On the Shift Operator, Graph Frequency and Optimal Filtering in Graph Signal Processing

Adnan Gavili and Xiao-Ping Zhang, Senior member, IEEE

Abstract

Defining a sound shift operator for signals existing on a certain graph structure, similar to the well-defined shift operator in classical signal processing, is a crucial problem in graph signal processing, since almost all operations, such as filtering, transformation, prediction, are directly related to the graph shift operator. We define a set of energy conservation shift operators that satisfy many properties similar to their counterpart in classical signal processing. Our definition of the graph shift operator negates the shift operators defined in the literature, such as the graph adjacency matrix and Laplacian matrix based shift operators, which modify the energy of a graph signal. We decouple the graph structure represented by eigengraphs and the eigenvalues of the adjacent matrix or the Laplacian matrix. We show that the adjacency matrix of a graph is indeed a linear shift invariant (LSI) graph filter with respect to the defined shift operator. We introduce graph finite impulse response (GFIR) and graph infinite impulse response (GIIR) filters and obtain explicit forms for such filters. We further define autocorrelation and cross-correlation functions of signals on the graph, which enable us obtain the solution to optimal filtering on graphs, i.e., the corresponding Wiener filtering on graphs and the efficient spectra analysis and frequency domain filtering in parallel with those in classical signal processing. This new shift operator based GSP framework enables the signal analysis along a correlation structure defined by a graph shift manifold as opposed to classical signal processing operating on the assumption of the correlation structure with a linear time shift manifold. We further provides the solution to the optimal linear predictor problem over general graphs. Several illustrative simulations are presented to validate the performance of the designed optimal LSI filters.

Index Terms

The authors are with the Department of Electrical and Computer Engineering, Ryerson University, 350 Victoria Street, Toronto, Ontario, Canada M5B 2K3 (E-mail: {adnan.gavili@ryerson.ca, xzhang@ee.ryerson.ca}); phone: +1 416-979-5000 ext. 6686 (main office), fax: +1 416-979-5280, Xiao-Ping Zhang is the corresponding author.
I. INTRODUCTION

Graph signal processing (GSP) is an emerging field, focusing on representing signals as evolving entities on graphs and analyzing the signals based on the structure of the graph [1]–[4]. The temporally evolving measured data from variety of sources in a network such as the measured data from sensors in wireless sensor networks, body area sensor network, transportation networks and weather networks are compatible with signal representation on certain graphs. For instance, a network of sensors implanted in a human body to measure the temperatures of different tissues can be viewed as a graph that the sensor nodes are the graph nodes and the graph structure shows the connection between the sensor nodes. Moreover, the measured temperatures by the nodes are considered to be the signals existing of the corresponding graph. Hence, GSP can be a powerful tool for analyzing and interpreting such signals existing on graphs.

Classical signal processing has provided a wide range of tools to analyze, transform and reconstruct signals regardless of the true nature of the signals evolution. Indeed, classical signal processing may not provide an effective way to represent and analyze the signals that exist on a graph structure. GSP is an attempt to develop a universal tool to process signals on graphs. More specifically, GSP benefits from algebraic and graph theoretic concepts such as graph spectrum, graph connectivity, etc., to analyze structured data [1], [5], [6]. When the structure of a graph is known, the common effort in GSP is to define a shift operator on the graph and then to introduce the concepts of filtering, transformation, denoising, prediction, compression and other operations similar to the conventional counterparts in classical signal processing. However, given the structure of the graph, obtaining the aforementioned operations depends on the definition of the shift operator on graph.

Two major approaches have been developed for signal processing on graphs. The first, and the most widely used, approach is to use the graph Laplacian matrix as the underlying building block for the definitions and tools in GSP [1]. The second approach is to use the adjacency matrix of the underlying graph as the shift operator on graph [2]–[4]. Both approaches define fundamental

\[^1\]Note that obtaining the structure of the graph is a different research question and is out of the scope of this paper.
signal processing concepts on graphs, such as filtering, transformation, downsampling, etc. However, these two approaches are different in graph spectral analysis.

In [7], [8], the authors address the problem of graph learning such that the obtained topology well explains the signal observations. In particular, they infer a graph such that the observed data forms a graph signal with smooth variations on the resulting topology by adopting a factor analysis model for the graph signal and impose a Gaussian probabilistic prior on the latent variables that control these graph signals. The authors of [9] investigate the relationship between the discrete graphical structure of the signal and the inverse of the generalized covariance matrix, called the precision or concentration matrix. The results of this study show that the inverse covariance matrix of indicator variables on the vertices of a graph reflects the conditional independence structure of the graph. One can use this information to calculate the Laplacian matrix corresponding to the graph underlying the signal and then reconstruct the graph adjacency matrix from the Laplacian matrix. One should note that this approach is valid only if the graph signal is assumed to be a random vector that has multivariate Gaussian distribution (called Gaussian-Markov-Random-Field model). The authors in [10], introduce a framework for signal graph wavelet transforms that leads to an efficient reconstruction scheme. Their study shows the connection between conventional and spectral graph wavelets. The idea of graph filter banks is developed in [11]. In this paper, the authors introduce the theory behind sampling operations on graphs, which lead to the design of critically-sampled wavelet-filter banks on graphs. Furthermore, they elaborate on describing downsampling-upsampling operation on graphs in which a set of nodes in the graph are first removed (downsampled) and then replaced back (upsampled) by inserting zeros. They also obtained the conditions on perfect reconstruction filter banks.

Moreover, when the structure of the graph, i.e., the adjacency matrix, is obtained, one has to define the first building block of signal processing called shift operator. It is defined in [1] as the translation on graph via generalized convolution with a delta centered at vertex \( n \). In [2], the graph shift operator is the adjacency matrix of the graph and simple justification of such a choice is presented. However, none of these operators satisfy the energy conservation property similar to its counterpart in classical signal processing. More specifically, applying the shift operator in [1], [2] to a graph signal several times will change the energy content of the graph signal and its frequency components. An isometric shift operator have recently been introduced.
in [12]–[14], which satisfies the energy conservation property. This shift operator is a matrix whose eigenvalues are derived from the graph Laplacian matrix. The limitation of this approach are that its phase shifts are structure-dependent and do not satisfy some other desired properties leading to computationally efficient spectral analysis.

Motivated by the graph shift matrix defined in [2], but fundamentally different, we introduce a unique set of shift operators on graphs that satisfy the properties of the shift operator in classical signal processing. The new shift operator does not have the fundamental drawback of the shift operators defined in the existing literature, i.e., the new shift operator preserves the energy content of the graph signal. We essentially decompose the graph adjacent matrix (the Laplacian matrix can be handled the same way) into two parts. The first part is the graph structure part represented by eigengraph, i.e., frequency components of a graph. The second part is the filtering part represented by eigenvalues of the adjacent matrix, which changes the amplitude of the frequency components. The eigenvalues of the new shift operators therefore only represent phase shift of frequency components that can be flexible constructed. A special construction of these phase shift eigenvalues with nice properties is given. We then elaborate on the structure of linear shift invariant (LSI) graph filters and show that any adjacency matrix can indeed be written as an LSI graph filter using the presented new shift operator. Furthermore, we define the graph finite impulse response (GFIR) and graph infinite impulse filters (GIIR), similar to the classical signal processing counterparts, and obtain an explicit form for such filters. Based on the defined shift operator, we introduce autocorrelation and cross-correlation functions on graph. We then formulate the optimal filtering and spectrum analysis on graphs, i.e., the corresponding Wiener-Hopf equation and Wiener filtering on graphs, and obtain the structure of such filters for any arbitrary graph structure. We finally elaborate on the best linear predictor graph filters and provide several illustrative simulation setups to verify the performance improvements of optimal filtering using our new graph shift operator.

The contribution of this paper can be summarized as follows:

- We define a general set of graph shift operators that satisfy the energy conservation property and other properties in classical signal processing. These shift operators only change the phase of frequency components. Especially, we design a specific shift operator with the desired periodicity property as in classical signal processing. The shift operation can then be considered as discrete-time lossless information flowing structure on a graph.
• We elaborate graph frequency components as a set of *eigengraphs* that represent basic correlation structures on a graph. Each of them can be considered a special graph signal that is a fixed point on a graph, i.e., a fixed point under the presented shift operator subject to only a phase shift, similar properties as in classical signal processing.

• We investigate the properties of the presented shift operator for *linear shift invariant filtering* and show that the adjacent matrix is indeed a LSI filter based on our new graph shift operator.

• We define autocorrelation and cross-correlation functions of a signal on graph. We then obtain a closed-form solution to the *Wiener* filtering problem and show that it has efficient power-spectrum representation similar to classical signal processing. Such a power spectral analysis can only be obtained using our new shift operator. This new shift operator based GSP framework enables the signal analysis along a correlation structure defined by a graph shift manifold as opposed to classical signal processing operating on the assumption of the correlation structure with a linear time shift manifold.

The paper is organized as follows. In section II, we discuss the basics of GSP and present a new set of shift operators. Section III introduces graph filters and Fourier transforms based on the new shift operator. We derive the optimal LSI graph filters in section IV. Section V presents the simulations and section VI concludes of the paper.

*Notations:* Matrices and vectors are represented by uppercase and lowercase boldface letters, respectively. Transpose and Hermitian (conjugate transpose) operations are represented by $(\cdot)^T$ and $(\cdot)^H$, respectively. The notation $I$ stands for the identity matrix, and $\bigcirc$ and $\ast$ are the circular and aperiodic convolution operators, respectively.

II. A NEW SET OF SHIFT OPERATORS AND GRAPH FREQUENCY COMPONENTS

A. Signals on Graph

Consider a dataset with $N$ distinct elements, where some information regarding the relations between data elements is available. One can represent such a dataset and the corresponding relational information as a graph. A graph can be denoted by a $G = \{\mathcal{V}, \mathcal{A}\}$, where $\mathcal{V} = \{\nu_0, \cdots, \nu_{N-1}\}$ is the set of all vertices of the graph, representing the elements in the dataset, and $\mathcal{A}$ is the weighted adjacency matrix that represents the relation between nodes. More specifically, if there is a relation between nodes $\nu_n$ and $\nu_m$, then $a_{n,m} = A(n,m) \neq 0$, otherwise $a_{n,m} = 0$. We note that the elements of the adjacency matrix $A$ is not restricted to a specific set of values.
In this paper, we consider a general graph, either directed or undirected, and assume that data elements take complex scalar values. We define a graph signal as a one-to-one mapping from the set of all vertices to the set of complex numbers:

$$x : \mathcal{V} \to \mathbb{C}, \nu_n \to x_n. \tag{1}$$

Without loss of generality, we represent a graph signal as a vector whose elements are complex numbers assigned to the nodes, $x = [x_1, \cdots, x_N]^T$, where $T$ stands for the transpose operator.

As a special case, a directed cyclic graph is shown in Fig. 1. Such a graph is compatible with the graph representation of a periodic time series signal, $x[n] = x[n + N]$, with $N$ signal points, i.e., one can assign a node to each signal point and the relation between the signal points is the causal relation that shows the time span. One can easily show that the graph adjacency matrix for the directed cyclic graph is given by

$$A = C = \begin{pmatrix}
0 & 0 & \cdots & 1 \\
1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 1 & 0
\end{pmatrix} \tag{2}$$

There are two fundamental components in GSP: the signals represented by the values on vertices, and the signal correlation structure represented by the connections between vertices.

**B. Graph Shift Operator, Information Flow and Filtering**

A graph shift operator allows us to define the notion of information flow over a graph. Indeed, it represents one elementary discrete step on how the information propagates (shifts) from one node to its neighbors. In classical signal processing, i.e., the case where the graph structure is a
cyclic graph, the information flow is restricted to be unidirectional, i.e., from each node to only its next neighbor. In a more complicated graph structure, the information flow will neither be restricted to unidirectional structure nor to a limited number of physical neighbors but depend the graph adjacency matrix. Therefore, the notion of shift operator on graph must be clarified.

1) Graph shift: In [3], the notion of shift operator is defined as a local operation that replaces a signal value at each node of a graph with the linear combination of the signal values at the neighbors of that node. Shift operator is a fundamental element in digital signal processing. Specifically, for a shift operator $\Phi$ and a graph signal $x$, the one-step shifted version of the graph signal, which is a new graph signal, is $\Phi x$. And the $n$-step shifted version of the signal is $\Phi^n x$.

2) Graph signal state change: A graph shift operator is a linear operator such that when it applies to a graph signal $x_n$ at state (or time) $n$, it changes the graph signal into a new graph signal $x_{n+1}$ at step $n+1$. We use the index $n$ to show the state of the graph signal. Equivalently, when a graph shift operator is applied to a graph signal, the state of the original graph signal is shifted to a new state by one unit of shift.

Note that this is similar to time series analysis, when a signal $x(t)$ is shifted in time by a certain amount $T$, $y(t) = x(t - T)$, the signal $x(\cdot)$ at time-state $t - T$ will be mapped to $y(\cdot)$ at time-state $t$. The state of the signal at a certain time stamp is updated due to the effect of the shift operator. Similarly, we defined the $(n + 1)$-th state of the vector of the graph signal, i.e., $x_{n_+}$, as the $n$-th time shifted version of $x_0$. For instance, when the graph shift operator applies to a graph signal, it changes the state of the graph signal at state $n$, i.e., $x_n$, and it changes the state of the graph signal to $n + 1$, i.e., $x_{n+1}$. More specifically, $x_{n+1} = \Phi x_n$.

Also note that in time series analysis, a time-state shift corresponds to a local shift in the cyclic graph. For a general graph shift, there is no such straightforward relationship.

3) Graph signal filtering: A linear filtering is defined by a matrix operation on the graph signal such that the result is also a graph signal. If we define the filter matrix as $H$, the filtered graph signal can be written as $Hx$. We will show later that if the filtering operation also satisfies the shift invariance property, the filter can be written as a polynomial of the new graph shift operator and the filter operation is indeed a modification of the amplitudes of existing signal frequency components, as in classical signal processing.
C. The New Graph Shift Operator

We now define a set of energy conservation shift operators for an arbitrary graph structure.

Definition: Given the adjacency matrix \( A \) for an arbitrary graph and assume that its eigen decomposition is \( A = V \Lambda V^{-1} = \sum_{i=1}^{N} \lambda_i v_i v_i^T = \sum_{i=1}^{N} \lambda_i \tilde{v}_i \), where \( V = [v_1 \ v_2 \ \cdots \ v_N] \) and \( (V^{-1})^T = [\tilde{v}_1 \ \tilde{v}_2 \ \cdots \ \tilde{v}_N] \), and \( v_i \) and \( \tilde{v}_i \) are \( N \times 1 \) column vectors of \( V \) and \( (V^{-1})^T \), respectively, we define the matrix \( A_\phi = V \Lambda_\phi V^{-1} = \sum_{i=1}^{N} \lambda_{\phi_i} \tilde{v}_i \) to be the shift operator such that

\[
\Lambda_\phi = \text{diag}(\lambda_{\phi_1}, \lambda_{\phi_2}, \cdots, \lambda_{\phi_N}),
\]

where \( \lambda_{\phi_k} = e^{j\phi_k}, \phi_k \) is an arbitrary phase in \([0, 2\pi] \) where \( \phi_k \neq \phi_l \) for \( k \neq l \), \( |\Lambda_\phi| = I, \|\cdot\| \) is defined as the point-wise absolute value operator. Thus,

\[
A = V \Lambda V^{-1} = V \Lambda_h \Lambda_\phi V^{-1}
= V \Lambda_h V^{-1} V \Lambda_\phi V^{-1}
= A_h A_\phi = A_\phi A_h,
\]

where \( \Lambda_h = \Lambda \Lambda_\phi^{-1} \) and \( A_h = V \Lambda_h V^{-1} \). In essence, the shift operator \( A_\phi \) preserves all the eigenvectors of the adjacency matrix \( A \), but replaces all the eigenvalues of \( A \) with pure phase shifts.

Definition: We further define a special new shift operator as

\[
A_e = V \Lambda_e V^{-1}, \lambda_{e_k} \lambda_{e_l}^* = e^{-j\frac{2\pi(k-l)}{N}}, \forall k, l = 1, \cdots, N,
\]

where \( \Lambda_e = \text{diag}(\lambda_{e_1}, \lambda_{e_2}, \cdots, \lambda_{e_N}) \). One can write

\[
\lambda_{e_k} = e^{j(\phi_{\text{const}} + \frac{2\pi k}{N})},
\]

where \( \phi_{\text{const}} \) can be any arbitrary constant phase shift. Without loss of generality, we will assume \( \phi_{\text{const}} = 0 \) in the rest of this paper. The shift operator \( A_e \) and \( A_\phi \) satisfies the following properties:

Property 1: \( \|A_\phi x\|^2 = \|x\|^2 \)

Property 2: \( A_e^N x = x \)

The first property is energy conservation. The second property specific for \( A_e \) is consistent with classic signal processing for an important phase shift property of the shift operator. We will
see additional important property of $A_e$ in filtering and spectral analysis in later section.

We note that our definition of the shift operator on graphs brought us the benefit to express the filtering operations in a more compact and meaningful form, similar to their counterparts in classical signal processing. For the choice of $\lambda_{e_k} = e^{-j\frac{2\pi k}{N}}$, the shift operator $A_e$ may not be sparse. Also $A_e(i, j)$ may not be real valued. Therefore, there is a need for a larger memory to save the corresponding operator and more mathematical operations if the filtering operation is employed in the shift domain. We emphasize that, the issues that we mentioned here does not cause any problem if the filtering operation is employed in the Fourier domain. We will also show that an LSI filtering with a non-sparse shift operator may be represented by a polynomial of a sparse graph operator and therefore has efficient shift domain implementation.

Remark: Most of existing shift operators in the literature do not satisfy the energy conservation property. For instance, in [2], the graph shift operator to be the adjacency matrix $A$ of the graph. When such a shift operator applies to a graph signal, the energy content of the graph signal changes. To show this, note that $A$ can be decomposed as $A = V\Lambda V^{-1}$. Applying the graph shift operator $n$ times to the graph signal $x$, results in $x_n = A^n x = V\Lambda^n V^{-1} x$. Since the magnitude of the diagonal elements of $\Lambda$ in general are not equal to 1, as $n$ becomes larger, some of the eigenvalues of $\Lambda^n$ grow exponentially and the other eigenvalues approach zero. This means that the energy content of the signal is not preserved.

We also note that there exist other definitions of the graph shift operator in the literature such as $\frac{1}{\lambda_{\max}(A)} A$ as the normalized shift operator [3] where $\lambda_{\max}(A)$ is the maximum eigenvalue of $A$, and Laplacian matrix based $D^{-1} A$ as the random walk shift operator or $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ as the symmetric normalized shift operator. The diagonal matrix $D$ is the degree matrix of the graph whose $i$-th diagonal element is defined by $D_{ii} = \sum_{j=1}^{N} A_{ij}$. Not only do these shift operators not preserve the energy, but also they actually filter the signals in that they modify the relative strength of different eigenvectors (frequency components). In [1], the translation on graph is defined via generalized convolution with a delta centered at vertex $n$. However, this translation operator aims to produce a geometrically localized shift in the vertex domain and does not preserve the energy. In [12], [14], a new isometric shift operator has recently been introduced that satisfies the energy conservation property with a similar general expression. It is indeed a special case of $A_\phi$. Its eigenvalues are derived from the eigenvalue of the graph Laplacian matrix. Note that in our definition of graph shift operator, the eigenvalues (phase shifts) are detached
from the eigenvalues of the graph adjacent matrix or Laplacian matrix and therefore are more flexible to accommodate other properties such as property 2 above. We will further show that our shift operator have properties leading to computationally efficient spectral analysis.

D. Frequency Content of Graphs, Eigengraphs and Graph Fourier Basis

Consider the graph adjacency matrix $A$ and its eigenvalue decomposition as $A = V \Lambda V^{-1}$, where $\Lambda$ is a diagonal matrix whose $i$-th diagonal element is the $i$-th eigenvalue of $A$. In graph theory, the eigenvalues of the graph adjacency matrix are called the spectrum of the graph [15].

1) Graph frequency content: Defining $V = [v_1 \ v_2 \ \cdots \ v_N]$ and $(V^{-1})^T = [\tilde{v}_1 \ \tilde{v}_2 \ \cdots \ \tilde{v}_N]$, where $v_i$ and $\tilde{v}_i$ are $N \times 1$ column vectors of $V$ and $(V^{-1})^T$, respectively, one can show that

$$A = V \Lambda V^{-1} = \sum_{i=1}^{N} \lambda_i v_i \tilde{v}_i^T = \sum_{i=1}^{N} \lambda_i \tilde{V}_i.$$  \hspace{1cm} (7)

The rank one matrix $\tilde{V}_i = v_i \tilde{v}_i^T$ is called the $i$-th eigengraph, or the $i$-th frequency component, of $A$ and it corresponds to the $i$-th graph frequency $\lambda_i$. Moreover, if none of the elements of $v_i$ and $\tilde{v}_i$ are zero, the corresponding eigengraph is a complete graph, meaning that all nodes are connected to each other. However, the original graph that is a linear combination of the eigengraphs, stated in (7), may not be complete.

Remark: The eigengraphs, or the frequency components, of the graph shift operator $A_\phi$ are the same as those of the adjacency matrix $A$ by definition.

2) Eigengraph structure: To elaborate more on eigengraph structures, let us define $v_i \triangleq [v_{i1} \ v_{i2} \ \cdots \ v_{iN}]^T$ and $\tilde{v}_i \triangleq [\tilde{v}_{i1} \ \tilde{v}_{i2} \ \cdots \ \tilde{v}_{iN}]$. The corresponding $i$-th eigengraph is given by the rank one matrix

$$\tilde{V}_i = v_i \tilde{v}_i^T = \begin{pmatrix}
v_{i1} \tilde{v}_{i1} & v_{i1} \tilde{v}_{i2} & \cdots & v_{i1} \tilde{v}_{iN} \\
v_{i2} \tilde{v}_{i1} & v_{i2} \tilde{v}_{i2} & \cdots & v_{i2} \tilde{v}_{iN} \\
\vdots & \vdots & \ddots & \vdots \\
v_{iN} \tilde{v}_{i1} & v_{iN} \tilde{v}_{i2} & \cdots & v_{iN} \tilde{v}_{iN}
\end{pmatrix},$$ \hspace{1cm} (8)

where $\tilde{V}_i(l, m) = v_{il} \tilde{v}_{im}$. The adjacency matrix of an eigengraph can be viewed as a state transition matrix, where the weight $v_{il} \tilde{v}_{im}$ is the transition weight from node $l$ to node $m$. For instance, the eigengraph of a three node graph and the state transition (bipartite) graph is shown...
in Fig. 2a and Fig. 2b. A more general $N$ node eigengraph transition state graph is shown in Fig. 2c. Note that for the $i$-th rank one eigengraph, the outgoing weight of node $l$ is $v_{il}$ and the incoming weight of node $m$ is $\tilde{v}_{im}$, see Fig. 2b and Fig. 2c. We note that in these figures, $w_{lm} = v_{il} \tilde{v}_{im}$ is the transition weight from node $l$ to node $m$ in a state transition from state $j$ to $j+1$.

Note that an eigengraph is a fixed point of a graph, i.e., $A \tilde{V}_i = \lambda_i \tilde{V}_i$. Although an eigengraph is generally a complete graph, a linear combination of the eigengraphs may not be complete, as is evident for the cyclic graph.

3) Graph Fourier basis and Graph Fourier transform (GFT): We refer to $\mathcal{F} = V^{-1}$ as the graph Fourier transform (GFT) operator since its rows, span a basis to represent the graph signal. The Fourier transform of a graph signal $x$ is $x_{\mathcal{F}} = V^{-1}x$. Thus $\mathcal{F}^{-1} = V$ is the inverse graph Fourier transform (IGFT) operator.

Note that the rows of $\mathcal{F} = V^{-1}$ are not orthogonal for a general shape graph. However, one can easily verify that the vector space $\text{Span}_k \{ \hat{v}_k \}$ of the columns of $(V^{-1})^T$ and the vector space $\text{Span}_k \{ v_k \}$ of the columns of $\mathcal{F}^{-1} = V$ construct a biorthogonal basis, i.e., $\hat{v}_i^T v_m = \delta_{i-m}$. We further note that $i$-th frequency components of a graph, i.e., eigengraph, is constructed by a pair of $\hat{v}_i$, $v_i$ meaning that they are constructed by the Fourier basis of the graph. This interpretation also confirms that the eigengraphs are frequency components of a graph, in which one can decompose a graph signal that is generated by the same graph structure, on those bases without any loss.

For biorthogonal GFT, we further define a dual GFT: $\tilde{\mathcal{F}} = V^H$ and an inverse dual GFT $\tilde{\mathcal{F}}^{-1} = V^{-H}$. As such, we have the inner product preservation: $\langle x, y \rangle = \langle \tilde{x}_{\mathcal{F}}, y_{\mathcal{F}} \rangle$, where $\tilde{x}_{\mathcal{F}} = \tilde{\mathcal{F}}x = V^Hx$.

Each graph structure has its own frequency contents. Only signals that conform to that structure can stay. In other words, the signal decomposition on the Fourier basis of the graph structure does not eliminate any frequency content of a graph signal, if the signal conforms to that graph structure.

The shift operator $A_{\phi}$ is a linear combination of eigengraphs, may not be a local (sparse) operator, meaning that most entries of this matrix may be non-zero. This means that the complexity of applying $A_{\phi}$ to a graph signal of size $N$ is of order of $O(N^2)$. However, once the signal is transformed to the Fourier domain, several other operations such as filtering will be
computationally efficient, as will be discussed later in section IV-B. Also, we will show in Theorem 3 that \( A_\phi \) can be represented as a polynomial of the adjacent matrix \( A \) in certain condition and thus has efficient local implementation.

**Remark:** (i) When \( A \) is not full rank, \( V^{-1} \) is composed of the biorthogonal basis with respect to the column of \( V \). All the definitions still hold true. Indeed, in such case, the graph structure is **band-limited**. Any graph signal that can survive the graph shift or filtering on this graph is also band-limited. (ii) For the adjacent matrix of the undirected graph, or the normalized Laplacian matrix, \( A \) is symmetric and the graph Fourier basis \( V \) will be unitary and the GFT becomes orthogonal.

---

**Fig. 2.** Eigengraph structure for a three node graph is shown in (a), where \( w_{lm}^i = v_{il} \tilde{v}_{im} \) is the signal transition weight from node \( l \) to node \( m \). The state transition representation of the \( i \)-th eigengraph is shown in (b). In this figure, \( v_{il} \) is the outgoing weight from node \( l \) and \( \tilde{v}_{im} \) is the incoming weight to node \( m \). Such a weight distribution preserves the rank one property of the \( i \)-th eigengraph. Part (c) is the generalization of the state transition representation of the \( i \)-th eigengraph for an \( N \) node graph structure.

4) **Linear operator, eigen-operator and graph shift operator:** We note that a linear operator on a graph signal can be defined as \( L \) such that if it applies to a graph signal \( x \), the result will also be a graph signal \( y \) in which \( y = Lx \). We define the operator \( W \) to be an **Eigen-operator**
if the following condition is satisfied

\[ W^kx = Wx, \quad \text{for all } k \in N. \] (9)

One can easily show that, eigengraph operators, i.e., \( V_i \), satisfy this property. The eigengraph operator \( V_i \) is the \( i \)-th basis for decomposition of the graph adjacency matrix \( A \). It means that, a graph structure, i.e., the graph adjacency matrix, is composed of a linear combination of \( n \) independent eigengraphs (as we assume that all eigenvalues of \( A \) are distinct, thus eigenvectors are linearly independent), and is indeed a frequency component of the graph. Moreover, applying an eigengraph operator to a graph signal, selects the corresponding frequency component of the graph signal. This operation is in accordance with the classical signal processing interpretation of a filter operation. More specifically, if a frequency selective filter applies to a signal several times, it returns the same frequency components of the signal similar to the case where the operator applies once. By defining the \( V_i \) as the \( i \)-th frequency component of the graph, we interpret the \( \lambda_i \) in \( A = \sum_{i=1}^{N} \lambda_i V_i \) as the significance of the corresponding frequency component. We further note that, indeed the frequency interpretation of time-series data comes from the graph structure of the time series data, not the time-correlation interpretation of data points.

We emphasize that, a graph shift operator should preserve the frequency contents of a graph. Therefore it should be an equal weighted linear combination of the eigengraphs with only phase shifts. In other words, \( A_{\phi} = \sum_{i=1}^{N} \alpha_i V_i \), where \( |\alpha_i| = 1. \) This left us with the choice that \( \alpha_i = e^{j\phi_i} \), where \( 0 \leq \phi_i < 2\pi \). We further note that, in order to have a graph shift operator with independent eigengraph representation, we assume that \( \phi_i \neq \phi_j \), for all \( i \neq j \). This result is in accordance with the definition of graph shift operator as we defined earlier.

**Example**: Consider the directed cyclic graph with three node as shown in Fig. 3. The
adjacency matrix $A_{\text{cyclic}}$ of this graph is given by

$$
A_{\text{cyclic}} = \begin{pmatrix} 0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \end{pmatrix} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\
1 & e^{j\frac{2\pi}{3}} & e^{j\frac{2\pi}{3}} \\
1 & e^{-j\frac{2\pi}{3}} & e^{-j\frac{2\pi}{3}} \end{pmatrix}.
$$

(10)

where $V^{-1}$ is the discrete Fourier (DFT) transform matrix. The eigengraphs and the state transition (bipartite) graphs of the graph structure in Fig. 3 are shown in Fig. 2.a and Fig. 2.b., where $v_{il} = e^{\frac{2\pi li}{N}}$, $\tilde{v}_{im} = e^{-\frac{2\pi mi}{N}}$. Also note that the $\Lambda_{\text{cyclic}}$ is of the form of the special $\Lambda_e$ defined in (6) and the adjacent matrix $A_{\text{cyclic}}$ is exactly $A_e$.

## III. Graph Filters based on the New Shift Operator

In classical signal processing, filters are referred to operators that apply on a signal as input, and produce another signal as output. Filters can be categorized into different classes, e.g., continues time or discrete time, linear and nonlinear, time invariant and time-varying, etc. The compatible category of filter classes to graph signals is discrete time linear filters. Linear filtering on graphs is represented by multiplying the input signal vector $x$ by a matrix $H \in \mathbb{C}^{N \times N}$, called filter matrix, that results the output signal vector $y = Hx$. This filter operates on graph signals similar to the shift operator, i.e., the filtered signal $y$ at the $i$-th vertex is a linear combination of the value of the original signal $x$. More specifically, $y(i) = \sum_{j=1}^{N} H(i, j)x(j)$. 

Fig. 3. Directed cyclic graph with three nodes, $v_{il} = e^{\frac{2\pi li}{N}}$, $\tilde{v}_{im} = e^{-\frac{2\pi mi}{N}}$. 
A. Linear Shift Invariant Graph Filters

If we consider the shift operator on a graph to be \( A_\phi \), then the linear shift invariant property (LSI) of filters is \( HA_\phi x = A_\phi Hx \). Indeed, this property implies that, the filter and the shift operator are commutable. It is straightforward to show that the following theorem in [2], [3] still hold for a graph LSI filter defined by shift operator \( A_\phi \).

**Theorem 1.** Every polynomial of a square matrix \( A_\phi \) is a graph LSI filter and every graph LSI filter is a polynomial of a square matrix \( A_\phi \).

This theorem shows that every LSI graph filter is a polynomial in the graph shift matrix, i.e.,

\[
H = h(A_\phi) = \sum_{k=0}^{L-1} h_k A_\phi^k
\]  

(11)

where \( h_k \) is called the \( k \)-th tap of the graph filter and \((L - 1)\) is the order of the polynomial representation of the LSI filter.

We therefore can prove the following theorem:

**Theorem 2.** Any arbitrary adjacency matrix \( A \) is an LSI filter under \( A_\phi \).

**Proof:** We know from the definition of the graph shift operator that \( A = A_h A_\phi = A_\phi A_h \), and hence the LSI filter \( H = A_h \).

**Remark:** Indeed, one can write \( A = \sum_{k=0}^{L-1} h_k A_\phi^k \). We emphasize that, for the case where the shift matrix is \( A_\epsilon \), one can show that \( h_k = \sum_{l=1}^{N} e^{2 \pi i k l / N} \lambda_l \), i.e., the \( k \)-th coefficient of the IDFT of the eigenvalue vector \( \lambda \) of \( A \), where \( \lambda_l \) is the \( l \)-th eigenvalue of \( A \). This result allows us to compute the filter coefficients more efficiently. Theorem 2 shows that the adjacent matrix \( A \) can be decomposed into two parts. The first part is the energy conservation graph shift operator \( A_\phi \), i.e., frequency components of a graph. The second part is the filtering part represented by eigenvalues of the adjacent matrix, which changes the amplitude of the frequency components.

**Theorem 3.** The graph shift operator \( A_\phi \) can be written as a polynomial of the graph adjacency matrix \( A \), if the eigenvalues of \( A \) are all distinct.

**Proof:** See Appendix A.

**Remark:** The adjacent matrix \( A \) is often sparse (local). We can design an LSI filter using \( A_\phi \)
since it has good mathematical properties. We can then convert the LSI filter as a polynomial of $A$ such that it has an efficient graph domain implementations. Note that using $A_\phi$ can have efficient in graph frequency domain as we will discuss in section [IV]

**Example**: Let us consider the discrete time circular convolution $y[n] = h[n] \otimes x[n] = \sum_{m \in \mathbb{N}} x[m]h[n-m]$ in classical signal processing for periodic time series data. Such an operator can be cast into the matrix form $y = Hx$ where $x$ and $y$ are the input and output signal vectors, respectively. The filter matrix $H$ has the following Toeplitz form

$$H = \begin{pmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & \cdots & h[1] & h[0] \end{pmatrix}.$$  \hspace{1cm} (12)

Note that, with a small abuse of notation, we will use $h[i]$ in the graph representation instead of $h[n]$ in the classical signal processing counterpart. Using some matrix calculation, one can show that the filter matrix can be written as a polynomial of the circulant adjacency matrix (2) as $H = h(C) = \sum_{k=0}^{N-1} h_k C^k$. Note that in the cyclic graph, the adjacent matrix is exactly the $A_e$ defined in (5), $A_e = C$. This means that the circular convolution is equivalent to the LSI graph filtering based on the graph representation of the periodic time series data. We note that every Toeplitz graph filter matrix can be considered as a linear time invariant filter for time series periodic data.

**Example**: Let us now consider filtering aperiodic time series data in classical signal processing. We show that such a filtering operation is also equivalent to the LSI graph filtering. To show this, let us start with the traditional signal processing filtering as $y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$. Without loss of generality, we assume that $x[n] \neq 0$, for $0 \leq n \leq N-1$ and $h[n] \neq 0$, for $0 \leq n \leq L-1$, and $L < N$. Defining $0_{1 \times L-1} \triangleq [0 \ 0 \ \cdots 0]$, $x \triangleq [x[0] \ x[1] \ \cdots x[N-1]]^T$ and $y \triangleq [y[0] \ y[1] \ \cdots y[N+L-1]]^T$, one can rewrite
the filtering equation as \( y = Hx \) where \( H_{(N+L-1)\times(N+L-1)} \) is defined in (13).

\[
H_{(N+L-1)\times(N+L-1)} =
\begin{pmatrix}
h[0] & 0 & 0 & \cdots & 0 & \cdots & 0 \\
h[1] & h[0] & 0 & \cdots & 0 & \cdots & 0 \\
h[2] & h[1] & h[0] & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & h[L-1] & \cdots & h[0]
\end{pmatrix}
\tag{13}
\]

The output of such a filtering operation, i.e., \( y = Hx \), is equivalent to that of the \( y = \tilde{H}x \), where \( \tilde{H} = \sum_{l=0}^{L-1} h_l C^l \). Note that, in this example \( C \) is the \((N+L-1)\times(N+L-1)\) circulant matrix defined in (2). One can easily show that \( \tilde{H} \) can be written as (14). This result shows that filtering aperiodic discrete time signals by the filter operator \( H \) is equivalent to filtering the zero-padded graph signal \( x \) by the graph filter \( \tilde{H} \). One should note that, convolution for aperiodic signals is equivalent to circular convolution of the zero-padded periodic versions of the input signal, and hence it is equivalent to graph filtering where the graph filter is defined by the filter matrix \( \tilde{H} \).

More specifically, let us consider the zero-padded periodic versions of the aperiodic signal \( x[n] \) and filter \( h[n] \) as \( \tilde{x}[n] = x[n+N+L] \) and \( \tilde{h}[n] = h[n+N+L] \) where

\[
\tilde{x}[n] = \begin{cases} 
  x[n] & n = 0, 1, \cdots N - 1 \\
  0 & n = N, N + 1, \cdots, N + L - 1 
\end{cases}
\]

\[
\tilde{h}[n] = \begin{cases} 
  h[n] & n = 0, 1, \cdots L - 1 \\
  0 & n = L, L + 1, \cdots, N + L - 1 
\end{cases}
\tag{15}
\]

then, \( x[n] * h[n] = \tilde{x}[n] \circ \tilde{h}[n] \), for all \( n \). Hence, filtering aperiodic time series data can be written
as the graph LSI filtering of the zero-padded graph signals.

**Theorem 4.** When \( L_{A_\phi} \leq L \), for an LSI filter there exists an equivalent form for the LSI filter in (11) as \( H = \tilde{h}(A_\phi) = \sum_{k=0}^{L_{A_\phi} - 1} \tilde{h}_k A_\phi^k \) where \( L_{A_\phi} \) is the degree of the minimal polynomial of \( A_\phi \). Moreover, there exist a closed-form expression for the filter taps \( \tilde{h}_k \) as a function of \( h_k \).

**Proof:** See Appendix B for a constructive proof.

Finite/infinite impulse response (FIR/IIR) filters are certain types of filters with great importance in classical signal processing and have simple frequency domain interpretation. We herein aim to bring those concepts to the graph signal processing as GFIR and GIIR filters where \( G \) stands for graph representation. As we have shown in Theorem 4, any LSI filter can be written as a polynomial of the graph shift operator with the maximum order of \( L_{A_\phi} - 1 \). We therefore can define the GFIR and GIIR filters.

**Definition:** We define a GFIR filter to be \( H = \sum_{k=0}^{L-1} h_k A_\phi^k \) where \( L < L_{A_\phi} \) and a GIIR filter to be \( H = \sum_{k=0}^{L-1} h_k A_\phi^k \) where \( L = L_{A_\phi} \).

**B. Linear Shift Variant Filters**

A time-varying filter for time series data (periodic or aperiodic), is not a Toepliz matrix and cannot be represented as polynomial of the shift matrix. Consider a general shift variant filter whose impulse response is represented by \( h[n, m] \), where \( n \) can be considered as the time instance and \( m \) stands for the amount of shift that is needed to be applied to the filter at time of \( n \). More specifically, \( h[n, m] = \mathcal{L}(n, \delta[n - m]) \) where \( \delta[\cdot] \) is the discrete Kronecker delta function and \( \delta[j] = 1 \) only for \( j = 0 \), and \( \mathcal{L}(\cdot) \) is the impulse response function. The output of a general linear time-varying filter can be written as \( y[n] = \sum_{k=0}^{N-1} h[n, m] x[m] \). Equivalently, one can rewrite such a filtering operation as the equivalent matrix form \( y = H x \) where

\[
H = \begin{pmatrix}
h[0, 0] & h[0, 1] & \cdots & h[0, N-1] \\
h[1, 0] & h[1, 1] & \cdots & h[2, N-1] \\
\vdots & \vdots & \ddots & \vdots \\
h[N-1, 0] & h[N-1, 1] & \cdots & h[N-1, N-1]
\end{pmatrix},
\]

represents a time-varying filter, equivalent to the general linear graph filter that is not a Toeplitz matrix and hence cannot be written as a polynomial of the shift operator.
C. Frequency domain interpretation of filtering

The derivation of the results presented earlier in this section is in the time domain for time series data or the shift domain for graph signals. One can also describe the filtering process, equivalently, in the frequency domain obtained by the Fourier transform operator. More specifically, if \( y = Hx \) is the filtering operation in the time/shift domain, it can also be represented in the frequency domain as \( y_F = H_F x_F \), where the subscript \( F \) stands for Fourier transformed versions of the corresponding signals/filters. Note that we have used \( y_F = V^{-1}y \), \( x_F = V^{-1}x \) and \( H_F = V^{-1}HV \). We emphasize that the filtering process \( Hx \) (matrix and vector multiplication) in the shift domain has a simpler representation in the Fourier domain as suggested by \( H_F x_F \). To show this, we note that

\[
H_F = \sum_{k=0}^{L-1} h_k \Lambda_{\phi}^k
\]

\[
= \begin{pmatrix}
\sum_{k=0}^{L-1} h_k (\lambda_{\phi_1})^k & 0 & \cdots & 0 \\
0 & \sum_{k=0}^{L-1} h_k (\lambda_{\phi_2})^k & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sum_{k=0}^{L-1} h_k (\lambda_{\phi_N})^k
\end{pmatrix}
\]

Therefore, the filtering \( H_F x_F \) in Fourier domain is a simple point-wise multiplication. More specifically, \( y(m) = \sum_{n=1}^{N} H(m, n)x(n) \). However, \( y_F(m) = H_F(m, m) \times x_F(m) \).

Note that for \( A_{\phi} \),

\[
H_F(m, m) = \sum_{k=0}^{L-1} h_k (\lambda_{\phi_m})^k = \sum_{k=0}^{L-1} h_k (e^{j\phi_m k})
\]

\[
= H_{DTFT}(\omega | \omega = \phi_m),
\]

and for \( A_e \), \( \phi_m = -\frac{2\pi m}{N} \), and thus

\[
H_F(m, m) = \sum_{k=0}^{L-1} h_k (\lambda_{e_m})^k
\]

\[
= H_{DTFT} \left( \omega | \omega = -\frac{2\pi m}{N} \right).
\]
Interestingly, with the new set of energy conservation shift operators, the GFT coefficient of a LSI filter $H$ can be computed using $L$-tap discrete-time Fourier transforms (DTFT).

**IV. Correlation Functions of Graph Signals and Optimal LSI Graph Filters**

In this section, we assume that the structure of the graph is known, meaning that the graph adjacency matrix $A$ and the related shift operator $A_\phi$ is given. Assuming that $A_\phi$ is known, we aim to obtain the structure of the graph LSI filters such that a certain set of constraints are satisfied. We discuss several filter design problems in GSP in the sequel that arise in classical signal processing.

**A. Wiener Filter for Directed Cyclic Graph Data (Time Series)**

We will first reformulate the time series signal Wiener filter using GSP representation and then generalize it to arbitrary graph signals. Consider the graph representation of the time series data in Fig. 1. Assume that $x$ is the graph signal and $y$ is a noisy measurement of the graph signal $x$: $y_i = x_i + n_i$, where $n_i$ is i.i.d. zero mean white Gaussian noise. A conventional question in denoising problems is to design an LSI filter such that the residual error $\|Hy - x\|_2^2$ is minimum. Strictly speaking, when $x$ and $y$ are given, we aim to solve the following optimization problem

$$\min_H \|Hy - x\|_2^2.$$  \hspace{1cm} (20)

Since $H$ is shift invariant, it can be written as $H = \sum_{k=0}^{L-1} h_k C^l$, where $C$ is defined earlier. Note that the filtered signal can be rewritten as $Hy = \sum_{k=0}^{L-1} h_k C^l y = Bh$ where

$$B_{N \times L} = [y \ C^1 y \cdots C^{L-1} y], \quad h = [h_0 \ h_1 \cdots h_{L-1}]^T.$$ \hspace{1cm} (21)

Rewrite (20) by replacing $Hy$ by its equivalent $Bh$,

$$\min_h \|Bh - x\|_2^2.$$ \hspace{1cm} (22)

Since $L = L_{A_\phi} \leq N$ is the degree of minimal polynomial of the graph shift matrix, hence we only consider the cases where $L \leq N$ (Note that when $N = L$, the matrix $B$ is full rank, hence the solution to the optimization problem (22) can be written as $h^o = B^{-1}x$). However, if $L < N$, the optimization problem (22) is overdetermined, meaning that there are more constraints than
variables and there is no solution to such a problem except for the degenerate case where some of the equations are linearly dependent. One possible approach to tackle this problem is to find a least square solution that can be obtained by solving

$$B^H B h = B^H x,$$  \hspace{1cm} (23)

where $H$ is the Hermitian operator. Such a solution has an interesting interpretation for time series data as will be shown in the sequel. We note that (23) can be written as

$$\begin{pmatrix}
    y^H y \\
y^H C^H y \\
\vdots \\
y^H (C^{L-1})^H y
\end{pmatrix} [y \ C y \cdots C^{L-1} y] h = 
\begin{pmatrix}
    y^H x \\
y^H C^H x \\
\vdots \\
y^H (C^{L-1})^H x
\end{pmatrix} x$$

or, equivalently, as (24). We note that the circulant matrix $C$ has the unitary property, i.e., $(C^H)^k C^k = I$ for all $k$. Moreover, we claim that $y^H (C^l)^H y$ is the autocorrelation of the vector $y$ at lag $l$. To show this, we first note that $C^l y$ is the circularly shifted version of the $y$ by amount $l$. Defining the autocorrelation function of $y$ as

$$R_{yy}(l) \triangleq \sum_{n \in \mathbb{N}} y_n y_{n+l}^*,$$ \hspace{1cm} (25)

one can easily show that

$$y^H (C^l)^H y = (C^l y)^H y = \sum_{n \in \mathbb{N}} y_n y_{n+l}^* = R_{yy}(l),$$ \hspace{1cm} (26)
where $y_k$ is the $(k \text{ mod } N)$-th element of the vector $y$ and $^*$ is the conjugation operator. We also define the cross-correlation between the input and output vectors $x$ and $y$, as

$$r_{xy}(l) \triangleq \sum_{n \in N} x_n y_n^* = y^H (C^l)^H x.$$ (27)

The linear equations (24) can hence be rewritten as

$$B^H B h = R_{yy} = r_{xy},$$ (28)

i.e.,

$$
\begin{pmatrix}
R_{yy}(0) & R_{yy}^*(1) & \cdots & R_{yy}^*(L-1) \\
R_{yy}(1) & R_{yy}(0) & \cdots & R_{yy}^*(L-2) \\
\vdots & \vdots & \ddots & \vdots \\
R_{yy}(L-1) & R_{yy}(L-2) & \cdots & R_{yy}(0)
\end{pmatrix}
\begin{pmatrix}
r_{xy}(0) \\
r_{xy}(1) \\
\vdots \\
r_{xy}(L-1)
\end{pmatrix}.
$$ (29)

Eq. (29) is indeed the Wiener-Hopf equation.

Note that the LSI property of graph filters for time series data leads to the Wiener filtering in the conventional signal processing. One can also compute the autocorrelation and cross-correlation more efficiently as

$$R_{yy}(m, l) = y_F^H \Lambda^m (\Lambda^*)^l y_F = \sum_{n=0}^{N-1} |y_F(n)|^2 \lambda_n^{m-l},$$

$$r_{xy}(l) = y_F^H \Lambda^l x_F = \sum_{n=0}^{N-1} y_F^*(n) x_F(n) (\lambda_n^*)^l,$$ (30)

which has lower computational complexity than calculating the autocorrelation and the cross-correlation using the definition directly. Note that $y_F = V^{-1} y$ and $x_F = V^{-1} x$ are the Fourier representations of the output and input signals, respectively, and $(\Lambda^H)^k \Lambda^k = I$.

The optimal LSI filtering (24) also has power spectrum representation, if $L = N$. Let us consider
B^H B = R_{yy} = \sum_{l=0}^{L-1} (C^l y)(C^l y)^H \\
= V \left( \sum_{l=0}^{L-1} (A^l y_F)(A^l y_F)^H \right) V^H \\
= VY_F \left( \sum_{l=0}^{L-1} (\lambda^l \lambda^H) \right) Y_F^* V^H, \hspace{1cm} (31)

where \( Y_F = \text{diag}(y_F) \) is a diagonal matrix whose \( i \)-th diagonal element is \( y_F(i) \), \( \lambda \) is a column vector whose \( i \)-th element is equal to \( \Lambda(i, i) \). Note that, we have used \( \Lambda^l y_F(A^l y_F)^H = \Lambda^l y_F y_F^H \Lambda^* = Y_F(\lambda^l \lambda^H)Y_F^* \).

**Lemma 1.** The identity \( \sum_{l=0}^{N-1} (\lambda^l \lambda^H) = NI \) holds true.

**Proof:** See Appendix C. Note that this \( \lambda \) is the same as \( \lambda_e \) we defined in (6).

We now multiply the left side of (29) by \( V^{-1} \) and since we have \( V^{-1} = V^H \) for the directed cyclic graph, one can rewrite the equation (29), for the case where \( L = N \), as

\[
NY_F Y_F^* V^H h = V^{-1} r_{xy} \Rightarrow N|Y_F|^2 h_F = r_{xy_F},
\]

where we have used \( V^H h = V^{-1} h = h_F \). Therefore, it is easy to verify that

\[
h_F(i) = \frac{r_{xy_F}(i)}{N|y_F(i)|^2}. \hspace{1cm} (33)
\]

This result is consistent with the power spectrum interpretation in the classical signal processing. Note that the property of \( \Lambda_e \) as shown in Lemma 1 is a key for the spectrum representation (33) to hold.

We will show that a similar structure exists for a general LSI filter for any arbitrary graph structure.

**B. Correlation Functions and Optimal (Wiener) Filtering for Arbitrary Graph Signals**

Arbitrary graph signals may have complex structures, e.g., directed or undirected, weighted or un-weighted, etc. As we defined the shift matrix to be the \( A_\phi \), we can construct a general LSI filter as a polynomial of the shift matrix, i.e., \( H = h(A_\phi) = \sum_{k=0}^{L-1} h_k A_k, \) where \( h_k \) is the
\[ \begin{pmatrix} y^H y & y^H A \phi y & \cdots & y^H (A \phi)^{L-1} y \\ y^H A^H y & y^H A^H A \phi y & \cdots & y^H A^H (A \phi)^{L-1} y \\ \vdots & \vdots & \ddots & \vdots \\ y^H (A^L \phi)^{L-1} y & y^H (A^L \phi)^{L-1} A \phi y & \cdots & y^H (A^L \phi)^{L-1} (A^L \phi)^{L-1} y \end{pmatrix} h = \begin{pmatrix} y^H x \\ y^H A^H x \\ \vdots \\ y^H (A^L \phi)^{L-1} H x \end{pmatrix}. \quad (35) \]

\( k \)-th filter tap. We also define
\[ h \triangleq [h_0 \ h_1 \ \cdots \ h_{L-1}]^T \]
as the vector of the filter. Consider again the denoising problem (22) given by
\[ \min_h \| B_{\text{new}} h - x \|_2^2, \quad (34) \]
where \( B_{\text{new}} = [y \ A \phi y \ \cdots \ A \phi^{L-1} y] \) and the solution to such a problem was obtained earlier. We proved that for the time series graph, the optimal solution is the Wiener filter given by (35). The Wiener filter structure depends on the availability of the autocorrelation function of the output data \( y \) and the cross-correlation of the input \( x \) and output \( y \). For a general graph, the autocorrelation and cross-correlation need to be defined on a particular graph structure.

**Definition**: We define the autocorrelation function of the signal \( y \) on an arbitrary graph with lag \( l \) as
\[ R^G_{yy}(l) \triangleq y^H (A^l \phi)^H y. \quad (36) \]
We also define the cross-correlation between the input and output vectors \( x \) and \( y \) at lag \( l \), as
\[ r^G_{xy}(l) \triangleq y^H (A^l \phi)^H x. \quad (37) \]
As can be seen, the autocorrelation and the cross-correlation on graphs are indeed the correlations between a graph signal and a graph shifted signal, where a shift is defined by the graph shift operator.

The linear equations (35) can hence be rewritten as
\[ R^G_{yy} h = r^G_{xy}, \quad (38) \]
i.e.,

\[
\begin{pmatrix}
R_{yy}^G(0) & R_{yy}^G(1) & \cdots & R_{yy}^G(L-1) \\
R_{yy}^G(1) & R_{yy}^G(0) & \cdots & R_{yy}^G(L-2) \\
\vdots & \vdots & \ddots & \vdots \\
R_{yy}^G(L-1) & R_{yy}^G(L-2) & \cdots & R_{yy}^G(0)
\end{pmatrix}
\begin{pmatrix}
h_0^G \\
h_1^G \\
\vdots \\
h_{L-1}^G
\end{pmatrix}
\]

which can be considered the Wiener-Hopf equation for graph signals.

We note that, a typical graph can have a large number of nodes, meaning that the size of the \( A_\phi \) matrix is large. The optimal filtering (39) needs the autocorrelation and cross-correlation functions. Since the computational complexity of calculating these functions are high (see the definition of the functions and the large multiplications of matrices), it is desirable to obtain the autocorrelation and cross-correlation functions in (39), using a similar equation as (30). To do so, let us consider

\[
A_\phi = V\Lambda_\phi V^{-1}
\]

where

\[
\Lambda_\phi = \begin{pmatrix}
\lambda_1 & 0 & 0 & \cdots & 0 \\
0 & \lambda_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & \lambda_{N-1} & 0 \\
0 & 0 & \cdots & \lambda_N
\end{pmatrix}
\]

Therefore, one can write

\[
R_{yy}^G(i, j) = \sum_{n=0}^{N-1} \tilde{y}_F^*(n)y_F(n)(\lambda_n^*)^i\lambda_n^j
\]

\[
r_{xy}^G(i) = \sum_{n=0}^{N-1} \tilde{y}_F^*(n)x_F(n)\lambda_n^*-i
\]

where we define

\[
y_F \triangleq V^{-1}y, \quad x_F \triangleq V^{-1}x, \quad \tilde{y}_F \triangleq V^Hy, \quad \tilde{x}_F \triangleq V^Hx.
\]

Note that, since \( \lambda_n^* \lambda_n = 1 \), then

\[
(\lambda_n^*)^i\lambda_n^j = \lambda_n^{i-j} = (\lambda_n^*)^{i-j},
\]
and the solution to (38) becomes similar to the Wiener filter for time series data. We further note that, for undirected graph $V^T = V^{-1}$ therefore, one can write

$$\tilde{y}_F^* = \left(V^H y\right)^* = \left(V^T y\right)^* = \left(V^{-1} y\right)^* = V^{-1} y^*$$

$$\tilde{x}_F = V^H x = (V^T)^* x = (V^{-1})^* x = \left(V^{-1} x^*\right)^*.$$  \hspace{1cm} (44)

Moreover, if the signal on the graph is real valued, then $\tilde{y}_F^*$ and $\tilde{x}_F$ can be written as

$$\tilde{y}_F^* = V^{-1} y^* = V^{-1} y = y_F$$

$$\tilde{x}_F = \left(V^{-1} x^*\right)^* = \left(V^{-1} x\right)^* = x_F^*.$$  \hspace{1cm} (45)

Hence, the autocorrelation and cross-correlation in the optimal filtering (39) can be simplified as

$$R_{xy}^G(i, j) = \sum_{n=0}^{N-1} (y_F(n))^2 \lambda_n^{j-i}$$

$$r_{xy}^G(i) = \sum_{n=0}^{N-1} y_F(n) x_F(n)^* \lambda_n^i.$$  \hspace{1cm} (46)

For a general shape graph and its related shift operator $A_\phi$, there may not be a power spectrum interpretation similar to the one in (32) for time series data. The reason is that, the identity $\sum_{l=0}^{N-1} (\lambda^l \lambda^l H) = NI$ may not hold for the shift operator $A_\phi$. However, if we choose the shift matrix to be $A_e$ for a general shape graph, that is defined in (6). From Lemma 1, we know that the equality $\sum_{l=0}^{N-1} (\lambda^l e^l H) = NI$ holds true. Therefore the same spectral representation, as in (32), holds true for a general shape graph, i.e.,

$$\tilde{h}_F(i) = \frac{r_{xy}(i)}{N |y_F(i)|^2},$$  \hspace{1cm} (47)

where $\tilde{h}_F = V^H h$.

**Remark:** We obtained the structure of the optimal graph filter earlier and showed that using the new graph shift operator, we can obtain the simple closed-form power spectrum solution (47), which is only possible by using (3) that satisfies the condition in Lemma 1. Any other graph shift operator does not lead to such efficient solution for the filtering problem such as (34). This is a major difference between our shift operator (3) and the shift operator proposed
C. Optimal LSI Prediction Filter for Arbitrary Graph Data

Another interesting problem in the design of LSI graph filters is to obtain the best LSI $q$-step ahead prediction filter. Such a problem can be formulated as

$$\min_h \|B_{\text{new}} h - A_{\phi}^q x\|_2^2,$$

where $B_{\text{new}}$ is defined earlier and $A_{\phi}^q x$ is the $q$-step ahead state of the input vector $x$. One can easily show that the optimal solution to such a problem satisfies

$$B_{\text{new}}^H B_{\text{new}} h = B_{\text{new}}^H A_{\phi}^q x.$$

Such a solution for the time series graph can be written as

$$h = \begin{pmatrix} R_{\text{xx}}(0) & R_{\text{xx}}^*(1) & \cdots & R_{\text{xx}}^*(L-1) \\ R_{\text{xx}}^*(1) & R_{\text{xx}}(0) & \cdots & R_{\text{xx}}^*(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{\text{xx}}^*(L-1) & R_{\text{xx}}^*(L-2) & \cdots & R_{\text{xx}}(0) \end{pmatrix} \begin{pmatrix} R_{\text{xx}}^G(L-q) \\ R_{\text{xx}}^G(L-q+1) \\ \vdots \\ R_{\text{xx}}^G(0) \\ \vdots \\ R_{\text{xx}}^G(L-q-1) \end{pmatrix},$$

which is the equation for the best linear $q$-step ahead prediction filter in the conventional signal processing. Moreover, (49) can be computed in the frequency domain as described in the previous subsection.

V. Simulations

Figs. 4 and 5 show the percentage of the reconstruction error for the Wiener filtering problem (34), for two different sets of noisy measurements. The dataset contains the average temperatures of 40 US states capitals, i.e., $x_t$ where $t \in \{1, 2, \cdots, M\}$, for a horizon of $M = 264$ consecutive days in 2015, and we consider a noisy measurements of those graph signals as $y_t = x_t + n_t$.

The average percentage of the reconstruction error is defined as $\frac{1}{M} \sum_{t=1}^{M} \frac{|x_t - H_0 y_t|}{|x_t|}$, where $H_0^*$ is the optimal graph filter obtained by the optimization problem (34). We consider three different approaches to construct the graph, i.e., the $k$-nearest neighbor method with $k = 9$. 

Wednesday 15th June, 2016 DRAFT
the exponentially distance-based weighted graph adjacency considered by [3], and the empirical covariance-based graph construction introduced in [16]. In all cases, the reconstruction error of Wiener filtering using our shift operator is much lower than the traditional adjacency-based operators.

The source of errors are due to the following fact. For large values of $L$ (the number of filter coefficients), the polynomial representation of the LSI filter $H = \sum_{l=0}^{L} h_l A^l$ dominates only the frequency components that have the largest eigenvalues of $A$. More specifically, adding further coefficients to the filter, does not improve the performance of the Wiener filtering since the frequency components corresponding to small eigenvalues are indistinguishable from the noise. We observe that after, almost, adding 15 taps, the performance of the adjacency-based approaches will be saturated. However, if we use our defined graph shift operator, adding more coefficients will improve the performance of the filtering operation as it allows us to use more coefficients to reduce the reconstruction error while keeping all frequency components of the graph. Furthermore, Fig. 4 and 5 show that, when the noise variance is larger, the performance of Wiener filtering will be affected and the percentage of the reconstruction error becomes larger, as we expect.

VI. CONCLUSIONS

In this paper, we define a new set of shift operators for graph signals satisfying the energy conservation properties as in classical signal processing by reset the eigenvalues of the adjacent matrix or Laplacian matrix flexibly. We show that such shift operators preserve the energy content of the graph signal in both shift and frequency domains. We further investigate the properties of LSI graph filters and show that any LSI filter can be written as a polynomial of the graph shift operator. We then categorize the LSI filters as GFIR and GIIR filters, similar to the classical FIR and IIR filters and obtain explicit forms for such filters. Based on these energy conservation shift operators, we further define autocorrelation and cross-correlation functions of signals on the graph. We then obtain the structure of the optimal LSI graph filters, i.e., Wiener filtering, through the construction of the Wiener-Hopf equation on graph. We show that only with the proposed graph shift operator, we can obtain the efficient spectra analysis and frequency domain filtering in parallel with those in classical signal processing. We further provides the solution to the optimal linear predictor problem over general graphs. Several illustrative simulations are
Fig. 4. Average reconstruction error for the Wiener filtering problem (34), where the noise variance is $\sigma^2 = 1$.

Fig. 5. Average reconstruction error for the Wiener filtering problem (34), where the noise variance is $\sigma^2 = 10$.

presented to validate the performance of designed optimal LSI filters.

Our new shift operator based GSP framework enables the signal analysis along a correlation structure defined by a graph shift manifold as opposed to classical signal processing operating on the assumption of the correlation structure with a linear time shift manifold.

**ACKNOWLEDGMENT**

We would like to thank Dr. José Moura and Dr. Antonio Ortega for discussions and feedbacks of many concepts and ideas in the early stage of manuscript development.
A. Proof of Theorem \[3\]

Let us assume that \( A_{\phi} \) can be represented as a polynomial of \( A \), i.e.,

\[
A_{\phi} = \sum_{k=0}^{N-1} g_k A^k. \tag{A.1}
\]

Since the Fourier basis of \( A \) and \( A_{\phi} \) are the same by the definition, we can diagonalize the two operator by multiplying \( V \) and \( V^{-1} \) to the two operator form right and left, respectively. Therefore, we can write

\[
\Lambda_{\phi} = \sum_{k=0}^{N-1} g_k \Lambda^k. \tag{A.2}
\]

This equation can also be written as a linear matrix equation as

\[
\begin{pmatrix}
1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{N-1} \\
1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{N-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \lambda_N & \lambda_N^2 & \cdots & \lambda_N^{N-1}
\end{pmatrix}
\begin{pmatrix}
g_0 \\
g_1 \\
\vdots \\
g_{N-1}
\end{pmatrix}
= \begin{pmatrix}
\lambda_{\phi_1} \\
\lambda_{\phi_2} \\
\vdots \\
\lambda_{\phi_N}
\end{pmatrix}, \tag{A.3}
\]

or more compactly, as

\[
Zg = \lambda_{\phi} \tag{A.4}
\]

where \( g = [g_0 \ g_1 \ \cdots \ g_{N-1}]^T \) and \( \lambda = [\lambda_{\phi_1} \ \lambda_{\phi_2} \ \cdots \ \lambda_{\phi_N}]^T \). We note that, \( Z \) is the well-known Vandermonde matrix and has full-rank iff \( \lambda_i \neq \lambda_j \), for \( i, j \in \{1, 2, \cdots, N\} \). Therefore, if \( Z \) is full-rank, then the linear equation (A.4) has a unique solution \( g = Z^{-1}\lambda \) (We emphasize that, there exists efficient recursive algorithms for obtaining the inverse of Vandermonde matrices with low computational complexity). This completes the proof.

B. Proof of Theorem \[4\]

The minimal polynomial of an \( N \times N \) matrix \( A_{\phi} \) is defined by a polynomial with minimum degree that satisfies \( p(A_{\phi}) = \sum_{i=0}^{L_{A_{\phi}}} \alpha_i A_{\phi}^i = 0_{N \times N} \). Note that for the degenerate case where \( A_{\phi} \) is not full-rank, \( L_{A_{\phi}} \neq N \). Without loss of generality, we assume that \( \alpha_{L_{A_{\phi}}} = 1 \). In order to
obtain the other \( \alpha_i \)'s, we first note that \( \sum_{i=0}^{L_{A_{\phi}}} \alpha_i A_{\phi}^i = -A_{\phi}^{L_{A_{\phi}}} \). We then write the diagonalized version of this equation as

\[
\sum_{i=0}^{L_{A_{\phi}}-1} \alpha_i A_{\phi}^i = -A_{\phi}^{L_{A_{\phi}}}. 
\]  

(A.5)

One can write this equation in a matrix form as

\[
\begin{pmatrix}
1 & \lambda_{\phi_1} & \lambda_{\phi_1}^2 & \cdots & \lambda_{\phi_1}^{L_{A_{\phi}}-1} \\
1 & \lambda_{\phi_2} & \lambda_{\phi_2}^2 & \cdots & \lambda_{\phi_2}^{L_{A_{\phi}}-1} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
1 & \lambda_{\phi_L A_{\phi}} & \lambda_{\phi_L A_{\phi}}^2 & \cdots & \lambda_{\phi_L A_{\phi}}^{L_{A_{\phi}}-1}
\end{pmatrix} \begin{pmatrix}
\alpha_0 \\
\alpha_1 \\
\vdots \\
\alpha_{L_{A_{\phi}}-1}
\end{pmatrix} = - \begin{pmatrix}
\lambda_{\phi_1}^{L_{A_{\phi}}} \\
\lambda_{\phi_2}^{L_{A_{\phi}}} \\
\vdots \\
\lambda_{\phi_L A_{\phi}}^{L_{A_{\phi}}}
\end{pmatrix},
\]  

(A.6)

or in a more compact form \( \tilde{Z} \alpha = \tilde{\lambda} \), where \( \alpha = [\alpha_0 \, \alpha_1 \, \cdots \, \alpha_{N-1}]^T \) and \( \tilde{\lambda} = -[\lambda_{\phi_1} \, \lambda_{\phi_2} \, \cdots \, \lambda_{\phi_{L A_{\phi}}}]^T \). The solution to the \( \alpha \)'s can be easily obtained as \( \alpha = \tilde{Z}^{-1} \tilde{\lambda} \).

We also note that for the special \( A_e \), we choose \( \lambda_{e_k} = e^{-j \frac{2 \pi k}{L_{A_e}}} \), \( \forall k \in \{1, \ldots, N\} \), if \( A \) is full-rank, otherwise we choose \( \lambda_{e_k} = e^{-j \frac{2 \pi k}{L_{A_e}}} \). We therefore obtain the closed-form solution for the \( \alpha \) as \( \alpha_k = -\sum_{l=1}^{N} \frac{2 \pi k}{L_{A_e}} \). That is \( \alpha_0 = -1 \) and \( \alpha_l = 0 \) for \( l \neq 0 \). This is also straightforward since \( \lambda_{e_k} = 1, \forall k \), i.e., \( A_e = I \).

Once we obtain \( \alpha_k \), we can write \( A_{\phi} = -\sum_{i=0}^{L_{A_{\phi}}-1} \alpha_i A_{\phi}^i \).

Now consider an LSI filter with length \( L = L_{A_{\phi}} + 1 \) as \( H = \sum_{k=0}^{L} h_k A_{\phi}^k \). Using the results obtained earlier, we can rewrite this LSI filter as

\[
H = \sum_{k=0}^{L_{A_{\phi}}} h_k A_{\phi}^k = \sum_{k=0}^{L_{A_{\phi}}-1} h_k A_{\phi}^k + h_{L_{A_{\phi}}} A_{\phi}^{L_{A_{\phi}}},
\]

\[
= \sum_{k=0}^{L_{A_{\phi}}-1} h_k A_{\phi}^k + h_{L_{A_{\phi}}} (- \sum_{k=0}^{L_{A_{\phi}}-1} \alpha_k A_{\phi}^k)
\]

\[
= \sum_{k=0}^{L_{A_{\phi}}-1} (h_k - h_{L_{A_{\phi}}} \alpha_k) A_{\phi}^k = \sum_{k=0}^{L_{A_{\phi}}-1} \tilde{h}_k A_{\phi}^k. 
\]  

(A.7)

This procedure can be repeated for any arbitrary \( L > L_{A_{\phi}} + 1 \). Therefore, we only consider LSI graph filters with \( L_{A_{\phi}} \) filter taps and we interchangeably use \( L \) instead of \( L_{A_{\phi}} \).

We finally note that, for the special case \( A_e = I \), meaning that if \( k > L_{A_{\phi}} \), then \( A_{\phi}^k = A_{\phi}^m \).
where \( m = (k \mod L_{A, \phi}) \).

C. Proof of Lemma [7]

To prove this lemma, let us rewrite \( L = \sum_{l=0}^{N-1} (\lambda l^H \lambda l^T) \) and note that \( \lambda_k \neq 0 \). One can easily show that for \( k \neq m \),

\[
L(k, m) = \sum_{l=0}^{N-1} (\lambda_k \lambda^*_m)^l = \frac{1 - (\lambda_k \lambda^*_m)^N}{1 - \lambda_k \lambda^*_m}.
\]  
(A.8)

Therefore, the conditions for \( L \) to be diagonal is \( L(k, m) = 0 \) for \( k \neq m \). This condition can be translated into

\[
\lambda_k \lambda^*_m \neq 1 \text{ and } (\lambda_k \lambda^*_m)^N = 1.
\]  
(A.9)

The result holds true iff \( \lambda_k \lambda^*_m = e^{-j \frac{2\pi (k-m)}{N}} \). One should note that, for time series data \( \lambda_i = e^{-j \frac{2\pi i}{N}} \), and hence \( \lambda_k \lambda^*_m = e^{-j \frac{2\pi (k-m)}{N}} \) is satisfied. Also note that for \( k = m \) we have \( L(k, k) = \sum_{l=0}^{N-1} (\lambda_k \lambda_k^*)^l = \sum_{l=0}^{N-1} (1)^l = N \), therefore, \( L = NI \).

REFERENCES

[1] D. Shuman, S. 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 Processing Magazine, vol. 30, no. 3, pp. 83–98, May 2013.

[2] A. Sandryhaila and J. Moura, “Discrete signal processing on graphs,” IEEE Transactions on Signal Processing, vol. 61, no. 7, pp. 1644–1656, April 2013.

[3] ——, “Discrete signal processing on graphs: Frequency analysis,” IEEE Transactions on Signal Processing, vol. 62, no. 12, pp. 3042–3054, June 2014.

[4] ——, “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure,” Signal Processing Magazine, IEEE, vol. 31, no. 5, pp. 80–90, Sept 2014.

[5] M. Puschel and J. Moura, “Algebraic signal processing theory: Foundation and 1-d time,” IEEE Trans. on Signal Process., vol. 56, pp. 3572–3585, Aug 2008.

[6] ——, “Algebraic signal processing theory: Cooley-tukey type algorithms for dcts and dsts,” IEEE Trans. on Signal Process., vol. 56, pp. 1502–1521, April 2008.

[7] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning graphs from signal observations under smoothness prior.” Submitted to IEEE Trans. Signal Process., 2014.

[8] ——, “Laplacian matrix learning for smooth graph signal representation,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 2015, pp. 3736–3740.
[9] P.-L. Loh and M. J. Wainwright, “Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses,” *Ann. Statist.*, vol. 41, no. 6, pp. 3022–3049, 12 2013. [Online]. Available: [http://dx.doi.org/10.1214/13-AOS1162](http://dx.doi.org/10.1214/13-AOS1162)

[10] N. Leonardi and D. Van De Ville, “Tight wavelet frames on multislice graphs,” *IEEE Trans. on Signal Process.*, vol. 61, pp. 3357–3367, July 2013.

[11] S. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” *IEEE Trans. on Signal Process.*, vol. 60, pp. 2786–2799, June 2012.

[12] B. Girault, P. Goncalves, and . Fleury, “Translation on graphs: An isometric shift operator,” *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2416–2420, Dec 2015.

[13] B. Girault, “Stationary graph signals using an isometric graph translation,” in *European Signal Processing Conference (EUSIPCO)*, Aug 2015, pp. 1516–1520.

[14] ——, *Signal Processing on Graphs - Contributions to an Emerging Field*, PhD thesis, Dec 2015.

[15] F. Chung, *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, vol. 92). Providence, RI: AMS Bookstore, 1997.

[16] C. Zhang, D. Florencio, and P. Chou, “Graph signal processing - a probabilistic framework,” *Microsoft Publisher, Technical Report*, April 2015.