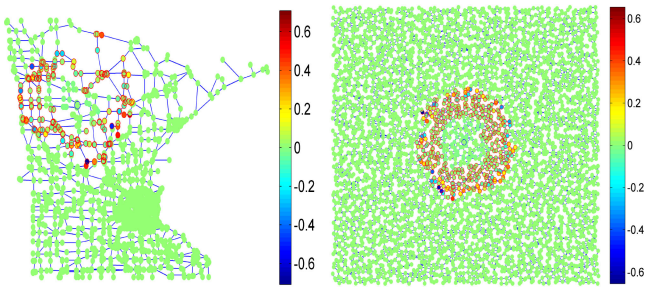


Fig. 4. Plotted are the difference in the $2r$-neighborhood of a random signal $\mathbf{x}_o$ and its local approximation $\mathbf{v}_{k,r}$ on the Minnesota traffic graph (left) and on the random graph RGG_{4096} (right), where $r = 6$, the center k is circled in blue, the vertices in the outlying region $B(k, 2r)\backslash B(k, r)$ are circled in red. In the simulation, the signal $\mathbf{x}_o$ has entries being randomly chosen from $[-1, 1]$, the analysis filter bank is the spline filter bank $(\mathbf{H}_{0,2}^{\mathrm{spln}}, \mathbf{H}_{1,2}^{\mathrm{spln}})$ of order 2, and $\mathbf{v}_{k,r}$ is given in (52). The maximal error between $\mathbf{x}_o$ and $\mathbf{v}_{k,r}$ on vertices in the r-neighborhood $B(k, r)$ and in the outlying region $B(k, 2r)\backslash B(k, r)$ are $0.0234, 0.7080$ for the Minnesota traffic graph, and $0.0049, 0.6545$ for the random graph RGG_{4096} respectively.

together

$$\mathbf{v}_r = \left(\sum_{k'\in V} \chi_{k'}^r\right)^{-1} \sum_{k\in V} \chi_k^r \mathbf{v}_{k,r} = \mathbf{J}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0 + \mathbf{H}_1^T\tilde{\mathbf{z}}_1) \quad (53)$$

to generate a good approximation,

$$\|\mathbf{v}_r - \tilde{\mathbf{x}}\|_p \le \delta_{r,\sigma}\|\tilde{\mathbf{x}}\|_p, \quad (54)$$

to the solution $\tilde{\mathbf{x}}$ of the least squares problem (49) in ℓ^p norm, where $\delta_{r,\sigma} \in (0, 1)$, the radius parameter $r \ge 1$ is chosen appropriately, and

$$\mathbf{J} = \left(\sum_{k'\in V} \chi_{k'}^r\right)^{-1} \sum_{k\in V} \chi_k^r (\chi_k^{2r}\mathbf{H}\chi_k^{2r})^\dagger \chi_k^{2r}. \quad (55)$$

Set filters

$$\mathbf{J}_0 = \mathbf{J}\mathbf{H}_0^T \text{ and } \mathbf{J}_1 = \mathbf{J}\mathbf{H}_1^T, \quad (56)$$

which have bandwidths $\sigma(\mathbf{J}_0, \mathbf{J}_1) \le \sigma + 4r$ and approximate synthesis filters $\mathbf{G}_0^L$ and $\mathbf{G}_l^L$ respectively when r is sufficiently large. Based on (52), (53) and (54), we propose an iterative distributed algorithm with initials $\tilde{\mathbf{z}}_0, \tilde{\mathbf{z}}_1 \in \ell^p$:

$$\begin{cases} \mathbf{v}^{(m)} = \mathbf{J}_0\tilde{\mathbf{z}}_0^{(m-1)} + \mathbf{J}_1\tilde{\mathbf{z}}_1^{(m-1)} \\ \mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} + \mathbf{v}^{(m)} \\ \tilde{\mathbf{z}}_0^{(m)} = \tilde{\mathbf{z}}_0^{(m-1)} - \mathbf{H}_0\mathbf{v}^{(m)} \\ \tilde{\mathbf{z}}_1^{(m)} = \tilde{\mathbf{z}}_1^{(m-1)} - \mathbf{H}_1\mathbf{v}^{(m)} \end{cases} \quad (57)$$

for $m \ge 1$, where

$$\mathbf{x}^{(0)} = \mathbf{0},\ \tilde{\mathbf{z}}_0^{(0)} = \tilde{\mathbf{z}}_0,\ \tilde{\mathbf{z}}_1^{(0)} = \tilde{\mathbf{z}}_1. \quad (58)$$

Recall that $(\mathbf{H}_0, \mathbf{H}_1)$ and $(\mathbf{J}_0, \mathbf{J}_1)$ have bandwidths σ and $\sigma + 4r$ respectively. Then the implementation of the above algorithm at each vertex k requires to store entries in the k-th row of filters $\mathbf{J}_0$ and $\mathbf{J}_1$ in the $(4r + \sigma)$-hop neighbor of the vertex k, entries in the k-th row of filters $\mathbf{H}_0$ and $\mathbf{H}_1$ in the σ-hop neighbor of the vertex k, and also the initial data $\tilde{\mathbf{z}}_0$ and $\tilde{\mathbf{z}}_1$ in the $(4r + \sigma)$-hop neighbor of the vertex k. In each iteration, the agent k requires $O((4r + \sigma)^d)$ manipulations and additions and it needs to share value of the vector $\mathbf{v}^{(m)}$ at the vertex k with its σ-hop neighbor, and values of the vectors $\tilde{\mathbf{z}}_0^{(m)}$ and $\tilde{\mathbf{z}}_1^{(m)}$ at the vertex k with its $(4r + \sigma)$-hop neighbor respectively.

Remark VII.1: Decompose $\mathbf{H} = \mathbf{D} + \mathbf{R}$ into a diagonal component $\mathbf{D}$ and the remainder $\mathbf{R}$. Then the classical Jacobi method to solve the linear system $\mathbf{H}\mathbf{x} = \mathbf{H}_0^T\tilde{\mathbf{z}}_0 + \mathbf{H}_1^T\tilde{\mathbf{z}}_1$ is

$$\mathbf{x}^{(m)} = \mathbf{D}^{-1}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0 + \mathbf{H}_1^T\tilde{\mathbf{z}}_1 - \mathbf{R}\mathbf{x}^{(m-1)}),\ m \ge 1. \quad (59)$$

The Jacobi method converges when $\mathbf{H}$ is diagonally dominated, which is not necessarily true for the case in our setting. We observe that for $r = 0$, the matrix $\mathbf{J}$ in (55) is equal to $\mathbf{D}^{-1}$. Hence the sequence $\mathbf{x}^{(m)}, m \ge 0$, in the proposed algorithm (57) and (58) with $r = 0$ is the same as the one in the Jacobi method (59) with initial $\mathbf{x}^{(0)} = \mathbf{0}$.

In the next theorem, we further show that the iterative algorithm (57) and (58) converges exponentially when r is appropriately selected.

Theorem VII.2: Let $(\mathbf{H}_0, \mathbf{H}_1)$ be a normal filter bank with ℓ^2-stability, $\kappa > 1$ be the condition number of the matrix $\mathbf{H} := \mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$ given in (28), and let $(\mathbf{G}_0^L, \mathbf{G}_1^L)$ be as in (42). Set

$$\delta_{r,\sigma} := \frac{(D_1(\mathcal{G}))^2(2\sigma + 1)^d\kappa^2}{\kappa - 1} \exp\left(-\frac{\theta}{2\sigma}r\right)(3r + 2\sigma + 1)^d, \quad (60)$$

where $\theta = \ln(\kappa/(\kappa - 1))$, $\sigma \ge 1$ is the bandwidth of the analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$, and d and $D_1(\mathcal{G})$ are the Beurling dimension and density of the graph $\mathcal{G}$ respectively. Take $\tilde{\mathbf{z}}_0, \tilde{\mathbf{z}}_1 \in \ell^p$, and let $\mathbf{x}^{(m)}, m \ge 0$, be as in (57) and (58). If the radius parameter r is so chosen that

$$\delta_{r,\sigma} \in (0, 1), \quad (61)$$

then $\mathbf{x}^{(m)}, m \ge 0$, converges to $\tilde{\mathbf{x}}$ in (48) exponentially,

$$\|\mathbf{x}^{(m)} - \tilde{\mathbf{x}}\|_p \le (\delta_{r,\sigma})^m\|\tilde{\mathbf{x}}\|_p,\ m \ge 0. \quad (62)$$

By (60) and (62) in Theorem VII.2, the iterative algorithm (57) and (58) has exponential convergence when a large radius

TABLE II
PERFORMANCE OF THE PROPOSED ITERATIVE DISTRIBUTED ALGORITHM TO RECOVER SIGNALS ON THE MINNESOTA TRAFFIC GRAPH

$E_{m,r}$ m \ r	0	1	2	3	4	6
1	1.1988	.6564	.3184	.1523	.0725	.0178
2	1.1662	.2315	.0518	.0136	.0029	.0002
3	1.4697	.1162	.0125	.0017	.0002	.0000
4	1.8674	.0567	.0026	.0002	.0000	.0000
5	2.5386	.0296	.0006	.0000	.0000	.0000
7	4.7088	.0082	.0000	.0000	.0000	.0000
10	12.7923	.0013	.0000	.0000	.0000	.0000
14	52.4180	.0001	.0000	.0000	.0000	.0000

TABLE III
PERFORMANCE OF THE PROPOSED ITERATIVE DISTRIBUTED ALGORITHM TO RECOVER SIGNALS ON THE RANDOM GEOMETRIC GRAPH RGG_{4096} IN FIGURE 2

$E_{m,r}$ m \ r	0	1	2	3	4	6
1	1.5267	.4674	.1487	.0437	.0159	.0018
2	2.8586	.1098	.0120	.0011	.0001	.0000
3	6.7494	.0374	.0014	.0000	.0000	.0000
4	16.4089	.0121	.0002	.0000	.0000	.0000
5	40.9430	.0041	.0000	.0000	.0000	.0000
8	672.8632	.0002	.0000	.0000	.0000	.0000

parameter r is chosen. The convergence rate of iterative algorithm depends on the radius parameter r, as observed from (60) and (62) in Theorem VII.2. Notice that graph filters $\mathbf{J}$, $\mathbf{J}_0$ and $\mathbf{J}_1$ have their bandwidths satisfying $\sigma(\mathbf{J}) \le 4r, \sigma(\mathbf{J}_0) \le 4r+\sigma$ and $\sigma(\mathbf{J}_1) \le 4r+\sigma$. Hence for a larger r, heavier burden arises at each iteration, which implies that each vertex in the graph $\mathcal{G}$ should have more data storages, better computing abilities and stronger communication capacities in real world applications. Shown in Tables II and III are the average $E_{m,r}$ of relative maximal reconstruction error $\|\mathbf{x}^{(m)} - \mathbf{x}\|_\infty / \|\mathbf{x}\|_\infty$ over 50 trials versus the number $m \ge 1$ of iterations and the radius parameter $r \ge 0$, where $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ with $n = 2$ are used as analysis filter banks, the signal $\mathbf{x}$ in Tables II and III is randomly selected on the Minnesota traffic graph and on the random geometric graph RGG_{4096} in Figure 2 respectively. This demonstrates that the iterative algorithm (57) and (58) converges faster for larger radius r, and the original signal can be well approximated in one step when a large radius r is chosen.

By (60) and Theorem VII.2, there is a radius parameter r_0 such that the iterative algorithm (57) and (58) converges exponentially whenever $r \ge r_0$. We emphasize that the above radius parameter r_0 can be selected to be *independent* of the order of the graph $\mathcal{G}$. Our simulation indicates that the iterative algorithm (57) and (58) with $r = 0$, i.e. the Jacobi iterative method in Remark VII.1, diverges for some bounded inputs on the Minnesota traffic graph and on some random geometric graphs; see the first column of Tables II and III.

The iterative algorithm (57) and (58) can be realized in a distributed manner, because all matrices $\mathbf{H}_l = (h_l(i,j))_{i,j\in V}$ and $\mathbf{J}_l = (j_l(i,i'))_{i,i'\in V}, l = 0, 1$, have small bandwidths. The detailed implementation is given in Algorithm 1, see Figure 5 for the block diagram, in which $\mathbf{H}_{l,k} = (h_l(k,i))_{i\in B(k,\sigma)}$ and $\mathbf{J}_{l,k} = (j_l(k,i))_{i\in B(k,4r+\sigma)}, l = 0, 1$. In Algorithm 1, every vertex $k \in V$ is required to store data of size $O((4r+\sigma)^d)$, to perform $O((4r+\sigma)^d)$ algebraic manipulations in each iteration, and to transmit data to its $(4r+\sigma)$-hop neighbors twice in each iteration. By the exponential convergence of the iterative algorithm (57) and (58), the total cost for each agent to implement Algorithm 1 is $O((4r+\sigma)^d \log(\|\mathbf{x}\|_\infty/\varepsilon))$ (independent on N), and the total computational cost for the whole graph is $O((4r+\sigma)^d \log(\|\mathbf{x}\|_\infty/\varepsilon)N)$, where ε is the stop criterion.

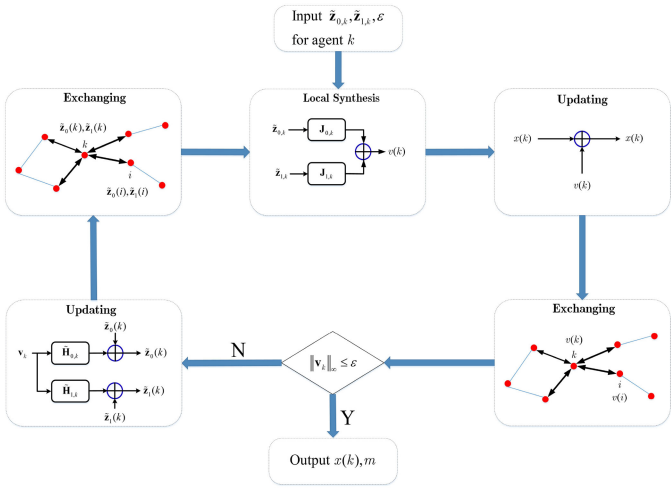


Fig. 5. Block diagram of the Algorithm 1.

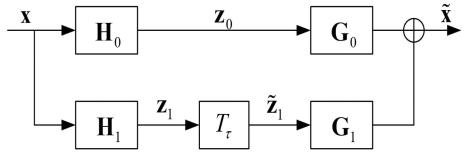


Fig. 6. Block diagram of an NSGFB-based denoising procedure, where $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are the noisy input and denoised output respectively, $(\mathbf{H}_0, \mathbf{H}_1)$ and $(\mathbf{G}_0, \mathbf{G}_1)$ are analysis filter bank and synthesis filter bank of the NSGFB respectively, and τ is the hard thresholding constant.

Algorithm 1: Iterative Distributed Algorithm on an Agent $k \in V$.

Inputs: stop criterion ε and observations $\tilde{\mathbf{z}}_{l,k} = (\tilde{z}_l(i))_{i\in B(k,4r+\sigma)}, l = 0, 1$.
Initialization: $x(k) = 0$ and $m = 0$.
Iteration:
1) Evaluate $v(k) = \langle \mathbf{J}_{0,k}, \tilde{\mathbf{z}}_{0,k} \rangle + \langle \mathbf{J}_{1,k}, \tilde{\mathbf{z}}_{1,k} \rangle$.
2) Update $x(k) = x(k) + v(k)$.
3) Send data $v(k)$ to all σ-hop neighbors, receive data $v(i), i \in B(k,\sigma)\backslash\{k\}$, from all σ-hop neighbors, and generate a vector $\mathbf{v}_k = (v(i))_{i\in B(k,\sigma)}$.
4) Evaluate $\tilde{z}_l(k) = \tilde{z}_l(k) - \langle \mathbf{H}_{l,k}, \mathbf{v}_k \rangle,\ l = 0, 1$.
5) Send data $\tilde{z}_0(k)$ and $\tilde{z}_1(k)$ to all $(4r+\sigma)$-hop neighbors, receive data $\tilde{z}_0(i)$ and $\tilde{z}_1(i), i \in B(k, 4r+\sigma)\backslash\{k\}$, from all $(4r+\sigma)$-hop neighbors, and generate vectors $\tilde{\mathbf{z}}_{l,k} = (\tilde{z}_l(i))_{i\in B(k,4r+\sigma)}, l = 0, 1$.
6) Evaluate $\|\mathbf{v}_k\|_\infty \le \varepsilon$. If yes, terminate the iteration and output $x(k)$ and m. Otherwise, set $m = m + 1$ and return to Step **1)**.
Outputs: $x(k)$ and m.

VIII. Distributed Denoising

For signal processing on an SDN without a fusion center, the recruiting of a distributed algorithm is indispensable since the whole adjacency matrix is not available at each agent due to the limited communication capacity of each agent on SDNs [7]. Given an NSGFB with analysis filter bank $(\mathbf{H}_0, \mathbf{H}_1)$ and synthesis filter bank $(\mathbf{G}_0, \mathbf{G}_1)$, we propose a denoising technique by applying hard thresholding operator $T_\tau, \tau \geq 0$, to the high-pass subband signal, where the hard thresholding function is $T_\tau(t) = tU(|t| - \tau)$ and the unit step function $U(t)$ takes value 1 for $t > 0$ and zero otherwise, cf. [27], [28], [30], [31]. Presented in Figure 6 is a block diagram of the proposed denoising algorithm. In this section, we demonstrate the performance of the proposed denoising procedure associated with spline/node-variant NSGFBs, which can be implemented in a distributed manner and hence it is beneficial to (local) noise suppression on SDNs (graphs) of very large size.

In the simulations, the noisy input is

$$\mathbf{x} = \mathbf{x}_o + \boldsymbol{\epsilon}, \tag{63}$$

where $\mathbf{x}_o = (x_o(i))_{i\in V}$ is the original graph signal and the random noise $\boldsymbol{\epsilon} = (\epsilon(i))_{i\in V}$ has value $\epsilon(i)$ at vertex $i \in V$ obeying uniform distribution on $[-\eta, \eta]$. In the simulations, we use the representative spline/node-variant NSGFBs. The spline NSGFBs contain analysis spline filter banks $(\mathbf{H}_{0,n}^{\mathrm{spln}}, \mathbf{H}_{1,n}^{\mathrm{spln}})$ in (3) and synthesis spline filter banks being either $(\mathbf{G}_{0,n}^{\mathrm{B,spln}}, \mathbf{G}_{1,n}^{\mathrm{B,spln}})$ in (34) or $(\mathbf{G}_{0,n}^{\mathrm{L,spln}}, \mathbf{G}_{1,n}^{\mathrm{L,spln}})$ in (47), where $n \geq 1$. They are abbreviated by NSGFB-Bn and NSGFB-Ln respectively. The above node-variant analysis filter bank (4) is not in the polynomial form (7) in general, however we can still construct well-localized synthesis filter bank $(\mathbf{G}_0^{\mathrm{L,nv}}, \mathbf{G}_1^{\mathrm{L,nv}})$ via solving the minimization problem (39), and we use the abbreviated notation NSGFB-NVF to represent the corresponding NSGFB.

The denoising procedure is performed by retaining the low-pass subband signal and applying the hard thresholding operation T_τ to the high-pass subband signal, where $\tau > 0$ is chosen appropriately. Thus the denoised output is

$$\tilde{\mathbf{x}} = \mathbf{G}_{0,n}^{\mathrm{B,spln}}\tilde{\mathbf{z}}_0 + \mathbf{G}_{1,n}^{\mathrm{B,spln}}\tilde{\mathbf{z}}_1 \tag{64}$$

for NSGFB-Bn,

$$\tilde{\mathbf{x}} = \mathbf{G}_{0,n}^{\mathrm{L,spln}}\tilde{\mathbf{z}}_0 + \mathbf{G}_{1,n}^{\mathrm{L,spln}}\tilde{\mathbf{z}}_1 \tag{65}$$

for NSGFB-Ln, and

$$\tilde{\mathbf{x}} = \mathbf{G}_0^{\mathrm{L,nv}}\tilde{\mathbf{z}}_0 + \mathbf{G}_1^{\mathrm{L,nv}}\tilde{\mathbf{z}}_1 \tag{66}$$

for NSGFB-NVF, respectively, where $\tilde{\mathbf{z}}_0 = \mathbf{z}_0, \tilde{\mathbf{z}}_1 = T_\tau(\mathbf{z}_1)$, and $\mathbf{z}_l = \mathbf{H}_{l,n}^{\mathrm{spln}}\mathbf{x}, n \geq 1$ in (64) and (65), or $\mathbf{z}_l = \mathbf{H}_l^{\mathrm{nv}}\mathbf{x}$ in (66), $l = 0, 1$ respectively. For the above denoising procedure, we use $20\log_{10}\|\mathbf{x}_o\|_p/\|\mathbf{x} - \mathbf{x}_o\|_p$ to measure the input ℓ^p-signal-to-noise ratio (ℓ^p-SNR) in dB, and $20\log_{10}\|\mathbf{x}_o\|_p/\|\tilde{\mathbf{x}} - \mathbf{x}_o\|_p$ to measure the output ℓ^p-SNR in dB, where $1 \leq p \leq \infty$.

In our simulations, the NSGFB-based denoising procedure is performed on a DELL PC by mimicking the distributed implementation in (18), (35) and Algorithm 1, except for the last two simulations on performance and time consumption to denoise signals residing on random geometric graphs of large order. In the spirit of reproducible research, we will make our matlab codes available to the public for academic purposes after the publication.

The Minnesota traffic graph is a test bed for various techniques in signal processing on graphs of medium order [10], [22], [24], [27]. The denoising performance of the proposed spline/node-variant NSGFBs on the Minnesota graph is presented in Table IV, where the original signal $\mathbf{x}_o$ is the blockwise constant function in Figure 2, the input and output ℓ^2-SNRs are the average values over 50 trials and the threshold value τ is selected to be 3η [24], [27], [44]. Shown also in Table IV are the performance comparisons with the biorthogonal graph filter bank (graphBior) in [23], the M-channel oversampled graph filter bank (OSGFB) in [24], the pyramid transform (PRT) in [27], the graph total variation regularization denoising method (GTVR) in [45], and the spectral graph trilateral filter denoising algorithm (SGTF) in [44], where the corresponding output ℓ^2-SNRs are calculated from the accompanying codes in these references. It indicates that the spline NSGFBs and the OSGFB outperform other four methods in the small noise scenario, the spline and node-variant based NSGFBs have the best performance in the moderate noise environment, and the SGTF stands out from the rest in the strong noise case.

TABLE IV
Denoising Performance on the Minnesota Traffic Graph Measured by the ℓ^2-SNR

η	1/32	1/16	1/8	1/4	1/2	1
Input	34.86	28.86	22.81	16.81	10.80	4.77
graphBior	34.40	28.91	23.99	18.20	12.81	7.37
OSGFB	38.32	**32.76**	24.56	16.78	12.53	4.71
PRT	35.28	29.60	23.74	18.52	15.61	12.87
GTVR	34.87	28.91	23.03	17.57	12.96	8.61
SGTF	34.92	28.96	23.48	18.93	**15.71**	**13.31**
NSGFB-B1	37.46	31.45	25.39	18.92	13.19	7.37
NSGFB-B2	37.19	30.79	24.94	18.57	13.34	7.65
NSGFB-L1	**38.45**	32.43	**26.37**	19.28	13.84	8.30
NSGFB-L2	37.19	30.71	24.88	18.21	13.35	7.85
NSGFB-NVF	38.29	32.26	26.22	**19.39**	13.85	8.16

TABLE V
Denoising Performance on the Random Geometric Graph RGG_{4096} Measured by the ℓ^2-SNR

η	1/32	1/16	1/8	1/4	1/2	1
Input	35.06	29.04	23.02	17.01	10.97	4.95
graphBior	33.82	28.61	23.27	18.20	13.21	8.34
OSGFB	31.69	26.37	20.79	16.40	13.40	11.13
PRT	32.73	27.45	22.38	17.48	13.96	11.63
GTVR	35.05	29.10	23.27	17.93	13.39	9.22
SGTF	35.08	29.13	23.52	18.16	14.23	**11.74**
NSGFB-B1	37.43	31.40	25.34	19.31	13.47	7.62
NSGFB-B2	36.65	30.63	24.89	19.37	13.80	8.25
NSGFB-L1	38.86	32.87	26.61	20.45	14.91	9.40
NSGFB-L2	36.08	29.97	24.27	19.16	13.92	9.05
NSGFB-NVF	**39.03**	**33.15**	**26.94**	**20.76**	**15.15**	9.63

We test the denoising performance of the proposed spline/node-variant NSGFBs on the random geometric graphs of medium order reproduced by the GSPToolbox [38]. Presented in Tables V and VI are the denoising performance of spline/node-variant NSGFBs and the performance comparison with graphBior, OSGFB, PRT, GTVR and SGTF on the random geometric graph RGG_{4096}, where $\mathbf{x}_o$ is the blockwise polynomial in Figure 2. In denoising procedure, the threshold value τ is selected to be 3η, and the ℓ^2-SNRs and ℓ^∞-SNRs are the average values over 50 trials. From the ℓ^2-SNR measurements in Table V, we notice that the NSGFB-NVF outperforms all other denoising methods in small and moderate noise scenarios, while the SGTF has the best performance in the strong noise scenario. From the

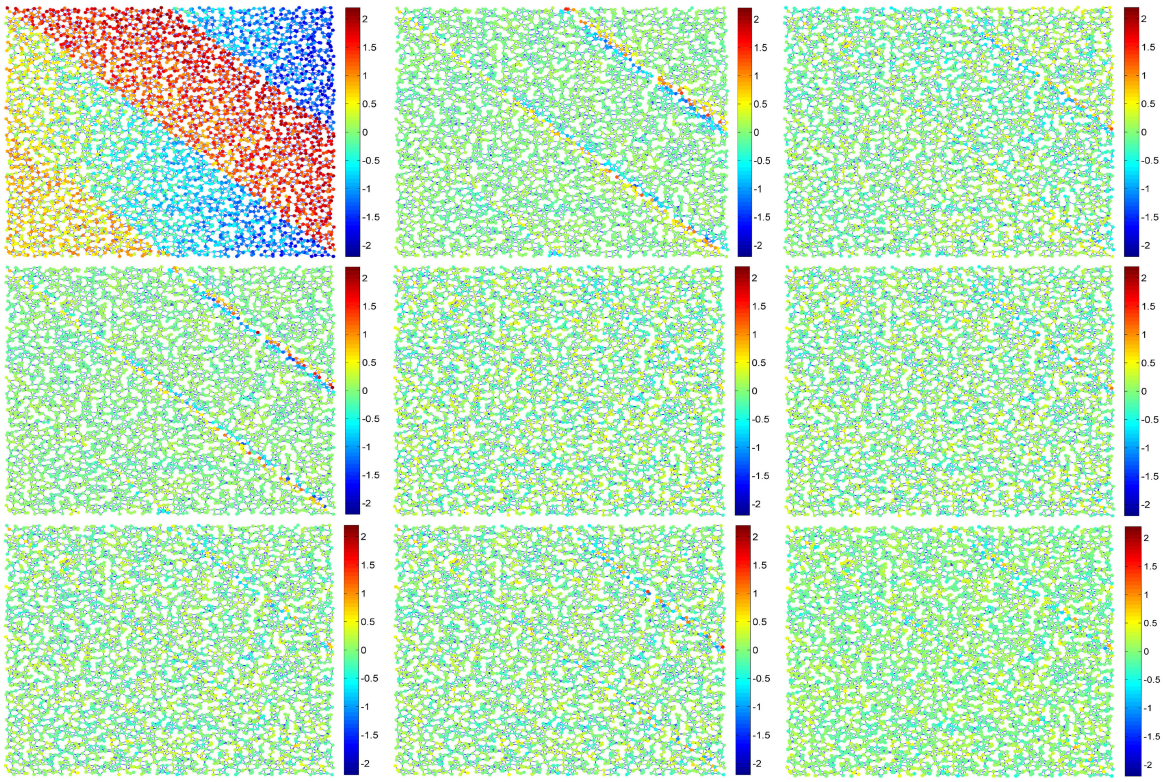


Fig. 7. Plotted are denoising performance comparison on the random geometric graph RGG_{4096}: from top left to top right are noisy signal on RGG_{4096}, residue through PRT, residue through GTVR respectively. From middle left to right are residue through SGTF, residue through NSGFB-B1, residue through NSGFB-B2 respectively. From bottom left to right are residue through NSGFB-L1, residue through NSGFB-L2, residue through NSGFB-NVF respectively.

TABLE VI
DENOISING PERFORMANCE ON THE RANDOM GEOMETRIC GRAPH RGG_{4096} MEASURED BY THE ℓ^∞-SNR

η	$1/32$	$1/16$	$1/8$	$1/4$	$1/2$	1
Input	34.90	28.88	22.85	16.83	10.81	4.79
graphBior	23.45	17.28	11.12	5.34	0.32	-4.00
OSGFB	20.59	14.32	6.83	0.99	-1.99	-2.71
PRT	23.29	17.28	11.22	5.32	0.19	-2.33
GTVR	**34.88**	**27.45**	**19.50**	10.58	3.60	2.01
SGTF	30.56	26.67	14.63	2.81	-1.76	-2.71
NSGFB-B1	31.84	25.16	18.71	**11.05**	**6.55**	**2.60**
NSGFB-B2	26.86	20.34	14.67	8.32	3.55	1.67
NSGFB-L1	29.28	22.70	16.19	9.28	4.08	0.35
NSGFB-L2	24.66	17.91	12.27	6.72	0.48	-0.52
NSGFB-NVF	30.29	22.64	16.60	10.42	4.46	0.55

ℓ^∞-SNR measurements in Table VI, we observe that the GTVR and the proposed spline/node-variant NSGFBs have comparable performances on denoising. In particular, the GTVR surpasses other denoising methods in small and moderate noise scenarios and the proposed spline NSGFBs stand out in the strong noise scenario. Also from Table V and VI, we see that for $p = 2$ and $p = \infty$, differences between the output ℓ^p-SNRs and the input ℓ^p-SNRs are in some range independent on the input noise level η. As ℓ^p-SNRs are in the logarithm scale of ℓ^p fidelity measurements, this confirms the conclusions in Propositions V.2 and VI.4 that the output error in the ℓ^p fidelity measurement is dominated by a multiple of the input noise.

Shown in Figure 7 is the comparison of the denoising performance of the proposed spline/node-variant NSGFBs, graphBior, SGTF, GTVR for the signal used for Tables V and VI with noise level $\eta = 1/2$ [24], [27]. We notice that all denoising techniques have satisfactory performance inside the strip where the signal has small variation, and that the proposed spline/node-variant NSGFBs have better performance on noise suppression than the other three methods do near the boundary of two adjacency strips where the signal has large variation.

We next demonstrate the denoising performance of the proposed spline /node-variant NSGFBs on some real dataset, particularly on the sea surface temperature dataset published by the Earth System Research Laboratory [46], [47]. The dataset is acquired at 100 station locations on the Pacific ocean from 170° west to 90° west and from 60° south to 10° north with dynamic range from -1.32 °C to 30.72 °C, and each station records monthly mean of the sea surface temperature within a time period of 1733 months from January 1870 to May 2014. The underlying graph $\mathcal{G}$ of the dataset has 100 vertices representing observation stations and edges given by 5-nearest neighboring stations in physical distances [48]. For $1 \leq t \leq 1733$, let $\mathbf{x}_o^t$ be the graph signal that contains the sea surface temperature of 100 observation stations at the t-th month, see Figure 8. In the simulation, subband signals at the low frequency are retained and other subband signals are proceeded by hard-thresholding with threshold $\tau = 3\eta$, except that for the GTVR, we use the algorithm in [45]. We compare our NSGFBs with the graphBior in [23], OSGFB in [24], PRT in [27], GTVR in [45] and SGTF in [44]. Shown in Table VII are the average input ℓ^2-SNRs with noise level $\eta = 1, 2, 5, 10$, and the average output ℓ^2-SNRs of $\mathbf{x}_o^t$, $1 \leq t \leq 1733$, where one denoising test is performed on a monthly signal on the sea surface temperature. It is observed that the proposed NSGFBs have better performance on

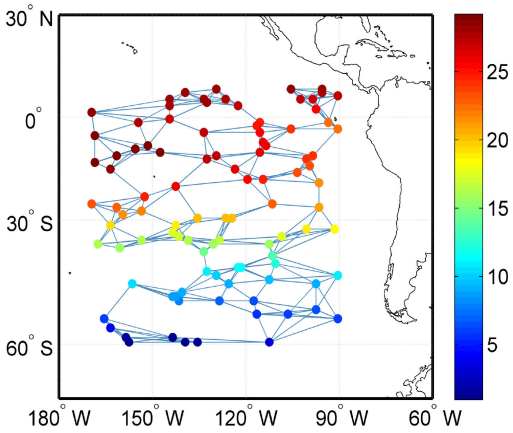


Fig. 8. The graph signal that contains monthly mean of the sea surface temperature of 100 observation stations in June 2013.

TABLE VII
DENOISING PERFORMANCE ON THE SEA SURFACE TEMPERATURE DATASET MEASURED BY THE ℓ^2-SNR

η	1	2	5	10
Input	31.20	25.18	17.25	11.21
graphBior	29.32	26.72	20.83	15.16
OSGFB	26.01	23.20	22.12	19.77
PRT	27.44	20.96	11.08	8.65
GTVR	30.48	25.61	18.96	15.21
SGTF	32.32	28.86	**24.08**	**20.22**
NSGFB-B1	32.77	27.66	19.98	13.98
NSGFB-B2	**33.01**	28.25	20.76	14.79
NSGFB-L1	32.59	29.10	22.34	16.50
NSGFB-L2	32.33	29.00	22.30	16.49
NSGFB-NVF	32.15	**29.22**	23.14	17.45

denoising than other methods for $\eta = 1, 2$, while the SGTF method stands out for $\eta = 5, 10$.

The proposed denoising algorithm is designed for the distributed implementation in an SDN with each agent has limited computing and communicating capability. However due to localization and sparsity of matrices $\mathbf{J}_l$ and $\mathbf{H}_l, l = 0, 1$, it can be implemented in a centralized facility (and also we believe in a parallel computing framework) very efficiently. So we may apply the proposed spline NSGFB-based denoising algorithm to denoise signals residing on random geometric graphs of large orders $10000 \leq N \leq 300000$, and we perform the simulations via Matlab2013b on a DELL T7910 workstation with 20 cores, where the workstation configuration is Intel(R) Xeon(R) CPU with two E5-2630 v4 processors (2.20 Hz) and 64G memory. Shown in Table VIII are average values of the ℓ^2-SNRs over 50 trials, where the upper row for each graph order N corresponds to the SNRs of input noisy signals while the lower row stands for the SNRs of denoised signals by NSGFB-L1. In Table VIII, the denoising performance for signals on a random geometric graph of order $N = 4096$ is copied from Table V, and in the synthesis procedure (65) we use the iterative algorithm (57) and (58) with the maximal signal adjustment $\|\mathbf{v}^{(m)}\|_\infty \leq 10^{-4}$ as a stopping criterion and the radius parameter $r = 2$. We observe that the stopping criterion is met after 4 iterations for all orders N and noise levels η listed in Table VIII. Also, we observe that the output ℓ^2-SNRs of spline NSGFBs have invisible change for the same input noise level when the order N of the graph

TABLE VIII
DENOISING PERFORMANCE ON RANDOM GEOMETRIC GRAPHS OF ORDER N MEASURED BY THE ℓ^2-SNR

N \ η	1/32	1/16	1/8	1/4	1/2	1
4096	35.06	29.04	23.02	17.01	10.97	4.95
	38.86	32.87	26.61	20.45	14.91	9.40
10000	35.04	29.02	23.00	16.98	10.95	4.95
	39.24	33.23	27.04	20.86	15.11	9.55
20000	35.01	28.98	22.95	16.95	10.92	4.90
	39.47	33.41	27.25	21.13	15.35	9.61
40000	35.04	29.02	23.00	16.99	10.96	4.94
	39.66	33.60	27.49	21.38	15.48	9.69
80000	35.04	29.02	23.00	16.98	10.96	4.94
	39.73	33.68	27.59	21.51	15.62	9.75
300000	35.04	29.02	23.00	16.98	10.96	4.94
	39.99	33.95	27.89	21.86	15.89	9.96

TABLE IX
RUNTIME IN SECONDS OF DENOISING ALGORITHMS ON RANDOM GEOMETRIC GRAPHS OF ORDER N

N	10000	20000	40000	80000	300000
Bior	184.57	759.01	5358.7	ERR	ERR
OSGFB	422.24	1754.2	8375.0	ERR	ERR
PRT	318.38	ERR	ERR	ERR	ERR
NSGFB-L1	0.0480	0.1059	0.2487	0.5008	2.8296
NSGFB-L2	0.1114	0.2532	0.5849	1.2539	7.2379
NSGFB-B1	0.0040	0.0080	0.0160	0.0349	0.1761
NSGFB-B2	0.0052	0.0101	0.0203	0.0423	0.2312
NSGFB-NVF	0.0537	0.1235	0.2760	0.6076	3.4792
GTVR	0.0565	0.1320	0.2864	0.6556	4.3919
SGTF	218.76	891.04	5360.7	ERR	ERR

$\mathcal{G}$ increases. This gives the prominent potential of the proposed algorithm (57) and (58) to denoise signals on graphs of very large order.

In addition to the satisfactory denoising performance on the random geometric graphs of large order, as shown in Table IX, the proposed denoising algorithms based on NSGFB-NVF, NSGFB-L1, NSGFB-B1 and NSGFB-B2 spend less time in their implementation than graphBior, OSGFB, PRT, GTVR and SGTF do, except that GTVR consumes less time than the iterative denoising algorithm based on NSGFB-L2 does. Here listed in Table IX are average running times in seconds over 50 trials to implement denoising algorithms on random geometric graphs of large order N, where noise level is $\eta = 1/2$ and the notion ERR means that either the running is above 10000 seconds or that algorithm is not applicable.

The Algorithm 1 is implementable on SDNs in which there is no fusion center and each agent is equipped with computing and communication subsystems. Due to our facility limitation, presented in the simulations above are the NSGFB-based denoising procedure performed on a PC or a workstation by mimicking the distributed implementation. We finish this section with a signal denoising simulation on random geometric graph of order N implemented on the Matlab distributed computing server (MDCS) configured across 3 computers (each with 3.6 GHz i7CPU and 8G-memory) that totally have 12 workers. The denoising performance of NSGFB-L1 is the same as that in Table VIII, while the average CPU time of operating one denoising test for $N = 10000, 20000, 40000, 80000, 300000$ are $39.30, 175.85, 720.23, 3083.40, 5.5451 \times 10^4$ seconds, respectively, cf. Table IX on the time consumption to implement signal

denoising on a workstation. The underlying reason could be the extra communication cost arisen in data exchanging between 12 workers in each iteration. We remark that the implementation in Algorithm 1 is designed to minimize the computing cost. The communication cost could be reduced if in Step 1) of Algorithm 1 the agent k evaluates all neighboring values $v(j)$ in its σ-hop neighborhood instead of $v(k)$ only and then the communication Step 3) in the middle of each iteration could be eliminated.

IX. Conclusions

The downsampling and upsampling procedure has been used in graph wavelet filter banks, while its proper definition is not obvious especially when the residing graph is of large order and complicated topological structure. For signal processing on graphs of large order, we consider an NSGFB that does not include the downsampling and upsampling procedure. This greatly simplifies the design of well-localized NSGFBs to satisfy the perfect reconstruction condition. The analysis filter bank in an NSGFB can be properly selected to have small bandwidth and to decompose a graph signal into two components carrying different frequency information. Synthesis filter banks can be constructed from the proposed algebraic and optimization approaches, which can be implemented in a distributed manner, and hence the design will be applicable for SDNs with each agent equipped with limited data processing and communication abilities. In this paper, NSGFBs are shown to be beneficial to (local) noise suppression and a distributed denoising technique based on NSGFBs is demonstrated to have satisfactory performance against bounded noises.

Exemplary analysis filter banks of NSGFBs in this paper are polynomials of a shift matrix of the graph, while the optimal graph filters associated with some fidelity measures are not of polynomial form in some applications. Future works will focus on the design of adaptive analysis filter banks for better frequency decomposition and various applications.

Appendix

A. Proof of Theorem IV.4

First we establish the lower bound estimate (29). Set

$$\mathbf{B} = \mathbf{I} - \mathbf{H}/\|\mathbf{H}\|_{\mathcal{B}_2}. \tag{A.1}$$

Then $\mathbf{B}$ has bandwidth 2σ, $\|\mathbf{B}\|_{\mathcal{B}_2} \le (\kappa-1)/\kappa$, and $\mathbf{H}^{-1} = (\|\mathbf{H}\|_{\mathcal{B}_2})^{-1}\sum_{n=0}^{\infty}\mathbf{B}^n$. Write $\mathbf{H}^{-1} = (g(i,j))_{i,j\in V}$. For $\kappa = 1$, we have

$$\mathbf{H}^{-1} = (\|\mathbf{H}\|_{\mathcal{B}_2})^{-1}\mathbf{I} \text{ and } \|\mathbf{H}^{-1}\|_{\mathcal{B}_p} = \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}. \tag{A.2}$$

Now we consider the case that $\kappa > 1$. Set $\theta = \ln(\kappa/(\kappa-1))$, and for $i, j \in V$ let $n_0(i,j)$ be the minimal integer such that $2n_0(i,j) \ge \rho(i,j)/\sigma$. Then

$$\begin{aligned}
|g(i,j)| &= (\|\mathbf{H}\|_{\mathcal{B}_2})^{-1}\left|\sum_{n=n_0(i,j)}^{\infty}\mathbf{B}^n(i,j)\right| \\
&\le (\|\mathbf{H}\|_{\mathcal{B}_2})^{-1}\sum_{n=n_0(i,j)}^{\infty}\|\mathbf{B}^n\|_{\mathcal{B}_2} \\
&\le \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\exp\left(-\frac{\theta}{2\sigma}\rho(i,j)\right),
\end{aligned} \tag{A.3}$$

where the equality follows from the observation that $\mathbf{B}^n$ have bandwidth $2n\sigma$, the first inequality holds by (14), and the second inequality is obtained from the definition of the minimal integer $n_0(i,j)$ and the following estimate $\|\mathbf{B}^n\|_{\mathcal{B}_2} \le \|\mathbf{B}\|_{\mathcal{B}_2}^n \le (1-1/\kappa)^n, n \ge 1$. By (A.3) and the second inequality in (14) we have

$$\begin{aligned}
\|\mathbf{H}^{-1}\|_{\mathcal{B}_p} &\le \|\mathbf{H}^{-1}\|_{\mathcal{B}_2} \\
&\quad\times \sup_{i\in V}\sum_{n=0}^{\infty}\sum_{2n\sigma\le\rho(i,j)<2(n+1)\sigma}\exp\left(-\frac{\theta}{2\sigma}\rho(i,j)\right) \\
&\le \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\sup_{i\in V}\sum_{n=0}^{\infty}e^{-n\theta}\mu\left(B(i,2(n+1)\sigma-1)\right) \\
&\le (2\sigma)^d D_1(\mathcal{G})\|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\sum_{n=0}^{\infty}(n+1)^d(1-\kappa^{-1})^n \\
&\le (2\sigma)^d D_1(\mathcal{G})\|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\left(\left(\frac{1}{1-t}\right)^{(d)}|_{t=1-\kappa^{-1}}\right) \\
&\le d!(2\sigma)^d D_1(\mathcal{G})\kappa^{d+1}\|\mathbf{H}^{-1}\|_{\mathcal{B}_2}
\end{aligned} \tag{A.4}$$

if $\kappa > 1$. Then

$$\begin{aligned}
\|\mathbf{x}\|_p &\le \|\mathbf{H}^{-1}\|_{\mathcal{B}_p}(\|\mathbf{H}_0^T\|_{\mathcal{B}_p}\|\mathbf{H}_0\mathbf{x}\|_p + \|\mathbf{H}_1^T\|_{\mathcal{B}_p}\|\mathbf{H}_1\mathbf{x}\|_p) \\
&\le d!2^d(\sigma+1)^{2d}(D_1(\mathcal{G}))^2\kappa^{d+1}\|\mathbf{H}^{-1}\|_{\mathcal{B}_2} \\
&\quad\times(\|\mathbf{H}_0\|_{\mathcal{B}_2}\|\mathbf{H}_0\mathbf{x}\|_p + \|\mathbf{H}_1\|_{\mathcal{B}_2}\|\mathbf{H}_1\mathbf{x}\|_p) \\
&\le d!2^{d+1}(\sigma+1)^{2d}(D_1(\mathcal{G}))^2\kappa^{d+2}\|\mathbf{H}\|_{\mathcal{B}_2}^{-1/2} \\
&\quad\times(\|\mathbf{H}_0\mathbf{x}\|_p^p + \|\mathbf{H}_1\mathbf{x}\|_p^p)^{\frac{1}{p}},
\end{aligned} \tag{A.5}$$

where the second inequality follows from (A.2) and (A.4), and the third one holds by (25), (28) and Proposition III.3. This proves (29), as C_p is the largest constant for (22) and (23) to hold.

Next we prove (30). Take $\mathbf{x} \in \ell^p, 1 \le p < \infty$, we have

$$\begin{aligned}
&(\|\mathbf{H}_0\mathbf{x}\|_p^p + \|\mathbf{H}_1\mathbf{x}\|_p^p)^{1/p} \\
&\le (\|\mathbf{H}_0\|_{\mathcal{B}_p}^p\|\mathbf{x}\|_p^p + \|\mathbf{H}_1\|_{\mathcal{B}_p}^p\|\mathbf{x}\|_p^p)^{1/p} \\
&\le D_1(\mathcal{G})(1+\sigma)^d(\|\mathbf{H}_0\|_{\mathcal{B}_2} + \|\mathbf{H}_1\|_{\mathcal{B}_2})\|\mathbf{x}\|_p \\
&\le 2D_1(\mathcal{G})(1+\sigma)^d\|\mathbf{H}\|_{\mathcal{B}_2}\|\mathbf{x}\|_p,
\end{aligned}$$

where the first inequality follows from the definition of the operator norm $\|\cdot\|_{\mathcal{B}_p}$ of a matrix, the second inequality holds by (14), and the last inequality holds by (25). This proves (30) for $1 \le p < \infty$, as D_p is the smallest constant for (22) and to hold.

The conclusion (30) for $p = \infty$ can be proved by a similar argument.

B. Proof of Proposition V.2

Set $\mathbf{z}_0 = \mathbf{H}_0\mathbf{x}$ and $\mathbf{z}_1 = \mathbf{H}_1\mathbf{x}$. Then

$$\begin{aligned}
\|\tilde{\mathbf{x}} - \mathbf{x}\|_p &\le \|\mathbf{G}_0(\mathbf{z}_0 - \Psi_0(\mathbf{z}_0))\|_p + \|\mathbf{G}_1(\mathbf{z}_1 - \Psi_1(\mathbf{z}_1))\|_p \\
&\le (\|\mathbf{G}_0\|_{\mathcal{B}_p} + \|\mathbf{G}_1\|_{\mathcal{B}_p})\epsilon \\
&\le D_1(\mathcal{G})(\tilde{\sigma}+1)^d(\|\mathbf{G}_0\|_\infty + \|\mathbf{G}_1\|_\infty)\epsilon,
\end{aligned} \tag{A.6}$$

where the first inequality follows from the perfect reconstruction condition (1) for the NSGFB constructed in Theorem V.1, the second one holds by (37), and the last estimate is true by Proposition III.3.

C. Proof of Theorem VI.1

By (42) and (A.3), we have

$$|g_l^L(i,j)| \le \|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}_l\|_\infty \sum_{\rho(k,j)\le\sigma} \exp\left(-\frac{\theta}{2\sigma}\rho(i,k)\right)$$
$$\le D_1(\mathcal{G})\|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}_l\|_\infty(\sigma+1)^d \times \exp\left(-\frac{\theta}{2\sigma}\rho(i,j)+\frac{\theta}{2}\right),\ i,j\in V,$$

where $l=0,1$. This proves (44).

D. Proof of Theorem VII.2

Set $\mathbf{y}^{(m)}=\tilde{\mathbf{x}}-\mathbf{x}^{(m)}$ and write $\mathbf{y}^{(m)}=(y^{(m)}(i))_{i\in V}, m\ge 0$. We claim that

$$\mathbf{y}^{(m)}=\mathbf{H}^{-1}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0^{(m)}+\mathbf{H}_1^T\tilde{\mathbf{z}}_1^{(m)}),\ m\ge 0. \tag{A.7}$$

The above claim holds for $m=0$, since

$$\mathbf{y}^{(0)}=\tilde{\mathbf{x}}=\mathbf{H}^{-1}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0+\mathbf{H}_1^T\tilde{\mathbf{z}}_1)=\mathbf{H}^{-1}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0^{(0)}+\mathbf{H}_1^T\tilde{\mathbf{z}}_1^{(0)})$$

by (48) and (58). Inductively for $m\ge 1$, we have

$$\mathbf{y}^{(m)}=\mathbf{H}^{-1}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0^{(m-1)}+\mathbf{H}_1^T\tilde{\mathbf{z}}_1^{(m-1)})-\mathbf{v}^{(m)}$$
$$=\mathbf{H}^{-1}(\mathbf{H}_0^T\tilde{\mathbf{z}}_0^{(m)}+\mathbf{H}_1^T\tilde{\mathbf{z}}_1^{(m)}),$$

where the first and second equality follows from the inductive hypothesis and (57) respectively. This completes the proof of Claim A.7.

Write $(\chi_k^{2r}\mathbf{H}\chi_k^{2r})^{-1}=(g_k(i,j))_{i,j\in B(k,2r)}$ and

$$\chi_k^r(\chi_k^{2r}\mathbf{H}\chi_k^{2r})^\dagger\chi_k^{2r}\mathbf{H}(\chi_k^{2r+2\sigma}-\chi_k^{2r})=(\tilde{g}_k(i,j))_{i,j\in V}, k\in V. \tag{A.8}$$

Following the argument used to prove (A.3), we have

$$|g_k(i,j)|\le\|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\exp\left(-\frac{\theta}{2\sigma}\rho(i,j)\right) \tag{A.9}$$

for all $i,j\in B(k,2r)$. By (A.8) and (A.9), we obtain

$$\tilde{g}_k(i,j)=0 \tag{A.10}$$

where either $i\notin B(k,r)$ or $j\notin B(k,2r+2\sigma)\backslash B(k,2r)$, and

$$|\tilde{g}_k(i,j)|\le\|\mathbf{H}^{-1}\|_{\mathcal{B}_2}\|\mathbf{H}\|_\infty\sum_{l\in B(j,2\sigma)}\exp\left(-\frac{\theta}{2\sigma}\rho(i,l)\right)$$
$$\le D_1(\mathcal{G})(2\sigma+1)^d\kappa\exp\left(-\frac{\theta}{2\sigma}r+\theta\right), \tag{A.11}$$

where $i\in B(k,r)$ and $j\in B(k,2r+2\sigma)\backslash B(k,2r)$.

Write $\mathbf{v}_k^{(m)}=(v_k^{(m)}(i))_{i\in V}, m\ge 1, k\in V$. By (57), (A.7), we have

$$\chi_k^r(\mathbf{v}_k^{(m)}-\mathbf{y}^{(m-1)})=\chi_k^r(\chi_k^{2r}\mathbf{H}\chi_k^{2r})^\dagger\chi_k^{2r}\times\mathbf{H}(\chi_k^{2r+2\sigma}-\chi_k^{2r})\mathbf{y}^{(m-1)}.$$

Combining the above equation with (A.10) and (A.11), we get

$$|v_k^{(m)}(i)-y^{(m-1)}(i)|=\left|\sum_{j\in B(k,2r+2\sigma)}\tilde{g}_k(i,j)y^{(m-1)}(j)\right|$$
$$\le D_1(\mathcal{G})(2\sigma+1)^d\kappa\exp\left(-\frac{\theta}{2\sigma}r+\theta\right)\times\left(\sum_{j\in B(i,3r+2\sigma)}|y^{(m-1)}(j)|\right),\ i\in B(k,r). \tag{A.12}$$

This together with (57) implies that

$$|y^{(m)}(i)|=|v^{(m)}(i)-y^{(m-1)}(i)|$$
$$\le\frac{1}{\mu(B(i,r))}\sum_{k\in B(i,r)}|v_k^{(m)}(i)-y^{(m-1)}(i)|$$
$$\le D_1(\mathcal{G})(2\sigma+1)^d\kappa\exp\left(-\frac{\theta}{2\sigma}r+\theta\right)\times\left(\sum_{j\in B(i,3r+2\sigma)}|y^{(m-1)}(j)|\right) \tag{A.13}$$

for all $i\in V$ and $m\ge 1$. Using the above componentwise estimate, we obtain

$$\|\mathbf{y}^{(m+1)}\|_p\le\delta_{r,\sigma}\|\mathbf{y}^{(m)}\|_p,\ m\ge 0. \tag{A.14}$$

Iteratively applying the above estimate proves (62).

ACKNOWLEDGMENT

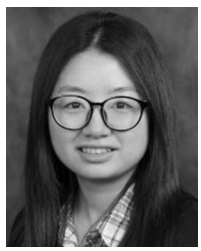
The authors would like to thank the associate editor and all anonymous reviewers for their constructive suggestions and comments to improve the manuscript.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, pp. 393–422, Mar. 2002.
[2] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.
[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, pp. 2292–2330, Aug. 2008.
[4] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Trans. Autom. Control*, vol. 53, no. 7, pp. 1616–1629, Aug. 2008.
[5] J. You and W. Wu, "Online passive identifier for spatially distributed systems using mobile densor networks," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2151–2159, Nov. 2017.
[6] R. Hebner, "The power grid in 2030," *IEEE Spectr.*, vol. 54, no. 4, pp. 51–55, Apr. 2017.
[7] C. Cheng, Y. Jiang, and Q. Sun, "Spatially distributed sampling and reconstruction," *Appl. Comput. Harmon. Anal.*, in press. [Online]. Available: https://doi.org/10.1016/j.acha.2017.07.007
[8] F. Chung and L. Lu, *Complex Graphs and Networks* (CBMS Regional Conference Series in Mathematics 107). Providence, RI, USA: Amer. Math. Soc., 2006.
[9] R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, vol. 26, pp. 53–94, Jul. 2006.
[10] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[11] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, Sep. 2014.
[12] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
[13] 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.
[14] I. Daubechies, *Ten Lectures on Wavelets* (CBMS-NSF Regional Conference Series in Applied Mathematics 61). Philadelphia, PA, USA: SIAM, 1992.
[15] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. New York, NY, USA: Academic, 2009.
[16] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.
[17] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *Proc. 20nd Annu. Joint Conf. IEEE Comput. Commun.*, vol. 3, 2003, pp. 1848–1857.
[18] M. Gavish, B. Nader, and R. R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 367–374.
[19] G. Shen and A. Ortega, "Transform-based distributed data gathering," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3802–3815, Jul. 2010.
[20] W. Wang and K. Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, 2006, pp. 161–164.
[21] D. K. Hammod, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, pp. 129–150, Mar. 2011.
[22] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2786–2799, Jun. 2012.
[23] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4673–4685, Oct. 2013.
[24] Y. Tanaka and A. Sakiyama, "M-channel oversampled graph filter banks," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3578–3590, Jul. 2014.
[25] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, "Spline-like wavelet filterbanks for multiresolution analysis of graph-structured data," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 268–278, Dec. 2015.
[26] H. Q. Nguyen and M. N. Do, "Downsampling of signals on graphs via maximum spanning trees," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 182–191, Jan. 2015.
[27] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, "A multiscale pyramid transform for graph signals," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 2119–2134, Apr. 2016.
[28] N. Tremblay and P. Borgnat, "Subgraph-based filterbanks for graph signals," *IEEE Trans. Signal Process.*, vol. 64, no. 15, pp. 3827–3840, Aug. 2016.
[29] N. Leonardi and D. Van De Ville, "Tight wavelet frames on multislice graphs," *IEEE Trans. Signal Process.*, vol. 16, no. 13, pp. 3357–3367, Jul. 2013.
[30] A. Sakiyama, K. Watanabe, and Y. Tanaka, "Spectral graph wavelets and filter banks with low approximation error," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 3, pp. 230–245, Sep. 2016.
[31] A. Sakiyama and Y. Tanaka, "Oversampled graph Laplacian matrix for graph filter banks," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6425–6437, Dec. 2014.
[32] M. S. Kotzagiannidis and P. L. Dragotti, "Sampling and reconstruction of sparse signals on circulant graphs—An introduction to graph-FRI," *Appl. Comput. Harmon. Anal.*, to be pubished. [Online]. Available: https://doi.org/10.1016/j.acha.2017.10.003
[33] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it?- A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.
[34] Q. Sun, "Localized nonlinear functional equations and two sampling problems in signal processing," *Adv. Comput. Math.*, vol. 40, pp. 415–458, Apr. 2014.
[35] S. Chen, A. Sandryhaila, and J. Kovačević, "Distributed algorithm for graph signal inpainting," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, 2015, pp. 3731–3735.
[36] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp 736–751, Dec. 2018.
[37] Seroul, R., "The Bezout Theorem," *Programming for Mathematicians*, New York, NY, USA: Springer Verlag, 2000, p. 10, Sec. 2.4.1.
[38] P. Nathanael *et al.*, "GSPBOX: A toolbox for signal processing on graphs," Aug. 2014, arXiv: 1408.5781.
[39] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Appl. Comput. Harmon. Anal.*, vol. 40, pp. 260–291, Mar. 2016.
[40] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.
[41] C. E. Shin and Q. Sun, "Stability of localized operators," *J. Funct. Anal.*, vol. 256, pp. 2417–2439, Apr. 2009.
[42] C. E. Shin and Q. Sun, "Polynomial control on stability, inversion and powers of matrices on simple graphs," *J. Funct. Anal.*, vol. 276, pp. 148–182, Jan. 2019.
[43] N. Motee and Q. Sun, "Sparsity and spatial localization measures for spatially distributed systems," *SIAM J. Control Optim.*, vol. 55, pp. 200–235, Jan. 2017.
[44] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, Jun. 2016.
[45] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Atlanta, GA, USA, 2014, pp. 872–876.
[46] "Sea surface temperature (SST) v2," Dec. 18, 2015. [Online]. Available: http://www.esrl.noaa.gov/psd/data/gridded/data.noaa.oisst.v2.html
[47] K. Qiu, X. H. Mao, X. Y. Shen, X. H. Wang, T. J. Li, and Y. T. Gu, "Time-varying graph signal reconstruction," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 870–883, Sep. 2017.
[48] D. Eppstein, M. S. Paterson, and F. F. Yao, "On nearest-neighbor graphs," *Discrete Comput. Geom.*, vol. 17, pp. 263–282, Apr. 1997.

Junzheng Jiang (M'19) received the B.S. degree in applied math from Guilin University of Electronic Technology, Guilin, China, in 2005, and the Ph.D. degree in information and communication engineering from Xidian University, Xi'an, China, in 2011. Since then, he has joined the Guilin University of Electronic Technology, where he is currently a Full Professor and an Advisor of Ph.D. student. He was a Visiting Scholar at the University of Central Florida, USA, from February 2016 to February 2017. His research interests include graph filter bank, distributed signal processing on graphs, time-vertex signal processing on graphs.



Cheng Cheng received the B.S. and M.S. degrees from Dalian Maritime University, Dalian, China, in 2009 and 2011, the Ph.D. degree in mathematics from University of Central Florida, Orlando, FL, USA, in 2017. She is currently a postdoctoral fellow at Statistical and Applied Mathematical Science Institute and Duke University. Her research interests include applied and computational harmonic analysis, sampling theory, phaseless sampling and reconstruction, and graph signal processing.

Qiyu Sun received the Ph.D. degree in mathematics from Hangzhou University, Hangzhou, China, in 1990. He is currently a professor of mathematics at the University of Central Florida, Orlando, FL, USA. His prior position was with the Zhejiang University, China, National University of Singapore, Singapore, Vanderbilt University, Nashville, TN, USA, and the University of Houston, Houston, TX, USA. His research interests include applied and computational harmonic analysis, sampling theory, phase retrieval and graph signal processing. He has published more than 100 papers and written a book *An Introduction to Multiband Wavelets* with N. Bi and D. Huang.